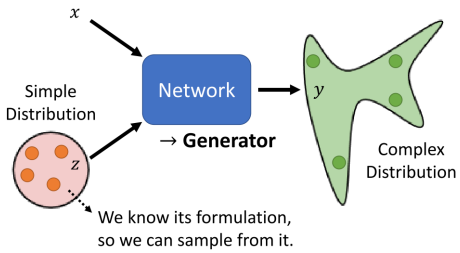
Network as Generator

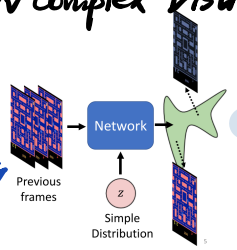
- Input x 需要加入一个从 Simple Distribution 中 sample 出的向量 z
  Add or Concat　已知分布 (Gaussian or Uniform)
- 加入的 z 的 Distribution 不同，Output 的 Complex Distribution 不同



Video Prediction

- Turn left or Turn Right
  → 不 Output 确定的结果，而是一个 Distribution
- 需要创造能力的 Task　包含了各种可能

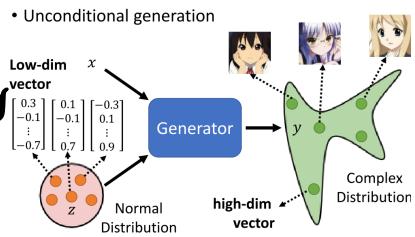
The same input has different outputs



GAN

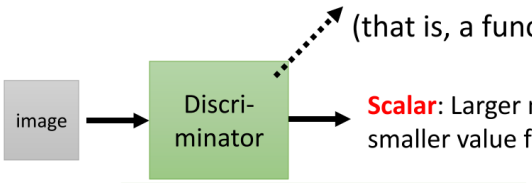
Unconditional Generation

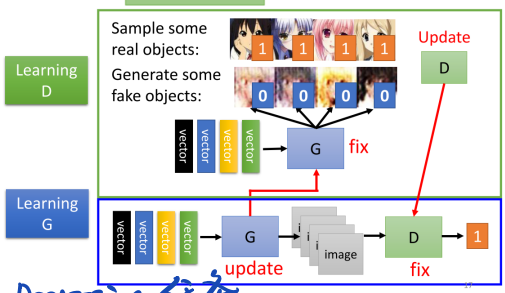
→ 不考虑 x，Input z，Output y
z 不同，y 不同　低维 → 高维
→ 简单 Distribution 就可以
不同的 Distribution 区别不大



Discriminator

Scalar 越大，Image 越真实



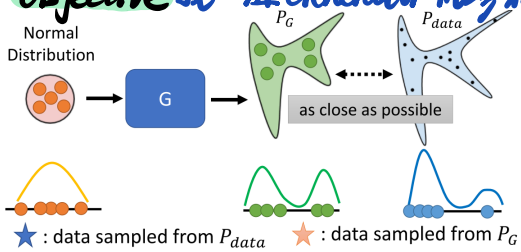


Algorithm

- 初始化 Generator 和 Discriminator
- In each iteration:
  - Fix Generator，update Discriminator　Classification or Regression 任务
    → 训练 Discriminator 学会分辨生成与真实的图片　给 Real 高分，给 Generated 低分
  - Fix Discriminator，update Generator　只调前几层的参数，不动 Discriminator 附近的参数
    → 训练 Generator 学会欺骗 Discriminator　Generator 和 Discriminator 合并起来
    中间某一个 Hidden Layer 的 Output 很宽
- 重复以上过程　用 Generator 生成的图片训练 Discriminator，再用 Discriminator 训练 Generator

Theory

Objective　找一组 Generator 的参数，使 P_G 与 P_{data} 的 Divergence 最小　只要能够 Sample 就能算

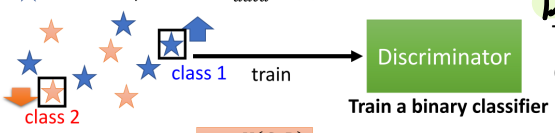


P_{data}：直接从 Database 中 Sample
P_G：从 Normal Distribution 中 sample vector 通过 Generator

Generator

$$G^* = \arg\min_G Div(P_G, P_{data})$$

Discriminator　Negative Cross Entropy

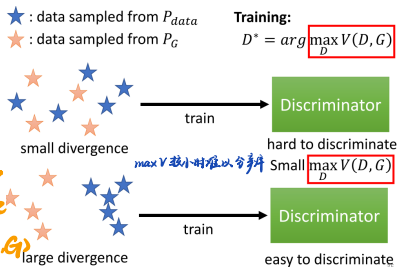


Training: $D^* = \arg\max_D V(D,G)$　The value is related to JS divergence.

