Self-Supervised GAN: Analysis and Improvement With Multi-Class Minimax Game

Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen,
Linxiao Yang, Ngai-Man Cheung

NIPS 2019

OUTLINE

- Background
- Proposed Method
- Experimental Results
- Conclusion

OUTLINE

- Background
- Proposed Method
- Experimental Results
- Conclusion

BACKGROUND

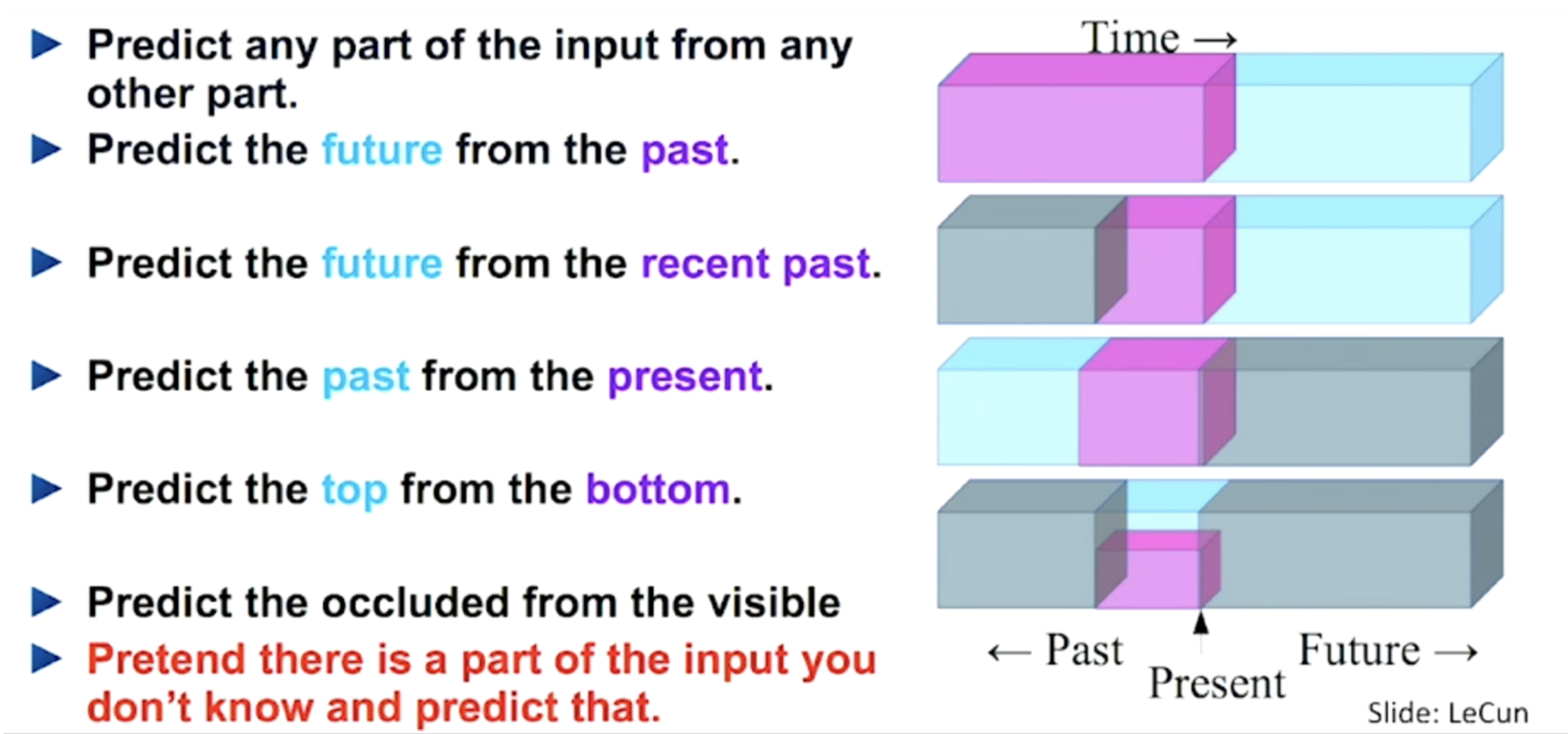
- Self-Supervised Learning
- Discriminator Forgetting

BACKGROUND

- Self-Supervised Learning
- Discriminator Forgetting

BACKGROUND

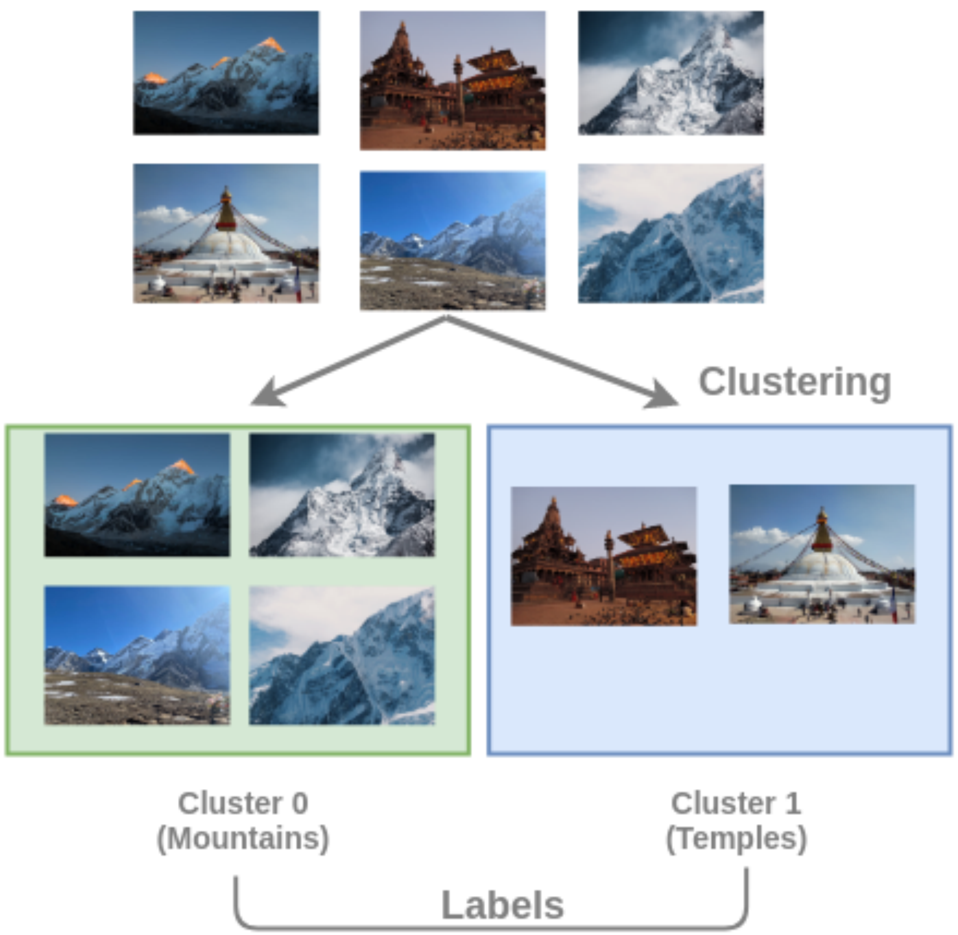
- Self-Supervised Learning
- Unlabeled data → Pretext
- Get supervision from the data itself.



