uOttawa

Generative Adversarial Nets

Shuang Xie 8912983
Jingyi Zou

GANs

Basic approach

Generative models

Discriminative models

How do GANs work

GANs

Basic approach

Generative models

Discriminative models

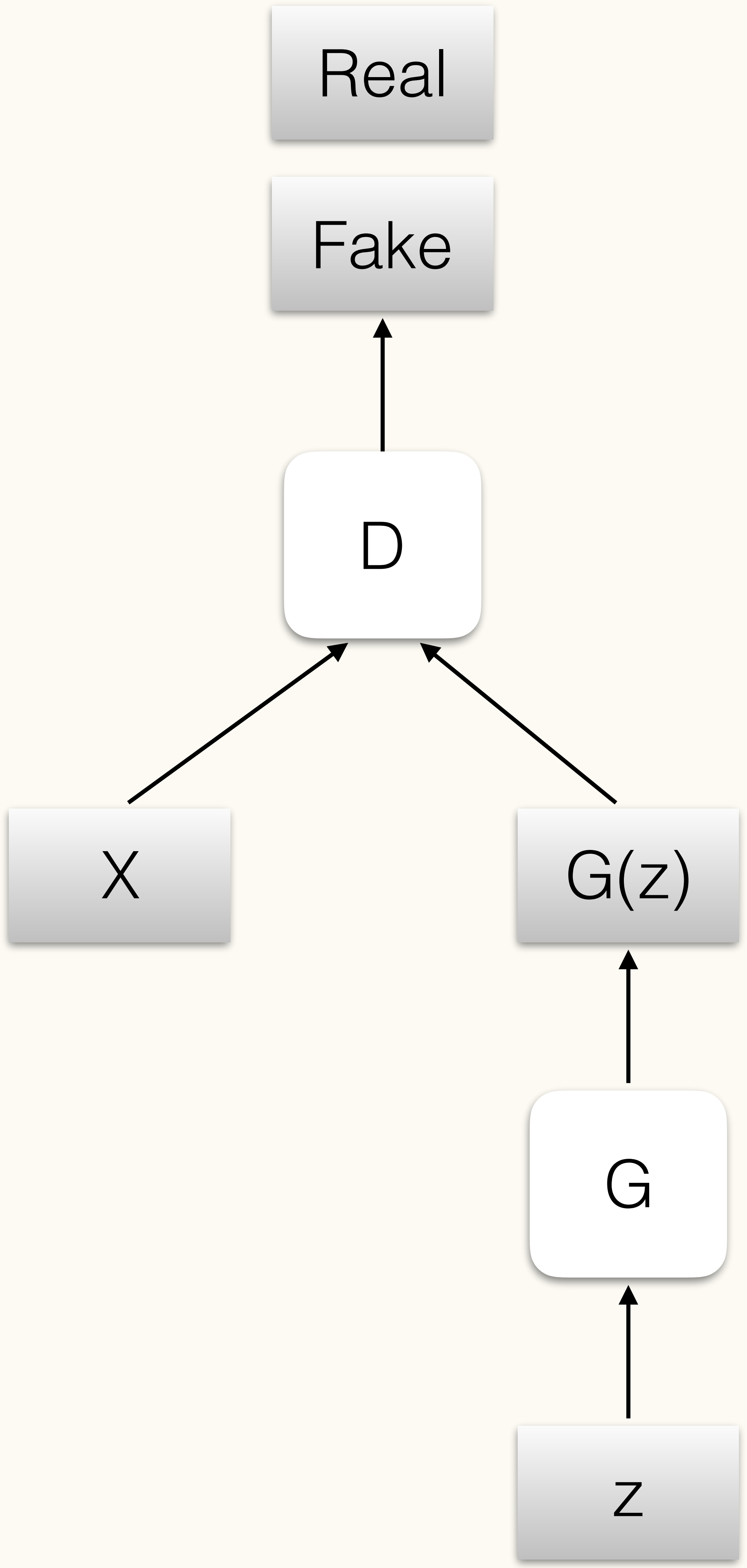
How do GANs work

Basic approach

Purpose:
Train a desire Generative Model

Idea:
Add a Discriminative Model

Basic approach



Basic approach

Set up **a game** between two neural nets

- the **generator** creates samples
- the **discriminator** classifies these samples as real or fake
- both **train together** together or in turns?

我觉得这边讲的时候可以把Generator和Discriminator的互相作用这一块着重讲一下：G的作用是尽可能生成可以以假乱真的图片，D的作用是尽可能分辨出图片的真伪。随着G生成的图片越来越真实，又会激励D的分辨能力变强；反之，D的分辨能力变强，又会激励G生成的图片更加可以以假乱真。G和D构成的是一个动态的“博弈过程”。

