ECE 884 Deep Learning

Lecture 22: GAN

04/15/2021

Review of last lecture

- Autoregressive Model
- Autoencoder
- Variational Autoencoder

Today’s lecture

- Generative Adversarial Networks (GANs)

GAN: Main Idea

- **Generator**: generates fake images to *fool* the discriminator.
- **Discriminator**: tries to *distinguish* between real and fake images.

- Train them against each other.
- Finally, the discriminator cannot distinguish between real and fake.
- It means the fake images look like real.

GAN: Main Idea

- **Generator**: a forger who wants to create a fake Picasso painting.
- **Discriminator**: an art dealer.

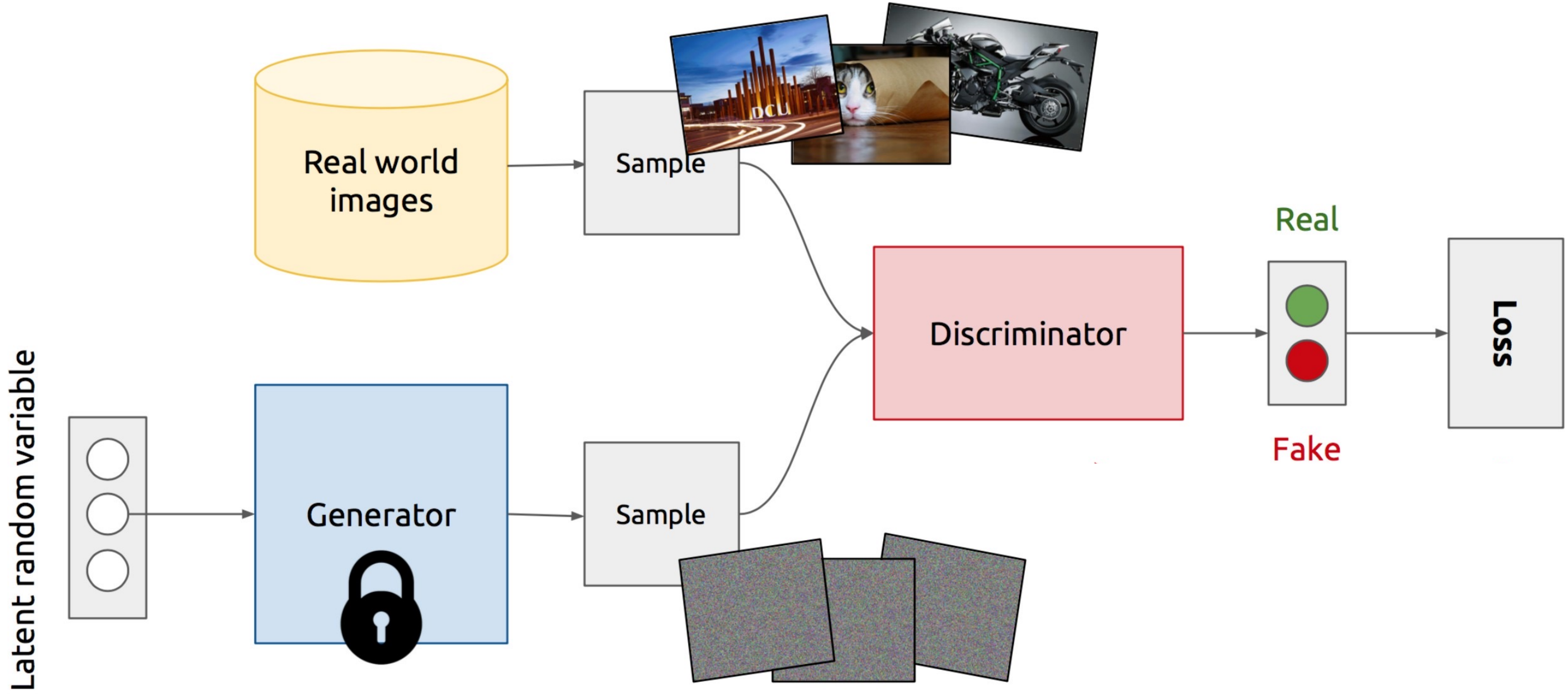
Which is real?

GAN: Main Idea

- **Generator**: a forger who wants to create a fake Picasso painting.
- **Discriminator**: an art dealer.

