Enhancing CGANs through Integrating Mutual Information

Qin Su

Problem Description

Generative Adversarial Networks (GANs) utilize two networks: a generator and a discriminator, in a minimax game to learn to generate new data samples that are indistinguishable from the real data [1]. Conditional GANs (CGANs) extend GANs by conditioning the generation process on additional information, improving control and relevance of generated samples [2]. Despite their successes, CGANs face challenges such as mode collapse and instability during training. Information Maximizing GANs (InfoGAN) is a generative adversarial network that also maximizes the mutual information between a small subset of the latent variables and the observation [3]. Incorporating mutual information can theoretically ensure that all modes of the data distribution are captured, enhancing diversity and fidelity of the generated samples [4]. This report explores enhancing CGANs by integrating mutual information to improve output diversity and quality.

Background and Motivation

GANs introduced the principle of adversarial training, involving a dynamic where two models, the generator (G) and the discriminator (D), are pitted against each other in a game-theoretic framework. Specifically, the discriminator's job is to accurately differentiate between real and fake data, whereas the generator strives to produce data so convincing that it becomes indistinguishable from real data. This interaction is succinctly captured by the GAN objective function:

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

where p_{data} is the real data distribution, and $p_z(z)$ is a prior over input noise variables. This groundbreaking strategy not only encourages the generation of new data that mirrors the original training dataset but also enhances its quality through continuous iterative training [1].

Building on the GAN model, CGANs were developed to generate data conditioned on additional information such as labels or features. This advancement enables targeted data generation, crucial for tasks requiring specified outcomes. The CGAN framework modifies the GAN objective to incorporate conditionality:

$$\underset{G_C}{min} \quad \underset{D_C}{max} V_C(D_C,G_C) = E_{x \sim p_{data}(x)}[\log D(x|y)] + E_{z \sim p_z(z)}\left[\log\left(1 - D\big(G(z|y)\big)\right)\right]$$

Here, y represents the conditioning variable, allowing for more directed and versatile data generation. This extension allows CGANs to direct the data generation process, highlighting the importance of efficiently processing conditional inputs, notably through convolutional neural networks [2]. CGANs have shown promise in tasks like image synthesis, where the model conditions on labels or attributes, as shown in Fig. 2.

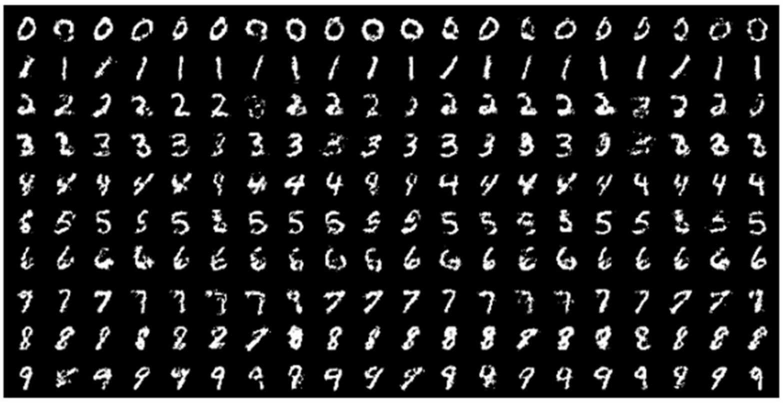
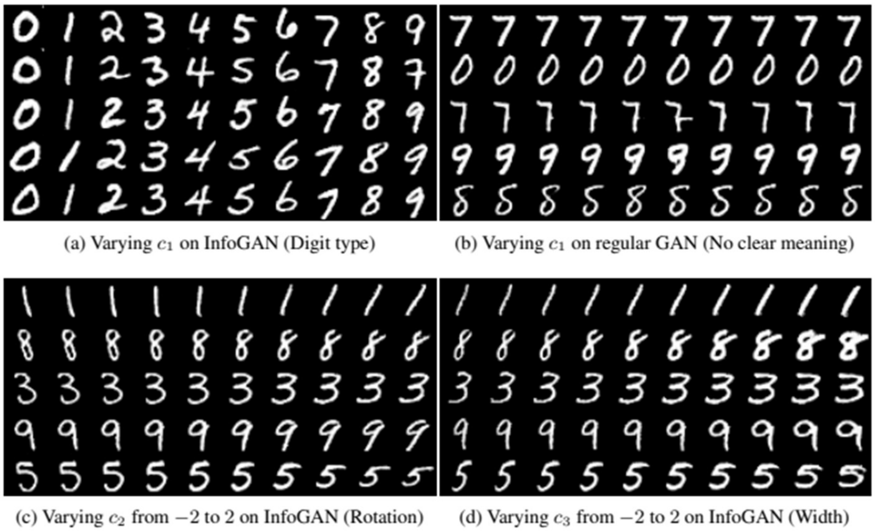


Fig. 1: Generated MNIST digits, each row conditioned on one label [2].