然后我觉得在这边可以把总的一个核心公式加上：min max...那个，然后再用数学方法解释一下G和D的作用。

GANs

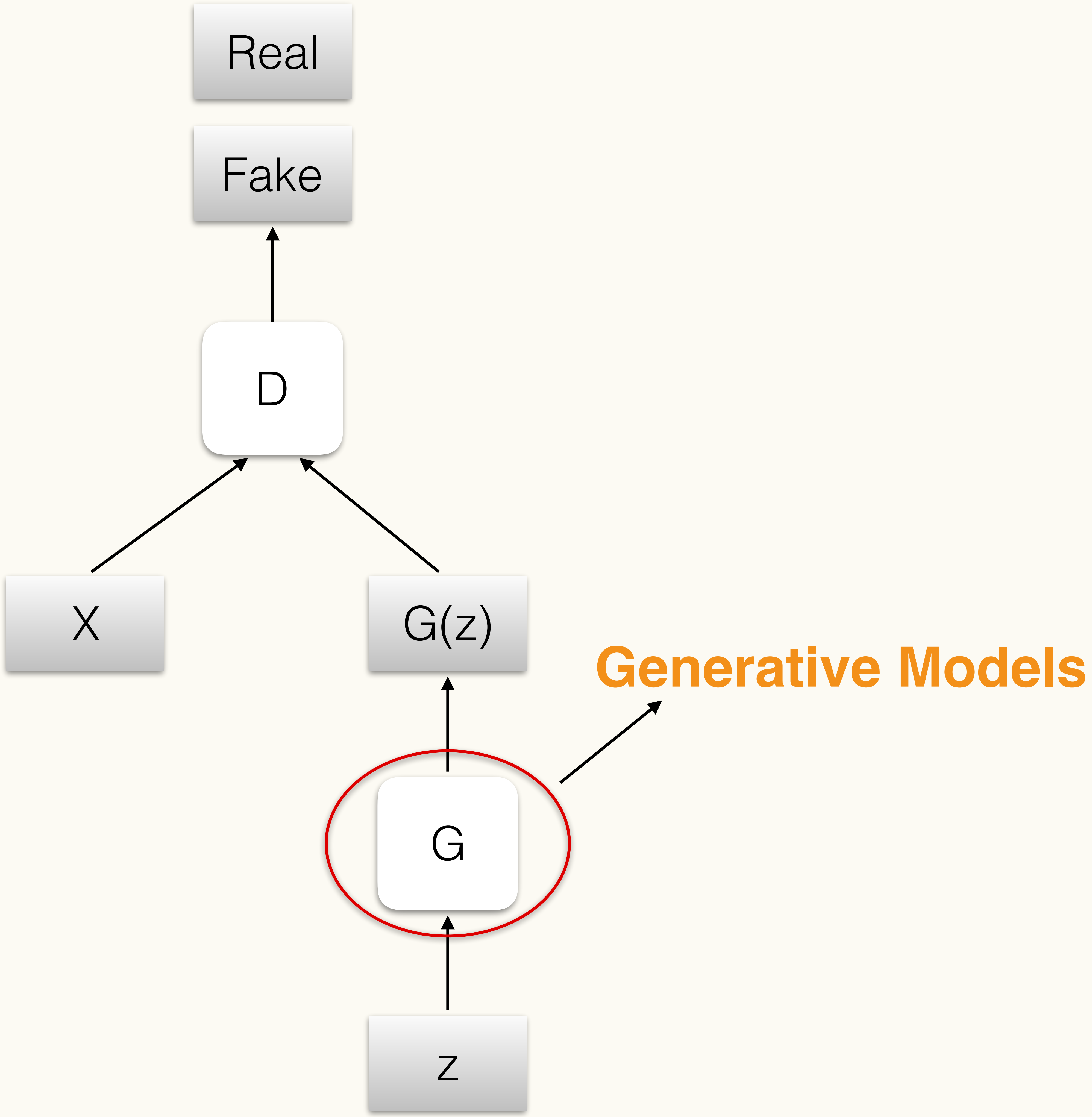
Basic approach

Generative models

Discriminative models

How do GANs work

GANs

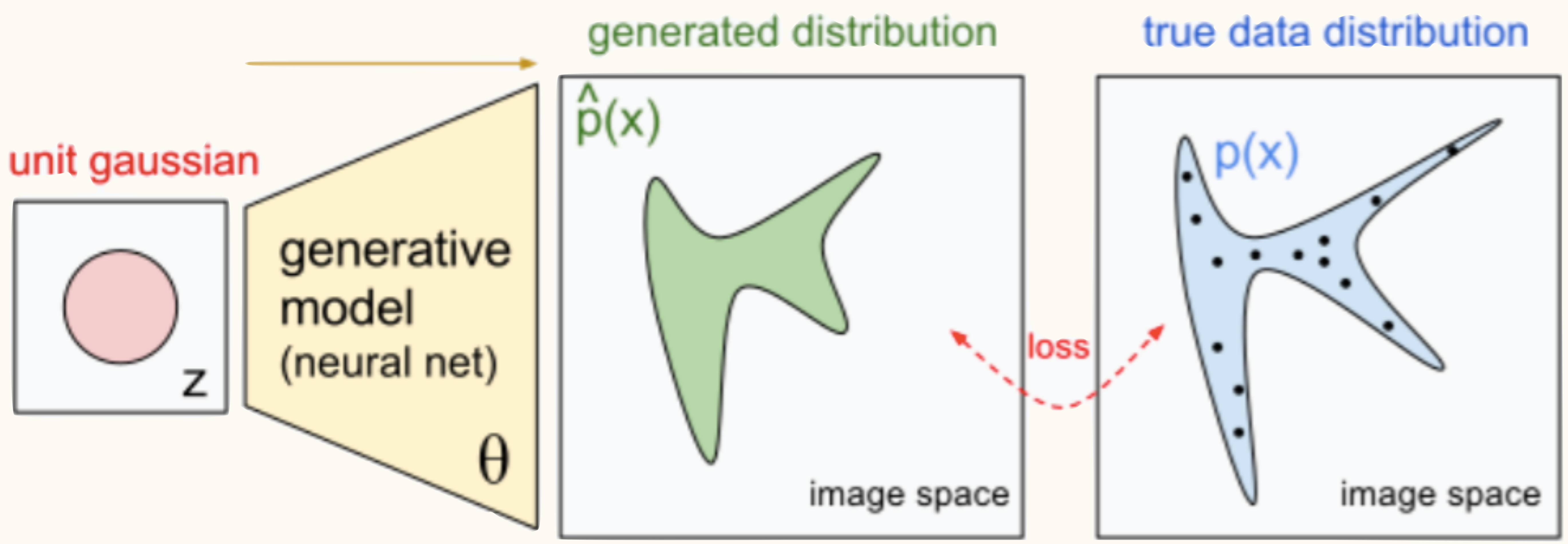


Generative Models

- G trying to recover the **training data distribution**
- The generative model can be thought of as analogous to a team of counterfeiters trying to **produce fake currency** and use it without detection

