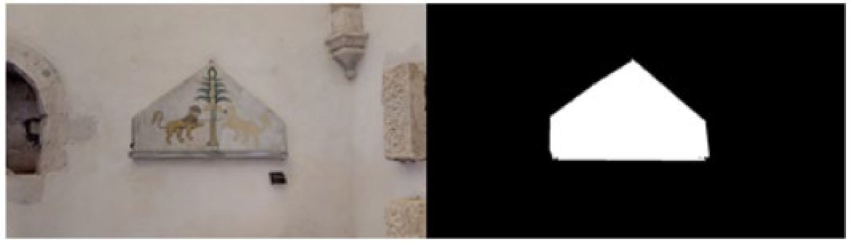
MCGAN: Mask Controlled Generative Adversarial Network for Image Retargeting

Jilyan Bianca Dy, Daniel Stanley Tan, Kai-Lung Hua

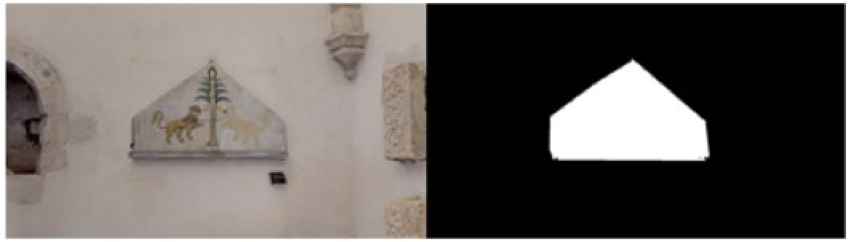
MCGAN: Mask Controlled Generative Adversarial Network for ~~Image Retargeting~~

Content-aware resizing



MCGAN: Mask Controlled Generative Adversarial Network for ~~Image Retargeting~~

Content-aware resizing



MCGAN: Mask Controlled Generative Adversarial Network for ~~Image Retargeting~~

Some model that produces an image

Content-aware resizing

What is image retargeting or content-aware resizing?

Resizing

Changes the scale of objects

Distorts the shape if aspect ratio is not preserved

But what if we want to change the size while preserving both?

Image Retargeting

Idea: Replicate rows or columns of pixels

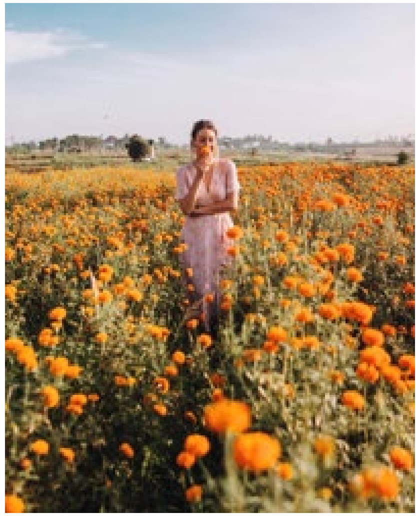
Seam Carving
(Method used in Photoshop)

Stretching artifacts

Image Retargeting

Why not replicate patches / objects?

Image Retargeting



Problem: It has no idea what it should and should not replicate

Our goal / contribution

- Reduce generation of unnatural objects due to lack of semantic understanding

Our goal / contribution

- Reduce generation of unnatural objects due to lack of semantic understanding
- Allow user to have control over desired object
  - (i.e., changing object location, replicating, removing)

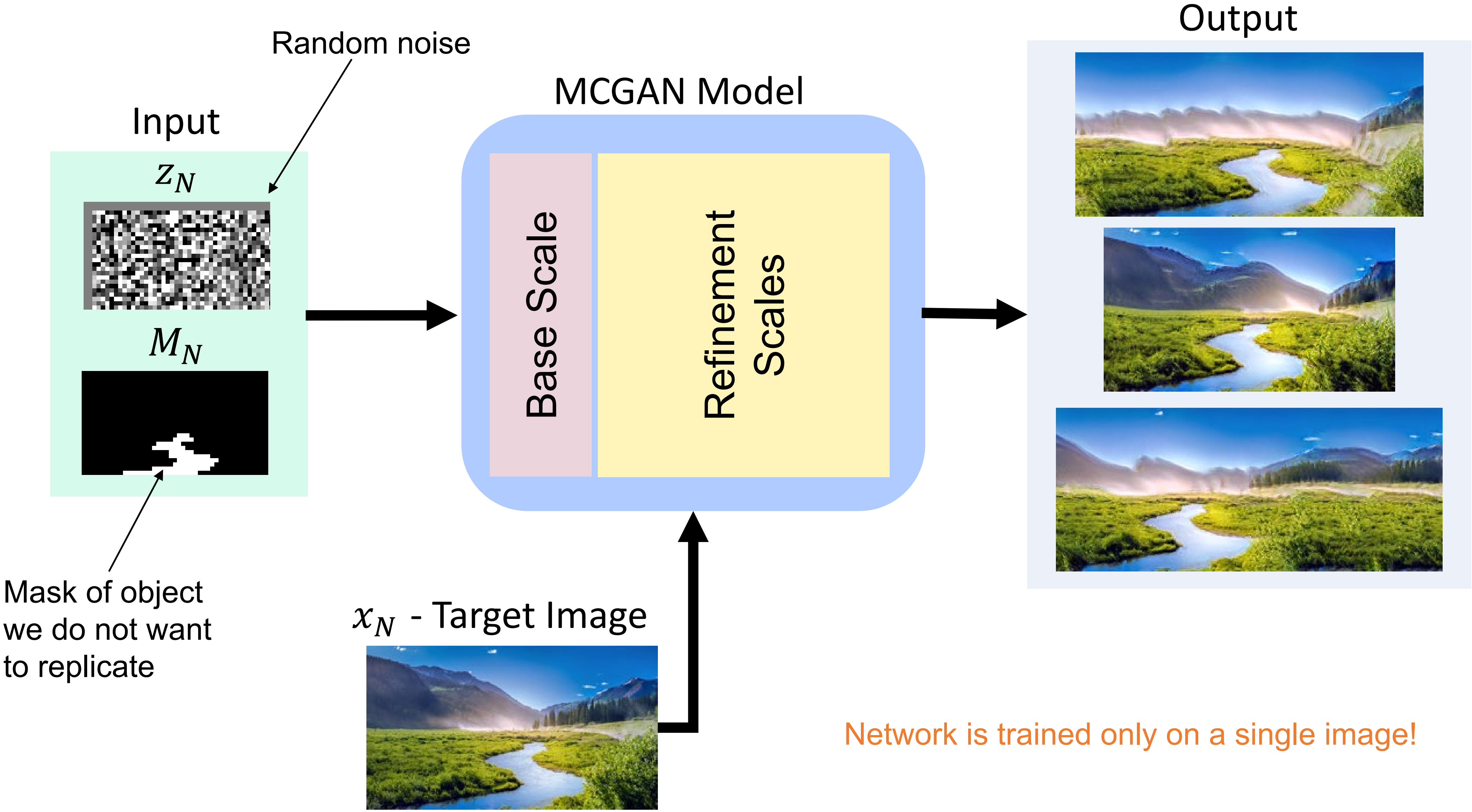
How can we de-associate the objects that can be replicated from those that should not be replicated?