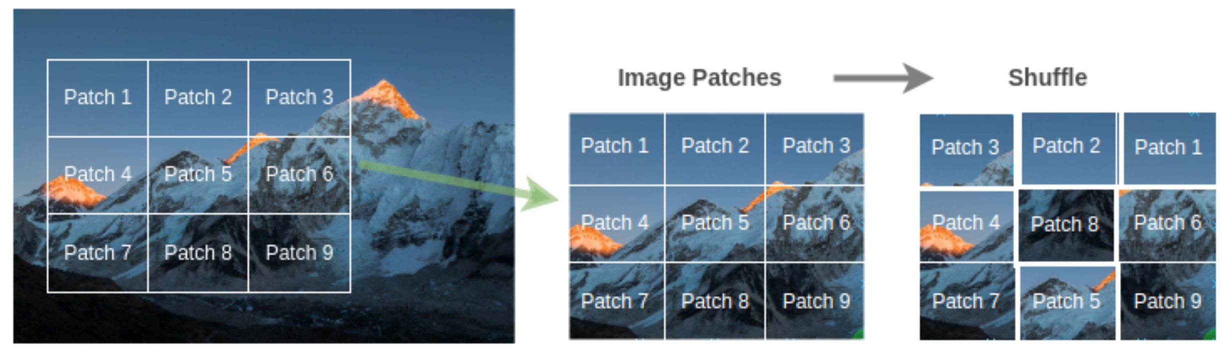
BACKGROUND

- Self-Supervised Learning

Clustering

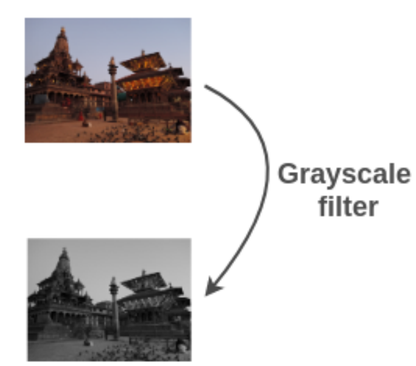


Jigsaw



Inpainting

Colorization

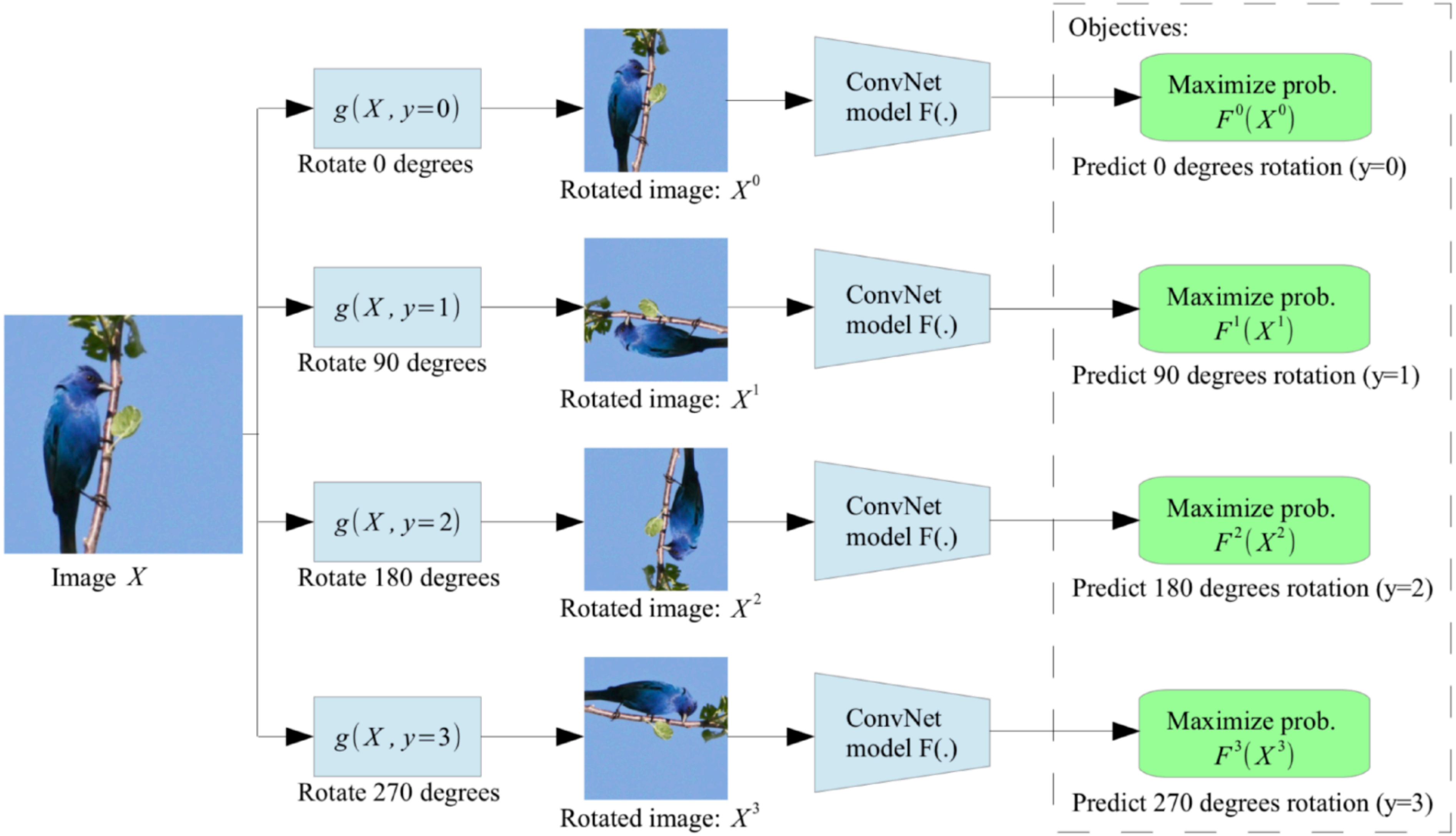


Blog: https://lilianweng.github.io/lil-log/2019/11/10/self-supervised-learning.html

BACKGROUND

➤ Self-Supervised Learning

- Rotation



[1] Unsupervised Representation Learning by Predicting Image Rotations (ICLR16)

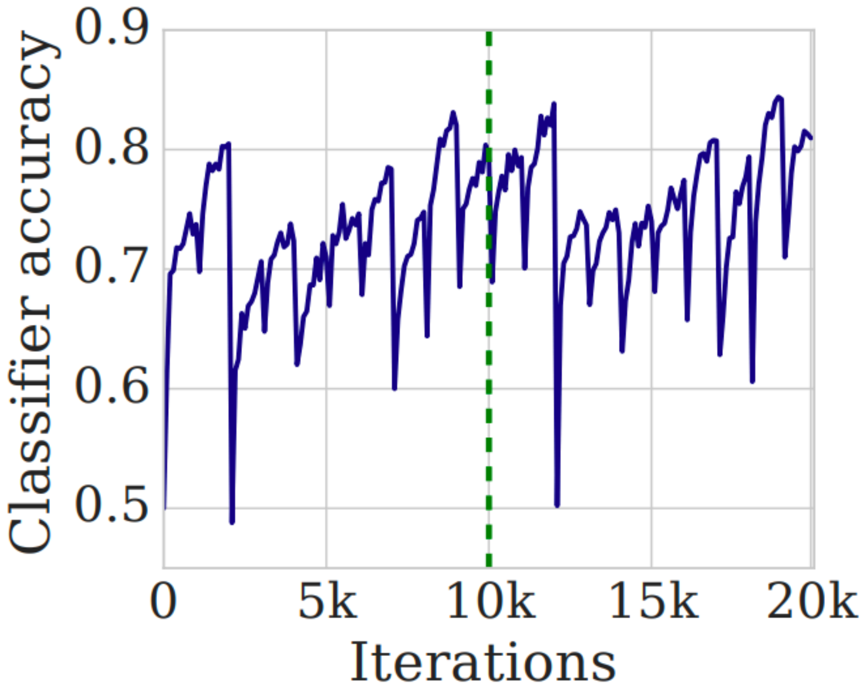
BACKGROUND

- Self-Supervised Learning
- Discriminator Forgetting

BACKGROUND

➤ Image Classifier Forgetting

- Experiment:
  - The task of “1 v.s. all” classification.
  - Each class, train 1k iteration.
  - Then move to the next class.
- Result:
  - Each time the task switches, accuracy drops.
  - After 10k iterations, the cycle of tasks repeats.