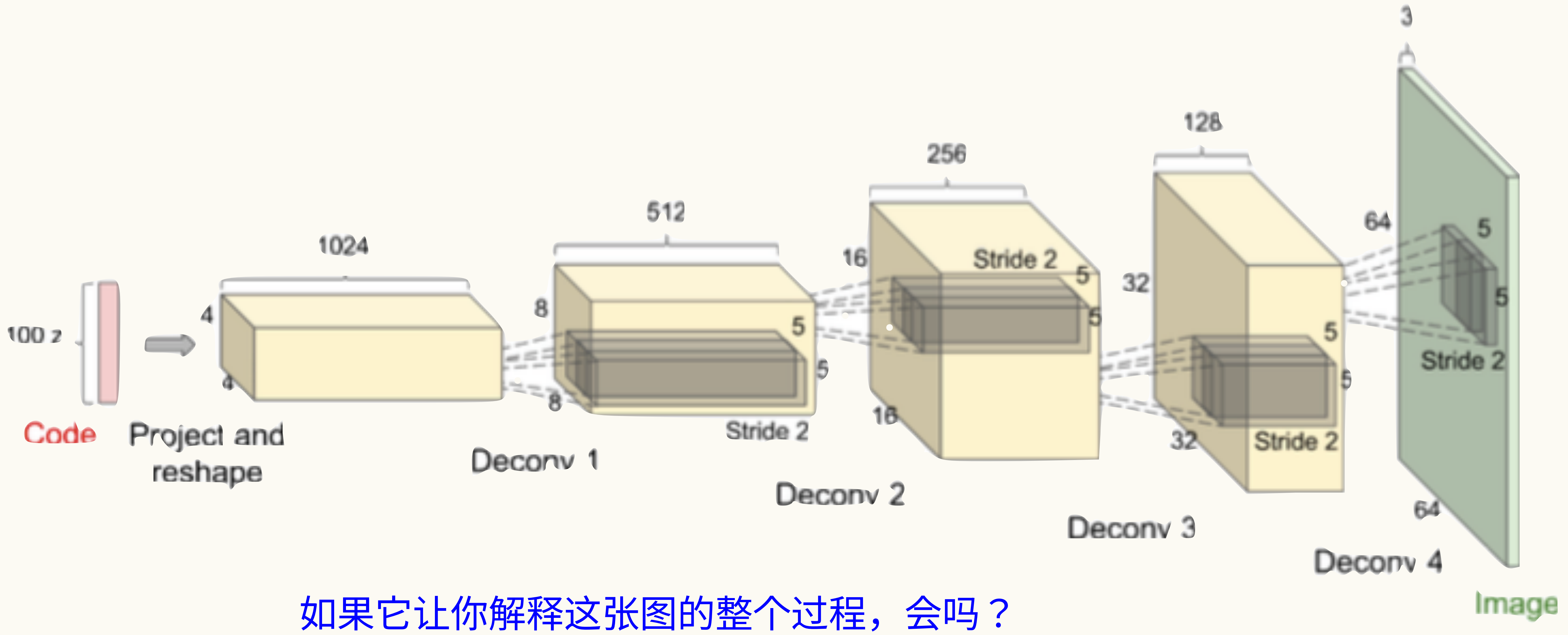
Generative Models

Training Approach



Generative Models

- Upsampling with fractionally-strided convolution
- Opposite of convolutional neural nets