- Train them against each other.
  - Forger mixes his fake with authentic paintings and ask art dealer to assess them.
  - Art dealer gives feedback about what makes a Picasso look like a Picasso.
  - Forger improves his competence at imitating the style of Picasso.
  - Art dealer improves his competence at distinguishing real and fake Picasso.
- Finally, the art dealer cannot distinguish between real and fake.

GAN: Model Overview



Training of GAN

Alternating minimization

Repeat the 2 steps:

1. Update the discriminator network;
2. Update the generator network.

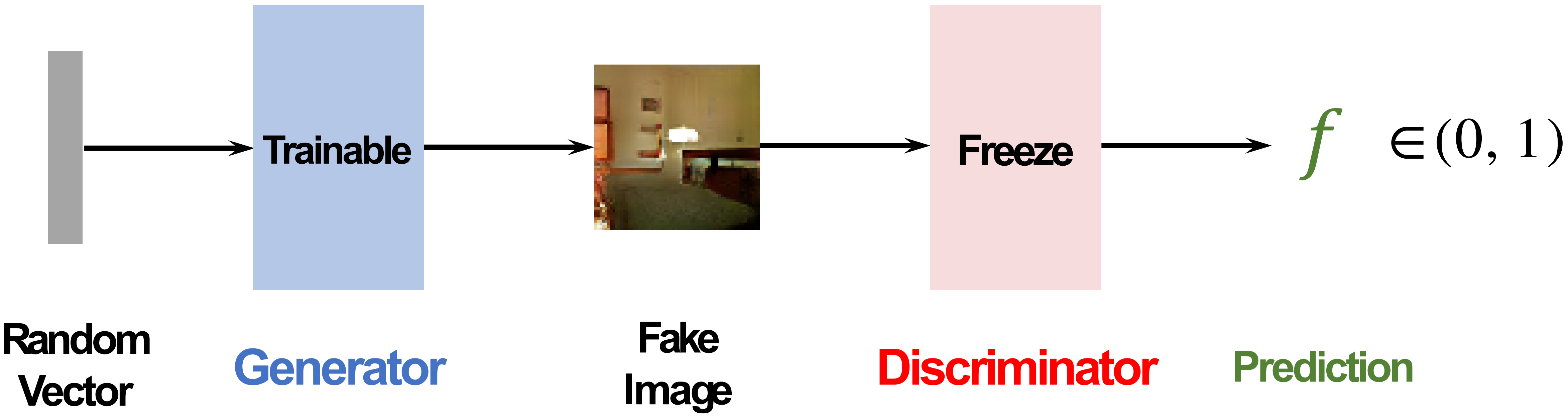
Update the Discriminator

Train a classifier

1. Generate a batch of fake images by the generator;
2. Randomly sample a batch of real images;
3. Inputs: $\mathbf{X} = [\text{real\_images}, \quad \text{fake\_images}]$;
4. Targets: $\mathbf{y} = [\text{True}, \cdots, \text{True}, \text{False}, \cdots, \text{False}]$;
5. Update the discriminator network using $\mathbf{X}$ and $\mathbf{y}$.

Update the Generator

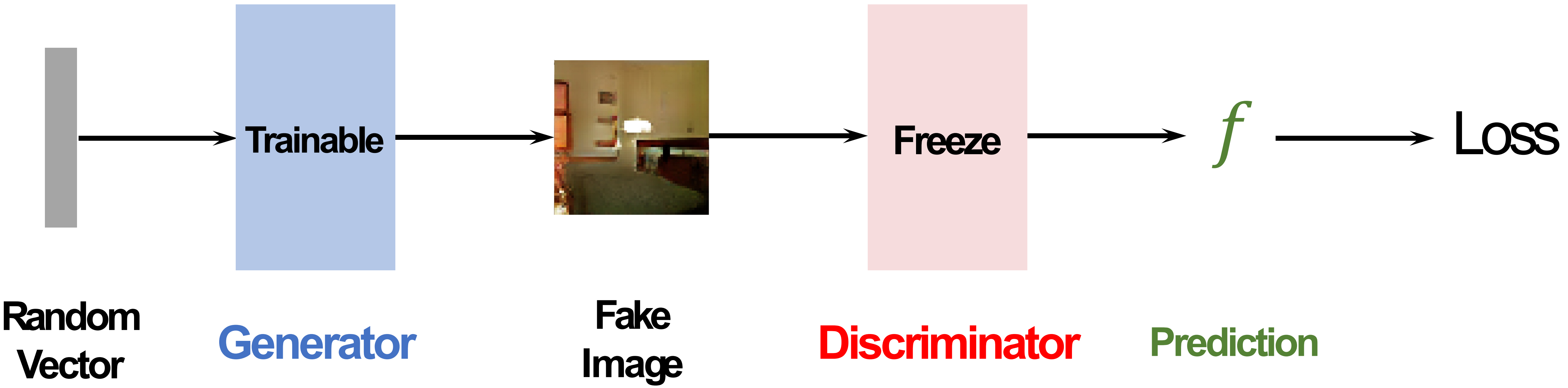
Connect the generator and discriminator (freeze discriminator's parameters).



