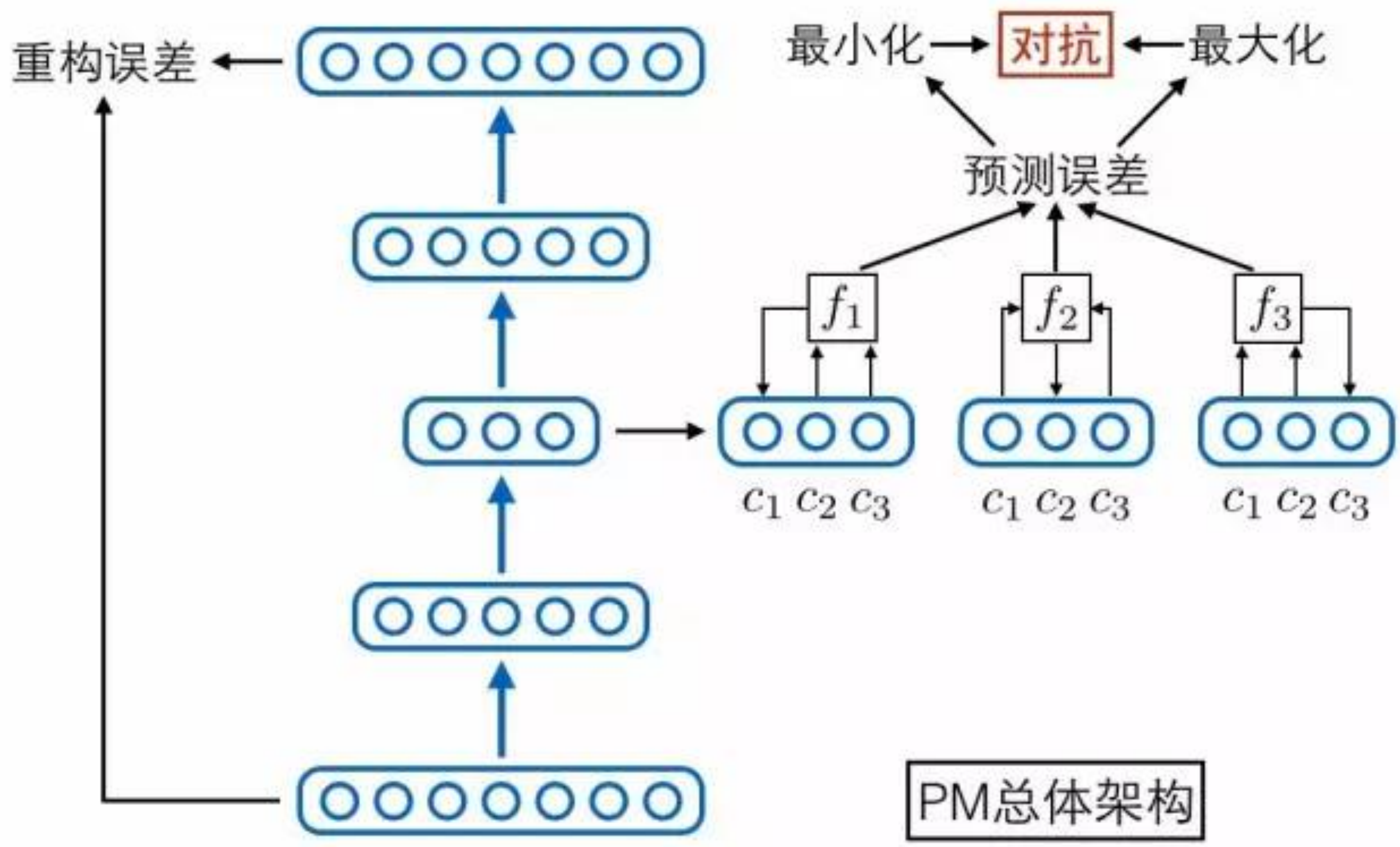
Generative Adversarial Networks (GANs)

Zhaokai Huang

Adversarial Training

- A phrase whose usage is in flux; a new term that applies to both new and old ideas

- Current usage: “Training a model in a worst-case scenario, with inputs chosen by an adversary”. Typically, ***generator + discriminator***.



Auxiliarily, to learn factorial codes by adversary.

Predictability Minimization (J Schmidhuber, 1992)

Roadmap

- Why study generative models?

- How do generative models work? GANs and its variants?

- Research frontiers

Why study generative models?

- Simulate possible futures for planning or simulated RL
- Realistic generation tasks
- Missing data
    - Semi-supervised learning

Why study generative models?

- Simulate possible futures for planning or simulated RL
- Realistic generation tasks
- Missing data
  - Semi-supervised learning

Creative applications of GANs to RL

- Adversarial Learning for Neural Dialogue Generation (Li, Jiwei, et al., 2017), getting rewards by ways of trained discriminator in GANs, rather than trial-and-errors.

The way to pass Turing Test?

- Generative Adversarial Imitation Learning (OpenAI, 2016). The agent learns from example demonstrations, eliminating the need to design a reward function.

Why study generative models?