如果它让你解释这张图的整个过程，会吗？

GANs

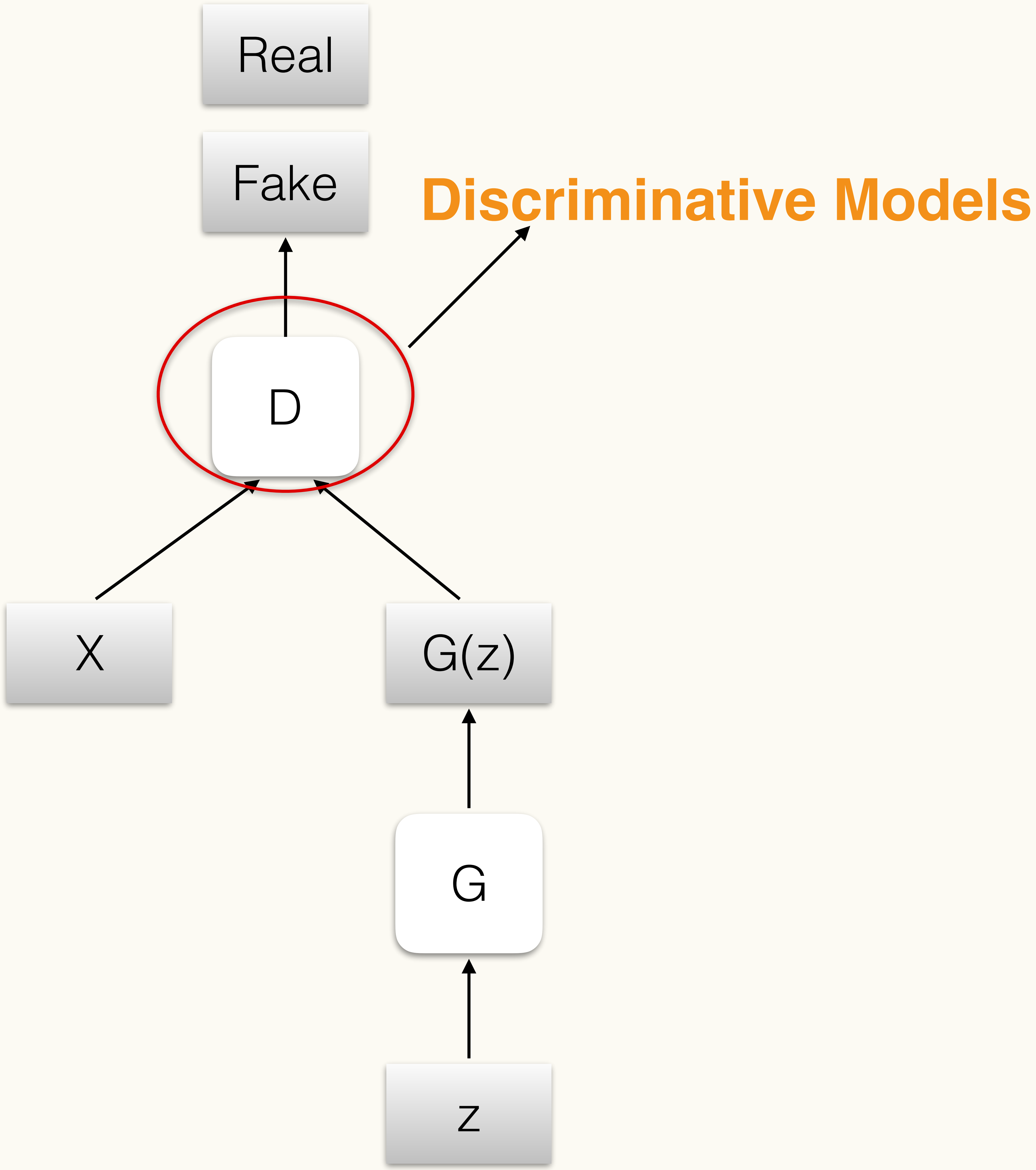
Basic approach

Generative models

Discriminative models

How do GANs work

GANs



Discriminative Model

- learns to determine whether a sample is from the model distribution or the data distribution
- estimates the **probability** that a sample came from the training data rather than G
- $$D(x) = \frac{P_{data}(x)}{P_{data}(x) + P_{model}(x)}$$ 这个公式需要稍微解释一下吗？