Update the Generator

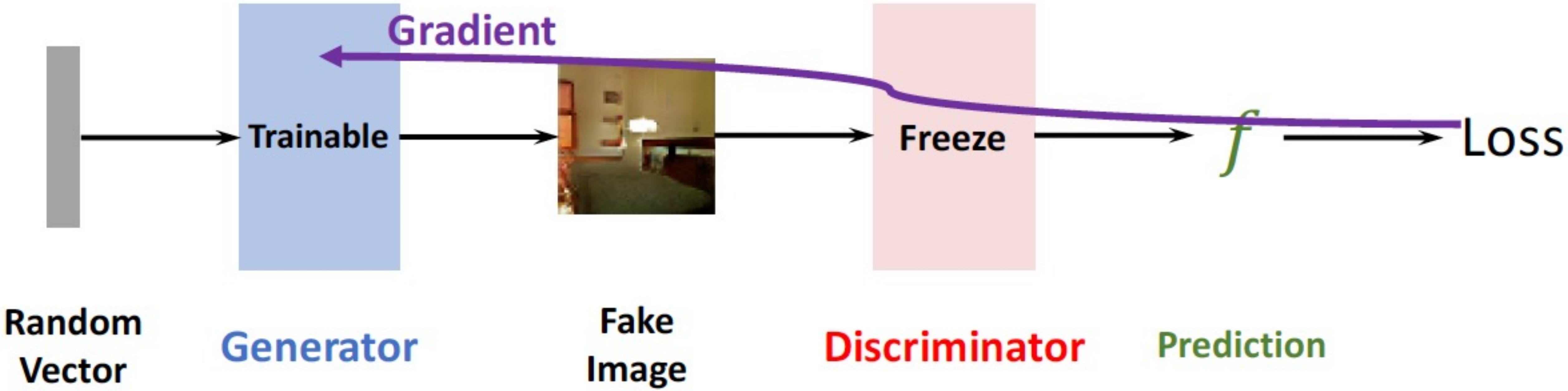
Connect the generator and discriminator (freeze discriminator's parameters).

Minimize $\mathrm{Loss} = \mathrm{Dist}(\mathrm{True}, f)$ w.r.t. generator. (Encourage f be **True.**)