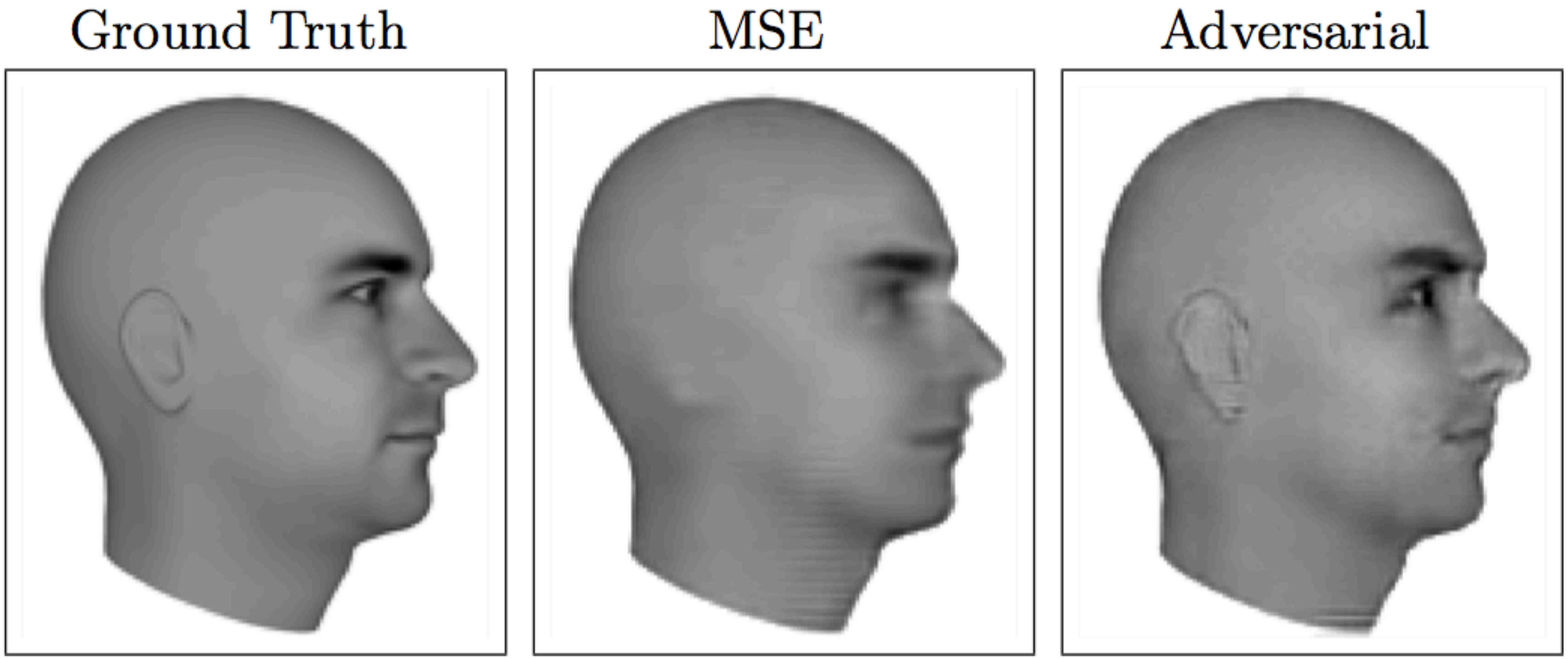
- Simulate possible futures for planning or simulated RL
- Realistic generation tasks
- Missing data
  - Semi-supervised learning

Single Image Super-Resolution

original | bicubic (21.59dB/0.6423) | SRResNet (23.53dB/0.7832) | SRGAN (21.15dB/0.6868)

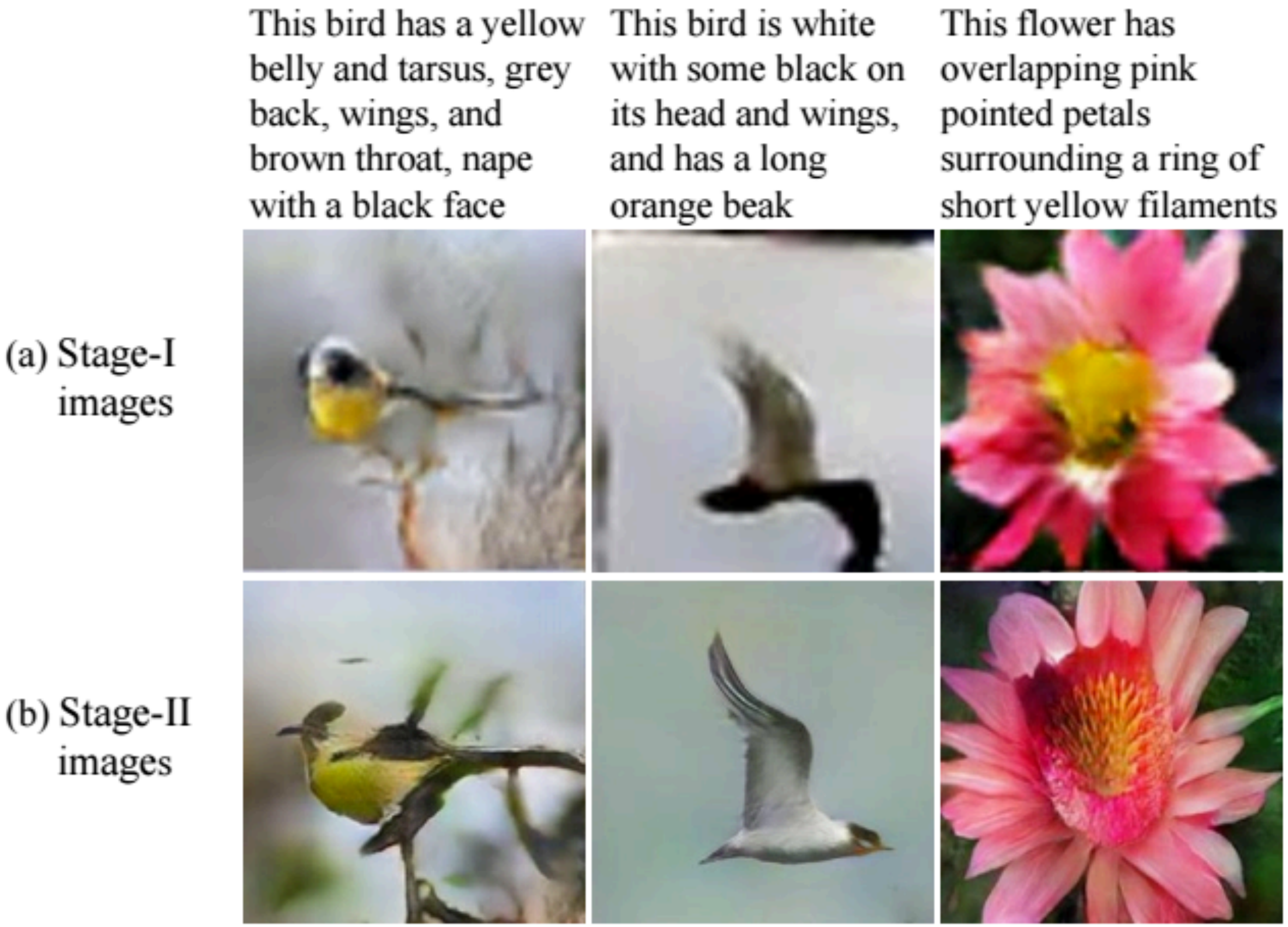
(Ledig et al 2016)