Objective Function for D

$$V(G,D) = E_{y\sim P_{data}}[\log D(y)] + E_{y\sim P_G}[\log(1-D(y))]$$

JS Divergence　Divergence 与 $\max_D V(D,G)$ 有关



$G^* = \arg\min_G \max_D V(G,D)$

$D^* = \arg\max_D V(D,G)$　The maximum objective value is related to JS divergence.

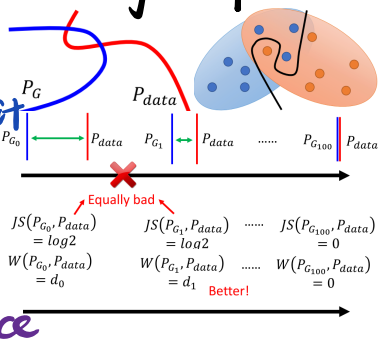
- Initialize generator and discriminator
- In each training iteration:
  - ***Step 1***: Fix generator G, and update discriminator D
  - ***Step 2***: Fix discriminator D, and update generator G

训练 Generator 需要求 Divergence 但无法直接计算
→ Divergence 和训练 Discriminator 的 Objective Function 相关
→ 使用 $\max_D V(D,G)$ 代替 $Div(P_G, P_{data})$　Discriminator 训练时目标函数的越小，分布越相似　最大值作为衡量两个分布间 Divergence 的方法
→ 通过设计 Objective Function 就可以使用不同的 Divergence 作衡量

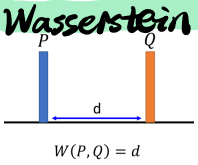
Tips for GAN

JS Divergence 的问题　P_G 和 P_{data} 通常没有 Overlap

- Nature of data　P_G 和 P_{data} 都是高维空间的低维映射，Overlap 可以忽略不计
- Sampling　即使有 Overlap，采样不够时能够轻易划分开，相当于没有 Overlap
  → 当两个 Distribution 没有 overlap，无法衡量距离，Accuracy 总是 100%（没有意义）



Wasserstein Distance



把土从 P 运到 Q 需要的平均距离

平均距离最小的移动方案 → Wasserstein Distance

有一个渐近的过程

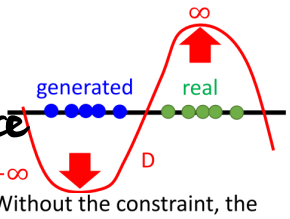
WGAN 使用 Wasserstein Distance 衡量 Pdata 和 PG 的 Divergence

Evaluate Wasserstein distance between P_{data} and P_G

$$\max_{D\in 1\text{-}Lipschitz}\{E_{y\sim P_{data}}[D(y)]-E_{y\sim P_G}[D(y)]\}$$

Discriminator 需要足够平滑
否则无法反映分布的距离 (与 JS Divergence 无异)

Restriction: 1-Lipschitz
→ 只有D足够平滑，结果才是 Wasserstein Distance
PG 和 Pdata 接近时 D 相差较小 (过大则不平滑) ∞
PG 和 Pdata 较远时 D 可以相差较大

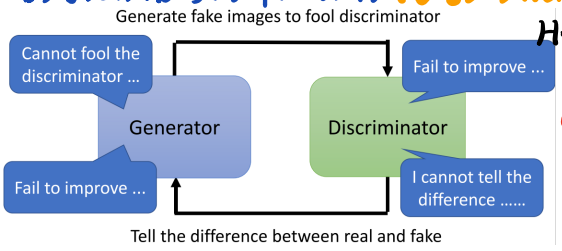


Without the constraint, the training of D will not converge.

