A SURVEY ON COLORIZATION WITH CGAN

STUDY REPORT ON CONDITIONAL GENERATIVE ADVERSARIAL NETWORKS

Ching Pui WAN
The Hong Kong University of Science and Technology
cpwan@ust.hk

June 5, 2019

ABSTRACT

cGAN has been applied on colorization in recent years. Various models and techniques are being developed for stochastic output. In this study, the designs of generator and discriminator will be explored. The encoder-decoder styled generator and U-Net generator are implemented and are to be compared. Regularization on generator, either by reconstruction loss or conditioning the discriminator, are implemented and explored. A few challenges including the generalization to unseen object, and the color ambiguity problem are briefly discussed. The implementation and exploration in this study pave the way to further study in larger scale image colorization.

1 Introduction

In conventional computer vision problem, colorization can be formulated as a problem of recovering the two color channels in the $L^*a^*b^*$ space (which is the colored image) given a channel of the brightness information (which is the grayscale image). Another approach can be operating on the RBG color space directly and left the conversion between color space to be done by the neural network.

Recent years, generative adversarial network (GAN) has attracted attention in machine learning community for its merit of unsupervised feature extraction and sample synthesis. Conditional GAN, a variant of GAN model, learns to generate samples with given attributes of which the input noise vector is conditioned to. Conditional GAN alleviates the problem of mode collapse in GAN, however, suffices to mode collapse with respect to the attributes that the input noise is conditioned to, leading to a situation that the input noise vector is always ignored. For example, a cGAN trained on digits may always return the same image of digit. Colorization, the process of mapping grayscale image to colored image, is an unconstrained problem for which each grayscale pixel may correspond to multiple choices of colored pixels. Incorporate noise vectors is necessary for the diversity of the colorization of an image. With appropriate network architecture, diverse colorization could be achieved.

On fully automatic colorization, previous works have feed the grayscale image as the input of a generator of "encoder-decoder" style equiped with either a U-net structure or concatenate the latent space with an additional classification network. The structures served to provide a feature-based representation for the generator instead of a noise vector used in GAN. The generator tends to generate samples by interpolating within the latent space of the structure, raising the dependency of the "encoder" part of the network and turning the problem to designing a better structure to encode the feature. A several variants have incorporated reference image, color stokes, or color palette for learning a better representation. The automatic colorization network may fail to generalize to unseen objects when the feature has not been captured, since the representation in the new object in latent space can be ambiguous and may not be captured by the trained samples.

In this study, a few models and techniques were reviewed. Implementation on some of the techniques were done and experimented.

2 Background

GANs are generative models that learn to map a latent variable z, initialized as a random noise, to a synthesized output y. Similarly, the conditional GANs learn to map the random noise z to the synthesized sample y under the condition of an additional input, such as a grayscale image in colorization task. The objective of the conditional GAN can be formulated as

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))] \tag{1}$$

Fixing the generator G, the discriminator D tries to maximize $\mathcal{L}_{cGAN}(G, D)$. Alternatively, for which fixing discriminator D, the generator G tries to fool the discriminator to classify fake samples as the real one, minimizing the second term of $\mathcal{L}_{cGAN}(G, D)$.

Note that the discriminator is also conditioned to the input in Equation 2. However, this is not necessary. An alternative formulation can be used:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x, z))] \tag{2}$$

Regularization on the generator can also be done on top of the cGAN optimization:

$$\mathcal{L}_{regularized_cGAN}(G, D) = \mathcal{L}_{cGAN}(G, D) + \lambda\mathbb{E}_{x,z}[l(x, G(x, z))] \tag{3}$$

for which l regularize the output of the generator. For example, the logistic loss of whether the deblurred image satisfies the blurring kernel [1], the cycle consistency loss in CycleGAN [2].

3 Attempts on cGAN models

A few models were attempted for colorization tasks in this study. They involve a specifically designed architecture. Encoder-docoder structure has widely applied to application of autoencoder for feature extraction. *U-Net* has been a popular architecture and had been applied on more areas such as segmentation[3], image style transfer, and image synthesis from scratch[4].

The models were tested on Cifar10 dataset [5]. The images were first converted to grayscale with the formula from the *CIE 1931 color space*[6]:

$$L = 0.2125R + 0.7154G + 0.0721B. \tag{4}$$

The task is to reconstruct the RGB space from these grayscale images. The conventional $L^*a^*b^*$ used for colorization was not used, instead, the relationship of brightness and color hue was left for the models to Figure out.

