한양대학교 인공지능연구실

Unsupervised Image-to-Image Translation Networks

Ming-Yu Liu, Thomas Breuel, Jan Kautz

NIPS 2017

조건희

Index

Introduction

Introduction

해결하려는 문제 : **2 개의 도메인** 간 Image-to-Image translation

Input Output

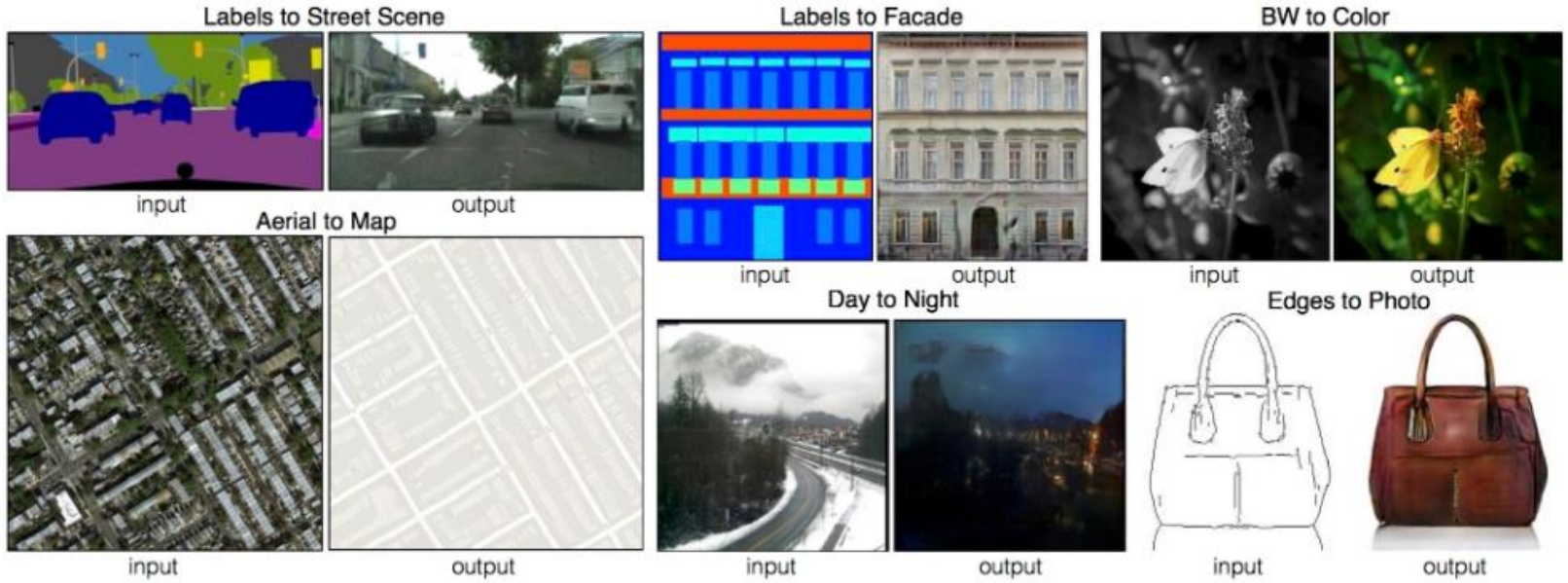
각 도메인 이미지의 pair data 가 존재하지 않음

→ **Unsupervised** problem!

Introduction

Image-to-Image translation

"Mapping an image in **one domain** to a corresponding image in **another domain**."



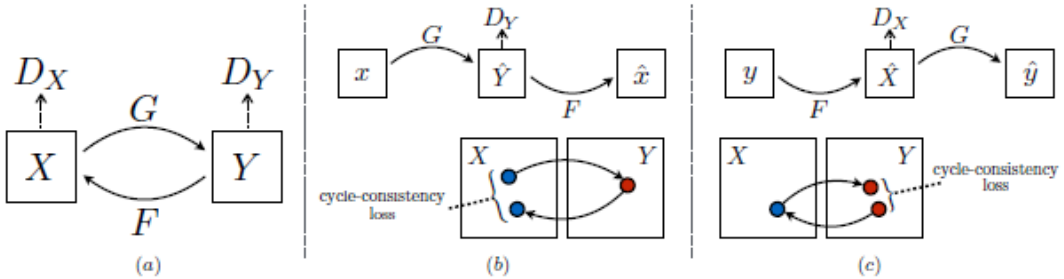
pix2pix[1]

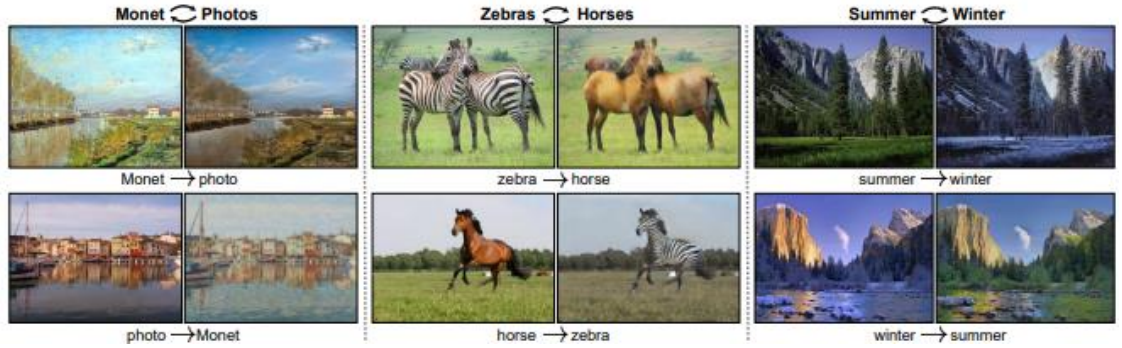
[1] : https://arxiv.org/pdf/1611.07004.pdf

Introduction

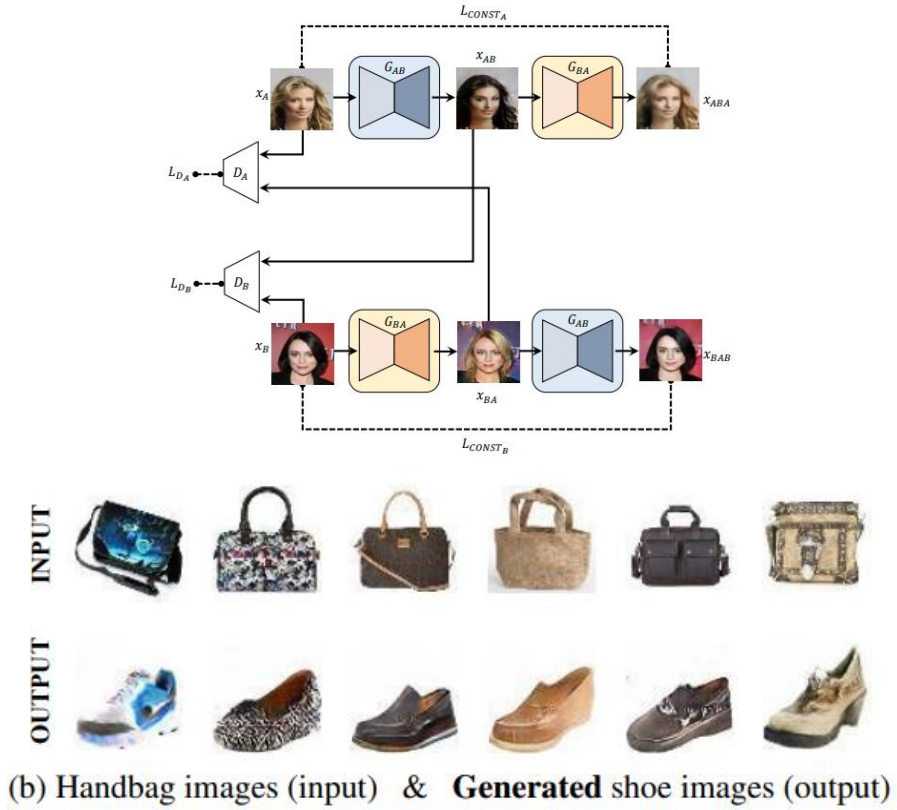
Image-to-Image translation

"Mapping an image in **one domain** to a corresponding image in **another domain**."





