Rethinking Categorical Generative Adversarial Networks in the context of Meta Pseudo Labels

2019122076 이상규

I. Introduction

Meta Pseudo Labels[1]는 최근 Semi-Supervised Learning으로써 우수한 performance를 보여주었다. 이는 Student-Teacher 구조를 갖는 Pseudo Labeling[2]과 유사하다. 그러나, teacher model을 pre-training 후 fix하고 pseudo label을 얻는 것이 아니라, teacher를 동시에 student와 함께 학습하는 방법이다. 이는 teacher의 Unsupervised Objective에 student의 labeled data에 대한 prediction loss를 feedback하는 방법으로 실현되었다. 기존의 Pseudo Labeling에서 student model의 performance가 teacher model의 performance보다 우수하기 어려웠던 문제를 효과적으로 해결했다는 점에서 큰 의의를 지닌다.

하지만 동시에 teacher와 student를 학습시키기 위해서 두 model을 memory에 유지하려면 큰 memory capacity를 요구했다. 저자들은 이로 인해 Model Parallelism으로 학습시켰으나 상대적으로 많은 overhead를 발생시켰다. 대안으로 저자들이 제시하는 Reduced Meta Pseudo Labels이 있지만 이 방법은 teacher model이 학습하는 과정에서 student model의 feedback이 온전히 반영되지 않는다. 생성한 pseudo label만을 student의 learning state에 따라서 calibration하는 것에 가깝기 때문에 완전한 Meta Pseudo Labels를 수행한 경우와 달리 형태상 유사한 Noisy Student[3]와 큰 performance 차이를 보여주지 못했다.

따라서 Meta Pseudo Labels의 접근법을 효율적으로 적용하기 위해서는 teacher model이 size가 작으면서도 student가 학습하기 충분한 수준의 pseudo label을 생성할 수 있어야 한다. 저자들은 Model Parallelism을 위해서 teacher와 student의 Architecture를 동일하게 유지하고 parameter weight만 달라지도록 구현하였다. 그러나 teacher model의 목적은 student model의 learning state에 맞추어 pseudo label을 생성하는 것이다. 따라서 Semi-Supervised Learning 자체를 잘 수행할 수 있으면서 동시에 상대적으로 작은 model size를 갖고 있다면, teacher model로서 필요한 조건은 만족한다고 할 수 있다.

이와 관련하여 Semi-Supervised Learning을 위해 Supervised case에서 우수했던 backbone을 사용하지 않고 GAN[4]의 Discriminator를 사용했던 시도[5, 6]가 있었다. Generator를 학습시키는 과정에서 Discriminator가 Unsupervised Learning을 통한 feature extractor로서 좋은 performance를 보이는 것이 관찰되었기 때문에[10, 16], 적절한 Objective를 설정하면 Semi-Supervised Learning에서도 적용될 수 있었다. Categorical GAN(CatGAN)[6]은 이러한 시도 중 하나였으며, Generator의 역할이 Discriminator의 학습 과정에서 일종의 Regularization으로 기능할 수 있음을 주장하였다. 저자들은 이를 Clustering에서 사용될 수 있는 Regularized Information Maximization[7]이 Neural Network에 적용될 수 있도록 일반화된 것이라고 설명한다.

하지만 GAN Architecture는 학습과정이 굉장히 불안정하기 때문에 Supervised Learning에서 사용되는 feature extractor에 비해 큰 input size를 처리할 수 있는 깊은 model의 학습이 어렵다. 이러한 문제는 GAN Architecture에서 지속적으로 제시되는 문제이며, 여전히 개선방법에 대해서 연구되고 있다[8]. 그러므로 현재 시점에서 Semi-Supervised GAN Objective를 통해 Discriminator의 model size를 늘려 GAN Architecture를 general purpose feature extractor로 학습시키는 것은 현실적으로 어려운 문제라고 할 수 있다.

그러나 Meta Pseudo Labels의 맥락으로 본다면 해당 접근법을 teacher model의 측면에서 응용할 수 있다. 즉, 최종 model로서 사용하기 힘들다면 pseudo label을 통해 student model에 자신의 performance를 'transfer'하는 auxiliary model으로 활용할 수 있다. 특히, 앞서 언급한 이유로 model size를 키울 수는 없지만 작은 size로도 좋은 performance를 보인다는 점은 memory capacity 문제에서 자유롭지 못한 Meta Pseudo

Labels 측면에서 상대적인 이점을 지닌다. 따라서 실험을 통하여 CatGAN Objective를 적절히 수정한다면 Meta Pseudo Labels를 적용할 수 있는지, 실제로 performance 측면에서 효과적인지를 확인해보고자 한다.