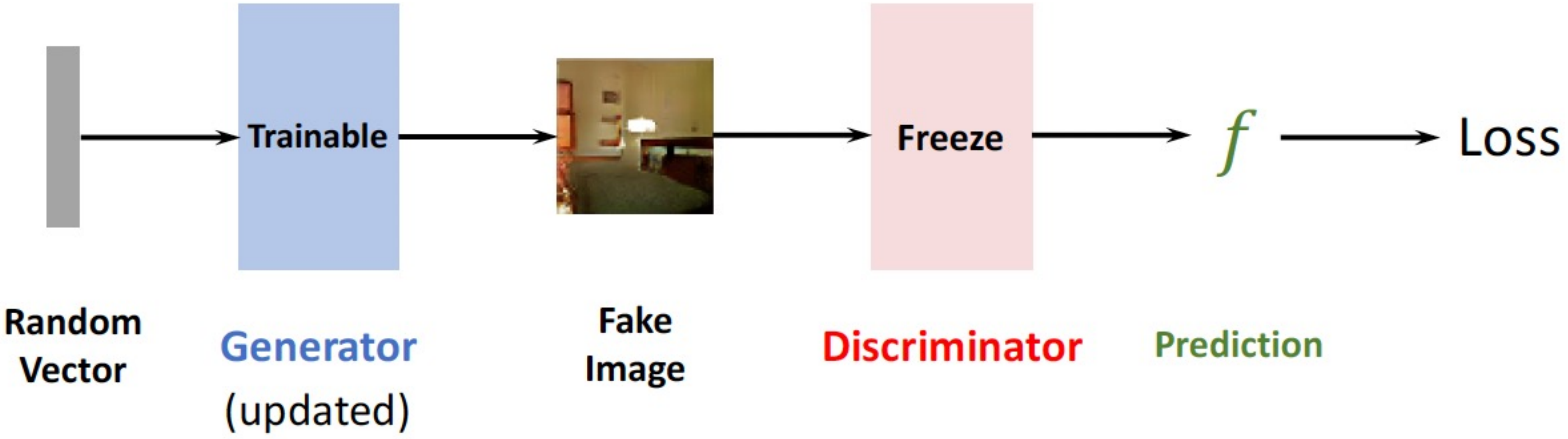
Update the Generator

- $\mathbf{W_G}$: parameters in the generator.
- Gradient: $\text{Grad} = \dfrac{\partial\, \text{Loss}}{\partial\, \mathbf{W_G}}$.



Update the Generator

- $\mathbf{W}_\mathrm{G}$: parameters in the generator.
- Gradient: $\mathrm{Grad} = \frac{\partial \, \mathrm{Loss}}{\partial \, \mathbf{W}_\mathrm{G}}$.
- Gradient descent: $\mathbf{W}_\mathrm{G} \leftarrow \mathbf{W}_\mathrm{G} - \alpha \cdot \mathrm{Grad}$.



Difficulties in Training GAN

Discriminator Shouldn’t Be Too Good

- **Generator**: a forger who wants to create a fake Picasso painting.
- **Discriminator**: an art dealer providing feedbacks.

What if the art dealer is 100% correct at judging Picasso painting?

Discriminator Shouldn't Be Too Good

- **Generator**: a forger who wants to create a fake Picasso painting.
- **Discriminator**: an art dealer providing feedbacks.

What if the art dealer is 100% correct at judging Picasso painting?

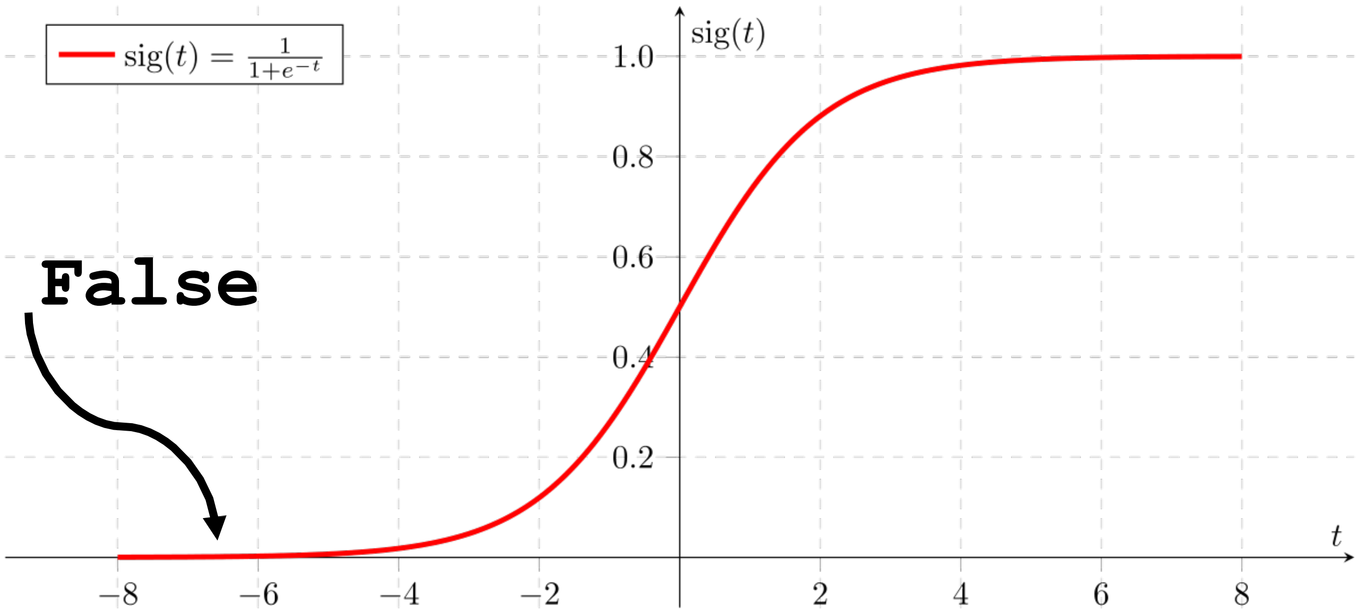
- Whatever forged painting sent to the art dealer is recognized as fake.
- The forger cannot learn anything from the *feedback*.
  - No positive case to follow.
- The forger need some success.
  - So he will know what kind of fake painting can fool the dealer.