CycleGAN [2]



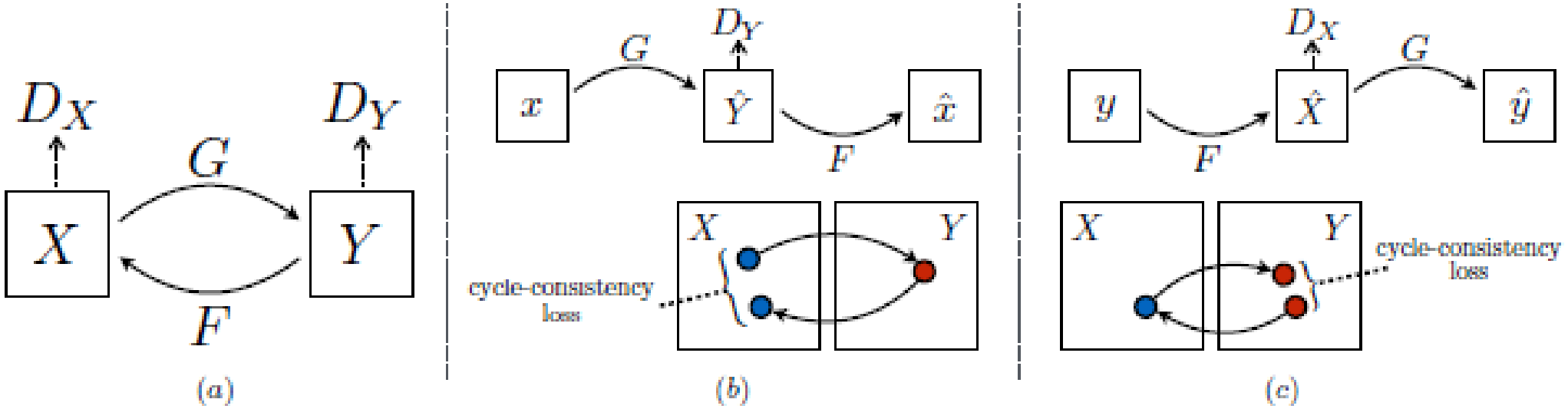
(b) Handbag images (input) & **Generated** shoe images (output)

DiscoGAN [3]

[2] : https://arxiv.org/pdf/1703.10593v6.pdf
[3] : https://arxiv.org/pdf/1703.05192.pdf

Introduction

CycleGAN 도 unpaired 이미지 변환에 대한 연구

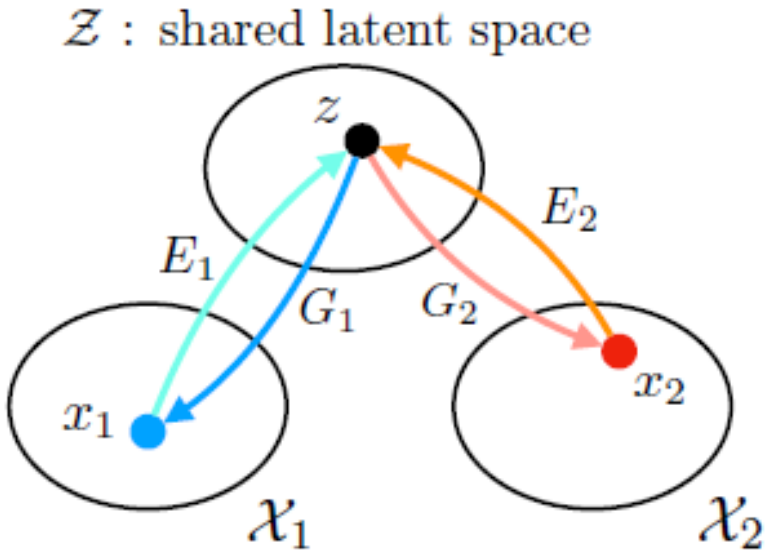


Cycle-consistency loss 이용

X 도메인 → Y 도메인 → X 도메인 으로 변환했을 때 다시 원본 이미지와 비슷해지도록

Introduction

이 연구에서는 어떻게 했을까?

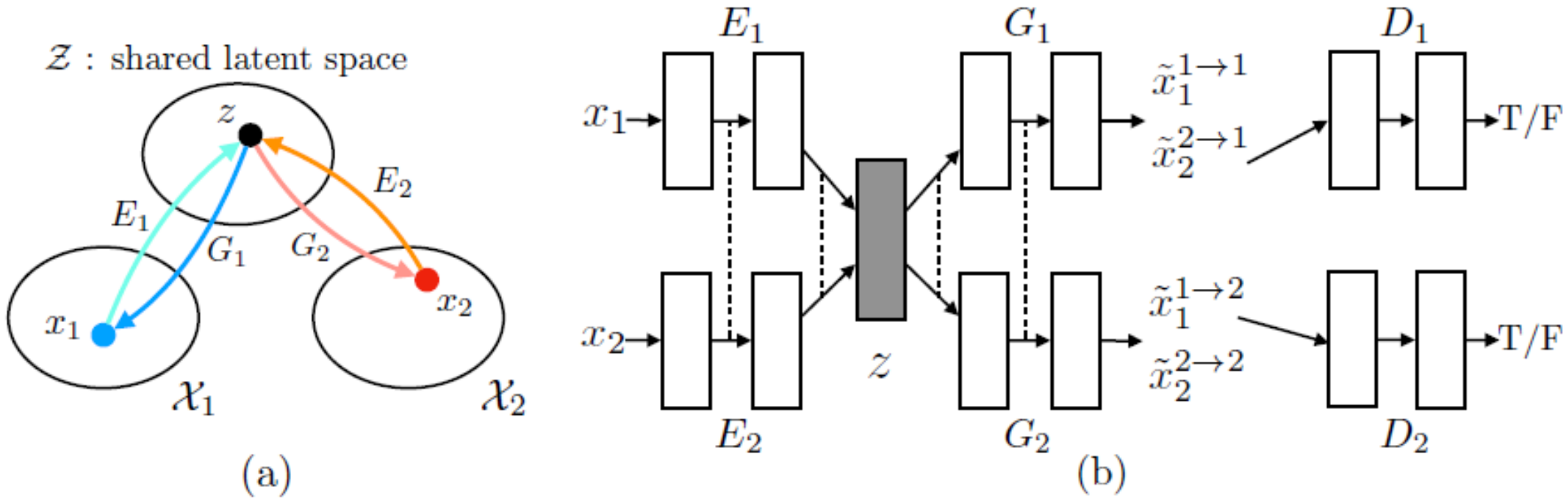


Shared latent space assumption

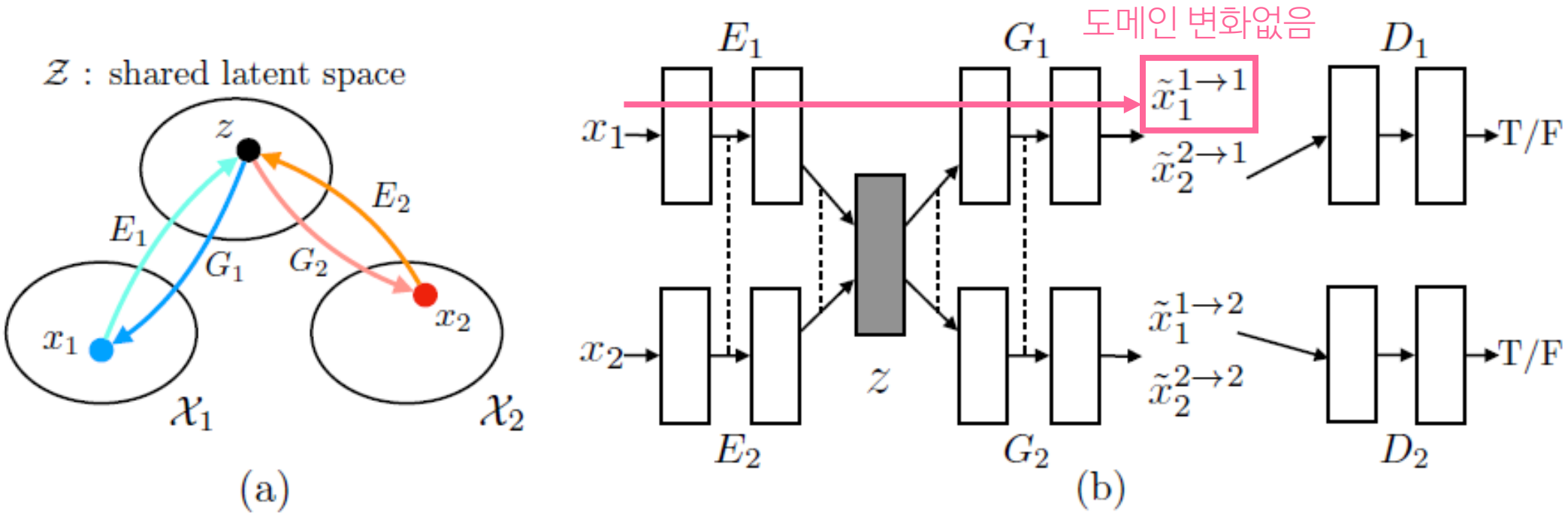
각 도메인에 대한 오토인코더(VAE) 를 학습할 때, 그 latent space 가 동일한 vector space 가 될 수 있지 않을까?