The images in Cifar10 dataset were rescaled to 64×64 pixels and normalized. A batch size of 64 was used. The generator and discriminator were optimized with Adam, sharing the same learning rate. The models were left running for at most 10 epochs over all the training data.

All models were trained on `Google Colab`, with a NVIDIA Tesla K80 GPU. The code can be found at the author's github: `https://github.com/cpwan/colorization-with-cGAN`

The conventional metric of colorization task involves computing the area under the curve of the cumulative error distribution over the ab space of the $L^*a^*b^*$ color space as introduced in [7]. However, the metric is not suitable in this study since one of the objective of using cGAN is to reproduce diverse samples. The metric became ill-posed since the model to be explored allows multiple colorization for each grayscale image. As an alternative, a visual exploration was given.

3.1 Baseline model

In this study, the baseline model consisted of a generator with an encoder-decoder structure and a discriminator as illustrated in Figure 1. The model employed the formulation in Equation 3. The grayscale image was fed to the encoder of the generator. An additional noise vector was concatenated with the encoded latent variable of the image in the bottleneck of the encoder-decoder. A colorized image was then generated in the decoder part of the generator from the encoded latent variable and the noise vector. The colorized image alone was then fed to the discriminator.

In the implementation of this study, the encoder-decoder structure in the generator employed Convolution-BatchNorm-LeakyReLu blocks in the encoder and Transposed_Convolution-BatchNorm-ReLu blocks in the decoder. The combination of the blocks were suggested in [8].

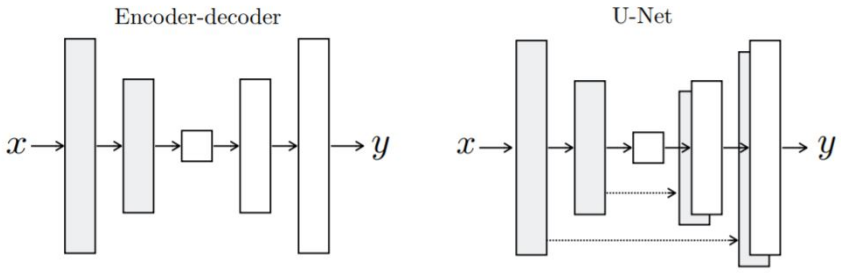


Figure 1: Choices of the generator. U-Net adds skip connection to encoder-decoder by stacking states of previous layers. Image retrieved from [4].

Figure 2: Colorization results of the baseline model at the 4th epoch. The sample generated from the colorization did not conform with the grayscale image.

The discriminator and the generator were both trained from scratch. In the first epoch, the image generated are blurred and monochromatic. As the training goes on, square patches appeared on the images generated and the images become finer in detail. However, the generated image did not conform to the original image as illustrated in Figure 2. The model reduced to a GAN treating the input image as a noise vector. The generator failed to generate good enough samples and the discriminator dominated the training, leading to mode collapse in the later epoch.

3.2 Regularization of the brightness channel

In colorization problem under the $L^*a^*b^*$ color space, two channels of the $L^*a^*b^*$ space are to be reconstructed from the L channel, which indicates the brightness. The input and output of the network for colorization problem should shares the same L channel so that the brightness is preserved. To preserve the brightness, it can be done by regularization with a reconstruction loss of the L channel. The regularization encourages the generated samples to be physically correct as suggested by PhysicsGan[1]. Similarly, a regularization can be done on the RGB image generated. In this study, the generated RGB image was converted to grayscale image with the Equation 4, then the converted synthesized samples were compared with the ground truth grayscale image, a Mean Square Error loss was then used for regularization. To control the weight of the regularization, different learning rate was attempted: the same learning rate as the generator, and a learning rate of 10 times lower. Adam optimizer was used.

In the experiments of this regularization, introducing regularization apparently provided diversity of the sample generated. The sample generated can slightly captured the form of the grayscale image, however, was not sharp enough to capture clearly the form of the grayscale image as shown in Figure 3

The model only generated a blurry colored sample instead of really performing colorization. The problem of deviating from the form of the object still persisted as in the baseline model.