Discriminator Shouldn't Be Too Good

Explanation: vanishing gradient

- Suppose the discriminator is perfect.
- Whatever the generators forged is recognized fake by the discriminator.
- The gradient is near zero.

Prediction = False



Discriminator Shouldn’t Be Too Bad

- **Generator**: a forger who wants to create a fake Picasso painting.
- **Discriminator**: an art dealer providing feedbacks.

What if the art dealer cannot distinguish between real and fake paintings?

- The art dealer’s judgement is almost *random guess*.
- The forger cannot learn anything from the *feedback*.
- When the forger’s skill is good, getting amateurish art dealer’s feedback is a not helpful.

Useful Tricks

Carefully tune the learning rates.

- If the discriminator improves too fast,
  - classification accuracy can be 100%,
  - vanishing gradient,
  - the generator is dead.

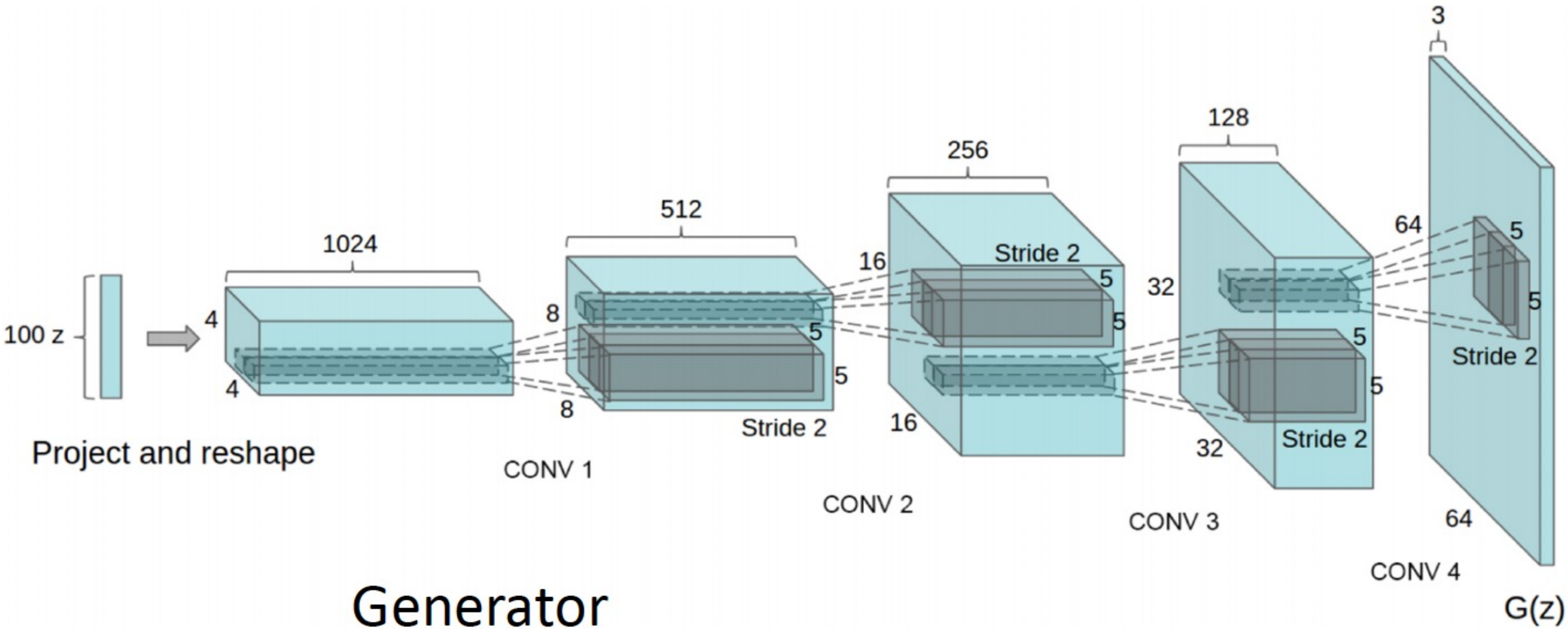
- If the discriminator improves too slowly,
  - the discriminator cannot provide useful feedback,
  - the generator has to wait for the discriminator,
  - slow convergence.