Next Video Frame Prediction



(Lotter et al 2016)

Text to Photo-realistic Image Synthesis (StackGan)



Why study generative models?

- Simulate possible futures for planning or simulated RL
- Realistic generation tasks
- Missing data
    - Semi-supervised learning

Semi-supervised learning

Idea: labeling sample from Generator with a new "generated" class ***y = K + 1.***

Result: On MNIST, achieve 99.14% accuracy with only ***10*** *labeled examples per class* (vs ***60,000*** labelled samples).

$$\begin{aligned}
L &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)}[\log p_{\text{model}}(y|\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim G}[\log p_{\text{model}}(y=K+1|\boldsymbol{x})] \\
&= L_{\text{supervised}} + L_{\text{unsupervised}}, \text{ where} \\
L_{\text{supervised}} &= -\mathbb{E}_{\boldsymbol{x},y\sim p_{\text{data}}(\boldsymbol{x},y)} \log p_{\text{model}}(y|\boldsymbol{x}, y<K+1) \\
L_{\text{unsupervised}} &= -\{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})} \log[1-p_{\text{model}}(y=K+1|\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim G}\log[p_{\text{model}}(y=K+1|\boldsymbol{x})]\},
\end{aligned}$$

(OpenAI, 2016)

Roadmap

- Why study generative models?
- <u>How do generative models work? GANs and its variants?</u>
- Research frontiers

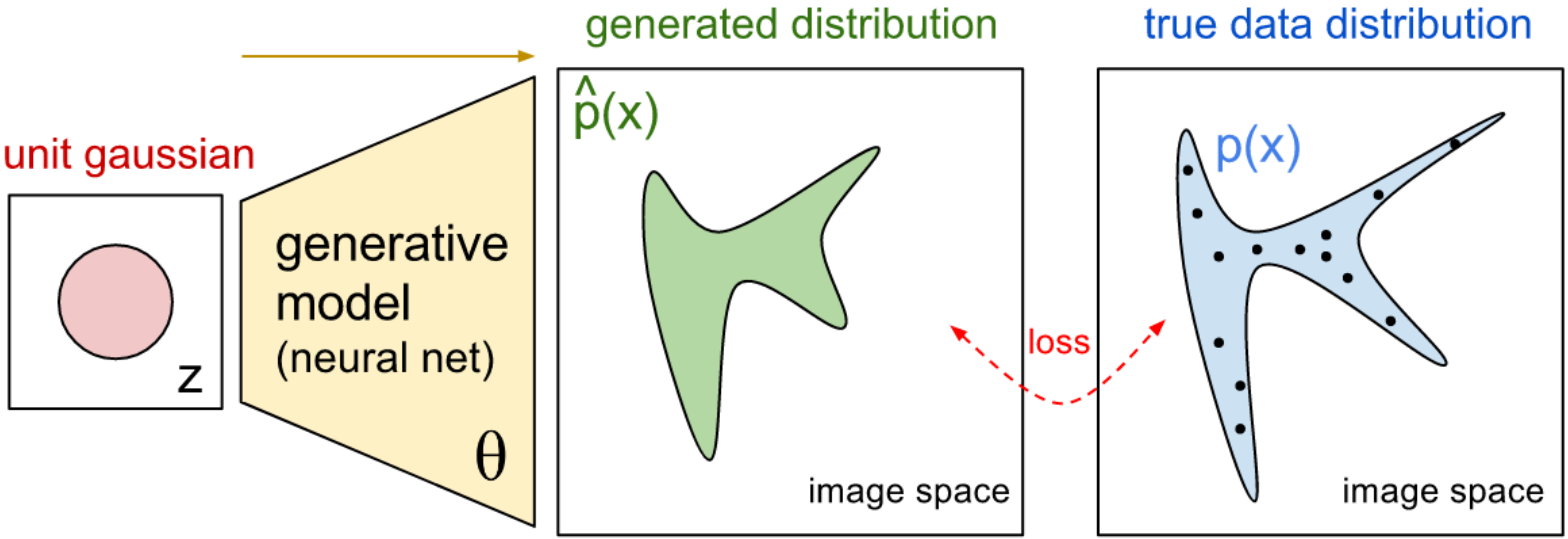
How do generative models work? GANs and its variants?