GANs

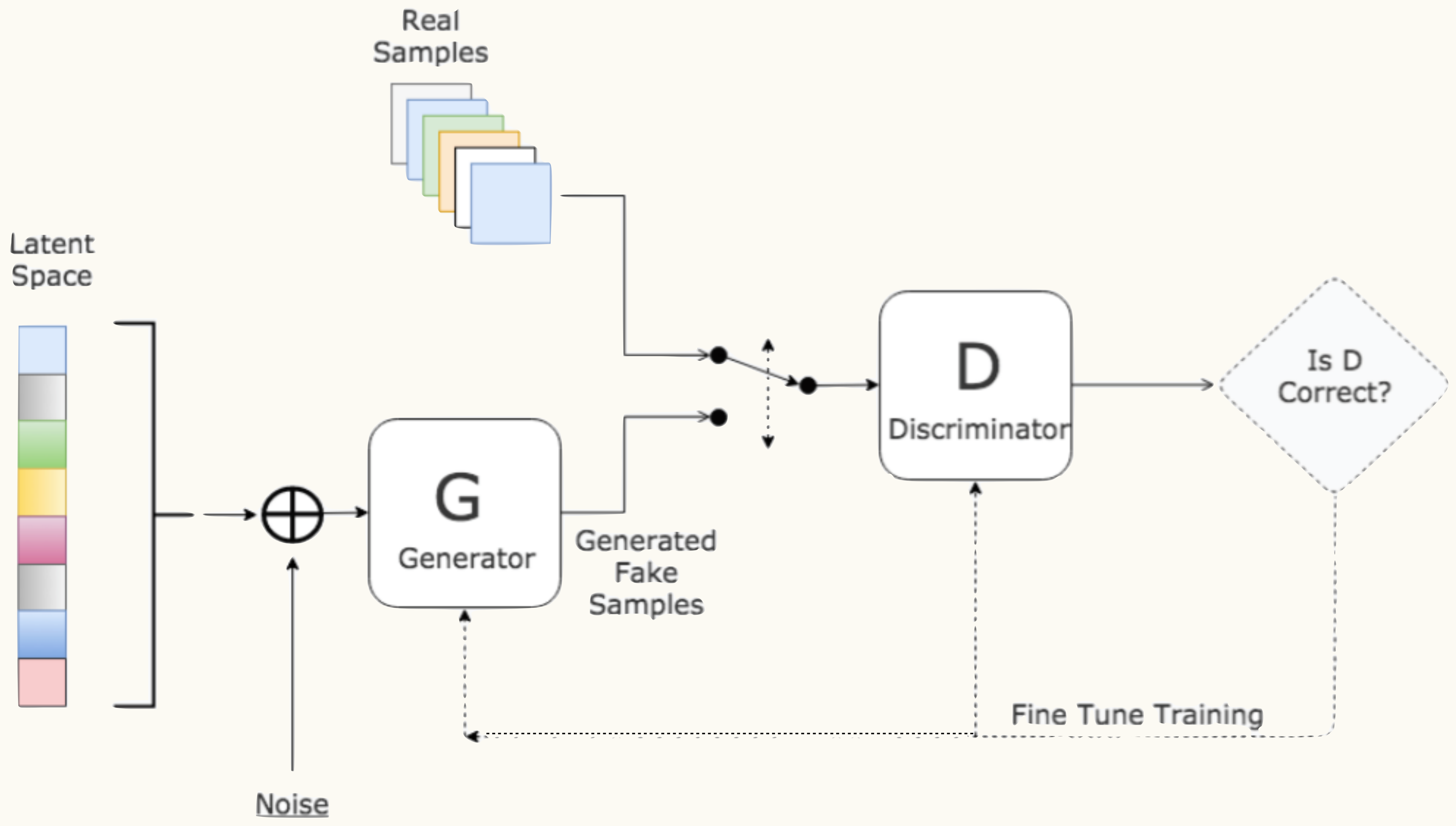
Basic approach

Generative models

Discriminative models

How do GANs work

Minimax Game



或者也可以在这边写那个总的核心公式，跟着图一起解释。

Minimax Game

Loss Function

$$J^{(D)} = -\frac{1}{2} E_{x \sim P_{data}} \log D(x) - \frac{1}{2} E_x \log(1 - D(G(z)))$$

$$J^{(G)} = -J^{(D)}$$

What is the problem?

Non-Saturating

Minimax Game

$$J^{(D)} = -\frac{1}{2}E_{x\sim P_{data}} \log D(x) - \frac{1}{2}E_x \log(1 - D(G(z)))$$

$$J^{(G)} = -\frac{1}{2}E_z \log D(G(z))$$

When the Discriminator is too smart,
G still has a learning signal

ALGORITHM

Minimax Game

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

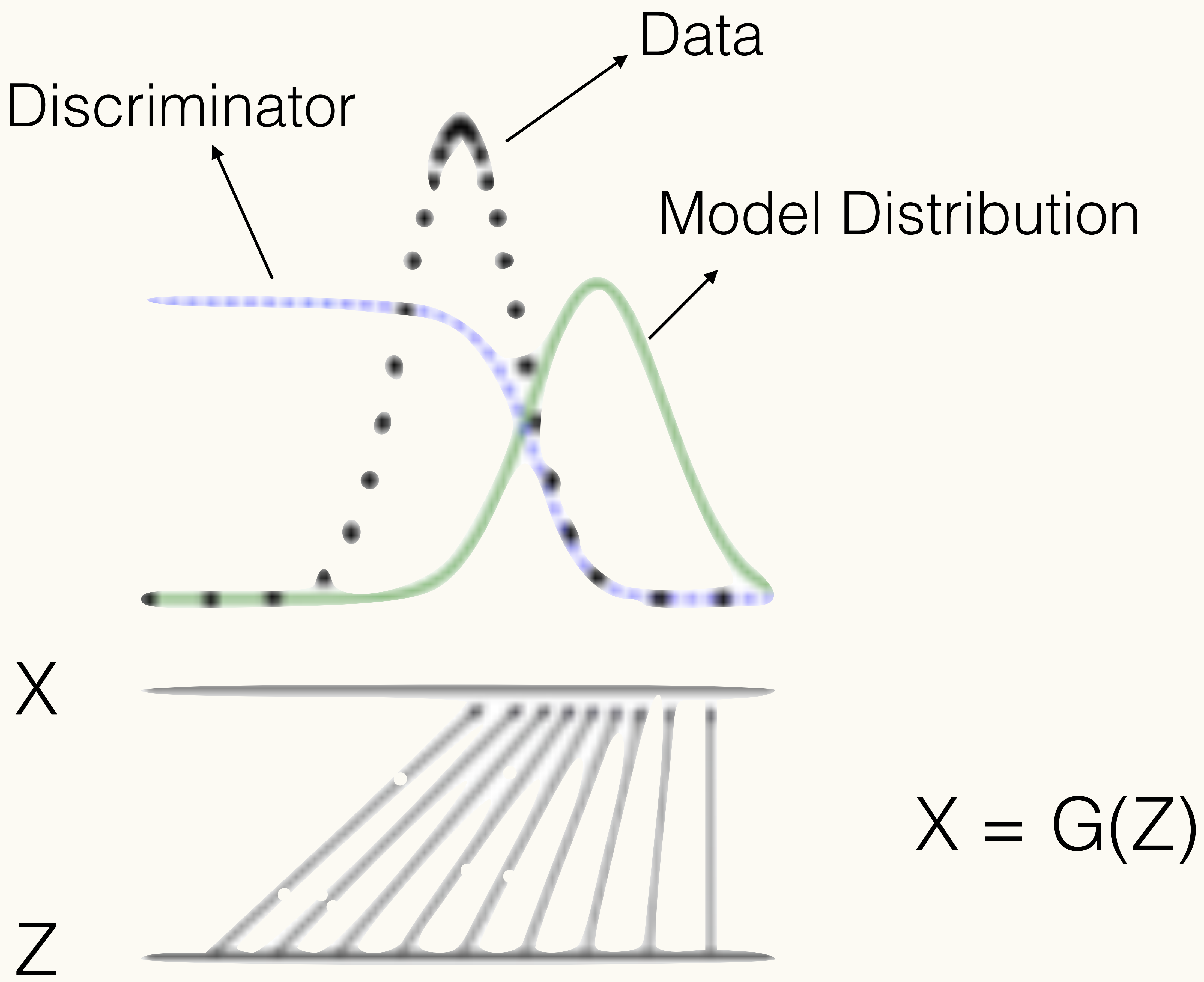
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

