DLOW: Domain Flow for Adaptation and Generalization

# **1. Introduction**

기존의 DA는 두가지의 문제점을 목표로 하고 있었다. 첫 번째는 label-rich data로부터 label-scarce data로의 adaptation과 두 번째로는 source domain의 image를 target domain의 image style로 흉내내는 image translation 문제가 존재하였다.

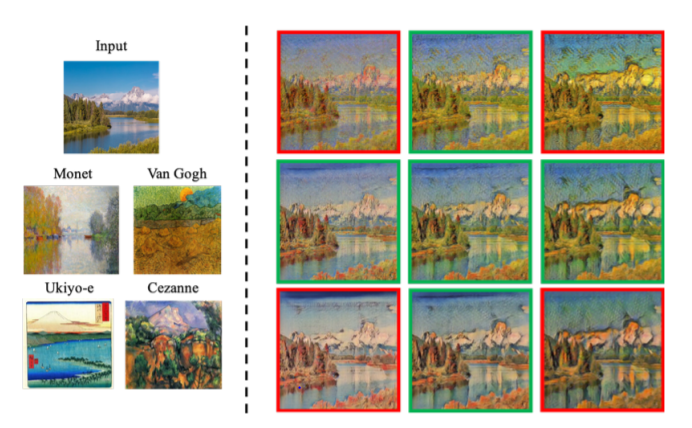
대부분의 선행연구들은 이러한 task들에서 target domain에 집중하였다. 즉, target domain의 data distribution에 well fitting시키는 것이 가장 큰 목적이었다.

그러나 저자는 intermediate domain(between source domain & target domain)에 집중하는 것을 제안하였다. 이는 image를 source domain에서 어떤 특정한 intermediate domain으로 translation을 가능하게 하였다. Source image에서 target image로 translation을 하면서 저자는 source domain에서 target domain으로 distribution을 이동 시키는 것을 자연스럽게 특성화 하는 일련의 이미지 시퀀스를 얻을 수 있다고 합니다.

이러한 DLOW model은 두 단계의 이점이 있습니다.

첫 번째로는 intermediate domain은 두 domain의 distribution의 차이를 연결시켜줄 수 있습니다. Intermediate domain으로의 translated image는 domain adaptation task를 더욱 쉽게 만들어 줄 수 있습니다. 더욱더 구해진 model은 unseen data에 대하여 generalized performance를 보여줄 수 있습니다. 이는 다양한 intermediate domain image덕분에 가능한 일입니다.

두 번째로는 DLOW model은 style generalization에 이용될 수 있습니다. 기존의 im2im model들은 오직 one-to-one mapping으로 source image에서 target style을 배우는 방식으로 학습이 되었습니다. 하지만 DLOW model은 target domain 뿐만 아니라 intermediate domain으로의 translation을 가능케 합니다.



해당 모델은 CylceGAN을 기반으로 하여 제시되었습니다. 여기에 Domainness variable을 conditional instance normalization(이 layer는 output image의 style에 영향을 줍니다)을 이용하여 translation network에 추가하였습니다. 반면에 discriminator가 output image의 relatedness를 다른 domain과 균형을 맞추기 위한 weight로도 사용합니다. 또한, multiple target domain에서는 domainness variable이 relatedness를 포함한 vector로 확장이 됩니다.

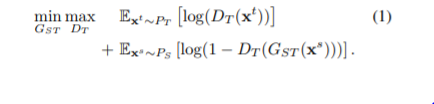
# **2. Domain Flow Generation**

## **2.1 Problem Statement**

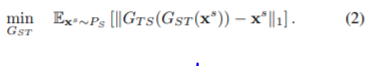
우리는 각각 다른 distribution 와 에서 sampling한 source domain S와 target domain T를 가집니다. 저자는 image level에서 domain shift problem을 해결할 것이라고 말하면서 기존의 방법과의 차별성을 다음과 같이 둡니다. 기존의 연구들은 source to target이었다면 저자가 제시한 방법은 source to all intermediate입니다.

## **2.2 The CycleGAN Model**

간단하게 설명하면 S 🡪 T 로 보내는 G\_ST function을 이용하여 다음 objective function을 푸는 것이다.