- General formulation
- Intuitions
- Adversarial Nets Framework
- DCGAN

How do generative models work? GANs and its variants?

- General formulation
- Intuitions
- Adversarial Nets Framework
- DCGAN

General formulation



How do generative models work? GANs and its variants?

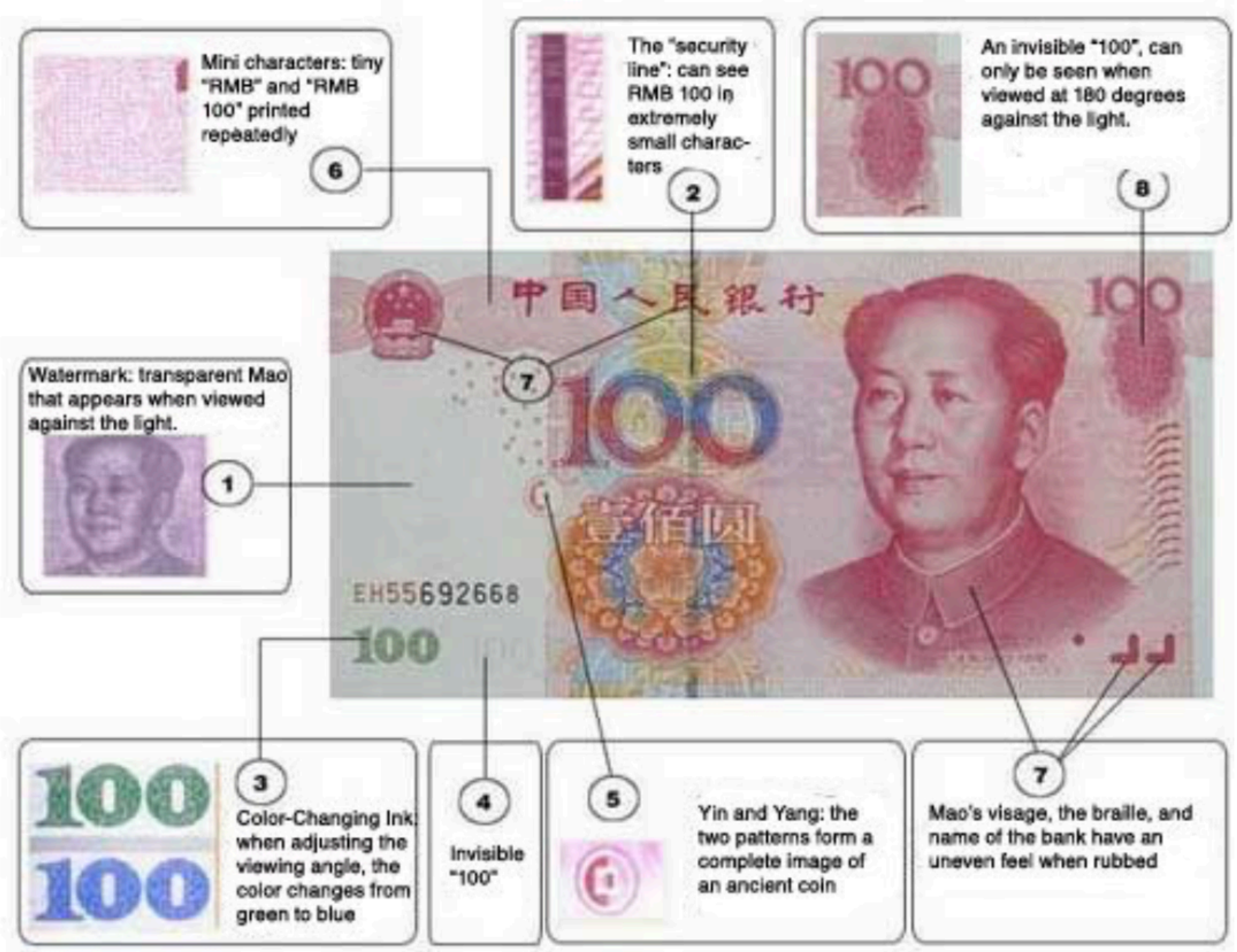
- General formulation
- Intuitions
- Adversarial Nets Framework
- DCGAN