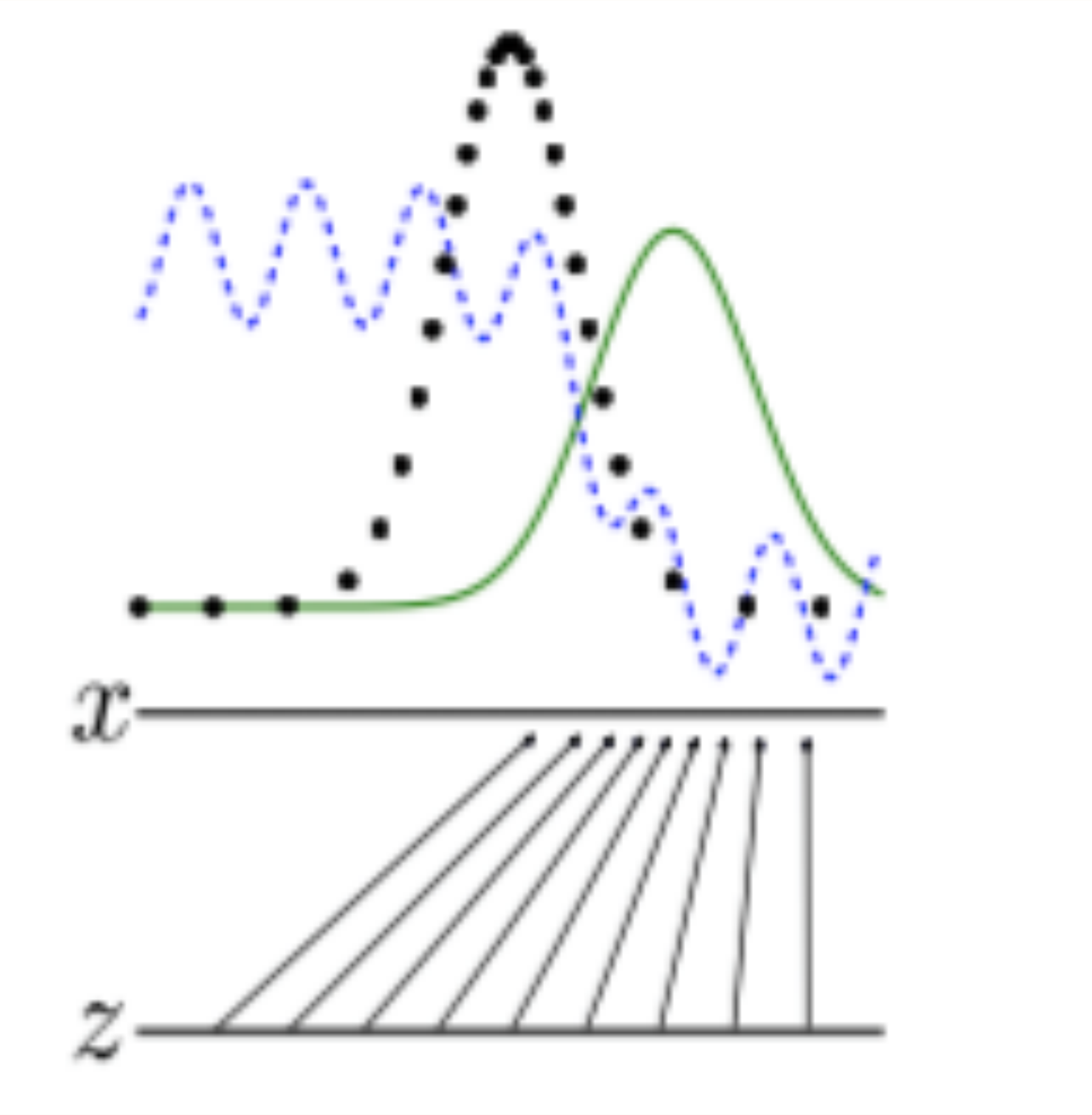
end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Minimax Game