또한, G(X^S)를 다시 source domain으로 transfer 시켜서 cycle consistency를 강화 시키는 loss를 추가하였다.



## **2.3 Modeling Intermediate Domains**

기존의 intermediate domain을 이용한 domain shift와는 다른 점은 image level에서 작동한다는 것이다. 이러한 경우에 저자가 제시한 방법은 model의 cross domain generalization 능력을 위한 딥러닝 기술들과 쉽게 결합이 가능하다는 것이다. (image level에서 보는 점은 여러 분야의 기술과 쉽게 접목이 가능하다.) 특히, domainness variable인 z를 0에서 1까지 변화시키면서 우리는 S 에서 T로 흘러가는 intermediate domain의 흐름을 얻을 수 있다. 이러한 흐름을 위해서 path 선택을 할 때 저자는 shortest geodesic path를 domain manifold에서 고른다. Loss는 다음과 같다.

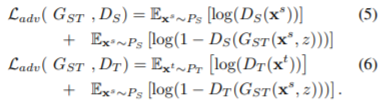


이 때 probability distribution의 distance는 JS divergence를 이용한다.

## **2.4 The DLOW Models**

DLOW model에서 generator는 를 바로 T로 이동시키지 않고 T의 방향으로 이동시킵니다. 이 때 가 z의 비율만큼 이동한 domain을 라 표현합니다.

**Adversarial loss** : 기존의 연구와 다르게 intermediate domain M^(z)가 추가되었으므로 Discriminator는 D\_S와 D\_T가 존재합니다. 식은 다음과 같습니다.



그리고 정리된 식은 다음과 같습니다.

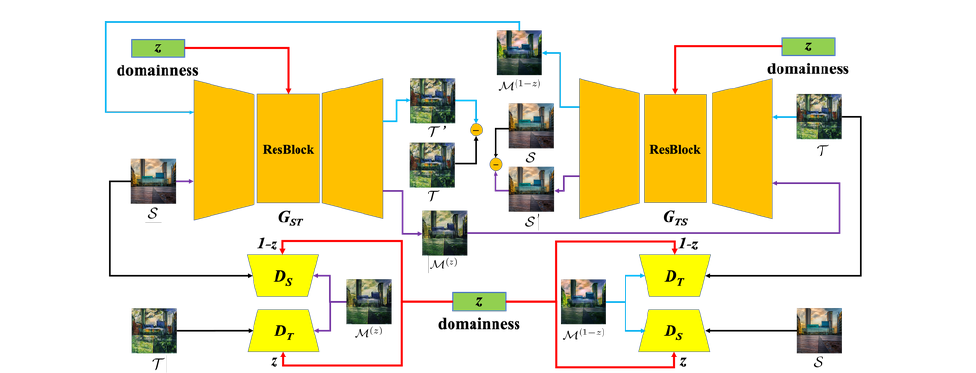


이를 이해하기 위해서는 만약 z=0.1일 경우 G\_ST(x^s)는 source domain S와 가깝기 때문에 D\_S에 대한 loss에 가중치가 크게 붙는다.(훨씬 잘 구분해야한다.) 반면에 D\_T는 멀기 때문에 가중치가 더 적게 붙는다.