Intuitions

Crook

Intuitions

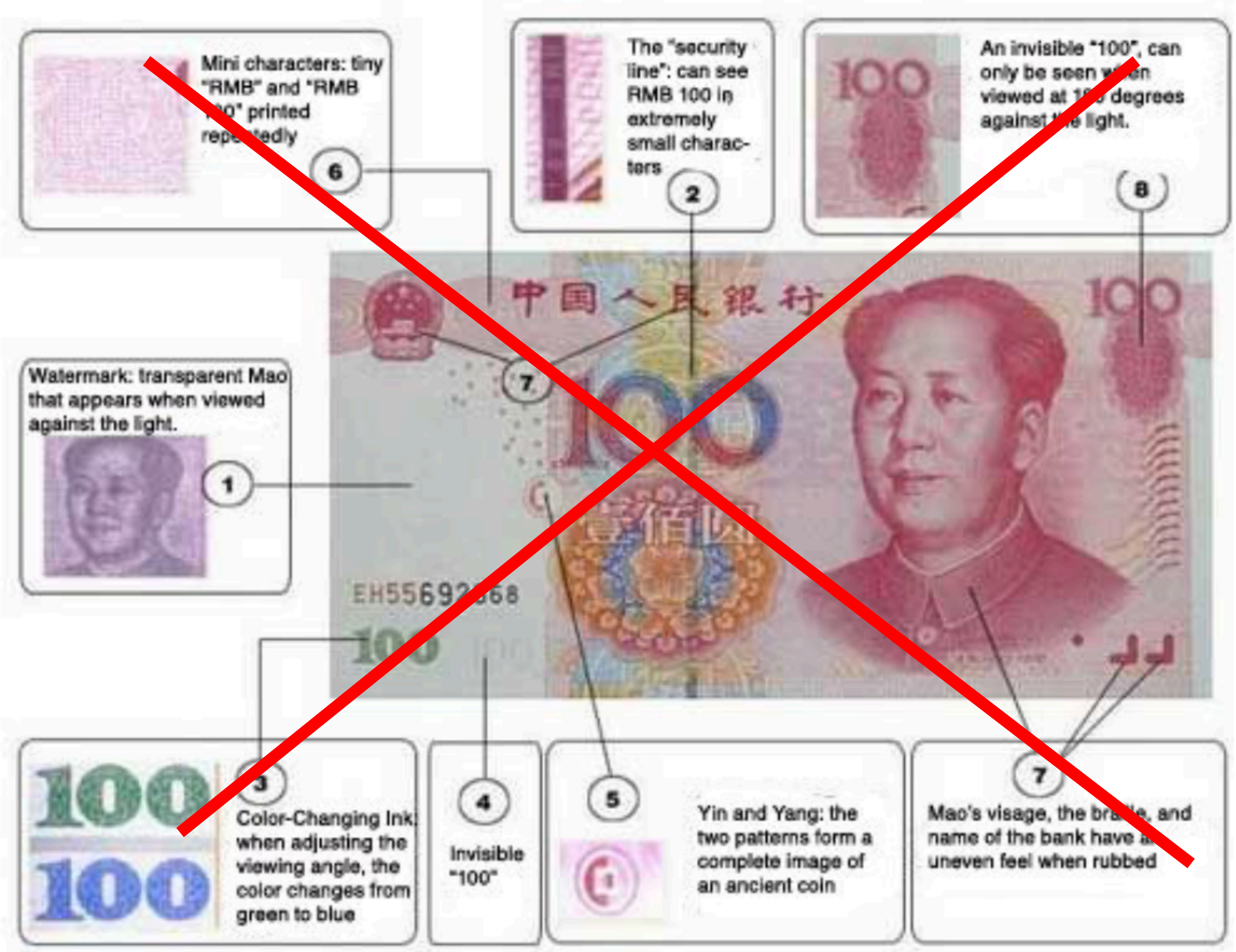
Crook



Teller

Intuitions

Crook



Teller

Intuitions

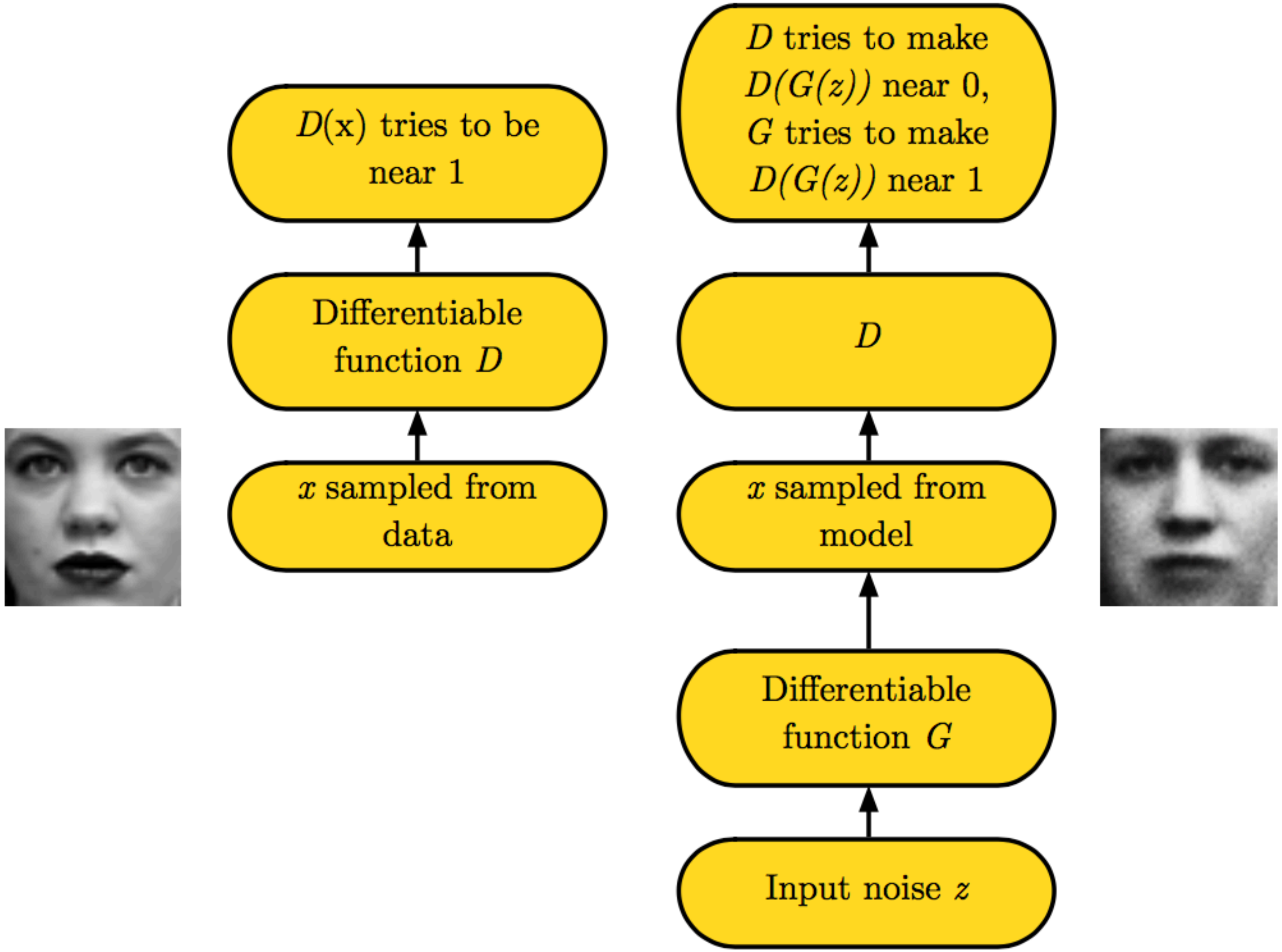
- Crook tries the best to cheat the teller by generating more realistic cash

- Teller tries the best to distinguish whether the cash is generated by crook or not

How do generative models work? GANs and its variants?

- General formulation
- Intuitions
- Adversarial Nets Framework
- DCGAN

Adversarial Nets Framework



Minimax Game (objective)

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

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

- Discriminator: real or fake, cross-entropy errors.