(a) Regular training.

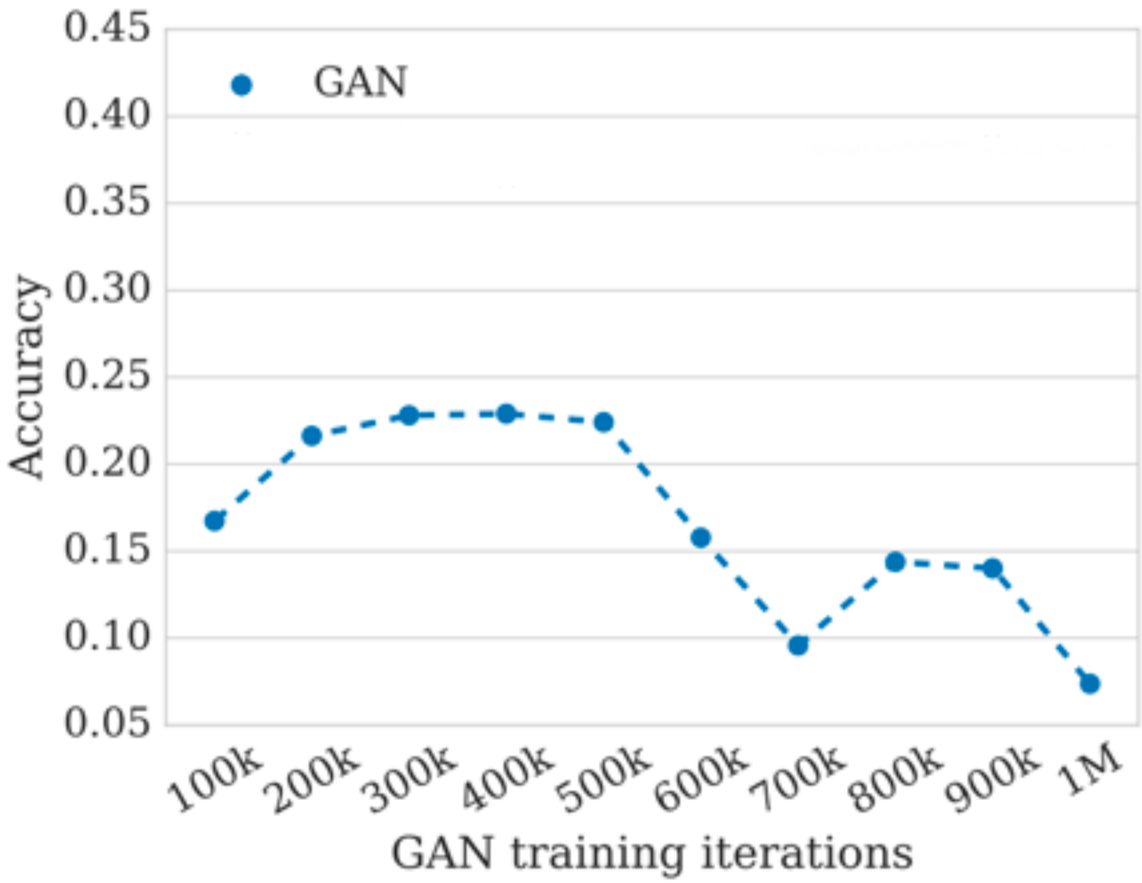
BACKGROUND

- ➤ Image Classifier Forgetting
- Conclusion:
  - Classifier fail to learn generalizable representations in a non-stationary environment.

BACKGROUND

➤ Discriminator Forgetting

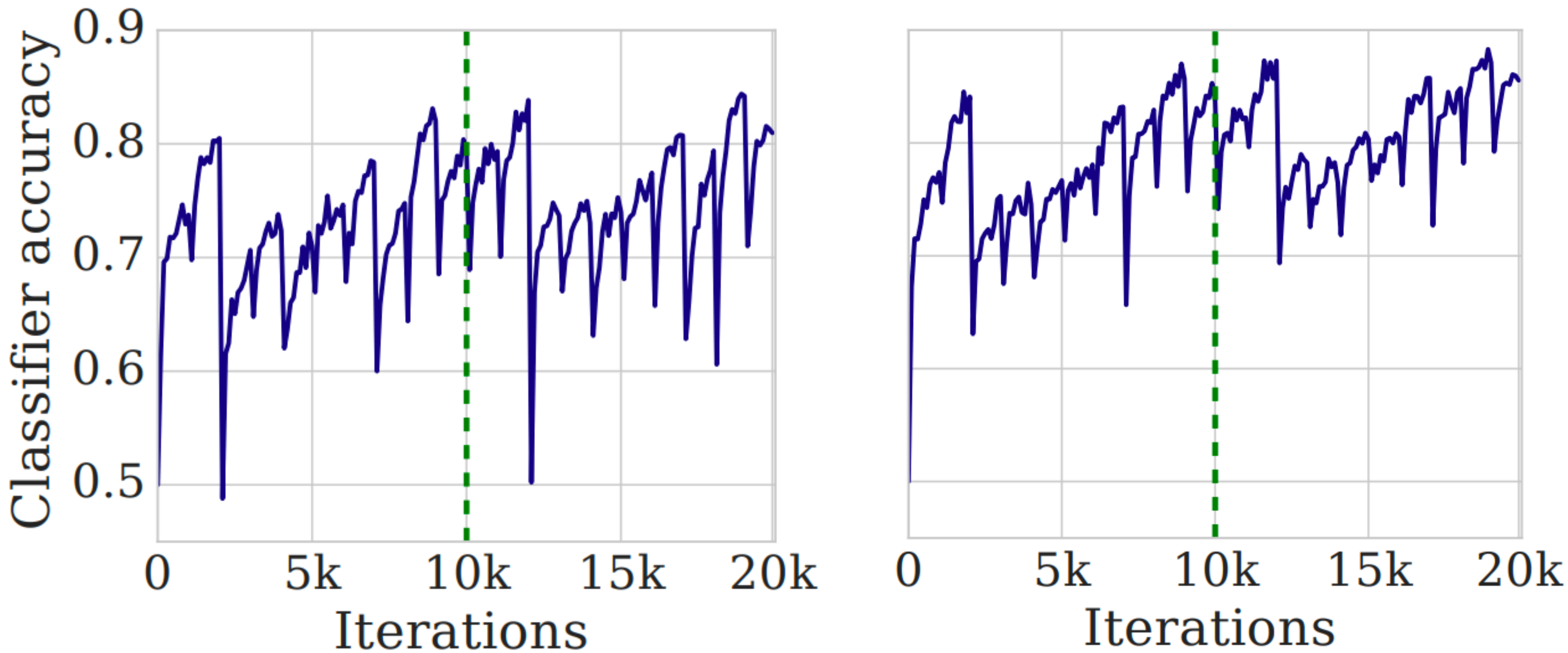
- Experiment:
  - Classifier trained with the final layer of a discriminator.