UNsupervised **I**mage-to-image **T**ranslation

UNsupervised Image-to-image Translation



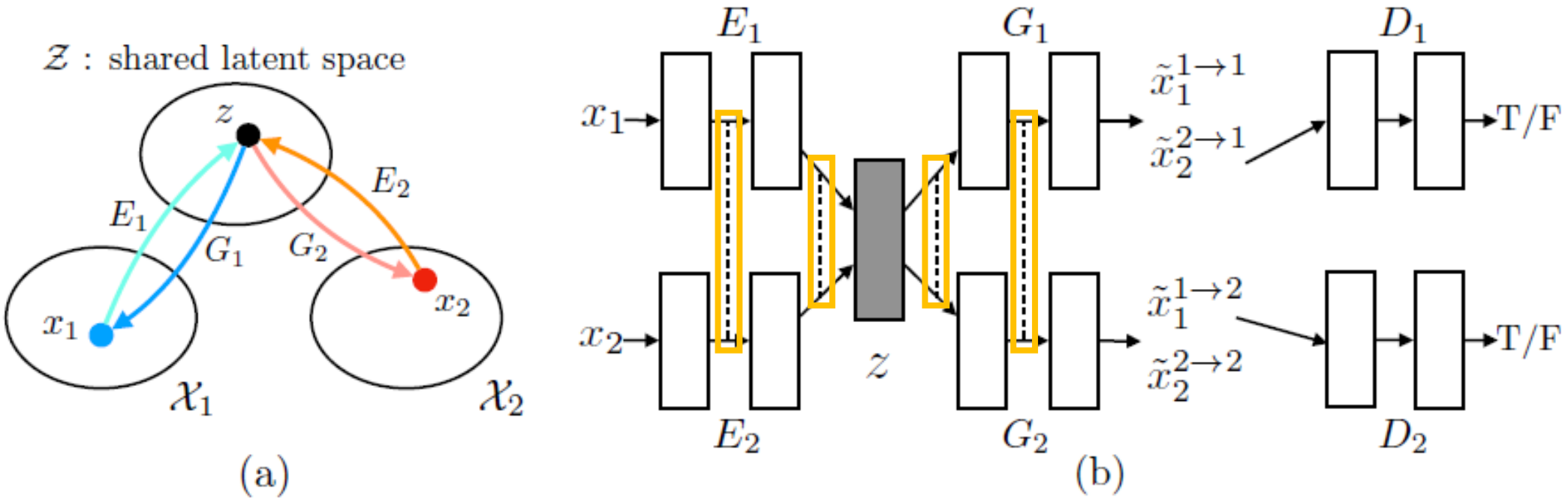
UNsupervised Image-to-image Translation



$\boldsymbol{x_1 \to E_1 \to G_1 \to \tilde{x}_1^{1\to1}}$: VAE[4]

[4] : https://arxiv.org/pdf/1312.6114.pdf

UNsupervised Image-to-image Translation

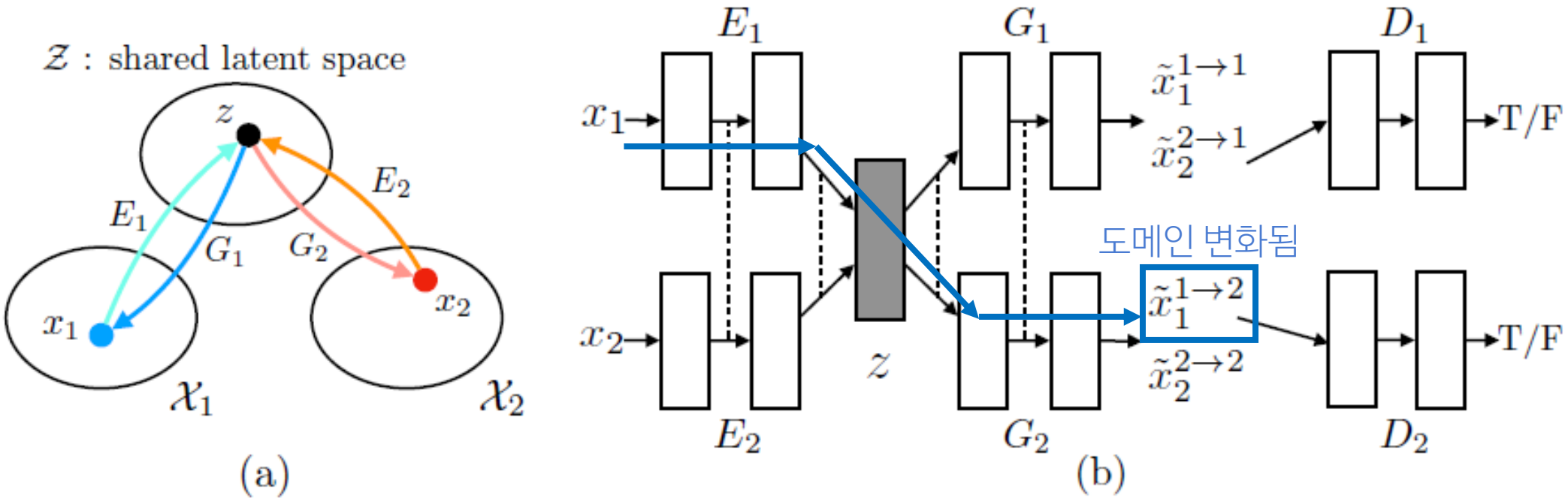


Weight sharing

각 도메인의 인코더 일부 레이어와 디코더(=generator) 일부 레이어의 weight 를 공유하여

두 도메인의 이미지가 common latent code 로 매핑되도록 함

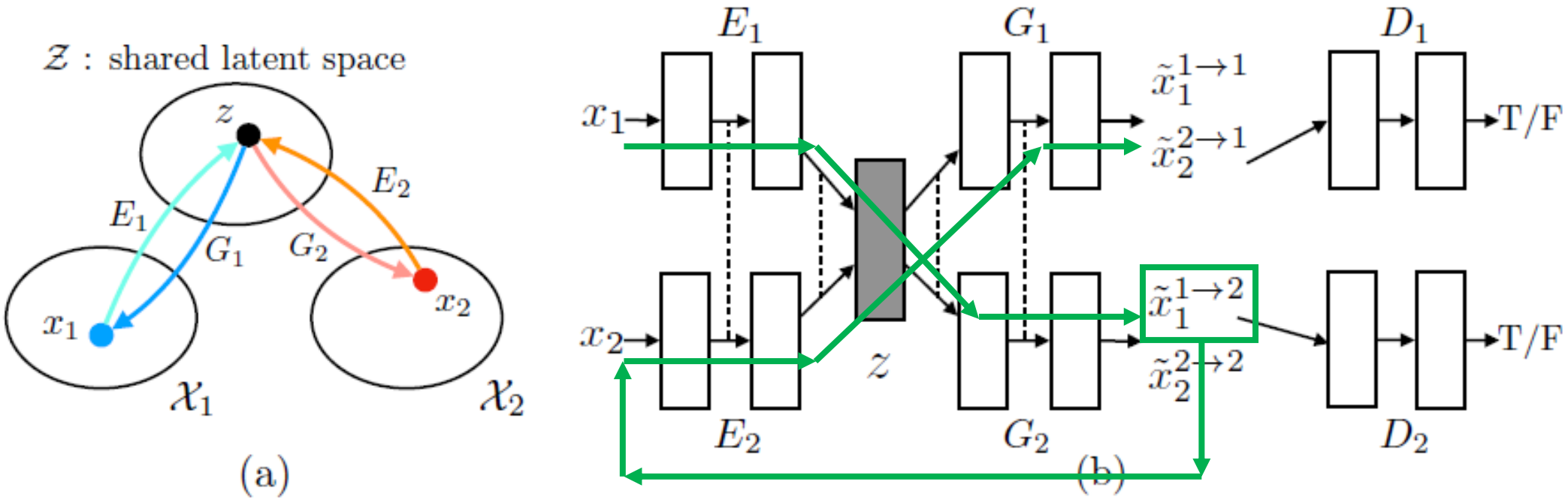
UNsupervised Image-to-image Translation



$$\boldsymbol{x_1 \to E_1 \to G_2 \to \tilde{x}_1^{1\to2} \to D_2}$$: GAN[5]

[5] : https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

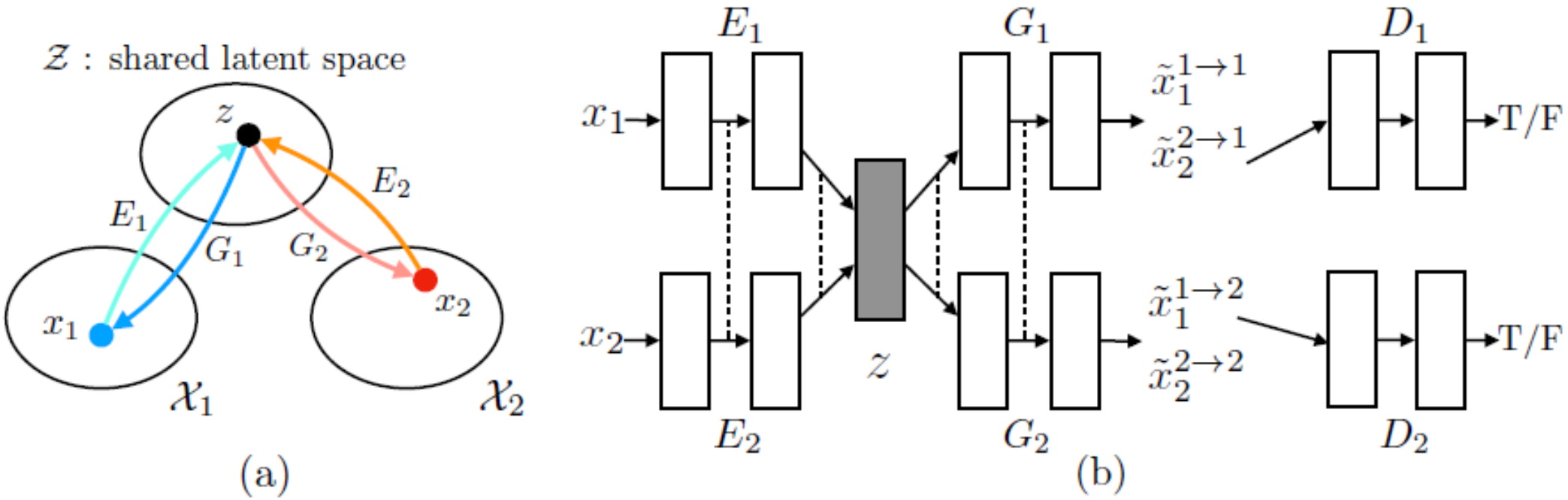
UNsupervised Image-to-image Translation



$$x_1 \to E_1 \to G_2 \to \tilde{x}_1^{1\to2} \to E_2 \to G_1 \to \tilde{x}_1^{1\to2\to1}$$

