Introduction

Generative Adversarial Networks (GANs) have been successful in image generation and semi-supervised learning (SSL). However, existing GANs for SSL face two critical challenges:

1. **Conflict in Roles**:
   - The discriminator in traditional GANs must identify fake samples and classify real samples into correct labels. These tasks are incompatible and often conflict.
   - For example, a discriminator trying to classify a fake sample may fail to identify its label correctly.
2. **Lack of Semantic Control**:
   - Existing generators cannot explicitly control the class of the generated samples.
   - Label information is ignored during generation, leading to the inability to disentangle class-specific features or styles.

Proposed Solution: Triple-GAN

The **Triple-GAN** framework resolves these issues by introducing a three-player game:

1. **Generator (G)**: Models the conditional distribution $p(x|y)p(x \mid y)p(x|y)$, generating data samples based on labels.
2. **Classifier (C)**: Models $p(y|x)p(y \mid x)p(y|x)$, predicting labels given input data.
3. **Discriminator (D)**: Distinguishes whether a pair $(x,y)(x, y)(x,y)$ is from the real data distribution or generated by GGG or CCC.

Key Contributions

1. **Novel Framework**: Triple-GAN separates roles between generator, classifier, and discriminator, ensuring each focuses on its primary task.
2. **Compatible Utilities**: Carefully designed loss functions ensure that the classifier and generator converge to the true data distribution independently.
3. **Improved Performance**: Empirically, Triple-GAN achieves state-of-the-art results in SSL while generating semantically meaningful and class-conditional samples.

Methodology

Factorizing the Joint Distribution

The true joint distribution $p(x,y)p(x, y)p(x,y)$ can be factorized in two ways:

1. p(x,y)=p(x)p(y|x)p(x, y) = p(x)p(y | x)p(x,y)=p(x)p(y|x), where p(y|x)p(y | x)p(y|x) is the conditional probability modeled by the classifier CCC.
2. p(x,y)=p(y)p(x|y)p(x, y) = p(y)p(x | y)p(x,y)=p(y)p(x|y), where p(x|y)p(x | y)p(x|y) is the conditional probability modeled by the generator GGG.

Three Players

Triple-GAN defines a three-player game:

1. **Generator GGG**:
   - Takes a class yyy and noise zzz as input and generates samples x∼G(y,z)x \sim G(y, z)x∼G(y,z).
   - Goal: Model p(x|y)p(x | y)p(x|y), producing realistic samples that match the data distribution for the given label.
2. **Classifier CCC**:
   - Takes input data xxx and predicts its label yyy.
   - Goal: Model p(y|x)p(y | x)p(y|x), accurately predicting labels for both real and generated samples.
3. **Discriminator DDD**:
   - Distinguishes between real and generated (x,y)(x, y)(x,y) pairs:
     - Real pairs come from the true data distribution p(x,y)p(x, y)p(x,y).
     - Fake pairs come from either GGG (pseudo-data) or CCC (pseudo-labels).

Loss Functions (Utilities)

The loss functions for each player are designed to achieve equilibrium, where GGG and CCC approximate the data distribution p(x,y)p(x, y)p(x,y).

1. **Discriminator Loss**:

U(D)=E(x,y)∼p(x,y)[log⁡D(x,y)]+αE(x,y)∼pC(x,y)[log⁡(1−D(x,y))]+(1−α)E(x,y)∼pG(x,y)[log⁡(1−D(x,y))],U(D) = \mathbb{E}_{(x, y) \sim p(x, y)}[\log D(x, y)] + \alpha \mathbb{E}_{(x, y) \sim p_C(x, y)}[\log(1 - D(x, y))] + (1 - \alpha) \mathbb{E}_{(x, y) \sim p_G(x, y)}[\log(1 - D(x, y))],U(D)=E(x,y)∼p(x,y)[logD(x,y)]+αE(x,y)∼pC(x,y)[log(1−D(x,y))]+(1−α)E(x,y)∼pG(x,y)[log(1−D(x,y))],

where:

- pC(x,y)=p(x)pC(y|x)p_C(x, y) = p(x)p_C(y | x)pC(x,y)=p(x)pC(y|x) represents pseudo-labels generated by CCC.
- pG(x,y)=p(y)pG(x|y)p_G(x, y) = p(y)p_G(x | y)pG(x,y)=p(y)pG(x|y) represents samples generated by GGG.

- α balances the importance of classification and generation tasks.

2. **Generator Loss**: The generator G minimizes the adversarial loss:

$$L_G = \mathbb{E}_{(x, y) \sim p_G(x, y)}[\log(1 - D(x, y))].$$

3. **Classifier Loss**: The classifier C minimizes two losses:
   - A **supervised loss** on labeled data (cross-entropy loss): $L_{\text{sup}} = \mathbb{E}_{(x, y) \sim p(x, y)}[-\log p_C(y|x)].$
   - An **adversarial loss** for pseudo-labeled data: $L_{\text{adv}} = \mathbb{E}_{(x, y) \sim p_C(x, y)}[\log(1 - D(x, y))].$

Training Algorithm

Triple-GAN is trained iteratively:

1. Sample batches of real labeled data, pseudo-labeled data from C, and generated data from G.
2. Update D using the discriminator loss $U(D)$.
3. Update C using the supervised loss L_{sup} and adversarial loss L_{adv}.
4. Update G using its adversarial loss L_G.

Theoretical Guarantees

1. **Equilibrium**: The loss functions are designed so that:
   - C approximates $p(y|x)$, while G approximates $p(x|y)$.
   - At equilibrium, both the generator and classifier converge to the true data distribution.
2. **Non-Competing Roles**: Unlike two-player GANs, where the generator and discriminator compete, Triple-GAN ensures cooperation between C and G.
3. **Pseudo Discriminative Loss**: Additional regularization ensures that C benefits from pseudo-labeled data, improving its performance.

Experiments

Datasets

The authors evaluate Triple-GAN on three datasets:

1. **MNIST** (handwritten digits): 50,000 training samples, 10,000 test samples.
2. **SVHN** (street view house numbers): 73,257 training samples, 26,032 test samples.
3. **CIFAR-10** (natural images): 50,000 training samples, 10,000 test samples across 10 classes.

Classification Results

1. Triple-GAN achieves state-of-the-art error rates for semi-supervised classification:
   - MNIST with 100 labeled samples: **0.91% error**.
   - SVHN with 1,000 labeled samples: **5.77% error**.
   - CIFAR-10 with 4,000 labeled samples: **16.99% error**.
2. Outperforms competitors like Improved-GAN and CatGAN.

Generation Results

1. **Image Quality**:
   - Triple-GAN generates clearer and more diverse images than baselines.
   - On CIFAR-10, the **inception score** improves significantly (from 3.87 to 5.08).
2. **Class-Conditional Generation**:
   - Generates samples that match the label semantics, even with limited supervision.

Latent Space Interpolation

- Demonstrates smooth transitions between images of different classes in the latent space, preserving semantic consistency.

Conclusion

Triple-GAN offers a unified framework that achieves:

1. **State-of-the-art classification performance** in semi-supervised settings.
2. **Semantically controlled generation**, enabling class-specific image synthesis.
3. **Disentanglement of class and style**, allowing better latent space interpolation.

By addressing the fundamental issues of two-player GANs, Triple-GAN advances the field of semi-supervised learning and generative modeling.