BACKGROUND

➤ Solution: self-supervised GAN (CVPR19)

- Image Classifier+ Self-Supervision (Rotation)



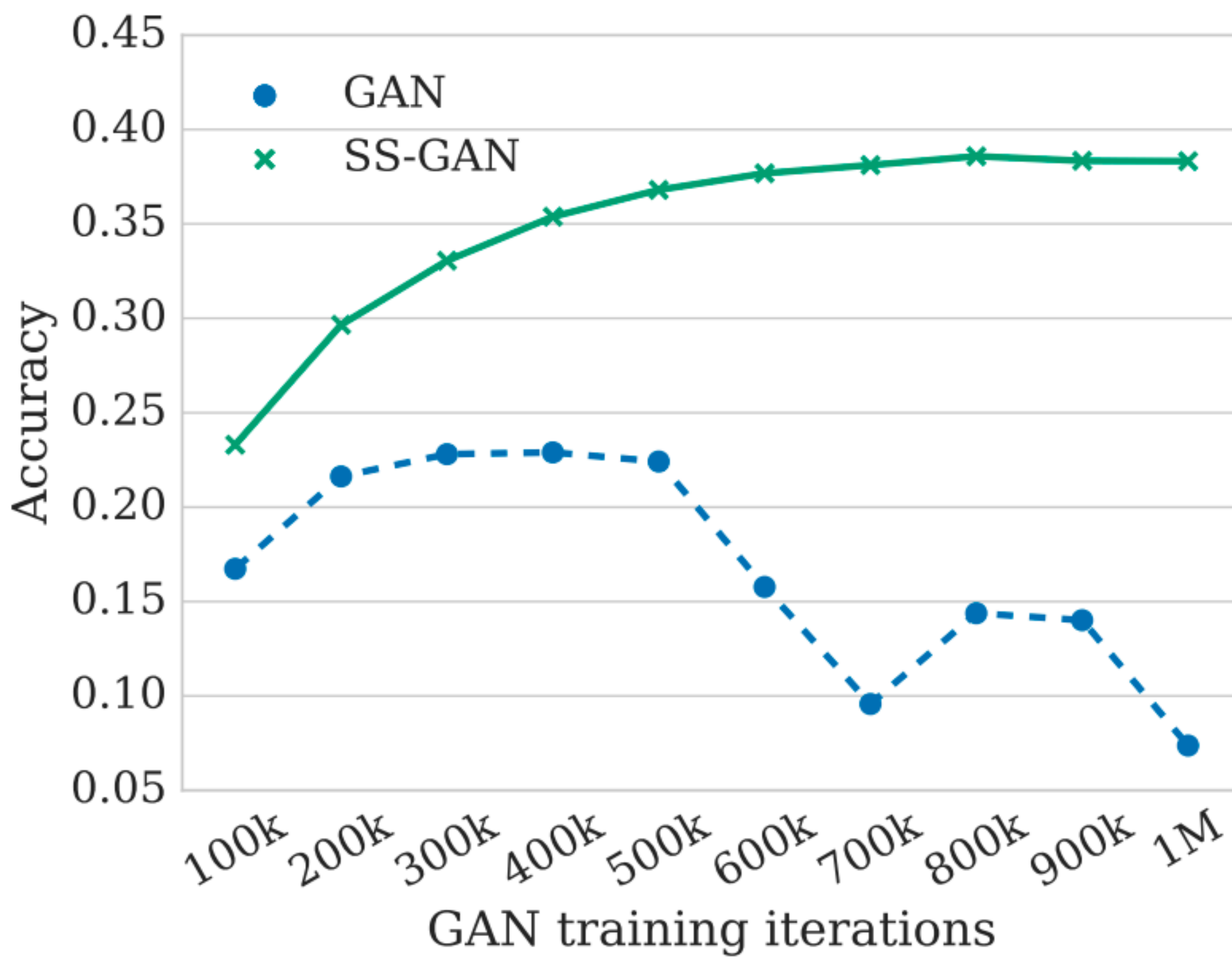
(a) Regular training. (b) With self-supervision.

[1] Self-Supervised GANs via Auxiliary Rotation Loss (CVPR19)

BACKGROUND

➤ Solution: self-supervised GAN (CVPR19)

- GAN+ Self-Supervision (Rotation)



BACKGROUND

➤ Solution: self-supervised GAN (CVPR19)

- Method

$$L_G = -V(G, D) - \alpha \mathbb{E}_{\boldsymbol{x} \sim P_G} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r) \right],$$

$$L_D = V(G, D) - \beta \mathbb{E}_{\boldsymbol{x} \sim P_{\text{data}}} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r) \right],$$

BACKGROUND

- ➤ Solution: self-supervised GAN (CVPR19)
- Method

$$L_G = -V(G, D) - \alpha \mathbb{E}_{\boldsymbol{x} \sim P_G} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r) \right],$$

$$L_D = V(G, D) - \beta \mathbb{E}_{\boldsymbol{x} \sim P_{\text{data}}} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r) \right],$$

Loss for GAN

BACKGROUND

➤ Solution: self-supervised GAN (CVPR19)

- Method

$$L_G = -V(G, D) - \alpha \mathbb{E}_{\boldsymbol{x} \sim P_G} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r) \right],$$
$$L_D = V(G, D) - \beta \mathbb{E}_{\boldsymbol{x} \sim P_{\text{data}}} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r) \right],$$

Loss for self-supervised learning

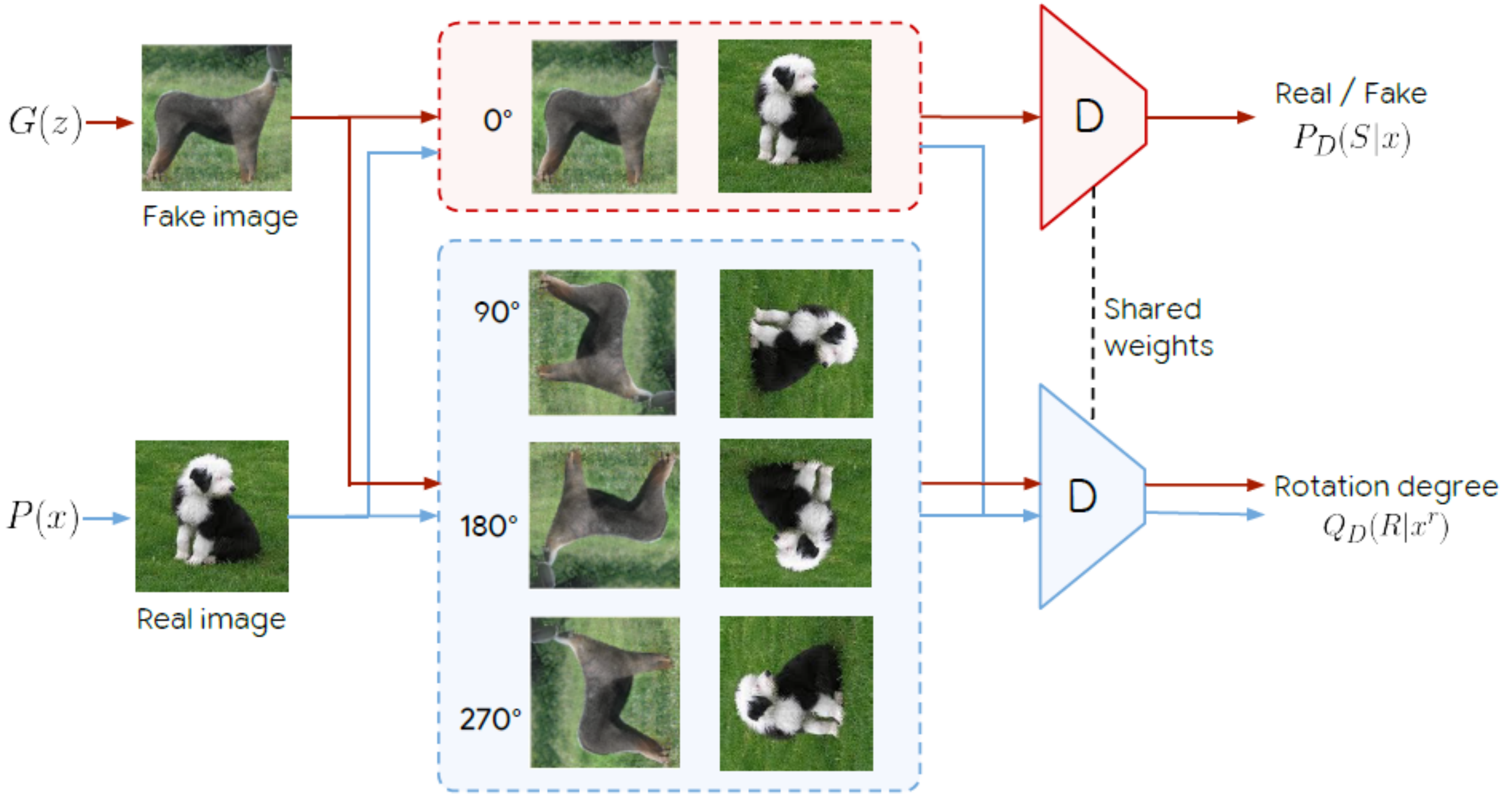
$\boldsymbol{x}^r$: Image $\boldsymbol{x}$ rotated by r degrees