: Cycle-Consistency [2]

UNsupervised Image-to-image Translation



$$\min_{E_1,E_2,G_1,G_2} \max_{D_1,D_2} \mathcal{L}_{VAE_1}(E_1,G_1) + \mathcal{L}_{GAN_1}(E_2,G_1,D_1) + \mathcal{L}_{CC_1}(E_1,G_1,E_2,G_2) + \mathcal{L}_{VAE_2}(E_2,G_2) + \mathcal{L}_{GAN_2}(E_1,G_2,D_2) + \mathcal{L}_{CC_2}(E_2,G_2,E_1,G_1)$$

UNsupervised Image-to-image Translation

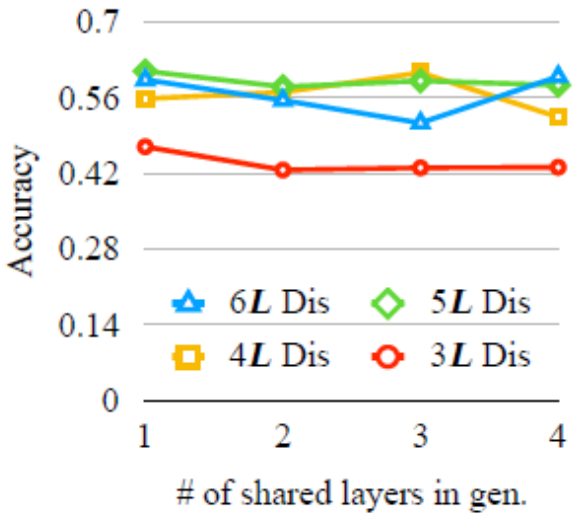
$$\mathcal{L}_{\text{VAE}_1}(E_1, G_1) = \lambda_1 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_1 \sim q_1(z_1|x_1)}[\log p_{G_1}(x_1|z_1)]$$
$$\mathcal{L}_{\text{VAE}_2}(E_2, G_2) = \lambda_1 \text{KL}(q_2(z_2|x_2)||p_\eta(z)) - \lambda_2 \mathbb{E}_{z_2 \sim q_2(z_2|x_2)}[\log p_{G_2}(x_2|z_2)]$$

$$\mathcal{L}_{\text{GAN}_1}(E_1, G_1, D_1) = \lambda_0 \mathbb{E}_{x_1 \sim P_{\mathcal{X}_1}}[\log D_1(x_1)] + \lambda_0 \mathbb{E}_{z_2 \sim q_2(z_2|x_2)}[\log(1 - D_1(G_1(z_2)))]$$
$$\mathcal{L}_{\text{GAN}_2}(E_2, G_2, D_2) = \lambda_0 \mathbb{E}_{x_2 \sim P_{\mathcal{X}_2}}[\log D_2(x_2)] + \lambda_0 \mathbb{E}_{z_1 \sim q_1(z_1|x_1)}[\log(1 - D_2(G_2(z_1)))]$$

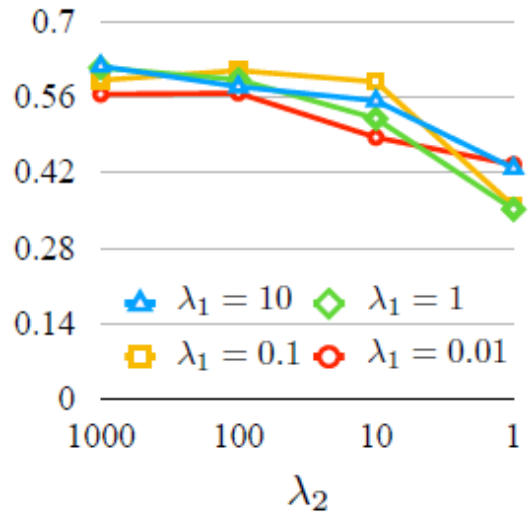
$$\begin{aligned}\mathcal{L}_{\text{CC}_1}(E_1, G_1, E_2, G_2) =& \lambda_3 \text{KL}(q_1(z_1|x_1)||p_\eta(z)) + \lambda_3 \text{KL}(q_2(z_2|x_1^{1\to 2}))||p_\eta(z)) - \\ & \lambda_4 \mathbb{E}_{z_2 \sim q_2(z_2|x_1^{1\to 2})}[\log p_{G_1}(x_1|z_2)] \\ \mathcal{L}_{\text{CC}_2}(E_2, G_2, E_1, G_1) =& \lambda_3 \text{KL}(q_2(z_2|x_2)||p_\eta(z)) + \lambda_3 \text{KL}(q_1(z_1|x_2^{2\to 1}))||p_\eta(z)) - \\ & \lambda_4 \mathbb{E}_{z_1 \sim q_1(z_1|x_2^{2\to 1})}[\log p_{G_2}(x_2|z_1)].\end{aligned}$$

Experiments

Experiments



(a) (b)



(c)

Method	Accuracy
~~Weight Sharing~~	0.569±0.029
~~Cycle Consistenc~~	0.568±0.010
Proposed	0.600±0.015

(d)

Figure 2: (a) Illustration of the Map dataset. Left: satellite image. Right: map. We translate holdout satellite images to maps and measure the accuracy achieved by various configurations of the proposed framework. (b) Translation accuracy versus different network architectures. (c) Translation accuracy versus different hyper-parameter values. (d) Impact of weight-sharing and cycle-consistency constraints on translation accuracy.

Experiments

Figure 3: Street scene image translation results. For each pair, left is input and right is the translated image.

Experiments

Figure 4: Dog breed translation results.

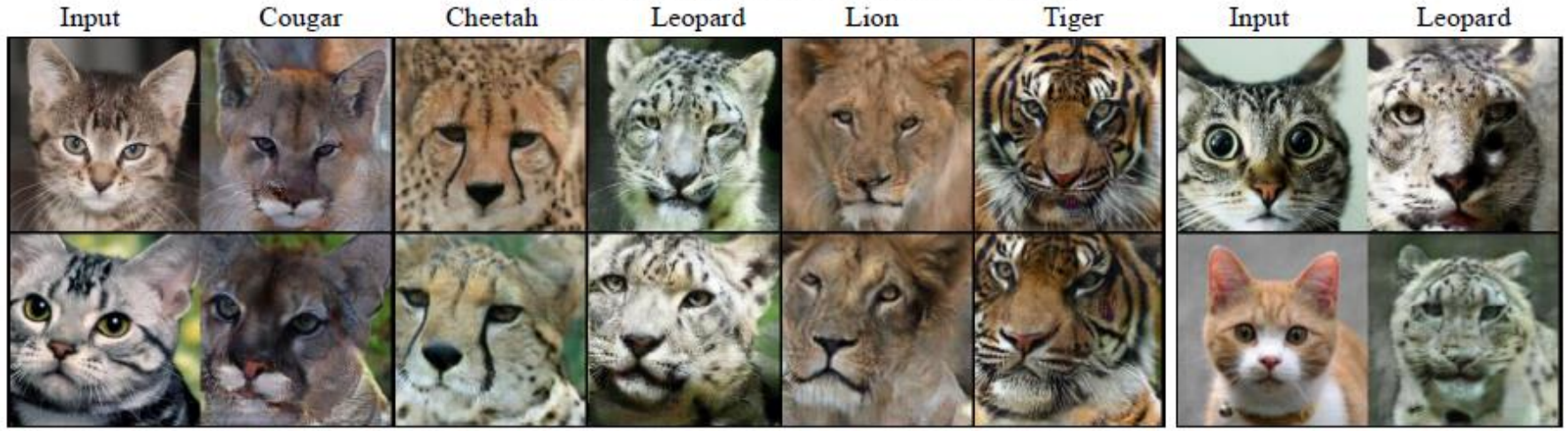


Figure 5: Cat species translation results.