Useful Tricks

Many other tricks…

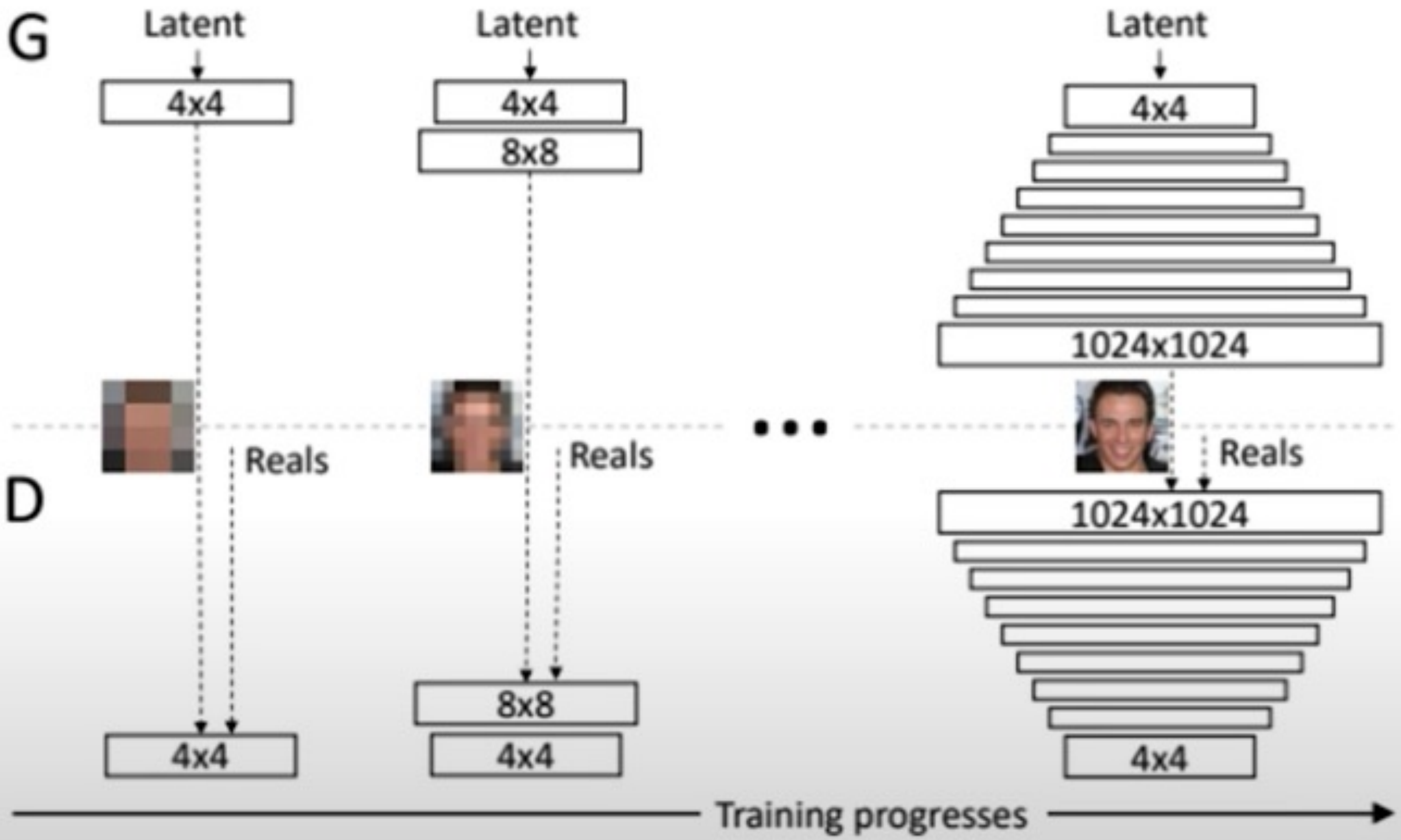
- Further reading:
- https://github.com/soumith/ganhacks