$$\mathcal{R} = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$$

BACKGROUND

➤ Solution: self-supervised GAN (CVPR19)

- Method



BACKGROUND

➤ Solution: self-supervised GAN (CVPR19)

- Method

$$L_G = -V(G, D) - \alpha \mathbb{E}_{\boldsymbol{x} \sim P_G} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r) \right],$$
$$L_D = V(G, D) - \beta \mathbb{E}_{\boldsymbol{x} \sim P_{\text{data}}} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r) \right],$$

- NetD:
  - Judge true/false on unrotated image.
  - Judge rotation angle on rotated images.

BACKGROUND

- Solution: self-supervised GAN (CVPR19)
- Method

$$L_G = -V(G, D) - \alpha \mathbb{E}_{\boldsymbol{x} \sim P_G} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r) \right],$$

$$L_D = V(G, D) - \beta \mathbb{E}_{\boldsymbol{x} \sim P_{\text{data}}} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r) \right],$$

- NetG and netD:
  - Collaborative on the rotation task.
  - Adversarial on the GAN task.

OUTLINE

- Background
- Proposed Method
- Experimental Results
- Conclusion

PROPOSED METHOD

- Problem of Auxiliary Rotation + GAN (CVPR19)

$$\max_{D,C} \mathcal{V}(D,C,G) = \mathcal{V}(D,G) + \lambda_d \underbrace{\left(\mathbb{E}_{\mathbf{x}\sim P_d^T} \mathbb{E}_{T_k \sim \mathcal{T}} \log \Big(C_k(\mathbf{x}) \Big) \right)}_{\Psi(C)}$$

$$\min_{G} \mathcal{V}(D,C,G) = \mathcal{V}(D,G) - \lambda_g \underbrace{\left(\mathbb{E}_{\mathbf{x}\sim P_g^T} \mathbb{E}_{T_k \sim \mathcal{T}} \log \Big(C_k(\mathbf{x}) \Big) \right)}_{\Phi(G,C)}$$

PROPOSED METHOD

➤ Problem of Auxiliary Rotation + GAN (CVPR19)

- C^*: the optimal classifier for self-supervised task

$$C_k^*(\mathbf{x}) = \frac{p_d^{T_k}(\mathbf{x})}{\sum_{k=1}^{K} p_d^{T_k}(\mathbf{x})}$$

$p_d^{T_k}(\mathbf{x})$: the probability of data sample

- $\min_G \mathcal{V}(D, C, G)$ equals to maximizing:

$$\Phi(G, C^*) = \frac{1}{K} \sum_{k=1}^{K} \left[\mathbb{E}_{\mathbf{x} \sim P_g^{T_k}} \log \left(\frac{p_d^{T_k}(\mathbf{x})}{\sum_{k=1}^{K} p_d^{T_k}(\mathbf{x})} \right) \right] = \frac{1}{K} \sum_{k=1}^{K} \mathcal{V}_{\Phi}^{T_k}(\mathbf{x})$$

PROPOSED METHOD

➤ Problem of Auxiliary Rotation + GAN (CVPR19)

- $\min_{G} \mathcal{V}(D, C, G)$ equals to maximizing:

$$\Phi(G, C^*) = \frac{1}{K} \sum_{k=1}^{K} \left[\mathbb{E}_{\mathbf{x} \sim P_g^{T_k}} \log \left(\frac{p_d^{T_k}(\mathbf{x})}{\sum_{k=1}^{K} p_d^{T_k}(\mathbf{x})} \right) \right] = \frac{1}{K} \sum_{k=1}^{K} \mathcal{V}_{\Phi}^{T_k}(\mathbf{x})$$

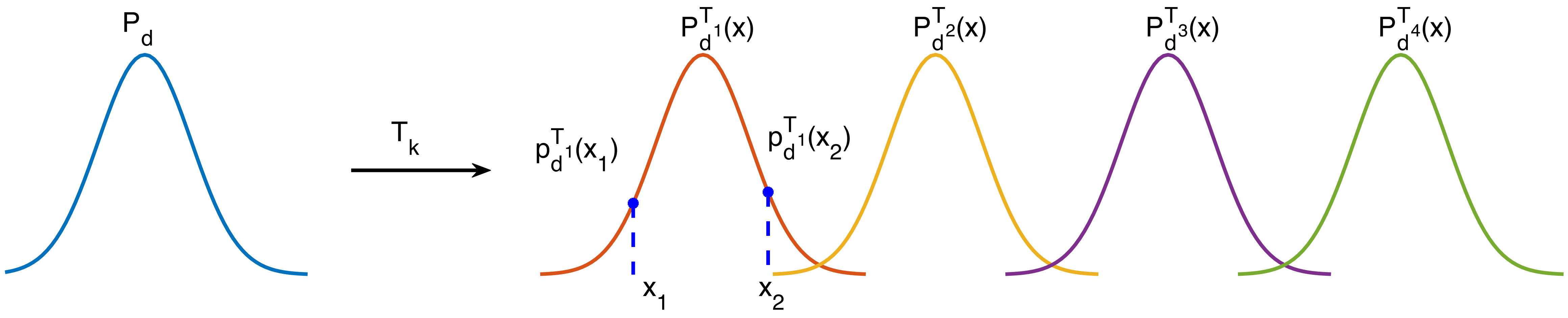
- A trick for netG to achieve the maximum is:

$$p_d^{T_1}(\mathbf{x}) \neq 0 \text{ and } p_d^{T_j}(\mathbf{x}) = 0, j \neq 1$$

- A “loophole”, without actually learning the data distribution.