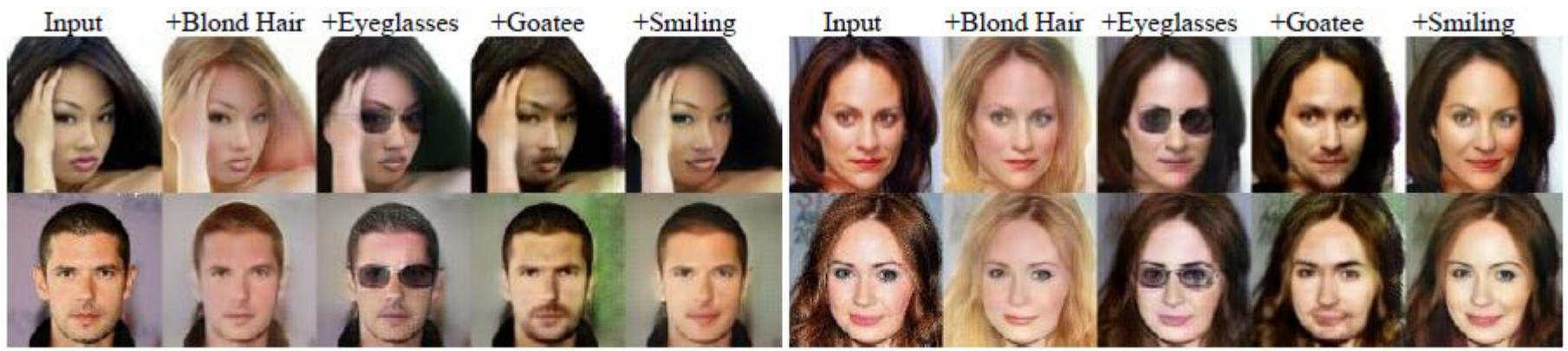
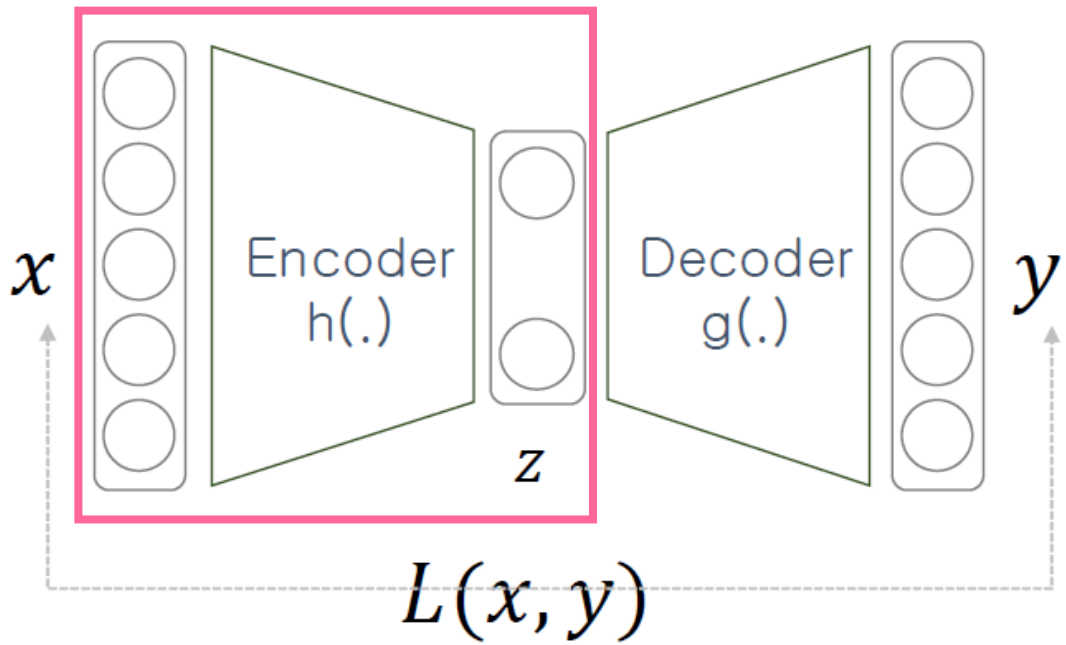


Figure 6: Attribute-based face translation results.

Variational **A**uto**E**ncoder

Variational AutoEncoder

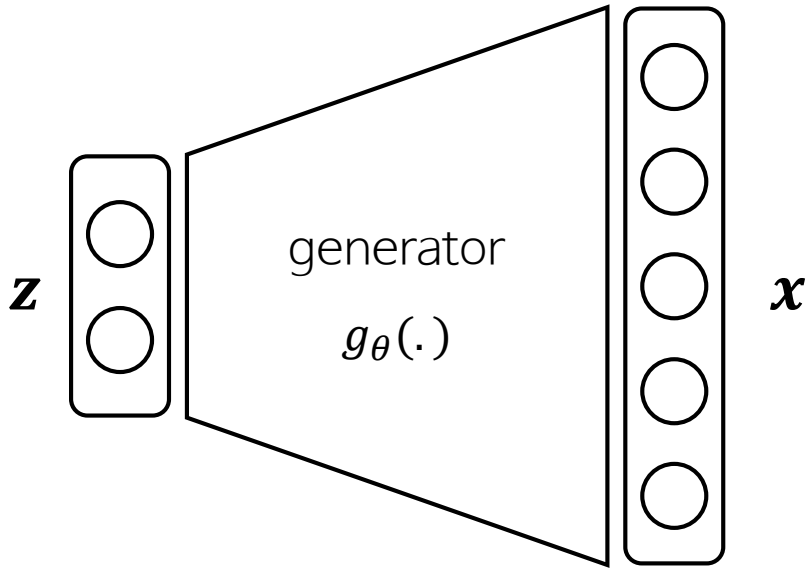
Auto Encoder



latent code 를 잘 만들기 위해서 디코더를 붙임

Variational AutoEncoder

VAE 는 generative model



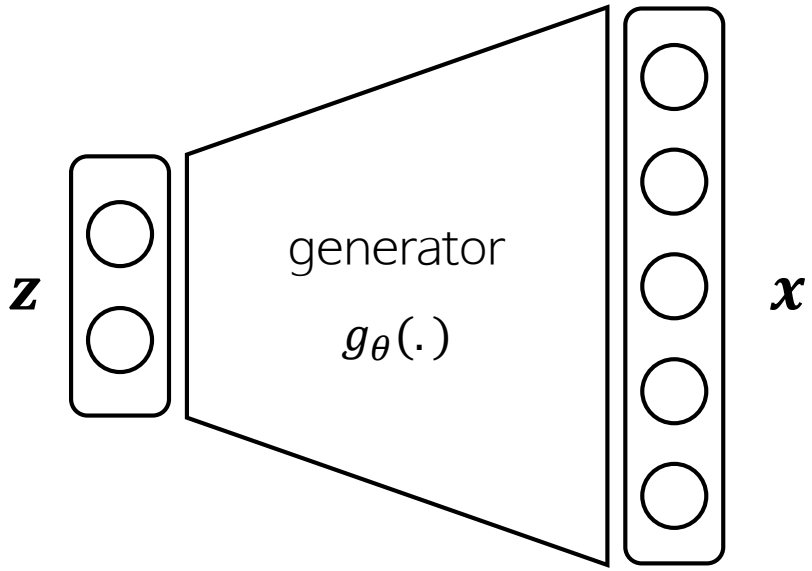
"Generative modeling" is a broad area of machine learning which deals with models of distributions P(X)

학습 데이터셋에 있는 데이터 샘플 x 와 비슷한 것을 생성하려고 함

"producing more examples that are **like** those already in a database, but **not exactly the same.**"

Variational AutoEncoder

VAE 는 generative model



We can formalize this setup by saying that we get examples X distributed according to some **unknown distribution $P_{gt}(X)$**, and our goal is to **learn a model P** which we can sample from, **such that P is as similar as possible to P_{gt}.**

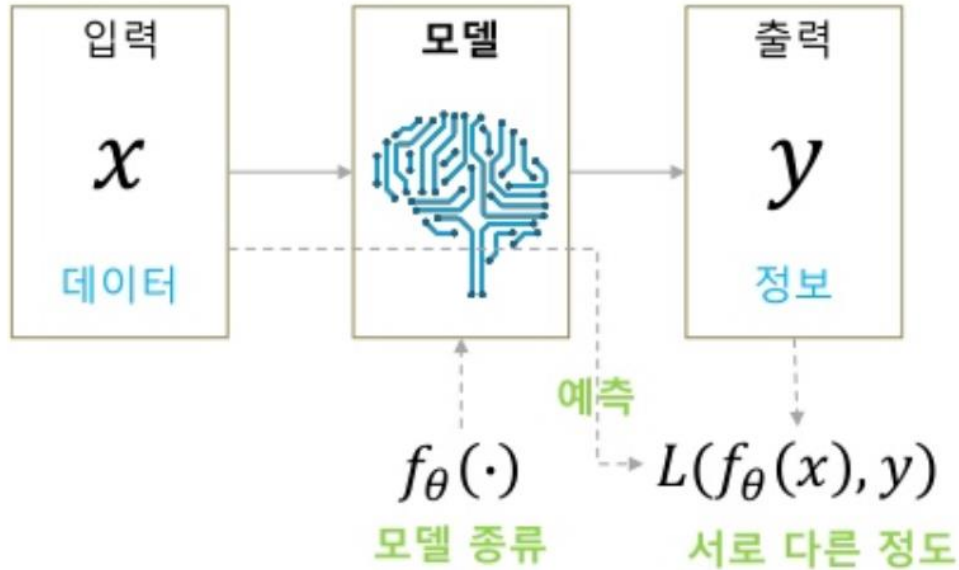
X 의 실제 분포(P_{gt})를 모르니 가능한 한 비슷한 분포(P)를 학습하려고 함

Variational AutoEncoder

네트워크 출력을 계속 “분포”라고 말하는 이유?