- Generator minimizes the log-probability of the discriminator being correct.
- Nash equilibrium is a saddle point of the discriminator loss.

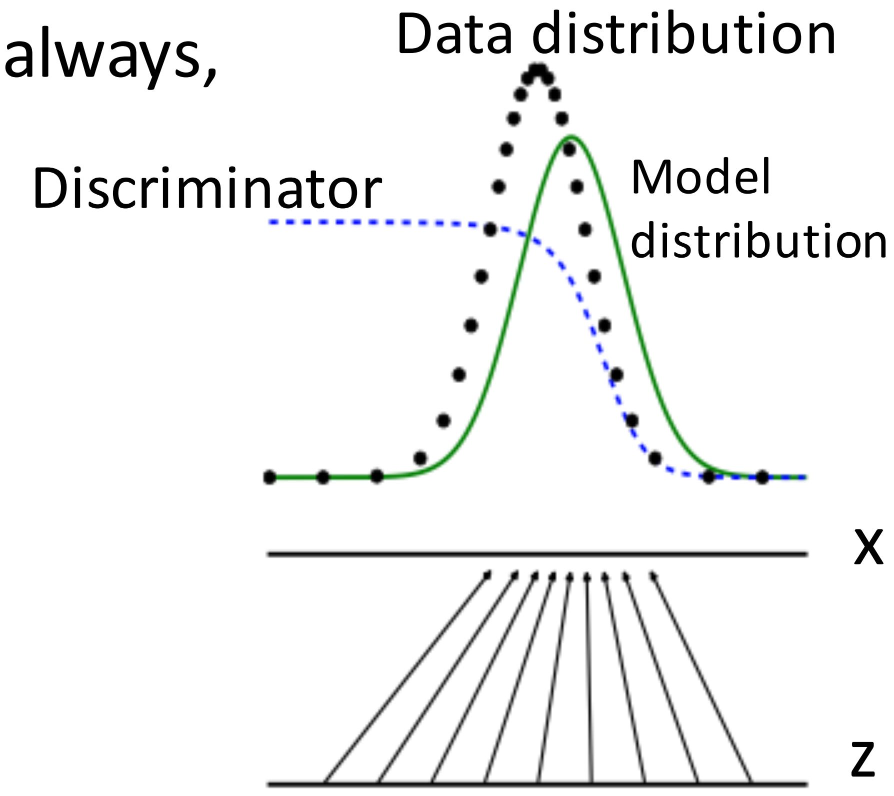
Discriminator Strategy

- Assume both densities are nonzero everywhere, $\frac{\delta}{\delta D(\boldsymbol{x})} J^{(D)} = 0$

- Optimal D(x) for any p_{data}(x) and p_{model}(x) is always,

$$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)}$$

- $C(G) = -\log(4) + 2 \cdot JSD\left(p_{\text{data}} \| p_g\right)$



Non-Saturating Game

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log D\left(G(\boldsymbol{z})\right)$$

- Generator maximizes the log-probability of the discriminator being mistaken.