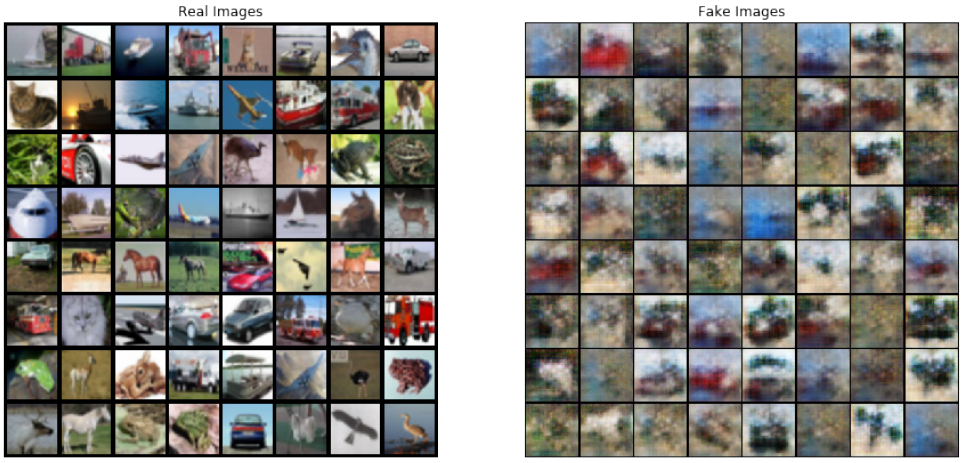


Figure 3: With regularization on the reconstructed grayscale image, the sample generated captured the form better, however, the form was blurry and the color was not consistent.

Figure 4: Colorization results of the U-Net model at the 8th epoch. The sample generated from the colorization conforms well with the grayscale image on the form of the object. However, color ambiguity problem persisted.

3.3 U-Net

The U-net models add skip-connections to the encoder-decoder structure by stacking the states in the previous layer to allow network to learn the residual part of the feature. This is illustrated in Figure 1.

In the implementation, the discriminator was kept the same as in the baseline model, feeding the colorized image alone. The generator employed a U-Net structure and the detail architecture is shown in Figure A.1 in the appendix. A noise vector was introduced in the bottleneck of the U-Net architecture.

The discriminator and the generator was again trained from scratch. After one epoch, the colorization results already conformed with the low level features (i.e. edges and corners) of the grayscale image. In the later training, the colorization results were smoother in tone graduation and the forms of the object were clearer. The resulting colorization at later epoch is shown in Figure 4. However, the color ambiguity problem existed. For example, the model was trained with images of horses in field, the model would colorized the background green even when the background should be the sky for a images capturing a horse from a lower angle.

However, the model diverged at later epochs. The generator failed to generate good enough colorization results, instead, generating unidentifiable noisy image. The model retrained itself after one to two epochs.

3.3.1 Effect of noise vector

Experiments had been done on the effect of the noise vector in the latent space at the bottleneck of the U-Net. In the first few epochs, there was no significant effect of the noise vector to the image generated from the model. In the later training, diversity among the noise vectors exhibited. These may due to that, in each epoch, the same image was fed with only one noise vector, giving only one representation for each image in the latent space. Hence, under different noise vector, the closest representation was such single representation. In contrast, in the later training, the colorization could be an interpolation of a couple of representation, giving a more diverse result.

As previously mentioned, the model diverged and retrained after a few epochs. An interesting observation was that, after these diverge-converge alternations, the diversity of the colorization increased. The results are captured in Figure 5

3.4 U-net with conditioned Discriminator

In this section, experiments were performed on a model having the same setup as in section 3.3 except that the discriminator was conditioned to the grayscale image.

In the implementation, the grayscale image was concatenated as an additional channel to the colorized images generated from the generator. The detailed architecture is shown in Figure A.1 in the appendix. Noise was added to the grayscale image before feeding to the discriminator to prevent the discriminator from being too strong.

In the experiments, the model with a conditioned discriminator produced generally produce a colorization result with sharp edges and forms. Compare to the model with unconditioned discriminator, the produced images were sharper, however, with high frequency noise. The problem of the high frequency noise was alleviated in the later epoch. Figure 6 shows the sharp but noisy colorization result in the first epoch. Figure 7 shows the sharp and clear colorization at a later epoch.

A remark is that, the model did not diverge during 10 epochs of training. Apparently, regularizing the generator with the conditioned generator is stabler than introducing reconstruction loss in the grayscale space. Further evidences are needed to verify the hypothesis.