Minimax
Game



Minimax Game

ALGORITHM

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

end for

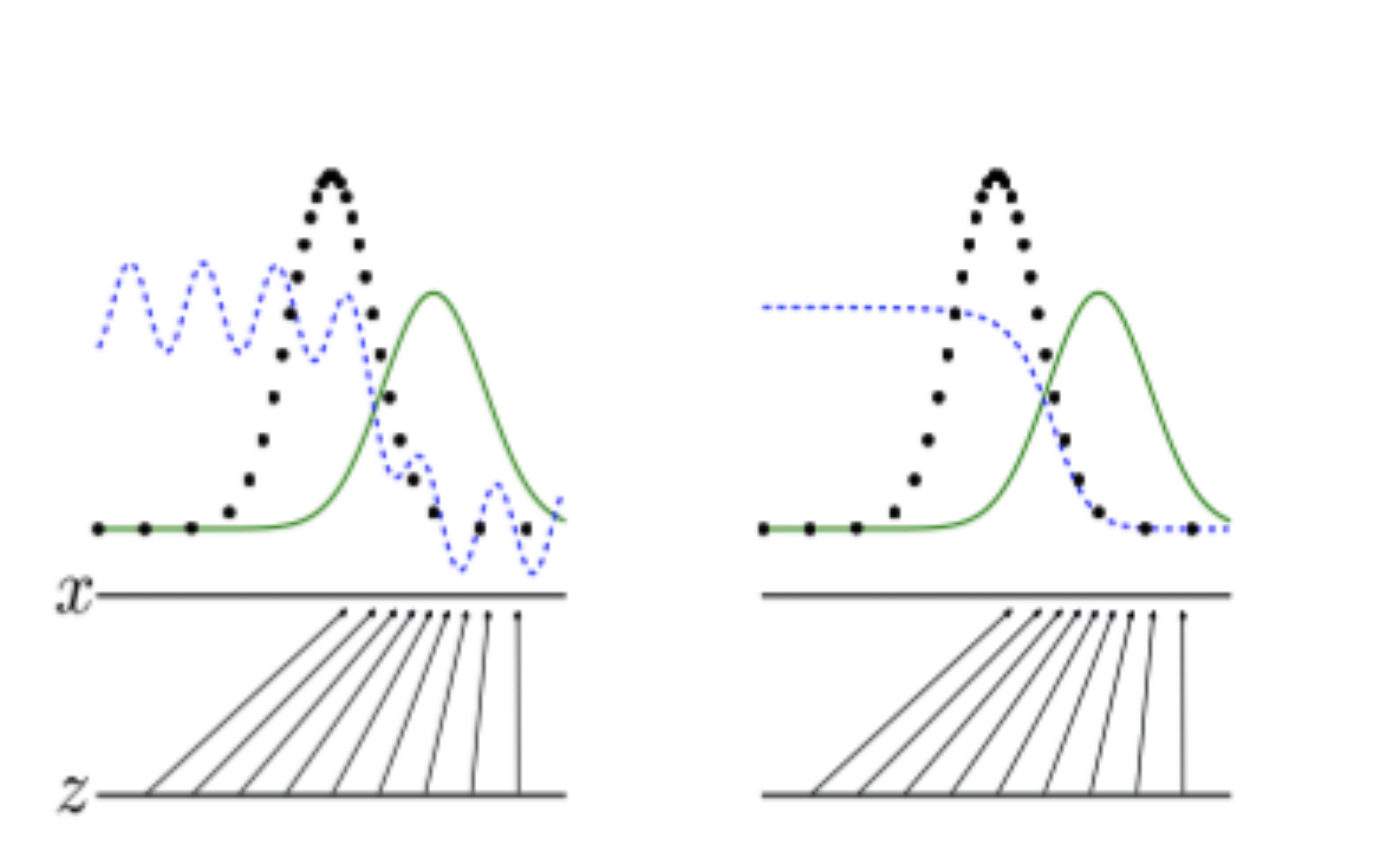
- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Minimax Game



ALGORITHM

Minimax Game

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

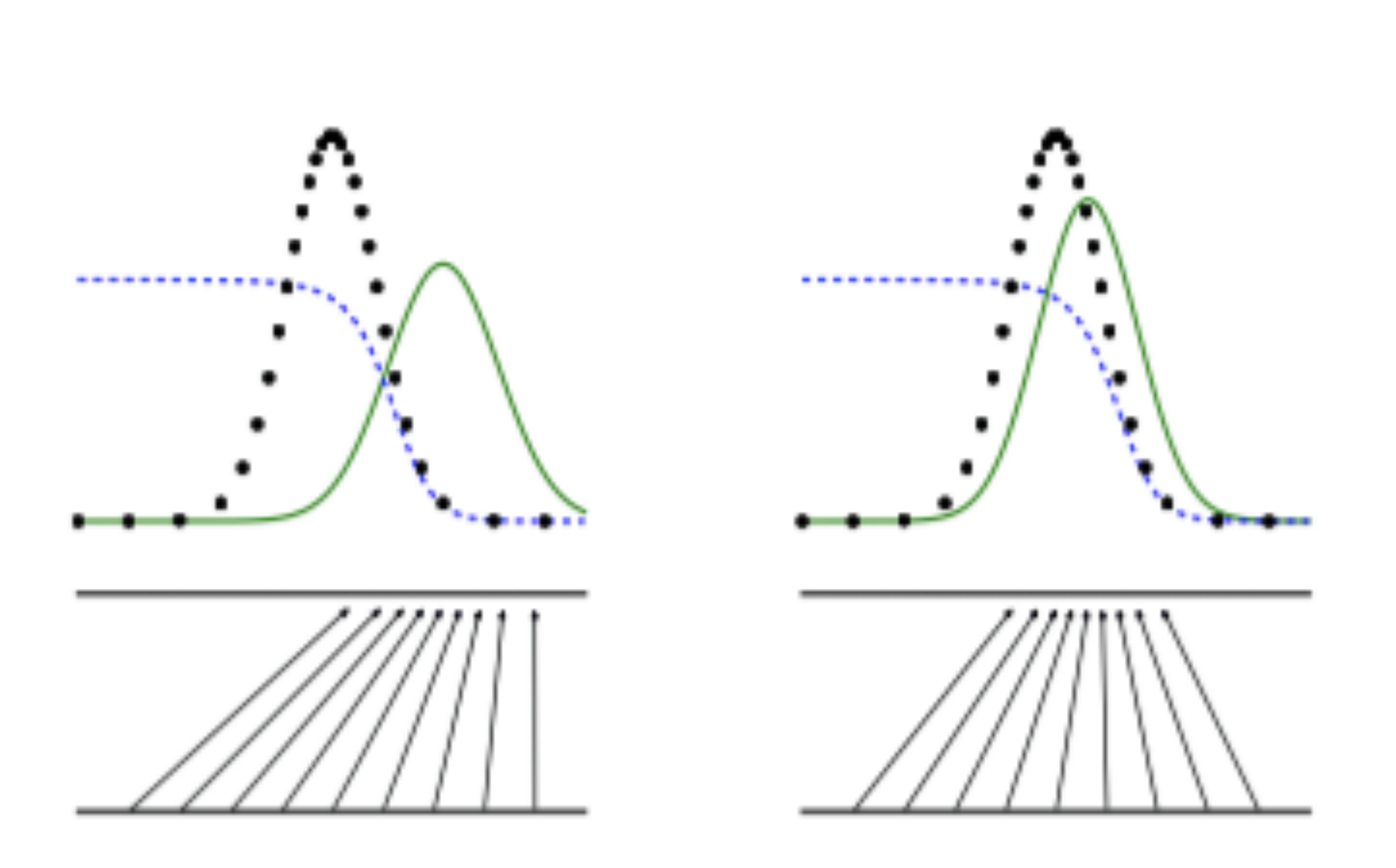
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