네트워크 출력을 Maximum likelihood 관점에서 해석.

Variational AutoEncoder



$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, L(f_\theta(x), y)$$

주어진 데이터를 제일 잘 설명하는 모델 찾기

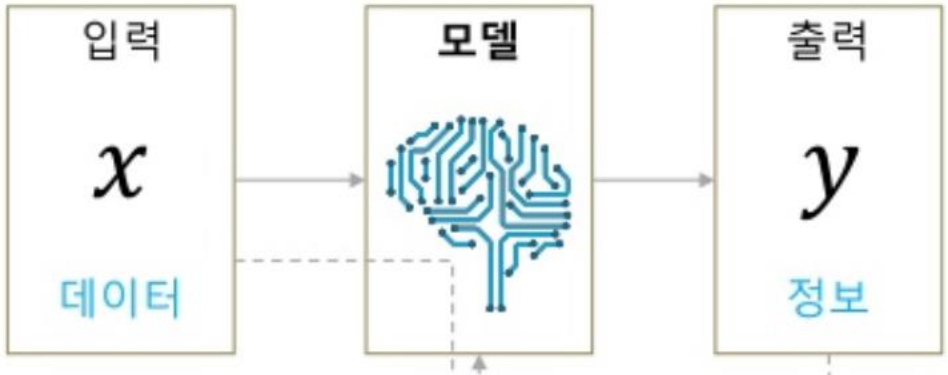
$$y_{new} = f_{\theta^*}(x_{new})$$

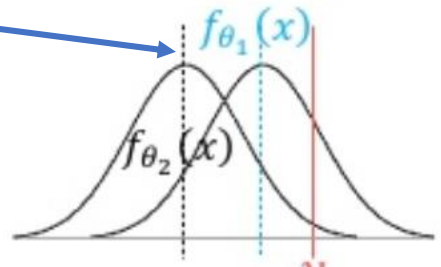
고정 입력, 고정 출력

Variational AutoEncoder

네트워크 출력에서
우리가 원하는 y 가 나올 확률(likelihood)이
높아지기(maximum)를 원함

가우시안이라고 가정하면
네트워크 출력값을
가우시안의 평균값으로 해석





$p(y|f_{\theta_2}(x)) < p(y|f_{\theta_1}(x))$

예측

$f_\theta(\cdot)$ 모델 종류

$p(y|f_\theta(x))$ 정해진 확률분포에서 출력이 나올 확률

이 관점에서는 y 에 대한 확률 분포가
무슨 모델인지 미리 정함
(가우시안, 베르누이 등)

Loss 에 log 가 붙은 이유는
Backpropagation 알고리즘을
적용하기 위한 조건에 맞추려고

$$\theta^* = \underset{\theta}{\operatorname{argmin}}[-\log(p(y|f_\theta(x)))]$$

주어진 데이터를 제일 잘 설명하는 모델 찾기

$$y_{new} \sim p(y|f_{\theta^*}(x_{new}))$$

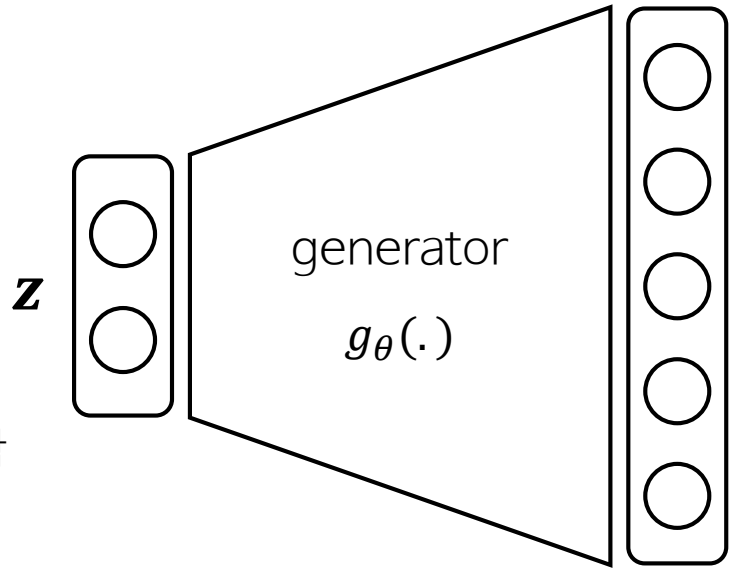
고정 입력, 고정/다른 출력

Variational AutoEncoder

$$z \sim p(z)$$

z의 확률분포는 다루기 쉬운 확률 분포 중 선택함

Normal / Uniform distribution



$$p(x|g_\theta(z))$$

g 가 z 라는 random variable 을 입력 받아 X 를 잘 만들어낼 확률을 높이도록 네트워크 파라미터 θ 를 조정

모든 training dataset에 대한 확률

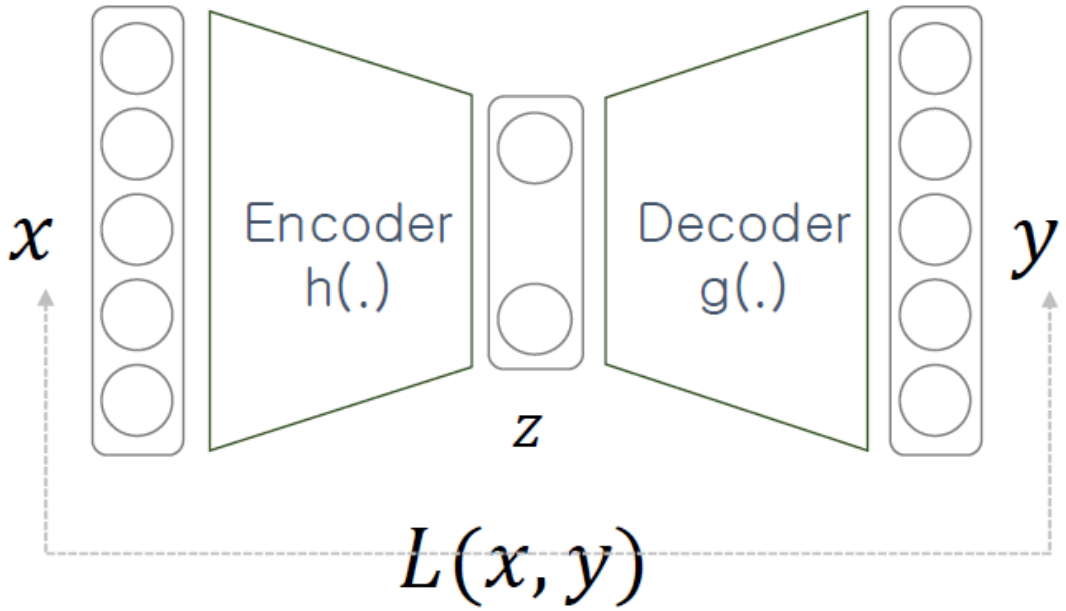
$$p(x) = \int p(x|g_\theta(z))p(z)dz$$

$$\approx \sum_i p(x|g_\theta(z_i))p(z_i)$$

Variational AutoEncoder

Manifold 가정

: 고차원의 벡터를 저차원으로 표현할 수 있다는 가정



오토인코더 구조에서는 z 를 latent vector 로 보고 manifold 상에 존재하는 것으로 해석하는데, 이 z의 분포를 단순한 normal distribution 이라고 가정하면 문제가 되지 않을까?

Variational AutoEncoder

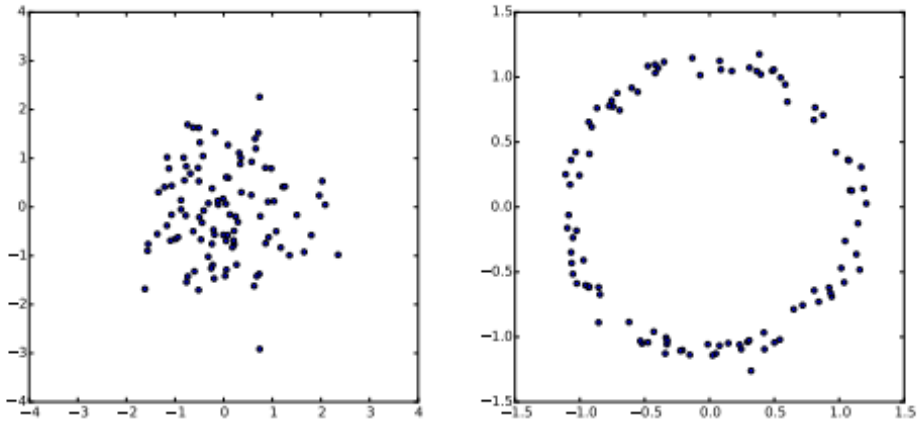


Figure 2: Given a random variable z with one distribution, we can create another random variable $X = g(z)$ with a completely different distribution. Left: samples from a gaussian distribution. Right: those same samples mapped through the function $g(z) = z/10 + z/||z||$ to form a ring. This is the strategy that VAEs use to create arbitrary distributions: the deterministic function g is learned from data.