4 Solving color ambiguity problem

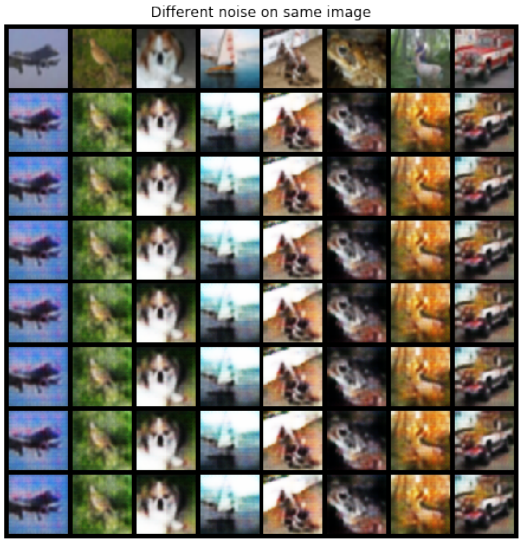
In section 3, the diversity of the cGAN output were explored. In contrast, this section discusses the approaches for solving the color ambiguity problem. The color ambiguity problem arises naturally in colorization task since multiple possible colorization can be made on the grayscale image. The "best" colorization depends heavily on the dataset fed into the model and the architecture the model employed. The data and the architecture builds together the likelihood of the colorization output with the grayscale image being the prior in a Bayesian's view. So, solving the color ambiguity problem corresponds to giving a colorization that fits the real colored image the best with the likelihood built by the dataset and the architecture. This section explores both direction for building the likelihood.

4.1 Architecture: Color Histogram

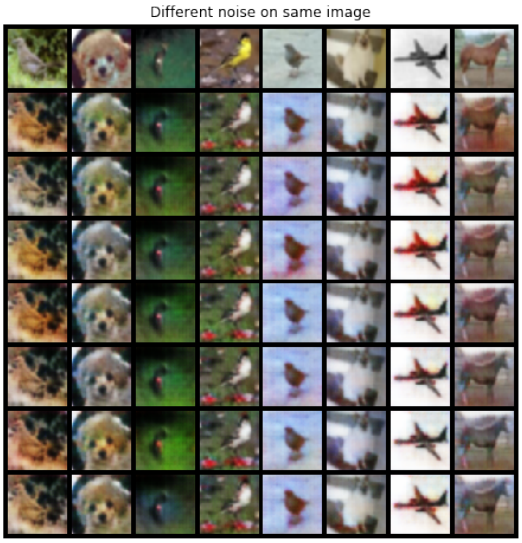
To better accommodate the span of the conditional distribution, task specific architectures were developed. In previous research, it was suggested to incorporate the color histogram of similar images into the latent variable in the bottleneck of the "encoder-decoder" styled generator. To quantify similarity of the images, classification could be incorporated in the cGAN. On top of the "encoder-decoder" styled generator, it was proposed to train an additional classifier on the classes of the object belonged to. A weighted color histogram among the classes predicted by the classifier was concatenated with the latent variable in the bottleneck of the generator. Impressive results on natural image was obtained in [7] by using the color histogram in the ab space of the $L^*a^*b^*$ color space. An interactive colorization on manga characters was archived in [9] by concatenating the classification results of manga characters and a user defined color histogram or color palette in the latent variable just before the decoder.

4.2 Data: Large scale colorization

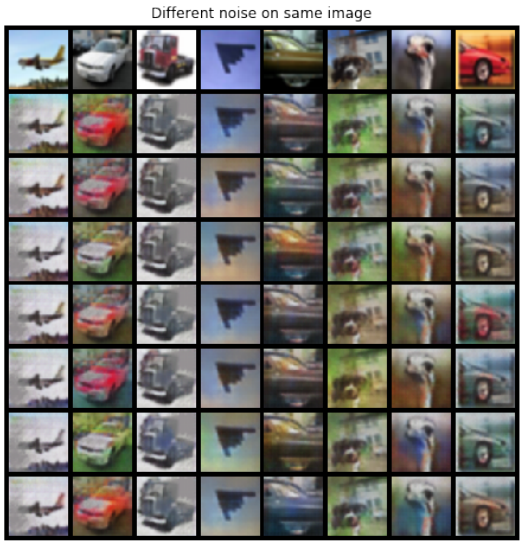
To capture the span of the conditional distribution, it might be a good idea to includes more data on training. However, there are limitations on encompassing information into neural network. With the standard GPU specification, there are 12 GB of memory for fitting the model. It is a challenge to capture larger dataset in the confined memory without sacrificing performance.



(a) In the first epoch, there are no visible difference among synthesized image under different noise vector.



(b) In the later epochs (shown in the Figure the third epoch), the diversity induced by the noise vector become more obvious.