Generative Adversarial Networks: DC-GAN



Generator

Progressive growing of GANs



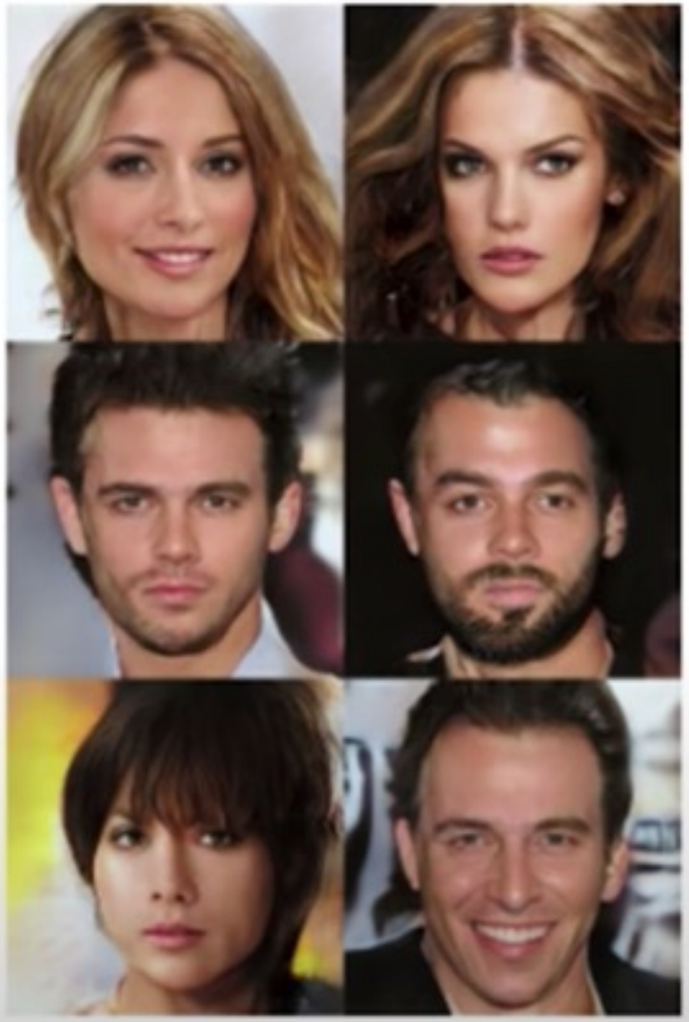
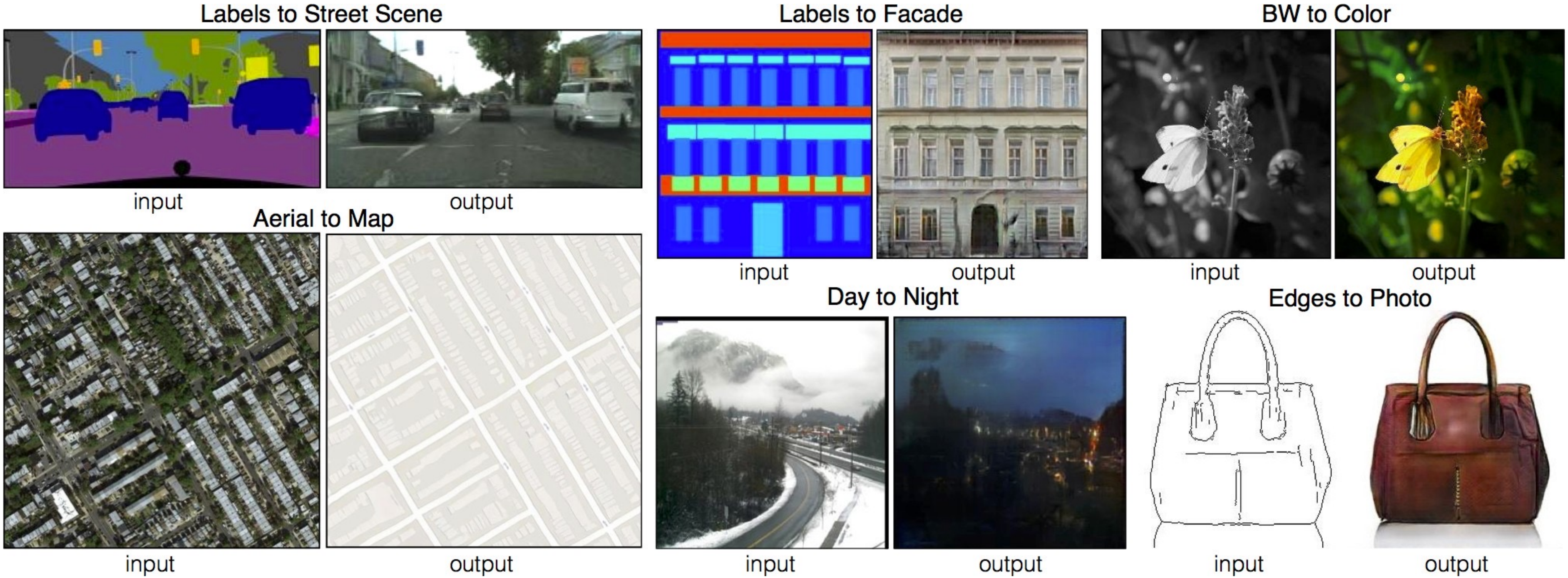