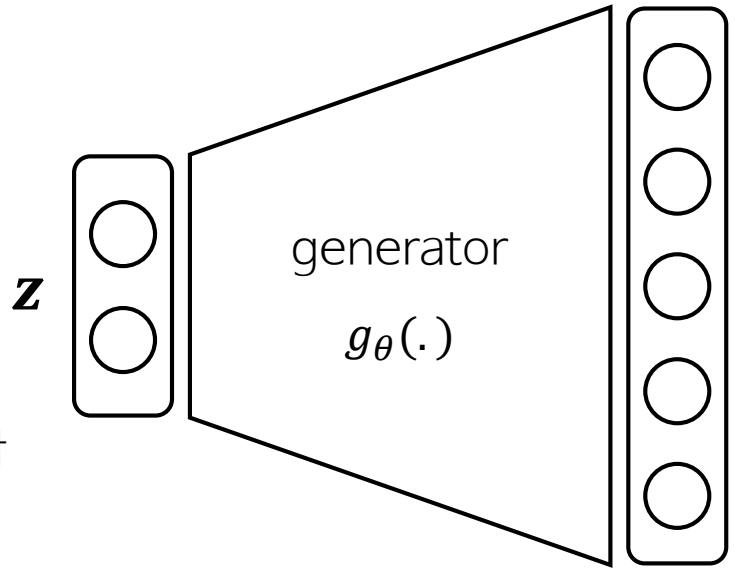
Generator 가 여러 개의 레이어를 가진 경우 처음 몇 개의 레이어가 latent space 로의 매핑을 수행해주고 나머지 레이어들이 그 latent vector 에 해당하는 이미지를 생성하게 된다고 함

Variational AutoEncoder

$$z \sim p(z)$$

z의 확률분포는 다루기 쉬운 확률 분포 중 선택함

Normal / Uniform distribution



$$p(x|g_\theta(z))$$

g 가 z 라는 random variable 을 입력 받아 X 를 잘 만들어낼 확률을 높이도록 네트워크 파라미터 θ 를 조정

모든 training dataset에 대한 확률

$$p(x) = \int p(x|g_\theta(z))p(z)dz$$

$$\approx \sum_i p(x|g_\theta(z_i))p(z_i)$$

Variational AutoEncoder

(a) 원하는 이미지

(c) 한 픽셀씩 shift된 이미지

(a) (b) (c)

(b) 앞쪽이 약간 잘린 이미지

Figure 3: It's hard to measure the likelihood of images under a model using only sampling. Given an image X (a), the middle sample (b) is much closer in Euclidean distance than the one on the right (c). Because pixel distance is so different from perceptual distance, a sample needs to be extremely close in pixel distance to a datapoint X before it can be considered evidence that X is likely under the model.

네트워크 출력에 대한 확률 모델을 가우시안으로 할 경우 MSE 관점에서 더 가까운 것으로 학습됨

(c) 가 의미적으로 (a) 에 더 가깝지만, MSE 는 (a)와 (b) 사이가 더 작음

$$\|x - g_\theta(z_b)\|^2 < \|x - g_\theta(z_c)\|^2$$

이러면 (b) 같은 출력이 likelihood 가 높으므로 학습이 제대로 되지 않음

Variational AutoEncoder

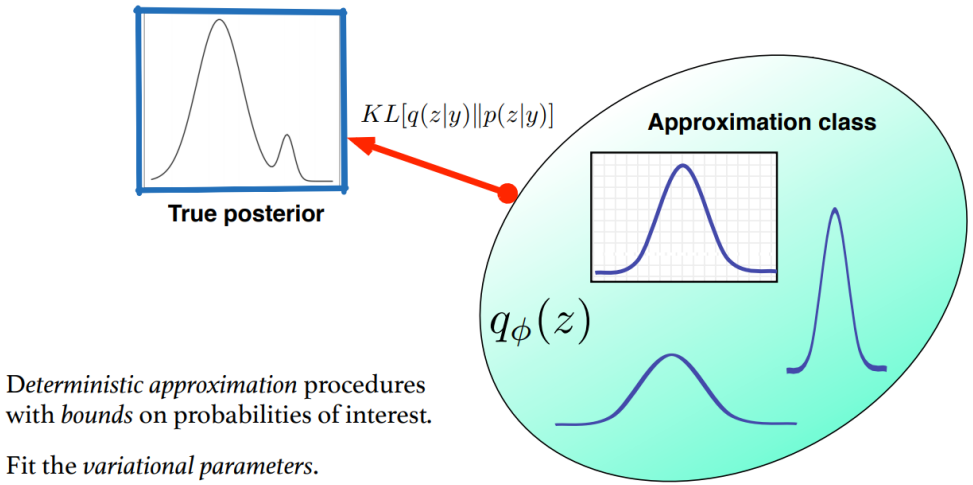
그래서 z 를 정규분포에서 샘플링하는 게 아니라 x 를 잘 생성해 줄 수 있는

이상적인 샘플링 함수 p(z|x)로부터 샘플링을 해보자

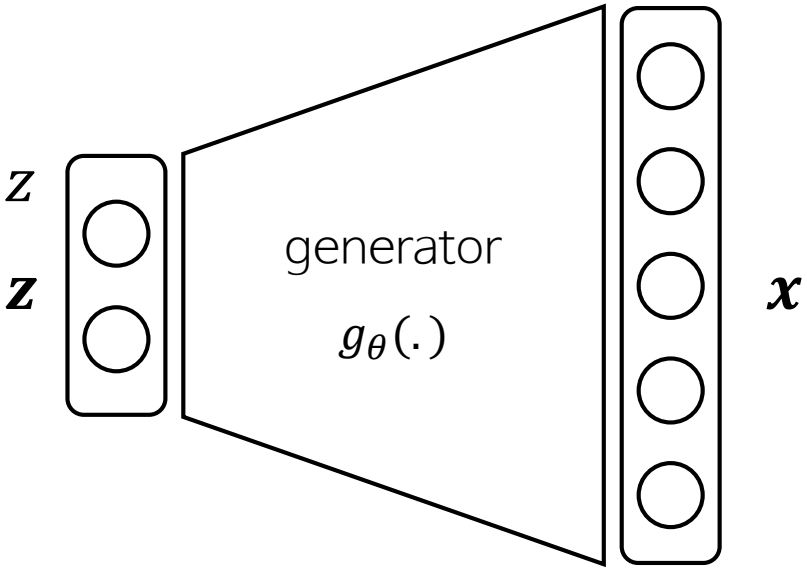
(x 를 evidence 로 줬으니까 x를 잘 만들겠지)

근데 p(z|x) 가 뭔지 모르니까 우리가 다루기 쉬운 확률 분포(예를 들면 가우시안) 중 하나를 가지고

파라미터를 조정해서 비슷하게 만들어 보자(=Variational Inference)



$$p(z|x) \approx q_\phi(z|x) \sim z$$



https://www.shakirm.com/papers/VITutorial.pdf

Variational AutoEncoder

$p(x), p(z|x), q_\phi(z|x)$ 사이의 관계식

$$\log(p(x)) = \int \log(p(x)) q_\phi(z|x) dz \quad \leftarrow \int q_\phi(z|x)dz = 1$$

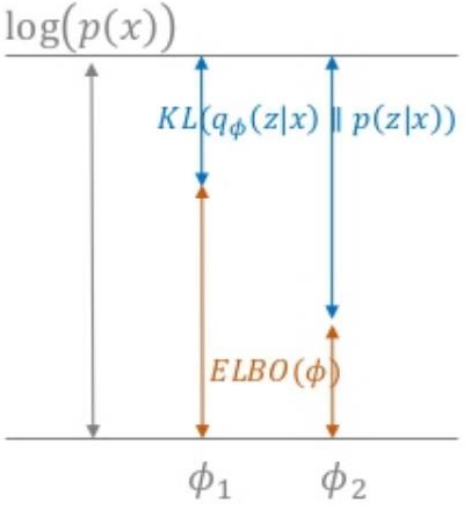
$$= \int \log\left(\frac{p(x,z)}{p(z|x)}\right) q_\phi(z|x)dz \quad \leftarrow p(x) = \frac{p(x,z)}{p(z|x)}$$