PROPOSED METHOD

➤ Problem of Auxiliary Rotation + GAN (CVPR19)

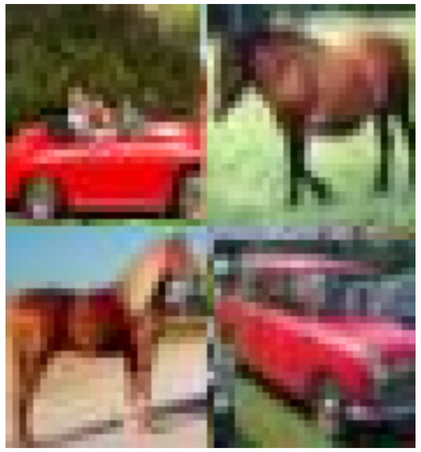


- The true distribution: P_d
- What netG generates: x_1, x_1, x_1 …

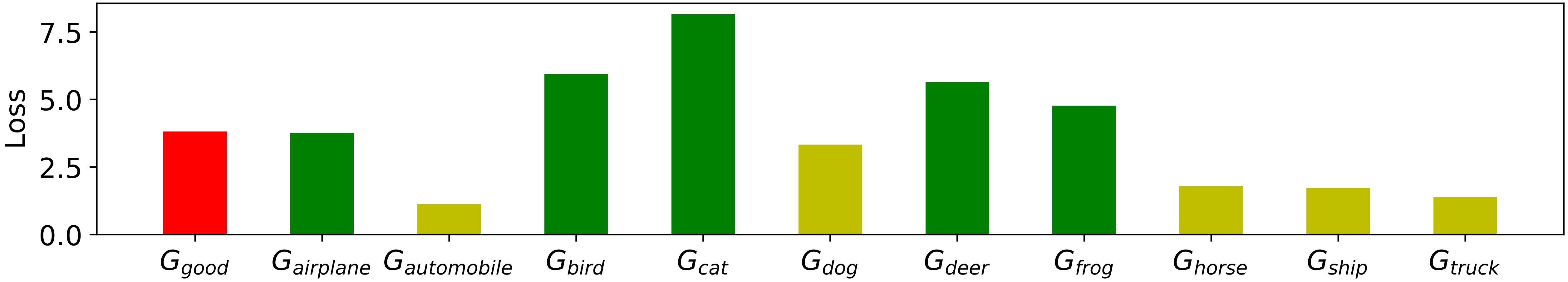
PROPOSED METHOD

➤ Problem of Auxiliary Rotation + GAN (CVPR19)

- For example: a mode-collapsed generator.
  - Samples from one class

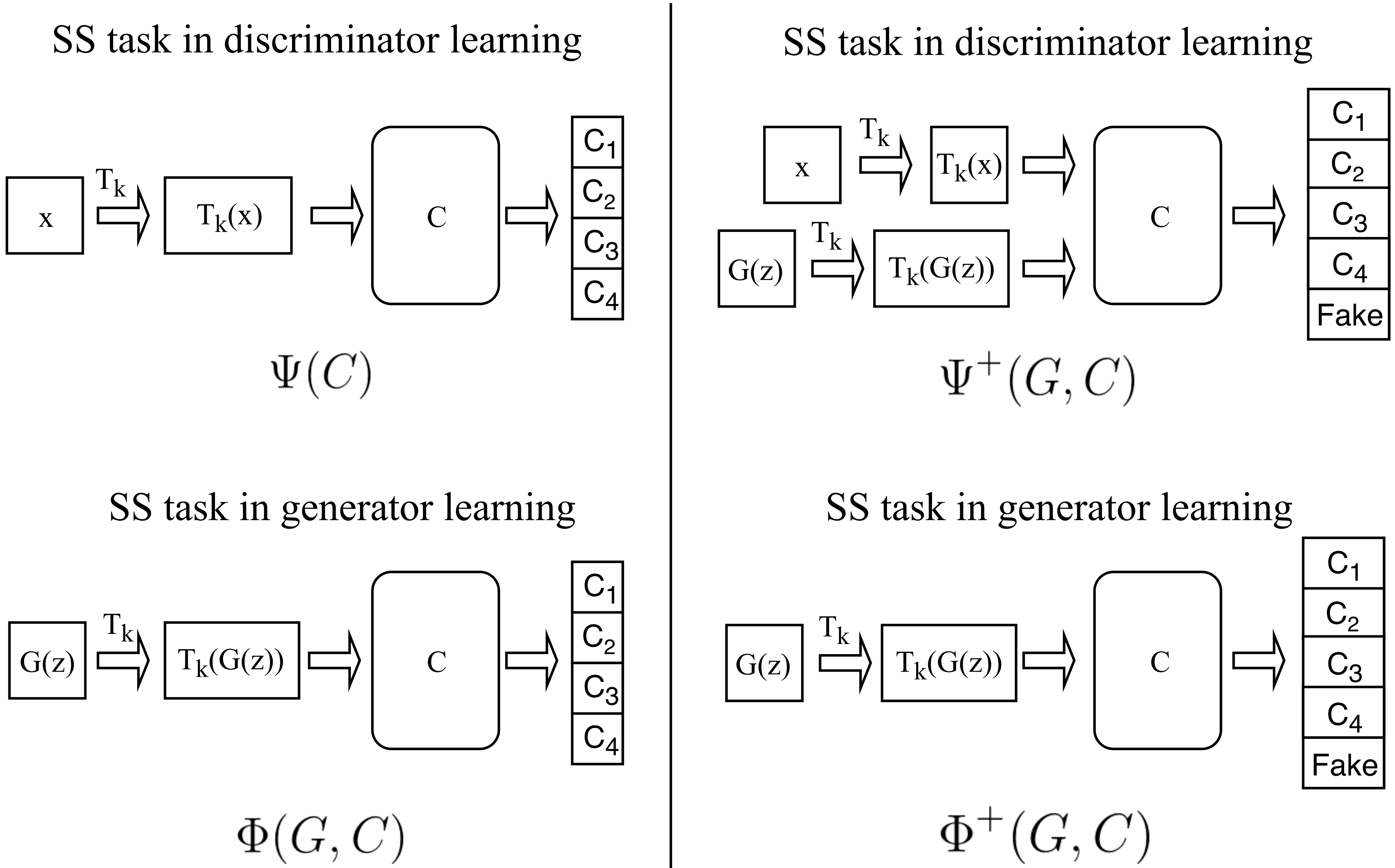


- $-\Phi(G, C)$ on different classes



PROPOSED METHOD

➤ Solution



(a) Original SSGAN

(b) Our proposal

PROPOSED METHOD

- Solution

$$\max_{D,C} \mathcal{V}(D,C,G) = \mathcal{V}(D,G) + \lambda_d \underbrace{\left(\mathbb{E}_{\mathbf{x}\sim P_d^T} \mathbb{E}_{T_k\sim\mathcal{T}} \log\Big(C_k(\mathbf{x})\Big) + \mathbb{E}_{\mathbf{x}\sim P_g^T} \mathbb{E}_{T_k\sim\mathcal{T}} \log\Big(C_{K+1}(\mathbf{x})\Big)\right)}_{\Psi^+(G,C)}$$

$$\min_{G} \mathcal{V}(D,C,G) = \mathcal{V}(D,G) - \lambda_g \underbrace{\left(\mathbb{E}_{\mathbf{x}\sim P_g^T} \mathbb{E}_{T_k\sim\mathcal{T}} \log\Big(C_k(\mathbf{x})\Big) - \mathbb{E}_{\mathbf{x}\sim P_g^T} \mathbb{E}_{T_k\sim\mathcal{T}} \log\Big(C_{K+1}(\mathbf{x})\Big)\right)}_{\Phi^+(G,C)}$$

PROPOSED METHOD

➤ Theoretical Analysis

Proposition 2 *For fixed generator G, the optimal solution C^* under Eq. 8 is:*

$$C_k^*(\mathbf{x}) = \frac{p_d^T(\mathbf{x})}{p_g^T(\mathbf{x})} \frac{p_d^{T_k}(\mathbf{x})}{\sum_{k=1}^K p_d^{T_k}(\mathbf{x})} C_{K+1}^*(\mathbf{x}) \tag{10}$$

where $p_d^T(\mathbf{x})$ and $p_g^T(\mathbf{x})$ are probability of sample $\mathbf{x}$ in the mixture distributions P_d^T and P_g^T respectively.