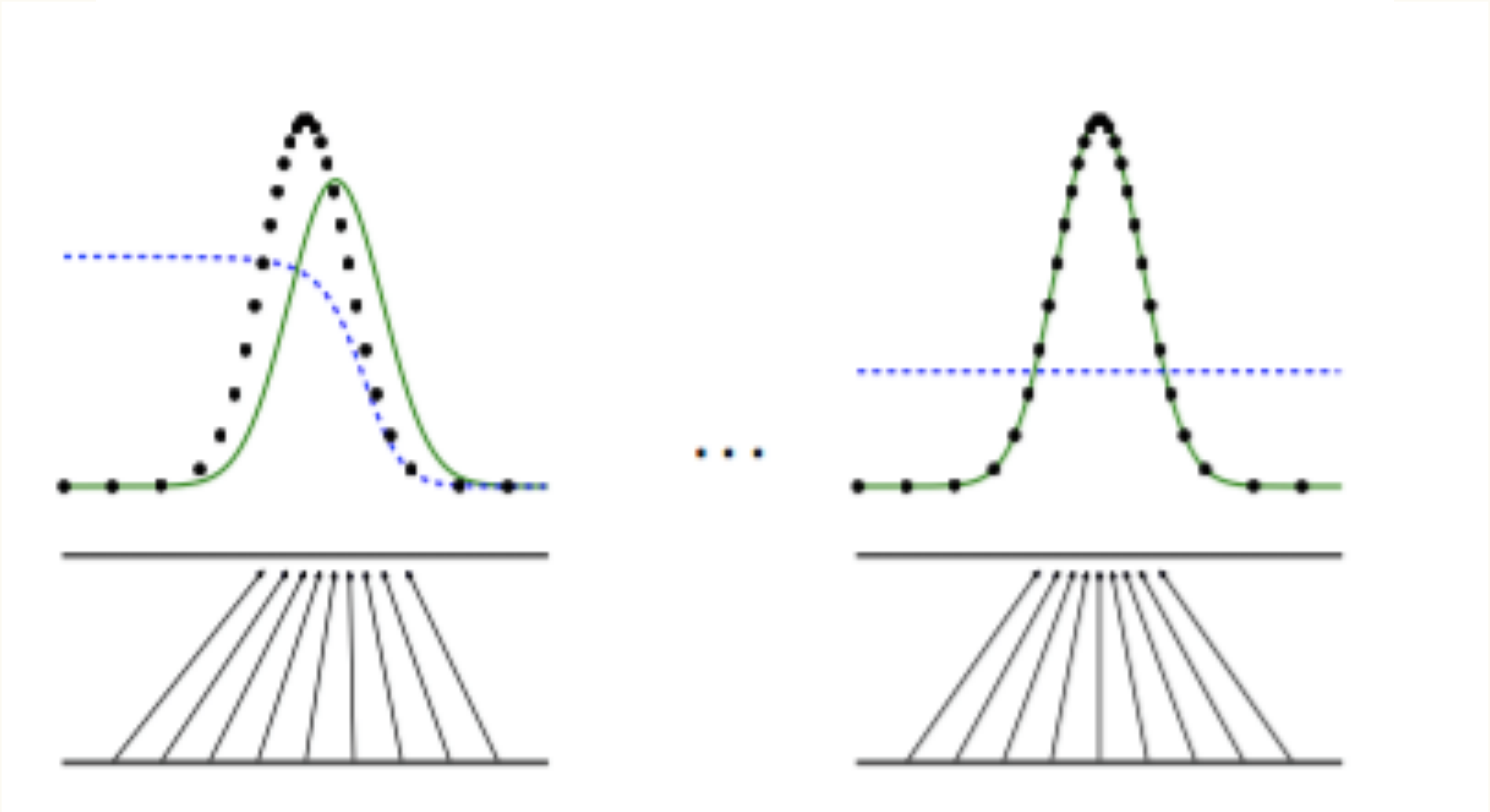
end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Minimax Game



Minimax Game



这边需不需要解释一下当Pg=Pdata的时候，min max...=-log 4？
即生成模型能完美地复制数据的生成过程。

Global optimality:

$$P_{data}(x) = P_{model}(x)$$

Thank you!

Questions?