Keeping the D smooth forces D(y) become ∞ and −∞

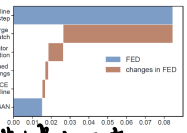
How to Realize?
- Original WGAN → Weight
- Improved WGAN → Gradient Penalty
- Spectral Normalization → Gradient Norm < 1 (SNGAN)

GAN 训练难点:
→ Generator 和 Discriminator 需要模逢对手 (match each other)
→ GAN for Sequence Non-differentiable

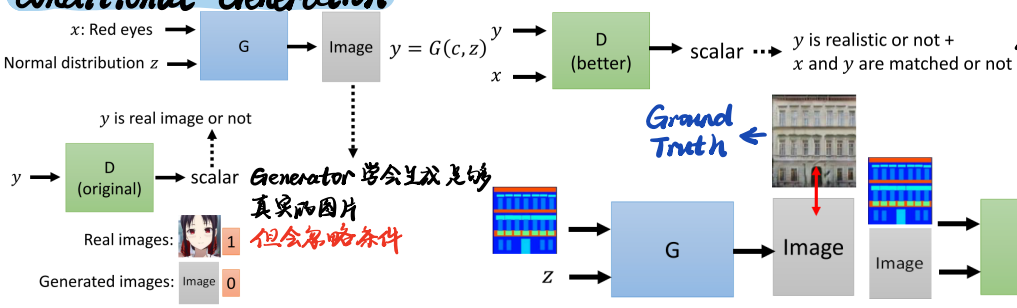


- Usually, the generator are fine-tuned from a model learned by other approaches.
- However, with enough hyperparameter-tuning and tips, ScarchGAN can train from scratch.

Training language GANs from Scratch
https://arxiv.org/abs/1905.09922



Conditional Generation

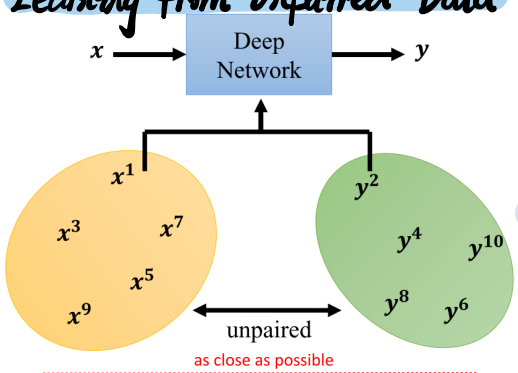


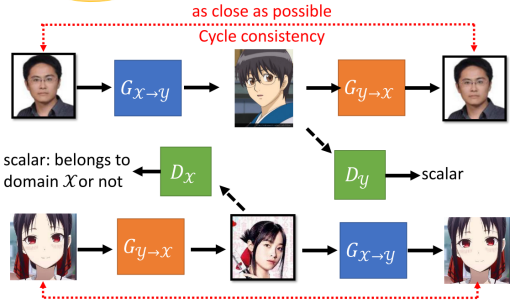
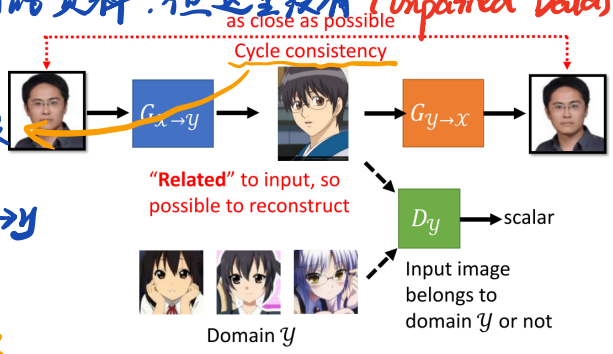
Generator 常会生成足够真实的图片
但会忽略条件

不仅判断是否真实，还要判别 x 与 y
即 Output 与 condition 是否匹配

Conditional GAN
+ Supervised
→ 可以得到更好的效果

Learning from Unpaired Data



从 Domain x → Domain y: Unsupervised Conditional Generation
原先从 Gaussian Distribution Sample → 从真人数据中 Sample
有可能 Generator 会忽略 Input，直接生成 Output
Conditional GAN 有成对的资料，但这里没有 (Unpaired Data)

Cycle GAN x→y, y→x
- Cycle Consistency
→ Input 与 Output 的相似度
- 训练时可以双向训练
$G_{x\to y}$、$G_{y\to x}$ & $G_{y\to x}$、$G_{x\to y}$
- Disco GAN、Dual GAN、Cycle GAN 做法几乎相同
Star GAN 多种风格间转换
- Text Style Transfer 文字风格转换
negative ↔ Postive