Theorem 2 *Given optimal classifier C^* obtained from multi-class minimax training $\Psi^+(G, C)$, at the equilibrium point, maximizing $\Phi^+(G, C^*)$ is equal to maximizing Eq. 11:*

$$\Phi^+(G, C^*) = -\frac{1}{K}\left[\sum_{k=1}^K \mathrm{KL}(P_g^{T_k} || P_d^{T_k})\right] + \frac{1}{K}\sum_{k=1}^K \left[\mathbb{E}_{\mathbf{x}\sim P_g^{T_k}} \log\left(\frac{p_d^{T_k}(\mathbf{x})}{\sum_{k=1}^K p_d^{T_k}(\mathbf{x})}\right)\right] \tag{11}$$

PROPOSED METHOD

➤ Theoretical Analysis

Proposition 2 *For fixed generator G, the optimal solution C^* under Eq. 8 is:*

$$C_k^*(\mathbf{x}) = \frac{p_d^T(\mathbf{x})}{p_g^T(\mathbf{x})} \frac{p_d^{T_k}(\mathbf{x})}{\sum_{k=1}^{K} p_d^{T_k}(\mathbf{x})} C_{K+1}^*(\mathbf{x}) \tag{10}$$

where $p_d^T(\mathbf{x})$ and $p_g^T(\mathbf{x})$ are probability of sample $\mathbf{x}$ in the mixture distributions P_d^T and P_g^T respectively.

Theorem 2 *Given optimal classifier C^* obtained from multi-class minimax training $\Psi^+(G, C)$, at the equilibrium point, maximizing $\Phi^+(G, C^*)$ is equal to maximizing Eq. 11:*

$$\Phi^+(G, C^*) = \boxed{-\frac{1}{K}\left[\sum_{k=1}^{K} \mathrm{KL}(P_g^{T_k} || P_d^{T_k})\right]} + \frac{1}{K}\sum_{k=1}^{K}\left[\mathbb{E}_{\mathbf{x}\sim P_g^{T_k}} \log\left(\frac{p_d^{T_k}(\mathbf{x})}{\sum_{k=1}^{K} p_d^{T_k}(\mathbf{x})}\right)\right] \tag{11}$$

new part

PROPOSED METHOD

➤ Theoretical Analysis

Proposition 2 *For fixed generator G, the optimal solution C^* under Eq. 8 is:*

$$C_k^*(\mathbf{x}) = \frac{p_d^T(\mathbf{x})}{p_g^T(\mathbf{x})} \frac{p_d^{T_k}(\mathbf{x})}{\sum_{k=1}^{K} p_d^{T_k}(\mathbf{x})} C_{K+1}^*(\mathbf{x}) \tag{10}$$

where $p_d^T(\mathbf{x})$ and $p_g^T(\mathbf{x})$ are probability of sample $\mathbf{x}$ in the mixture distributions P_d^T and P_g^T respectively.

Theorem 2 *Given optimal classifier C^* obtained from multi-class minimax training $\Psi^+(G, C)$, at the equilibrium point, maximizing $\Phi^+(G, C^*)$ is equal to maximizing Eq. 11:*

$$\Phi^+(G, C^*) = -\frac{1}{K}\left[\sum_{k=1}^{K} \mathrm{KL}(P_g^{T_k} || P_d^{T_k})\right] + \frac{1}{K}\sum_{k=1}^{K}\left[\mathbb{E}_{\mathbf{x}\sim P_g^{T_k}} \log\left(\frac{p_d^{T_k}(\mathbf{x})}{\sum_{k=1}^{K} p_d^{T_k}(\mathbf{x})}\right)\right] \tag{11}$$

new part

$\mathrm{KL}(P_g^{T_k} || P_d^{T_k}) = \mathrm{KL}(P_g || P_d)$: rotation is an affine transform.

KL divergence is invariant under affine transform.

PROPOSED METHOD

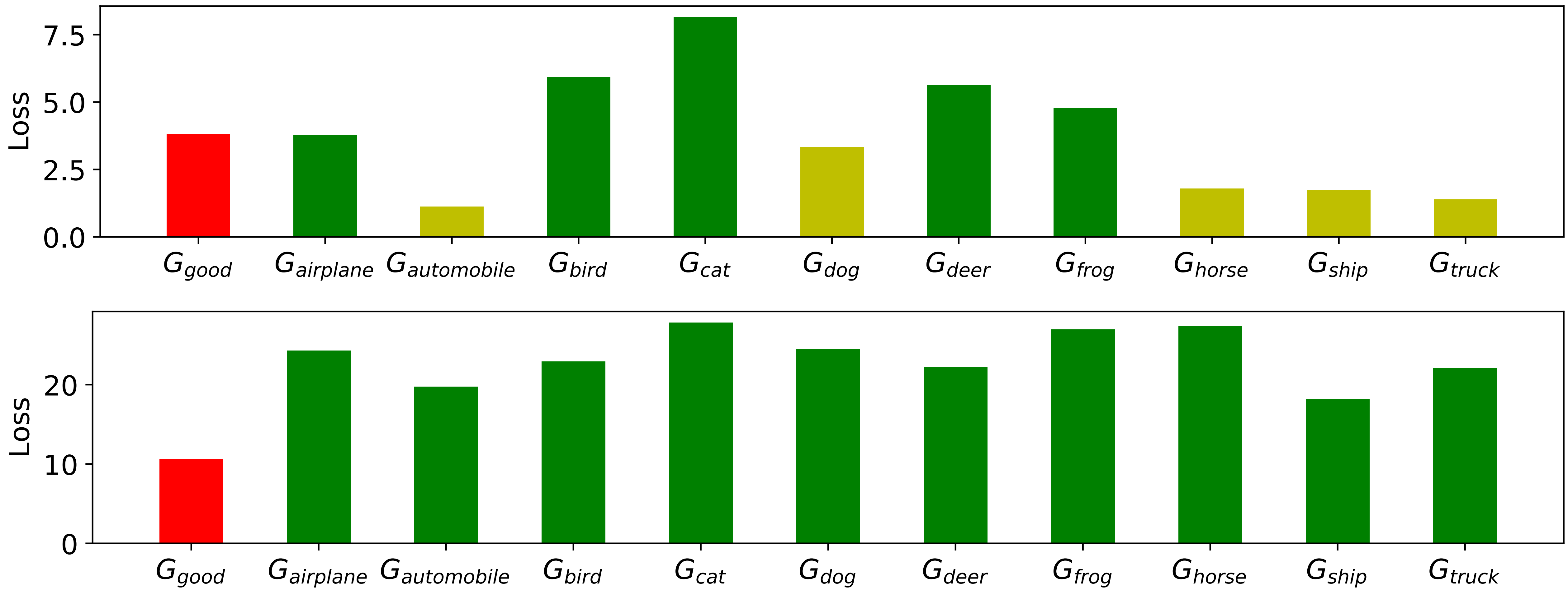
➤ Theoretical Analysis

- Proposed SS tasks work together to improve the matching of P_g and P_d by leveraging the rotated samples
- NetG has more feedbacks