II. Hypothesis

Meta Pseudo Labels Objective 중, 생성된 pseudo label에 대한 student model로부터 발생하는 feedback이 teacher model에 영향을 주는 Gradient term은 다음과 같다.

$$\nabla_{\theta_T}\mathcal{L}_T = h_t \cdot \nabla_{\theta_T} CE\big[T(x_u|\theta_T)||\hat{y_u}\big],\ where\ (x_l, y_l) \sim \chi^L,\ x_u \sim \chi^U$$
$$s.t.\ hard\ pseudo\ label\ \hat{y_u} \sim P(\cdot|x_u, D),\ h_t = \eta_S \cdot \Big(\big(\nabla_{\theta_{S_{t+1}}} CE\big[S(x_l|\theta_{S_{t+1}})||y_l\big]\big)^T \cdot \nabla_{\theta_{S_t}} CE\big[S(x_u|\theta_{S_t})||\hat{y_u}\big]\Big)$$

이 때 'feedback coefficient'로 불리는 scalar h_t를 제외해서 본다면, 실질적으로 teacher parameter에 영향을 주는 term은 자신이 생성한 hard pseudo label에 대한 unlabeled data에서의 prediction, 다시 말해 soft pseudo label과의 Cross-Entropy임을 알 수 있다. 즉, feedback coefficient는 유도 과정에서는 teacher의 parameter에 의존하지만 실제 teacher의 Gradient 계산 시에는 training step마다 변화하는 Unsupervised Learning hyperparameter라고 할 수 있다. 그러므로 CatGAN Objective에 pseudo label에 대한 student model의 feedback을 반영하기 위해서는 teacher의 Unsupervised Learning term에 대해서만 고려해도 충분하다.

따라서 우리가 CatGAN Objective에 이러한 feedback을 반영하기 위해서는 Objective 중 Unsupervised Learning term을 살펴보면서 어떻게 이를 변경할 수 있는지 판단해야 한다. 이때, CatGAN의 Semi-Supervised Objective를 각 term의 목적에 따라서 재구성하여 표현하면 다음과 같이 나타낼 수 있다.

$$\mathcal{L}_{supervised} = \min_D \lambda \cdot \mathbb{E}_{(x_l,y_l)\sim\chi^L}\big[CE[D(x_l)||y_l]\big]$$
$$\mathcal{L}_{unsupervised} = \max_D H_{\chi^U}\big[p(y_u|D)\big] - \mathbb{E}_{x_u\sim\chi^U}\big[H[p(y_u|x_u, D)]\big]$$
$$\mathcal{L}_{adversarial} = \min_G \max_D \mathbb{E}_{z\sim p(z)}\big[H[p(y|G(z), D)]\big] - H_G\big[p(y|D)\big]$$

Unsupervised Objective는 2개의 term을 갖고 있는데, 이는 Discriminator의 unlabeled dataset에 대한 prediction의 Marginal Entropy와 Joint Entropy이다. 이에 따르면 Marginal Entropy는 maximize하면서 Joint entropy는 minimize하는 것을 목적으로 하고 있다. Marginal Entropy를 maximize하는 것은 Discriminator가 unlabeled data에 대해 uniform한 class prediction을 갖게 만들어서 predicted class의 imbalance를 피하기 위해서 존재한다. Joint Entropy를 minimize하는 것은 특정 sample이 주어졌을 때 그 sample의 class를 하나로 특정하기 위해서 존재한다. 이는 Adversarial Objective에서 Generator가 생성하는 image에 대한 Entropy를 maximize하는 것과 반대되어 unlabeled data에 대한 regularization의 효과도 유도된다.