- Heuristically motivated; generator can still learn even when discriminator successfully rejects all generator samples

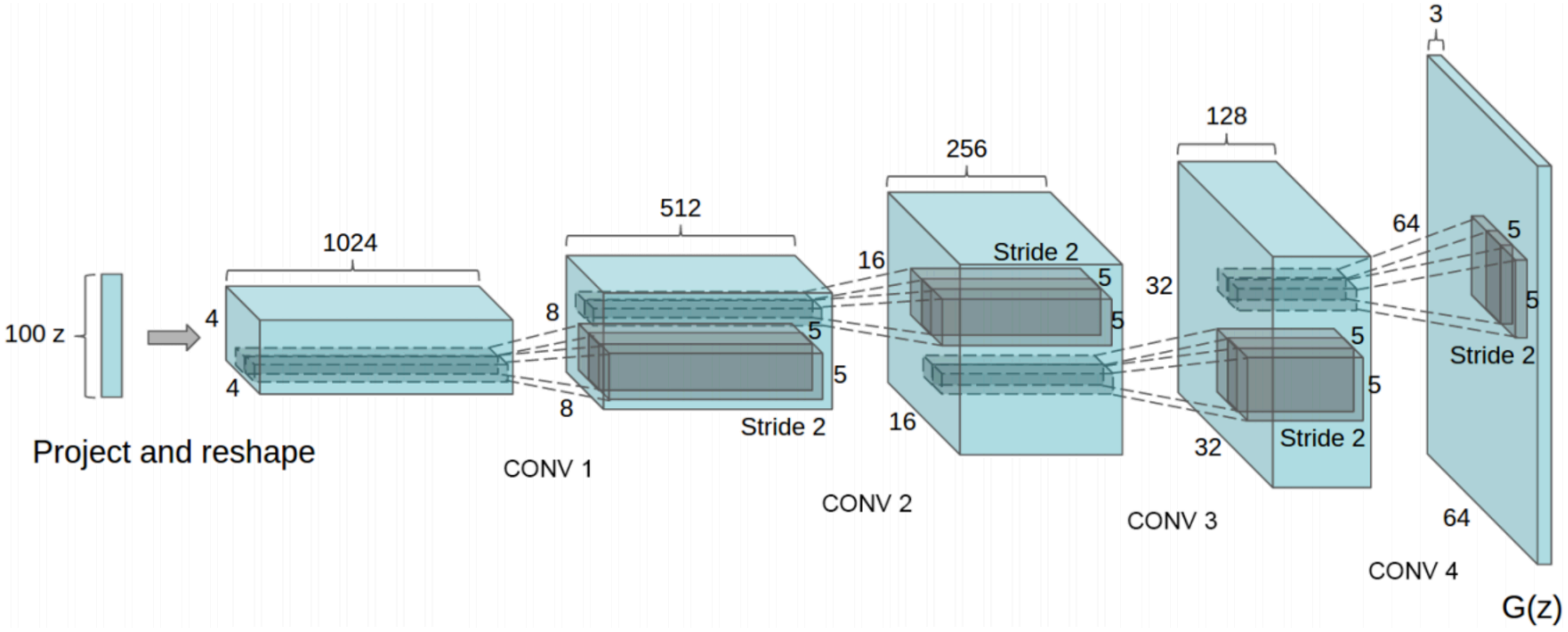
Training Procedure

- Use SGD-like algorithm of choice (Adam) on two minibatches simultaneously:
  - A minibatch of training examples
  - A minibatch of generated samples
- Optional: run *k* steps of one player for every step of the other player.

How do generative models work? GANs and its variants?