PROPOSED METHOD

➤ Experiments

- G_{good} (balanced generator) has the lowest loss



OUTLINE

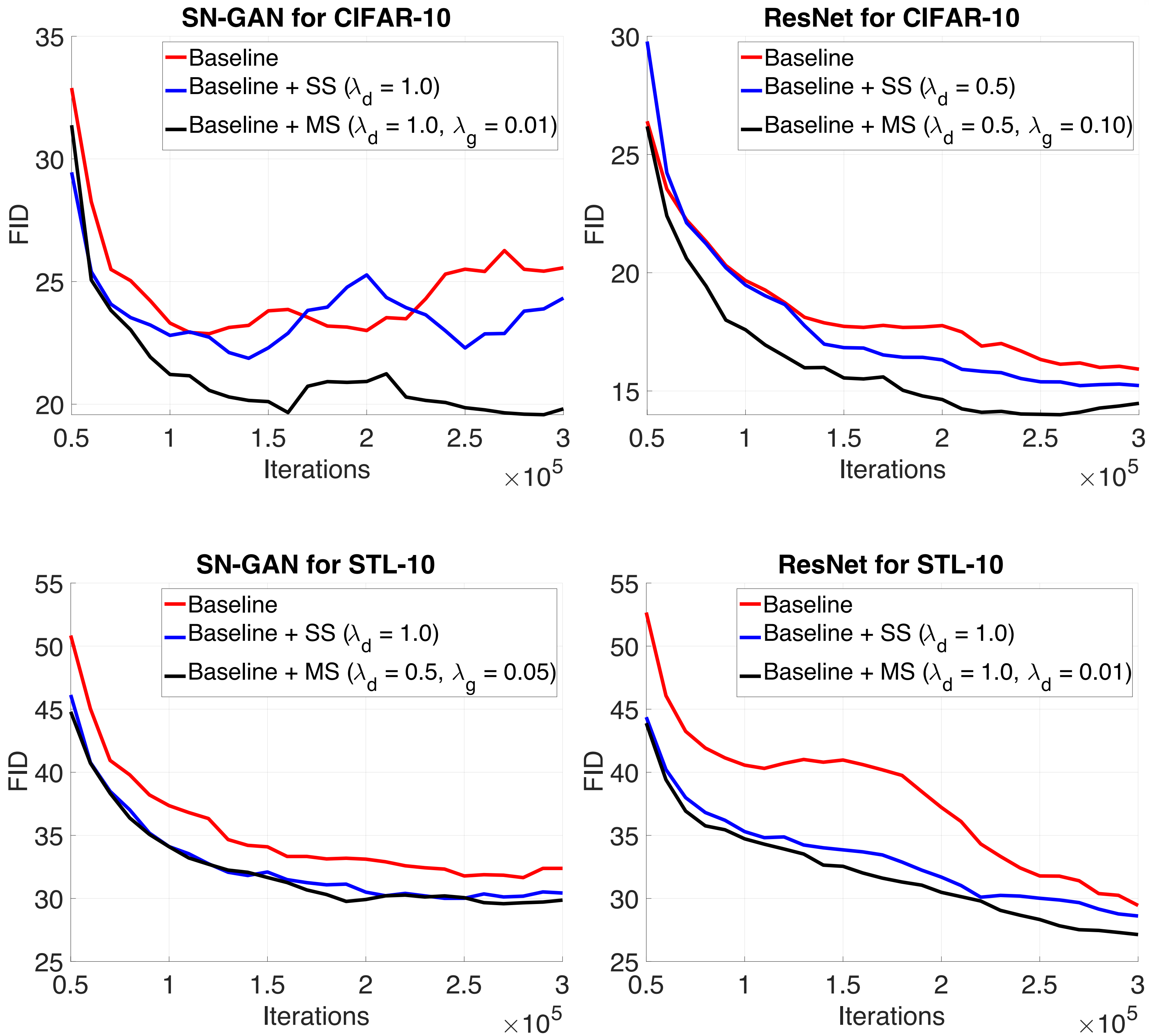
- ➤ Background
- ➤ Proposed Method
- ➤ Experimental Results
- ➤ Conclusion

EXPERIMENTAL RESULTS

- Metric: Fréchet Inception Distance (FID)
- Dataset: CIFAR-10, STL-10

EXPERIMENTAL RESULTS

- ➤ SS: CVPR19
- ➤ MS: Proposed



EXPERIMENTAL RESULTS

Table 1: Comparison with other state-of-the-art GAN on CIFAR-10 and STL-10 datasets. We report the best FID of the methods. Two network architectures are used: **SN-GAN** networks (CNN architectures in [30]) and **ResNet**. The FID scores are extracted from the respective papers when available. **SS** denotes the original SS tasks proposed in [4]. **MS** denotes our proposed self-supervised tasks. '*': FID is computed with 10K-10K samples as in [4]. All compared GAN are unconditional, except SAGAN and BigGAN. SSGAN$^+$ is SS-GAN in [4] but using the best parameters we have obtained. In SSGAN$^+$ + MS, we replace the original **SS** in author's code with our proposed **MS**.

	SN-GAN		**ResNet**		
Methods	**CIFAR-10**	**STL-10**	**CIFAR-10**	**STL-10**	**CIFAR-10***
GAN-GP [30]	37.7	-	-	-	-
WGAN-GP [30]	40.2	55.1	-	-	-
SN-GAN [30]	25.5	43.2	21.70 ± .21	40.10 ± .50	19.73
SS-GAN [4]	-	-	-	-	15.65
Dist-GAN [41]	22.95	36.19	17.61 ± .30	28.50 ± .49	13.01
GN-GAN [42]	21.70	30.80	16.47 ± .28	-	-
SAGAN [47] (cond.)	-	-	13.4 (best)	-	-
BigGAN [2] (cond.)	-	-	14.73	-	-
SSGAN$^+$	-	-	-	-	20.47
Ours(SSGAN$^+$ + MS)	-	-	-	-	19.89
Dist-GAN + SS	21.40	29.79	14.97 ± .29	27.98 ± .38	12.37
Ours(Dist-GAN + MS)	**18.88**	**27.95**	**13.90 ± .22**	**27.10 ± .34**	**11.40**

EXPERIMENTAL RESULTS

- Dataset: CIFAR-100, ImageNet 32×32

Datasets	SS	MS
CIFAR-100 (10K-5K FID)	21.02	19.74
ImageNet 32×32 (10K-10K FID)	17.1	12.3

- Dataset: Stacked MNIST (stacking 3 random digits)

Table 3: Comparing to state-of-the-art methods on Stacked MNIST with tiny $K/4$ and $K/2$ architectures [29]. We also follow the same experiment setup of [29]. Baseline model: Dist-GAN. **SS**: proposed in [4]; **MS**: this work. Our method **MS** achieves the best results for this dataset with both architectures, outperforming state-of-the-art [41, 17] by a significant margin.

Arch	Unrolled GAN [29]	WGAN-GP [13]	Dist-GAN [41]	Pro-GAN [17]	[41]+SS	Ours([41]+MS)
K/4, #	372.2 ± 20.7	640.1 ± 136.3	859.5 ± 68.7	859.5 ± 36.2	906.75 ± 26.15	926.75 ± 32.65
K/4, KL	4.66 ± 0.46	1.97 ± 0.70	1.04 ± 0.29	1.05 ± 0.09	0.90 ± 0.13	0.78 ± 0.13
K/2, #	817.4 ± 39.9	772.4 ± 146.5	917.9 ± 69.6	919.8 ± 35.1	957.50 ± 31.23	976.00 ± 10.04
K/2, KL	1.43 ± 0.12	1.35 ± 0.55	1.06 ± 0.23	0.82 ± 0.13	0.61 ± 0.15	0.52 ± 0.07

OUTLINE

- Background
- Proposed Method
- Experimental Results
- Conclusion

CONCLUSION

- Theoretical analysis on auxiliary self-supervised + GAN
- Propose multi-class minimax game