이때, Marginal Entropy term은 Meta Pseudo Labels 측면에서는 큰 관련이 없다. 이는 Discriminator가 unlabeled data에 대해서 효과적인 prediction을 수행하도록 돕기 보다는 overfitting을 방지한다고 보아야 한다. 왜냐하면 Marginal Entropy term은 negative form의 Kullback-Leibler divergence로 다시 표현할 수 있는데, 이를 maximize하는 것은 prior distribution을 Uniform distribution으로 가정하고 prediction distribution과의 차이를 minimize하는 것과 동일하기 때문이다. 즉, unlabeled dataset의 distribution을 uniform이라 가정하고 Discriminator가 가정에서 크게 벗어나는 수준으로 unlabeled data에 class를 부여하지 않도록 predicted class balance를 prior distribution과 유사하게 만드는 것이다. 이는 unlabeled data이기 때문에 아무런 정보 없이 distribution을 추론하는 것이 불가능하므로 합리적인 가정이라고 할 수 있다.

$$H_{\chi^U}\left[p(y_u|D)\right] = \log C - D_{KL}(p(y_u|D)||U)$$

그러나 Joint Entropy term에 대해서는 생각해볼 필요가 있다. 해당 term을 minimize하는 것은 실제 data가 Discriminator를 지난 확률 $P(y|D,x_u)$가 특정한 class y에 대해서만 maximize되는 것을 의미한다. 이를 learning step t에서 본다면 step $t+1$에서는 step t에서 생성된 hard pseudo label의 결과를 강화하는 것과 같다. 왜냐하면 hard pseudo label은 각 unlabeled sample x_u에 따라 도출된 $P(y|D,x_u)$ 중 최대의 y로 결정되므로, 이 결정된 y를 더욱 강화하는 것은 다른 y에 대한 $P(y|D,x_u)$가 작아지는 것을 의미하기 때문이다. 즉, 우리가 Joint Entropy term을 minimize하는 것은 결과적으로 step t의 soft pseudo label의 distribution을 hard pseudo label의 distribution과 유사도록 변화시키는 것과 같다.

따라서, Joint Entropy term을 minimize하는 것은 현재 step t의 hard pseudo label에 대한 soft pseudo label의 Cross-Entropy를 minimize하는 문제로 치환할 수 있다. 즉, Meta Pseudo Labels의 Objective term과 유사하게 이를 변경할 수 있다. 이러한 관점에서 보면 Meta Pseudo Labels Objective는 CatGAN의 Unsupervised Objective의 일반화라고 볼 수 있다. 고정된 scalar가 아니라 step t마다 정해지는 hyperparameter, feedback coefficient h_t를 통해서 Unsupervised Loss의 크기를 결정하는 것이다.

$$\min_D \mathbb{E}_{x_u\sim\chi^U}\left[H[p(y_u|x_u,D)]\right] \approx \min_D \mathbb{E}_{x_u\sim\chi^U}\left[CE\left[D(x_u)||\hat{y_u}\right] \rightarrow \min_D h_t \cdot \mathbb{E}_{x_u\sim\chi^U}\left[CE\left[D(x_u)||\hat{y_u}\right]\right.\right.$$
$$s.t.\ \textit{hard pseudo label}\ \hat{y_u} \sim P_{x_u\sim\chi^U}(\cdot|x_u,D),\ \textit{arbitrary scalar}\ h_t$$

만약 이 일반화가 적절하다면 Vanilla Objective performance보다 더 나은 performance를 보여줄 것이다. 또한, Meta Pseudo Label의 Objective와 유사한 형태이므로 student의 performance가 단순한 Pseudo Labeling에 비해 상승되어야 할 것이다. 따라서 이러한 가설이 적절한 추론인지 실험을 통해서 확인해보고자 한다. 이를 위해 최종적으로 실험 시에 사용되어야 할 Objective는 다음과 같이 정리할 수 있다.