However, the generation quality often varies significantly depending on the complexity of the conditioning data and the architecture used. Mutual information, a measure of the amount of information one random variable contains about another, can be leveraged to optimize the information flow between the conditioning labels and the generated outputs, as shown in Fig. 2.



(a) Varying c_1 on InfoGAN (Digit type)

(b) Varying c_1 on regular GAN (No clear meaning)

(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Fig. 2: Manipulating Latent Codes on MNIST: Each figure demonstrates variations in a specific latent code from left to right, with other codes and noise held constant [3].

Different rows showcase various random samples of these fixed elements:

(a): Each column displays five samples from a single category in c1, with a row depicting generated images across 10 categories in c1. Here, c1 primarily correlates with distinct digit types.

(b): Variation in c1 on a GAN without information regularization leads to non-interpretable outcomes.

(c): A low value of c2 causes the digits to lean to the left, whereas a high value of c2 makes them tilt to the right.
(d): c3 adjusts the digit width smoothly.
Panel (a) is reordered for clarity as the categorical code is not inherently ordered. Unlike CGANs, the generated digits are organized with labels that match the conditions.

This project is motivated by the hypothesis that enhancing CGANs with mutual information can lead to more stable and high-quality generation, particularly in multimodal data settings. Simultaneously, integrating CGANs with InfoGAN could allow the original InfoGAN to generate ordered digits not by categorical code but through conditional labels.

Methods
The approach involves two key methodological advancements:

1. Architectural Enhancements: Elements from Information Maximizing GANs (InfoGAN) are integrated into the CGAN architecture. This involves adapting the generator of the CGAN to maximize mutual information between a subset of latent codes and the generated outputs. Therefore, we propose to solve the following information-regularized minimax game [3] [4]:
$$\underset{G_C}{min}\ \underset{D_C}{max} V_{C-Inf}\ (D_C, G_C) = V_C(D_C, G_C) + \lambda I(c; G(z|y, c)$$
Utilizing the architecture of InfoGAN, which reveals key structured semantic features from the latent space, is designed to enhance both the control and diversity of the data generated.

2. Mutual Information Maximization: In practice, the mutual information term $I(c; G(z|y, c)$ is hard to maximize directly as it requires access to the posterior $P(c|x)$. Fortunately, we can obtain a lower bound of it by defining an auxiliary distribution $Q(c|x)$ to approximate $P(c|x)$ and we can define a variational lower bound, $L_I(G_C|Q)$, of the mutual information $I(c; G(z|y, c)$ [3].

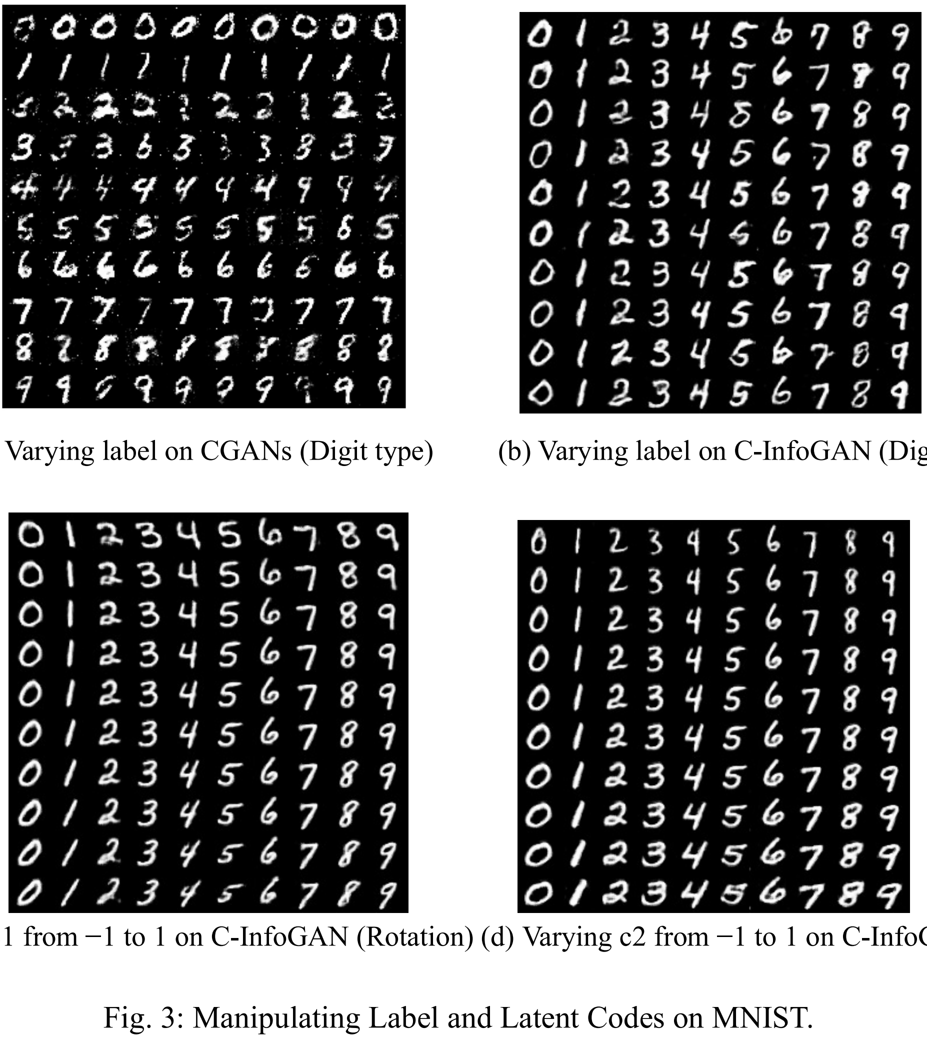
Hence, CGANs with mutual information maximization (C-InfoGAN) is defined as the following minimax game with a variational regularization of mutual information and a hyperparameter λ [3]:
$$\underset{G_C, Q}{min}\ \underset{D_C}{max} V_{C-InfoGA}\ (D_C, G_C) = V_C(D_C, G_C) + \lambda L_I(G_C|Q)$$