- General formulation
- Intuitions
- Adversarial Nets Framework
- DCGAN

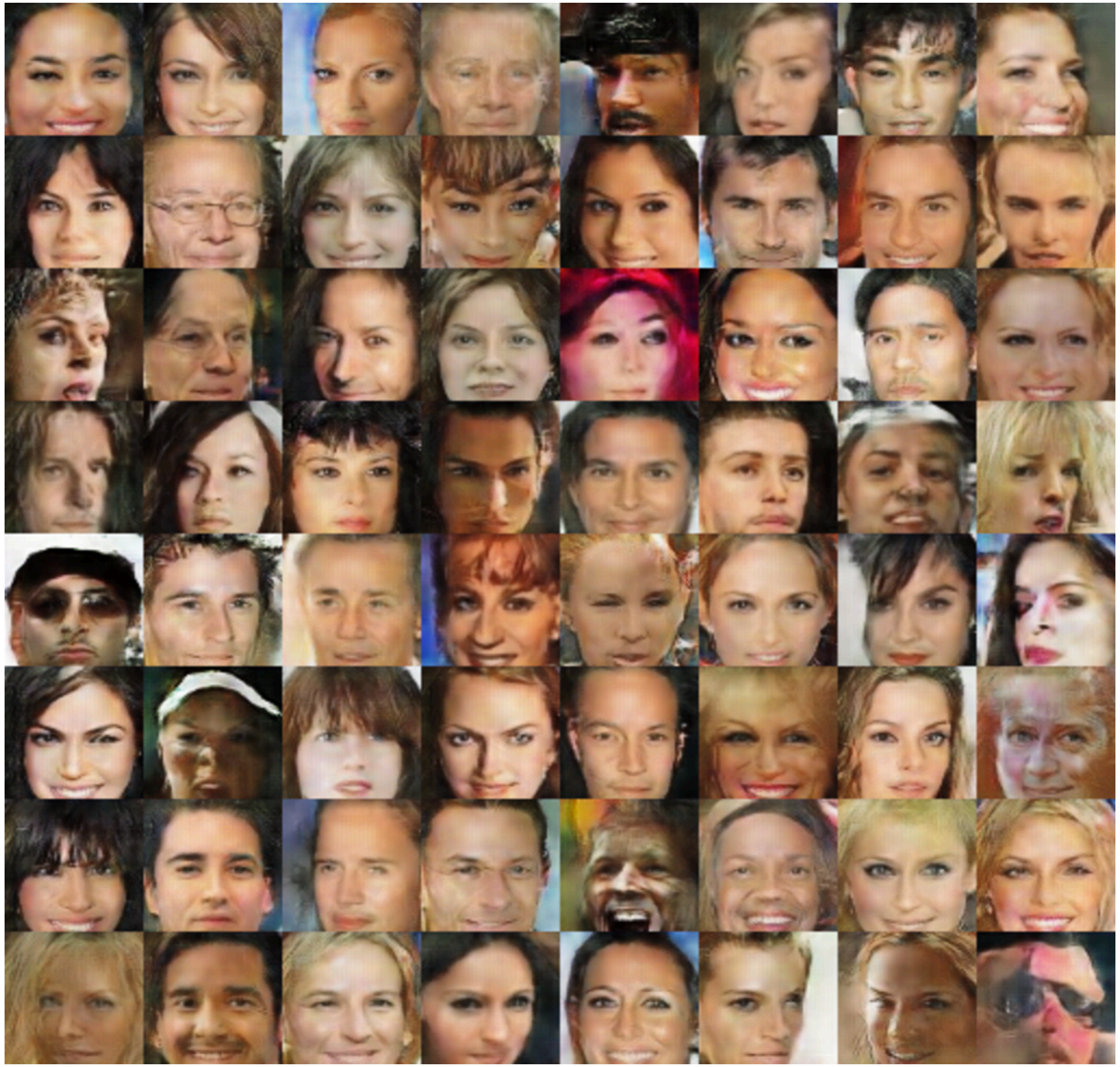
DCGAN



(Radford et al 2015)

DCGANs for MINIST

DCGANs for celebA



Roadmap

- Why study generative models?
- How do generative models work? GANs and its variants?
- Research frontiers

Research frontiers

- Non-convergence

- Wasserstein GAN

Research frontiers

- Non-convergence
- Wasserstein GAN

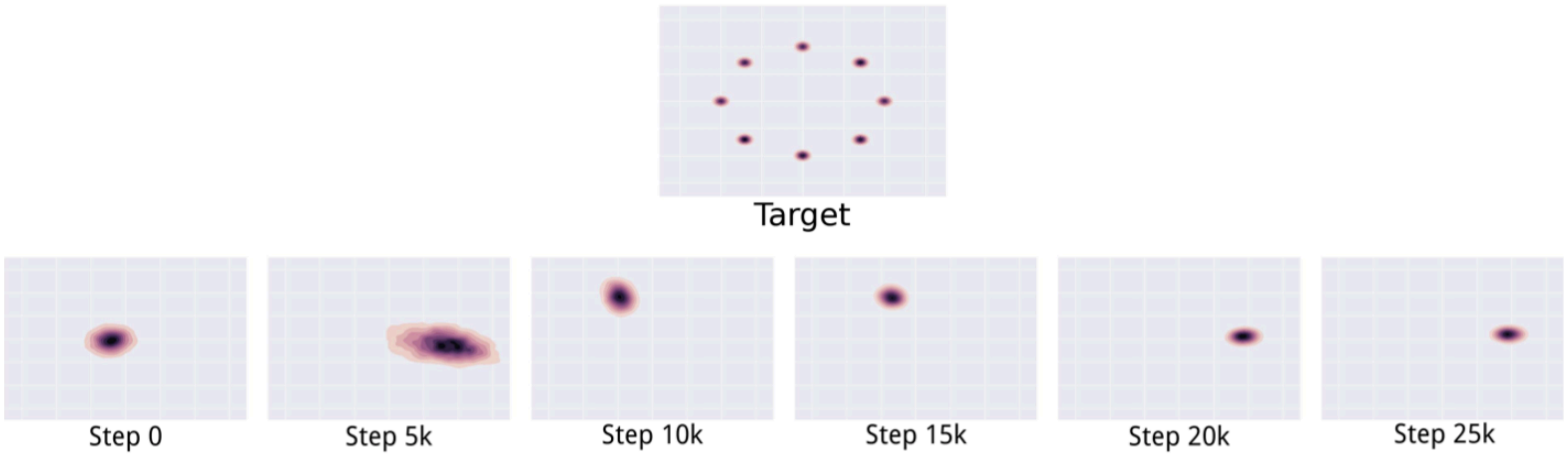
Non-convergence

- Game solving algorithms may not approach an equilibrium at all
- We represent *G* and *D* as highly non-convex parametric functions
- Mode collapse: most severe form of non-convergence

Mode collapse

$$\min_G \max_D V(G, D) \neq \max_D \min_G V(G, D)$$

Minibatch Features



Research frontiers

- Non-convergence

- Wasserstein GAN

Wasserstein GAN

- Jensen-Shannon Divergence -> Wasserstein Divergence