$$\mathcal{L}_G = \min_G \mathbb{E}_{z\sim p(z)}\left[H[p(y|G(z),D)]\right] - H_G\left[p(y|D)\right]$$
$$\mathcal{L}_D = \min_D \lambda \cdot \mathbb{E}_{(x_l,y_l)\sim\chi^L}\left[CE[D(x_l)||y_l]\right] + \left(h_t \cdot \mathbb{E}_{x_u\sim\chi^U}\left[CE[D(x_u)||\hat{y_u}]\right] - H_{\chi^U}\left[p(y_u|D)\right]\right) - \mathbb{E}_{z\sim p(z)}[H[p(y|G(z),D)]]$$
$$\mathcal{L}_S = \min_S \mathbb{E}_{x_u\sim\chi^U}\left[CE[(S(x_u))||\hat{y_u}]\right]$$
$$s.t.\ \textit{hard pseudo label}\ \hat{y_u} \sim P_{x_u\sim\chi^U}(\cdot|x_u,D),\ h_t = \eta_S \cdot \left(\left(\nabla_{\theta_{S_{t+1}}} CE\left[S(x_l|\theta_{S_{t+1}})||y_l\right]\right)^T \cdot \nabla_{\theta_{S_t}} CE\left[S(x_u|\theta_{S_t})||\hat{y_u}\right]\right)$$

III. Experiments

실험을 위한 dataset으로는 STL-10[9]을 사용하였다. 해당 dataset은 10개의 class에 대한 labeled image가 500장씩 주어지며, 총 100,000장의 unlabeled data가 주어진다. unlabeled data는 어느 정도 labeled data의 distribution과 유사하지만, labeled data에서 주어진 class에 속하지 않는 image도 포함되어 있다. 해당 dataset은 PyTorch의 Torchvision package API를 통하여 사용되었다.

teacher model로서 CatGAN의 Objective를 사용하는 DCGAN[10] Architecture를 선택하였다. 기본적으로 DCGAN Architecture는 real, fake를 구분하기 위해 최종 output이 Sigmoid output으로 채택했으므로 최종 output 단을 Softmax output으로 교체하였다. 한편, 비교군을 위한 student model들로 ResNet-50[11]과 Xception[12]을 채택하였다. 실험 환경의 문제로 인해 큰 image size에 대한 실험이 불가능하므로 목적 image size에 맞게 각 모델의 첫 convolution layer의 kernel size가 original Architecture에 비해서 축소되었다. Hypothesis의 확인을 위해서 training method를 바꾸어 가며 다음과 같이 총 4가지 case에 대한 performance를 비교하였다.

1. labeled data만을 통해 student model의 Supervised Learning

2. CatGAN을 teacher model로 Semi-Supervised Learning 후 Pseudo Labeling training

3. student model과 동일한 teacher model를 사용하여 Meta Pseudo Labels training

4. 수정된 CatGAN Objective를 통해 student model과 teacher model를 Meta Pseudo Labels training

각 조건에서의 Model 별 Top-1 accuracy를 정리하면 다음과 같았다.

Method	Model		Top-1 accuracy
Supervised	ResNet-50		51.18%
	Xception		57.19%
Pseudo Labeling	CatGAN		65.15%
	Student	ResNet-50	51.81%
		Xception	63.75%
Meta Pseudo Labels	ResNet-50		55.80%
	Student	Before fine-training	25.58%
		fine-trained	28.49%
	Xception		66.31%
	Student	Before fine-training	65.09%
		fine-trained	69.26%
	CatGAN(Student: ResNet-50)		**70.21%**
	Student	Before fine-training	62.37%
		fine-trained	**65.15%**
	CatGAN(Student: Xception)		68.85%
	Student	Before fine-training	69.51%
		fine-trained	**72.73%**

공통적으로 input image는 64 x 64로 random crop된 이후 channel-wise하게 mean 0.5, standard deviation 0.5로 Normalize되었다. 이외의 data augmentation은 적용되지 않았다. 원문의 Meta Pseudo Labels의 경우 UDA Objective[13]가 사용되었으나, 이는 learning method 간의 비교의 단순화를 위해 모든 경우에서 생략되었다. Semi-Supervised Objective의 parameter λ는 모두 3으로 적용되었다. 한편, memory capacity 문제로 인해 CatGAN과 ResNet-50을 training하는 경우는 batch size 100, Xception은 batch size 50으로 진행되었다. Optimizer는 모든 경우에서 Adam[17]을 사용했으며, learning rate η와 β_1은 각각 0.0002, 0.5로 지정되었다. 이외의 learning policy는 모든 경우에서 적용되지 않았다.