(c) After diverge-converge alternation, the diversity induced by the noise vector become more obvious (shown in the Figure the ninth epoch).

Figure 5: Colorization results of the U-Net model with different noise vector in latent space. The first row is the original color scale image. The following 7 rows are the colorization results with different noise vector.

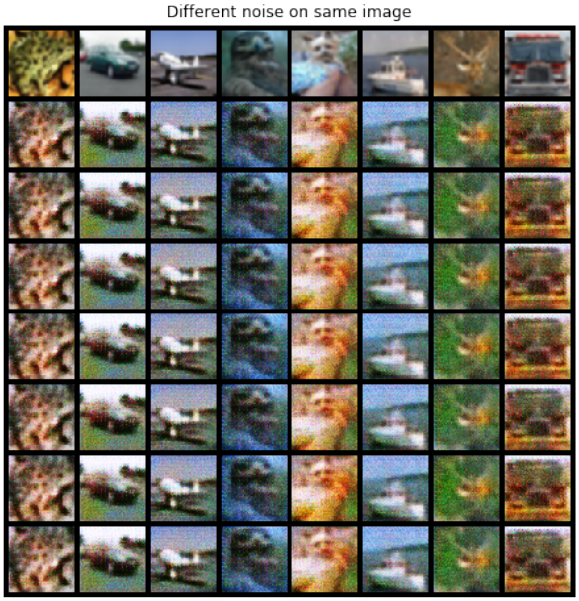


Figure 6: Results for model with the U-Net generator and a conditioned discriminator. In the first epoch, the colorized images were sharp but noisy.

Figure 7: Colorization results of the U-Net with conditioned discriminator at the 8th epoch. The sample generated from the colorization was sharper than the model with unconditioned discriminator.

The current deep generative models are in a End-to-End fashion that data are incorporated implicitly in the architecture. The latent space can be regarded as the index of the data for which synthesized samples are generated by interpolating the index in the latent space through the network. One alternative regarding very large scale neural network problem is to separate data and architecture, and proceed in a Neural Turing Machine [10] or Meta Learning [11] fashion. Training an architecture to retrieve similar images and retrain the cGAN with the images for color consistency and efficiency of training.

A proposed pipeline for future research on large scale colorization is first to learn a representation with a state of art architecture, then the representation is to be used in image retrieval to collect image of similar context from a large dataset, such as ImageNet, the images retrieved is then fed to retrain a conditional GAN for colorization task.

The key step is to retrain the conditional GAN from a new set of relevant images. Since a deep generative model that learns an auto-regressive prior has been shown to be beneficial to a better generalization. Various approach that learnt an auto-regressive prior distribution can be explored, such as VGrow[12], VQ VAE [13]. To ensure efficiency, the models may be modified in a Neural Turing Machine fashion that separates data and algorithm. Recent advances on Neural Turing Machine [10] and formulation like MANN [11] can be explored.

5 Conclusion

In this study, two major choices of generator for colorization tasks were re-implemented and their effects were explored. U-net captures the low level feature such as edges and forms very well in colorization tasks. Regularization of cGAN was also explored through regularizing the reconstruction loss of the generator, and employing an conditioned discriminator for which the grayscale image was conditioned to. The conditioned discriminator may be helpful for a stable training. Further verification of the visual observation need to be done on other dataset and a different random seed initialization.

The color ambiguity problem was briefly explored without concrete implementation. Previous research suggested that incorporating color histogram into latent space could be beneficial. The development of meta learning suggested future direction to very large scale colorization. Further exploration on the large scale colorization through meta learning and auto-regressive learning of noise vector would be worthy.

References

[1] J. Pan, Y. Liu, J. Dong, J. Zhang, J. Ren, J. Tang, Y.-W. Tai, and M.-H. Yang, "Physics-based generative adversarial models for image restoration and beyond," *arXiv preprint arXiv:1808.00605*, 2018.

[2] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[5] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," tech. rep., Citeseer, 2009.

[6] C. CIE, "International commission on illumination proceedings, 1931," *Cambridge University, Cambridge*, 1932.

[7] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European conference on computer vision*, pp. 649–666, Springer, 2016.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[9] C. Furusawa, K. Hiroshiba, K. Ogaki, and Y. Odagiri, "Comicolorization: semi-automatic manga colorization," in *SIGGRAPH Asia 2017 Technical Briefs*, p. 12, ACM, 2017.