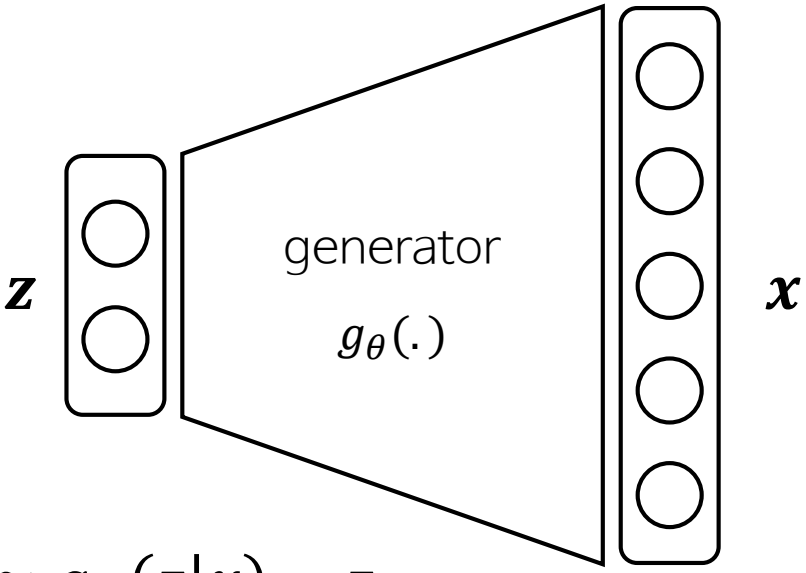
$$= \int \log\left(\frac{p(x,z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x)dz$$

$$= \underbrace{\int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz}_{ELBO(\phi)} + \underbrace{\int \log\left(\frac{q_\phi(z|x)}{p(z|x)}\right) q_\phi(z|x)dz}_{KL\left(q_\phi(z|x) \,\|\, p(z|x)\right)}$$

두 확률분포 간의 거리≥0

Evidence Lower Bound (ELBO)





KL term 을 최소화하는 것은 ELBO 를 최대화 하는 것과 동일함

$$p(z|x) \approx q_\phi(z|x) \sim z$$

Variational AutoEncoder

$$\log(p(x)) = ELBO(\phi) + KL(q_\phi(z|x)|p(z|x))$$

$$q_{\phi^*}(z|x) = \underset{\phi}{\operatorname{argmax}}\, ELBO(\phi)$$

이상적인 샘플링 함수를 approximation 하기 위해 파라미터 ϕ 를 조절하여 ELBO 를 최대화 해야함

$$ELBO(\phi) = \int \log\left(\frac{p(x,z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz$$

$$= \int \log\left(\frac{p(x|z)p(z)}{q_\phi(z|x)}\right) q_\phi(z|x)dz$$

$$= \int \log(p(x|z)) q_\phi(z|x)dz - \int \log\left(\frac{q_\phi(z|x)}{p(z)}\right) q_\phi(z|x)dz$$

$$= \mathbb{E}_{q_\phi(z|x)}[\log(p(x|z))] - KL(q_\phi(z|x)||p(z))$$

앞 슬라이드에서의 KL과 인자가 다른 것에 유의

ELBO 를 다시 전개한 최종수식

Variational AutoEncoder

Optimization problem 1 on ϕ

$$q_{\phi^*}(z|x) = \underset{\phi}{\operatorname{argmax}}\, ELBO(\phi)$$

$$ELBO(\phi) = \mathbb{E}_{q_\phi(z|x)}\left[\log\left(p(x|z)\right)\right] - KL\left(q_\phi(z|x)\middle||p(z)\right)$$

Optimization problem 2 on θ (Maximum likelihood 관점)

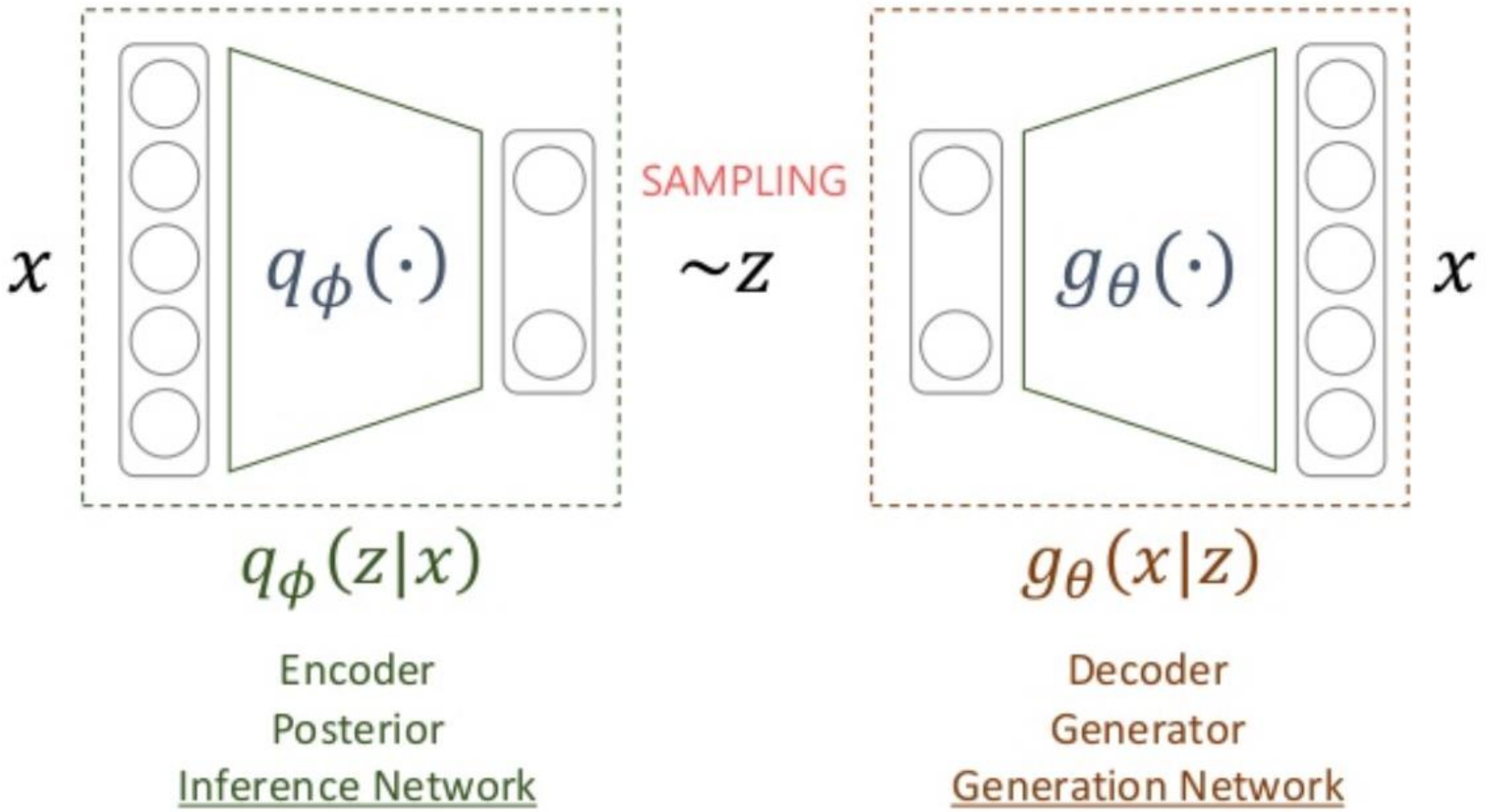
$$\arg\min_\theta \left(-\sum_i \log\left(p(x_i)\right)\right) = \arg\min_\theta \left(-\sum_i \left\{ELBO(\phi) + KL(q_\phi(z|x)||p(z|x)\right\}\right)$$

Final Optimization problem

$$\arg\min_{\phi,\theta} \sum_i -\mathbb{E}_{q_\phi(z|x_i)}\left[\log\left(p(x_i|g_\theta(z))\right)\right] + KL(q_\theta(z|x)||p(z))$$

Variational AutoEncoder

이상적인 샘플링함수를 approximation 하는 함수가 $q_\phi(.)$



The mathematical basis of VAEs actually has relatively little to do with classical autoencoders.

Variational AutoEncoder

Variational inference를 위한 approximation class 중 선택

원 데이터에 대한 likelihood 선택

다루기 쉬운 확률 분포 중 선택

$$L_i(\phi, \theta, x_i) = -\mathbb{E}_{q_\phi(z|x_i)}\left[\log\left(p_\theta(x_i|z)\right)\right] + KL\left(q_\phi(z|x_i)\middle||p(z)\right)$$

Reconstruction Error

- 현재 샘플된 z에 대한 negative log likelihood
- x_i에 대한 복원 오차 (AutoEncoder 관점)

Regularization

- 현재 샘플된 z에 대한 대한 추가 조건
- 샘플링되는 z들에 대한 통제성을 prior를 통해 부여, Variational distribution q(z|x)가 p(z)와 유사해야 한다는 조건을 부여

Q & A

References

- Paper

  https://arxiv.org/pdf/1703.00848.pdf

- VAE 관련
  - Tutorial on Varialtional Autoencoders

    https://arxiv.org/pdf/1606.05908.pdf
  - 오토인코더의 모든 것 - 2/3

    https://www.youtube.com/watch?v=rNh2CrTFpm4&list=PLD7i1I-WJa6Qq4uTcVQNsmm8c3HrPfeNe&index=2

Thank you