Case 1의 경우 labeled data가 적어 빠른 수렴을 보였기 때문에, ResNet-50은 100 epochs으로 진행되었다. 같은 update 횟수를 유지하기 위해서 Xception은 50 epochs을 진행하였다. ResNet-50을 기준으로 Case 2, 4의 경우는 공통적으로 CatGAN을 training 시에 총 500 epochs을 진행했으며, Case 2의 Pseudo Labeling training task에서는 Case 1과 같이 100 epochs을 진행하였다. Xception의 경우는 Case 1과 유사한 이유로 ResNet-50 case의 절반의 epochs으로 진행하였다. Case 4의 경우 최종적으로 labeled data에 대한 fine-training을 요구하므로, ResNet-50의 경우 50 epochs, Xception의 경우 25 epochs을 진행하였다. 또한 Case 3의 경우 teacher model size가 Case4보다 크다는 문제로 인해 batch size를 절반으로 줄여야 했으므로, update 횟수를 유지하기 위해서 모든 epochs가 Case 4의 절반이 되었다.

한편 Meta Pseudo Labels case에서 feedback coefficient는 원문과 다른 방식으로 구현되었다. 원문에서는 teacher를 update하는 과정에서 사용된 Gradient를 다시 사용하여 효율성을 높였다. 또한 Gradient들의 dot product가 아닌 cosine similarity를 통해서 feedback coefficient를 계산했다. 그러나 본 실험의 목적이 state-of-the-art 달성이 아니기 때문에, 조금 더 단순화한 형태로 구현하기로 하였다. 이미 계산된 Gradient를 사용하는 것이 아니라 각 step마다의 student의 labeled data에 대한 loss의 증감분을 coefficient에 그대로 반영하였다. 이 경우에는 coefficient의 range가 원문의 경우보다 크기 때문에 초기 학습에서 불안정할 수는 있으나, 실제 feedback coefficient의 방향에 비례하기 때문에 근사하여 사용할 수 있다.

$$\eta_S \cdot \left(\left(\nabla_{\theta_{S_{t+1}}} CE\left[S(x_l|\theta_{S_{t+1}}) || y_l \right] \right)^T \cdot \nabla_{\theta_{S_t}} CE\left[S(x_u|\theta_{S_t}) || \hat{y_u} \right] \right) \propto \left(CE\left[S(x_l|\theta_{S_t}) || y_l \right] - CE\left[S(x_l|\theta_{S_{t+1}}) || y_l \right] \right)$$

IV. Discussion

실험의 결과에서 볼 때, 수정된 Objective를 통해 CatGAN을 Meta Pseudo Labels teacher로서 활용하는 것은 효과적이었다. Vanilla CatGAN Objective의 경우보다 Meta Pseudo Labels을 적용한 것이 Discriminator 자체의 performance가 증가했음을 알 수 있다. 동시에 Pseudo Labeling을 적용해서 student를 training하는 경우보다 Meta Pseudo Labels를 통해 student를 training하는 경우가 우수했다. 특히 performance transfer 측면에서 Meta Pseudo Labels는 teacher보다 student가 우수해질 수 있음을 확인할 수 있었다. 또한 student와 동일한 Architecture를 teacher로 사용하는 Meta Pseudo Labels보다 model size가 확연히 작음에도 불구하고, CatGAN을 teacher로 사용하는 경우의 performance가 더 우수한 것을 확인할 수 있었다.

그러나 해당 실험으로 완전한 결론을 내리는 것은 아직 부족하다. 큰 input size를 받는 big teacher를 사용할 경우의 performance와 비교할 때는 다른 결론이 나올 수 있다. big teacher를 작은 input size를 사용하는 GAN으로 완전히 대체할 수 있다는 결론을 내기 위해서는 해당 경우와의 비교가 필요하다. 그럼에도 불구하고, memory capacity로 인해 resized된 작은 input size를 사용하는 small teacher로 pseudo label을 생성해야 하는 경우에는 GAN을 Semi-Supervised teacher로서 사용하는 것이 효과적이라는 결론을 내릴 수 있다.

추가적으로 원문에서는 teacher에 UDA Objective를 추가하여 더 나은 Unsupervised Learning을 유도하였다. 이는 [14]에서 classical한 data augmentation이 GAN을 통한 data augmentation과 독립적으로 기능한다는 점이 확인되었기 때문에 UDA Objective를 추가해도 performance 측면에서는 방해가 되지 않을 것이다. 하지만 Discriminator의 performance가 빠르게 향상될 경우 Generator의 학습이 어려울 수 있다는 문제가 있기 때문에 실험을 통한 검증이 필요하다. 그러나 [15]와 같은 경우 좋은 Discriminator를 얻기 위해서 좋은 Generator가 요구되지 않는다는 주장을 제시하므로 이는 치명적인 문제가 아닐 수 있다.