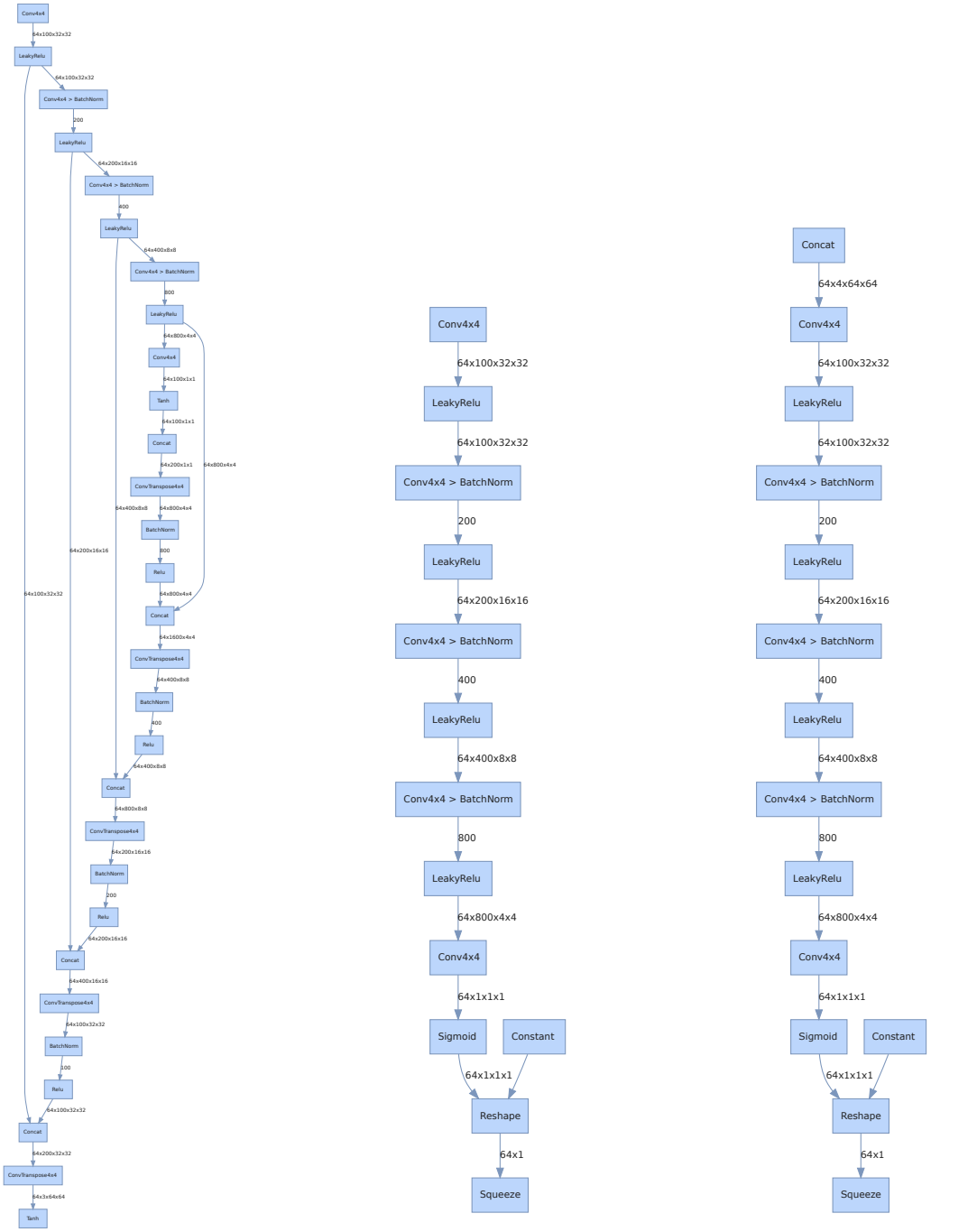
[10] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[11] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," *arXiv preprint arXiv:1605.06065*, 2016.

[12] G. Yuan, J. Yuling, W. Yang, W. Yao, Y. Can, and Z. Shunkang, "Deep generative learning via variational gradient flow," *arXiv preprint arXiv:1901.08469*, 2019.

[13] A. van den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems*, pp. 6306–6315, 2017.

A Architecture of the cGAN with U-Net Generator and Discriminators



(a) U-Net Structure of the Generator (b) Unconditioned Discriminator (c) Conditioned Discriminator

Figure A.1: Architecture of the cGAN with U-Net Generator and Discriminators