Experimental Settings and Results
Datasets: The study utilized the MNIST dataset for digit generation.

Training Details: Since GAN is known to be difficult to train, we design our experiments based on existing techniques introduced by DC-GAN [5], which are enough to stabilize C-InfoGAN training and we did not have to introduce new trick. The training configuration included the use of the Adam optimizer, with specific batch sizes and learning rates adjusted over the course of training [6].

The main aim of the experiments is to see if C-InfoGAN can not only make the generated images more diverse and clearer, but also can learn disentangled and interpretable representations. This involves using the generator to change one latent factor at a time to check if this change affects only one aspect of the image.

To disentangle digit shape from styles on MNIST, the experiments choose class labels encoded as one-hot vectors and two continuous codes that can capture variations that are continuous in nature: c1, c2 ∼ Uniform (−1, 1). In Fig. 3, we show that each row conditioned on one label in (a), each column conditioned on one label in (b) and each figure demonstrates variations in a specific latent code from left to right, with other codes and noise held constant in (c) and (d).



(a) Varying label on CGANs (Digit type)

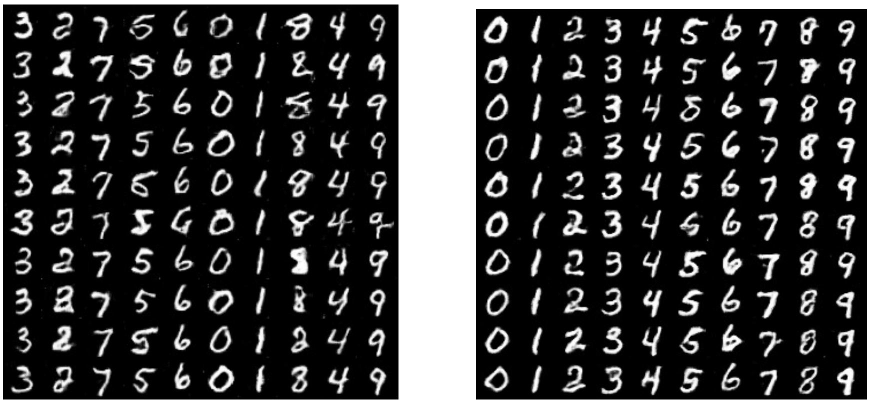
(b) Varying label on C-InfoGAN (Digit type)

(c) Varying c1 from −1 to 1 on C-InfoGAN (Rotation)

(d) Varying c2 from −1 to 1 on C-InfoGAN (Width)

Fig. 3: Manipulating Label and Latent Codes on MNIST.

The second goal is to confirm that C-InfoGAN could allow the original InfoGAN to generate ordered digits not by categorical code but through conditional labels, as shown in Fig. 4.



(a) Varying categorical code on InfoGAN (Digit type)   (b) Varying label on C-InfoGAN (Digit type)

Fig. 4: Manipulating Categorical Codes and Label on MNIST.

Conclusion

The experiments demonstrate that integrating mutual information leads to observable improvements in the diversity and accuracy of the generated images across different conditions. Challenges such as computational complexity and the need for fine-tuning the balance between adversarial and information-theoretic components in the loss function were noted. Future work could explore further architecture optimizations and the application of these methods to other types of data beyond images.

References

[1] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). "Generative Adversarial Nets." Neural Information Processing Systems (NIPS).

[2] Mirza, M., & Osindero, S. (2014). "Conditional Generative Adversarial Nets." arXiv preprint arXiv:1411.1784.

[3] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, “InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets,” in Advances in Neural Information Processing Systems, 2016.

[4] Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., & Hjelm, R. D. (2018). "Mutual Information Neural Estimation." International Conference on Machine Learning (ICML).

[5] Radford, A., Metz, L., & Chintala, S. (2015). "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks." arXiv preprint arXiv:1511.06434.

[6] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training GANs,” in Advances in Neural Information Processing Systems, 2016.