한편 GAN을 사용한 Meta Pseudo Labels이 어느정도 효과적이었다는 점에서 Objective의 Marginal Entropy term에 대해서도 개선의 여지가 있다. 최초에는 Marginal Entropy를 사용해 pre-trained student를 얻은 후, unlabeled dataset에 대한 prediction을 수행한다. 그 후 전체 unlabeled dataset에 대한 predicted distribution을 새로운 prior distribution으로 반영하여 Noisy Student와 같이 iterative하게 training을 진행한다. 즉, uniform distribution을 prior distribution으로 고정하지 않고 pre-trained student의 predicted distribution을 prior distribution으로서 update한다. 이를 통해 update된 prior distribution과의 Kullback-Leibler divergence를 minimize하도록 Objective를 수정하는 것이다. STL-10의 경우는 어느 정도 균등한 distribution을 갖고 있으나, dataset이 불균등한 경우에는 이렇게 training을 반복하는 것이 효과적일 것이라 추측할 수 있다.

$$H_{\chi^U}\left[p(y_u|D) \right] \approx -D_{KL}(p(y_u|D)||U) \rightarrow -D_{KL}(p(y_u|D)||\mathcal{U}) \ where \ \mathcal{U} \sim P_{x_u \sim \chi^U}(\cdot|x_u, \tilde{S})$$

V. Reference

[1] Hieu Pham, Zihang Dai, Qizhe Xie, Minh-Thang Luong and Quoc V. Le, "Meta Pseudo Labels", arXiv, 2003.10580, 2021.

[2] D. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks", International Conference on Machine Learning Workshops (ICMLW), 2013.

[3] Qizhe Xie, Minh-Thang Luong, Eduard Hovy and Quoc V. Le, "Self-training with Noisy Student improves ImageNet classification", arXiv, 1911.04252, 2020

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio, "Generative Adversarial Networks", arXiv, 1406.2661, 2014.

[5] Augustus Odena, "Semi-Supervised Learning with Generative Adversarial Networks", arXiv, 1606.01583, 2016.

[6] Jost Tobias Springenberg, "Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks", arXiv, 1511.06390, 2016.

[7] Krause, Andreas, Perona, Pietro, and Gomes, Ryan G, "Discriminative clustering by regularized information maximization", In Advances in Neural Information Processing Systems (NIPS) 23, MIT Press, 2010.

[8] Tero Karras, Timo Aila, Samuli Laine and Jaakko Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation", arXiv, 1710.10196, 2018.

[9] Adam Coates, Honglak Lee and Andrew Y. Ng, "An Analysis of Single Layer Networks in Unsupervised Feature Learning", AISTATS, 2011.

[10] Alec Radford, Luke Metz and Soumith Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", arXiv, 1511.06434, 2016.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun, "Deep Residual Learning for Image Recognition", arXiv, 1512.03385, 2015.

[12] François Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions", arXiv, 1610.02357, 2017.

[13] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong and Quoc V. Le, "Unsupervised Data Augmentation for Consistency Training", arXiv, 1904.12848, 2020.

[14] Christopher Bowles, Liang Chen, Ricardo Guerrero, Paul Bentley, Roger Gunn, Alexander Hammers, David Alexander Dickie, Maria Valdés Hernández, Joanna Wardlaw and Daniel Rueckert, "GAN Augmentation: Augmenting Training Data using Generative Adversarial Networks", arXiv, 1810.10863, 2018.

[15] Zihang Dai, Zhilin Yang, Fan Yang, William W. Cohen and Ruslan Salakhutdinov, "Good Semi-supervised Learning that Requires a Bad GAN", arXiv, 1705.09783, 2017.

[16] Xin Mao, Zhaoyu Su, Pin Siang Tan, Jun Kang Chow and Yu-Hsing Wang, "Is Discriminator a Good Feature Extractor?", arXiv, 1912.00789, 2020.

[17] Diederik P. Kingma and Jimmy Bam, "Adam: A Method for Stochastic Optimization", arXiv, 1412.6980, 2017.