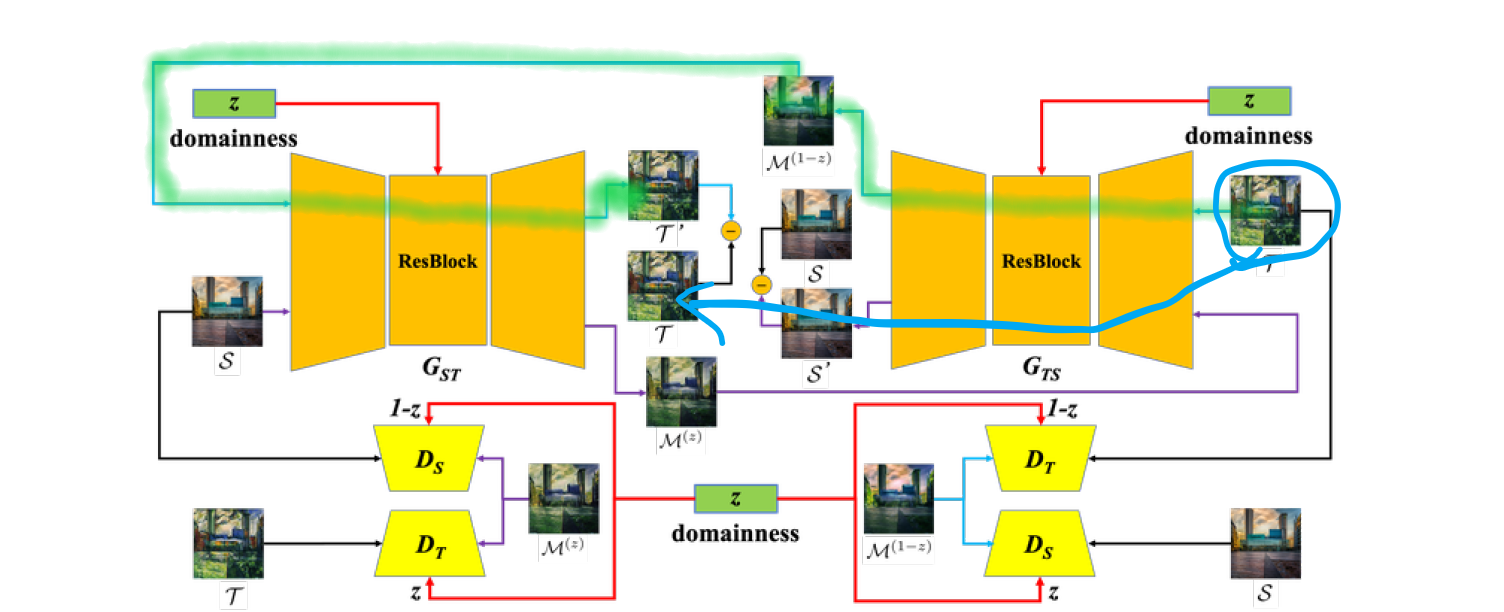
**Image Cycle Consistency loss**: 를 기억한 다음 cycle consistency loss는 다음과 같다.

로 정의가 된다.

**Full Objective**: full objective function은 다음과 같다.



전체 레이어에 대한 설명을 덧 붙이자면 처음에 Domainness variable 를 deconvolution layer를 이용하여 (1, 16, 1, 1)-dimensional vector로 변환해주어서 CN layer에 입력해준다. 또한, discriminator에 balance weight로 입력을 해준다. 이는 위의 eq(7)에서 확인할 수 있다.



또한, cycle consistency를 위해 사용되기도 한다. Cycle flow의 일부는 다음과 같다. 또한, MIXUP 논문에서 영감을 받아 저자는 를 의 분포를 따르게끔 지정해주었다. 여기서 이고 는 number of total iteration이다. 즉, 처음에 는 시작시에 매우 작은 숫자이며 점점 iteration이 증가할수록 증가한다. 이는 uniform sampling 보다 더욱 안정된 training을 제공한다.

## **2.5 Boosting Domain Adaptation Models**

기존에 논의하던 내용이랑 동일한 내용이다. DA를 위해서 대신 domainness variable 를 적용시킨 에서 sampling한 data를 이용해 DA를 진행하는 점이 boost의 개념이다. 이 때 이다. 이 때이다. Target domain과 조금 더 가깝고 비슷한 domain이므로 training에 좋은 영향을 줄 수 있다.

## **2.6 Style Generalization**

기존의 img2img의 method는 target style이 fixed style이므로 오직 한 가지 style로만 translation을 진행할 수밖에 없었다. 그러나 DLOW model에서는 를 변환시키면서 다양한 스타일을 만들어낼 수 있다. 또한, multiple target domain이 주어졌을 경우

# **3. Conclusion**

하기 싫다. 그런데 StarGAN이 생각나는 이유가 뭘까. Low-level에서 StarGAN을 분석한 논문 같은 느낌이기도 하다.