Image Super-Resolution: Low-Res to High-Res

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

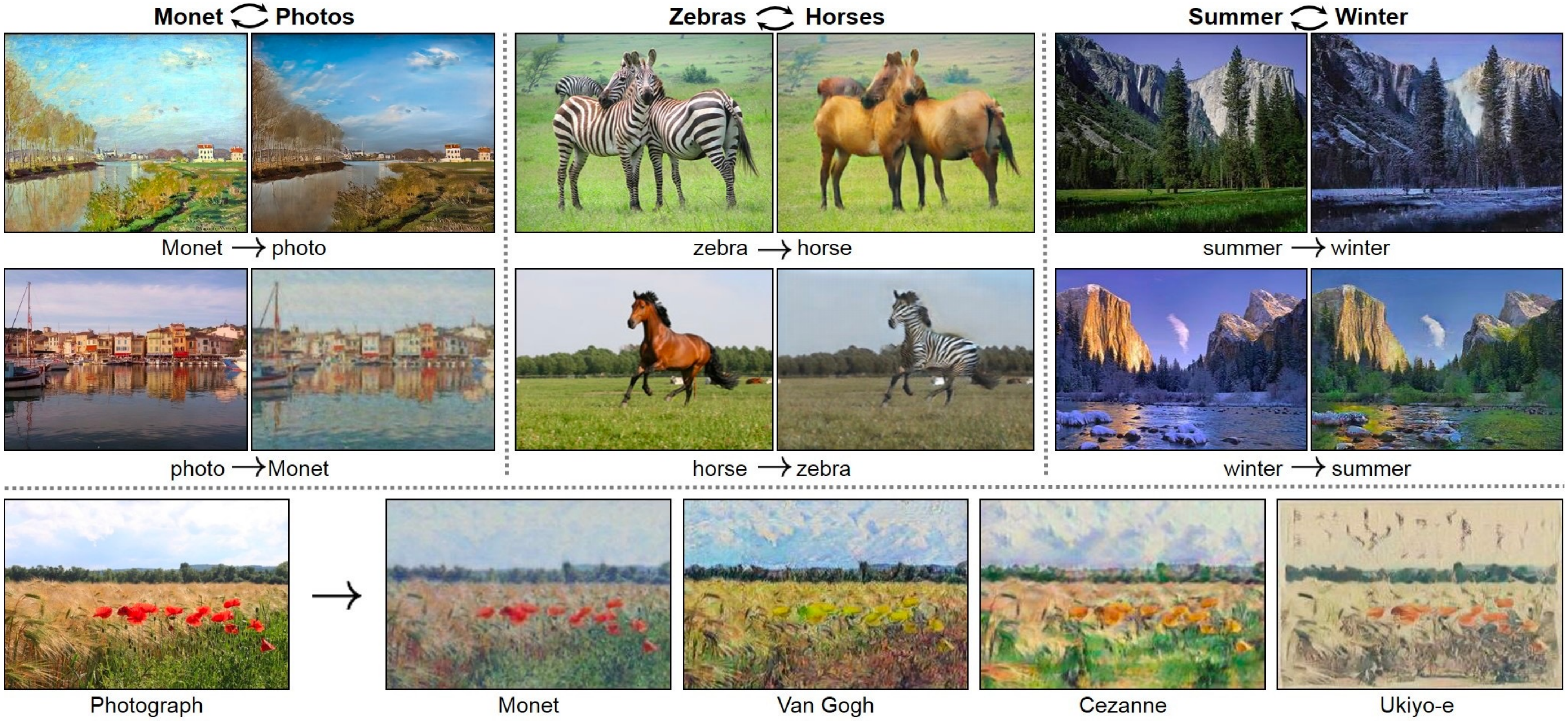
Ledig et al, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”, CVPR2017

Image-to-Image Translation: Pix2Pix



Isola et al, "Image-to-Image Translation with Conditional Adversarial Nets", CVPR2017

Unpaired Image-to-Image Translation: CycleGAN



Zhu et al, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", ICCV 2017

Any Question ?