Should not be replicated

Can be replicated

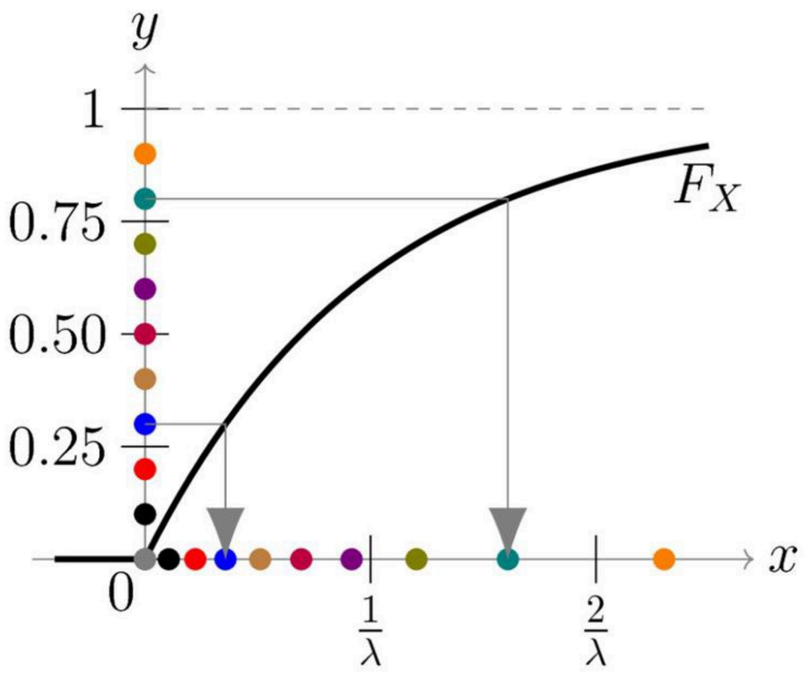
Make it interactive and use masks!

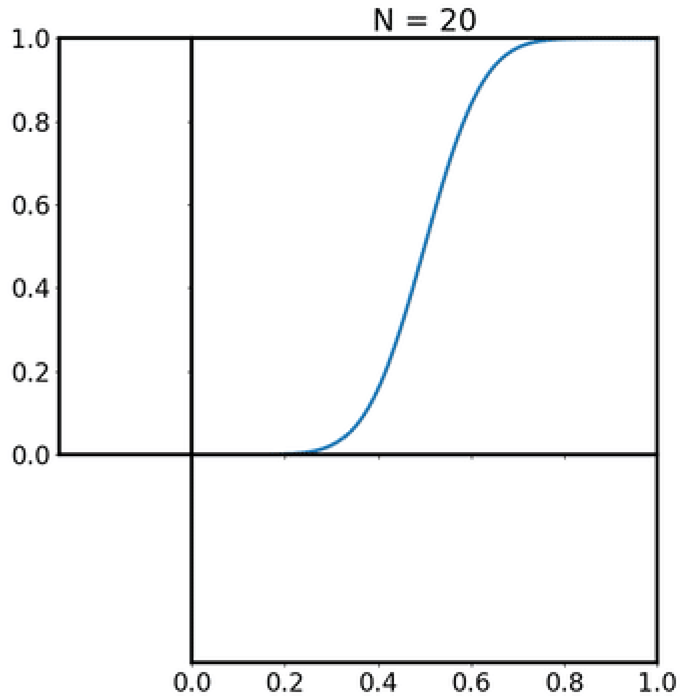
Method



Network is trained only on a single image!

How can inputting random noise produce anything useful?

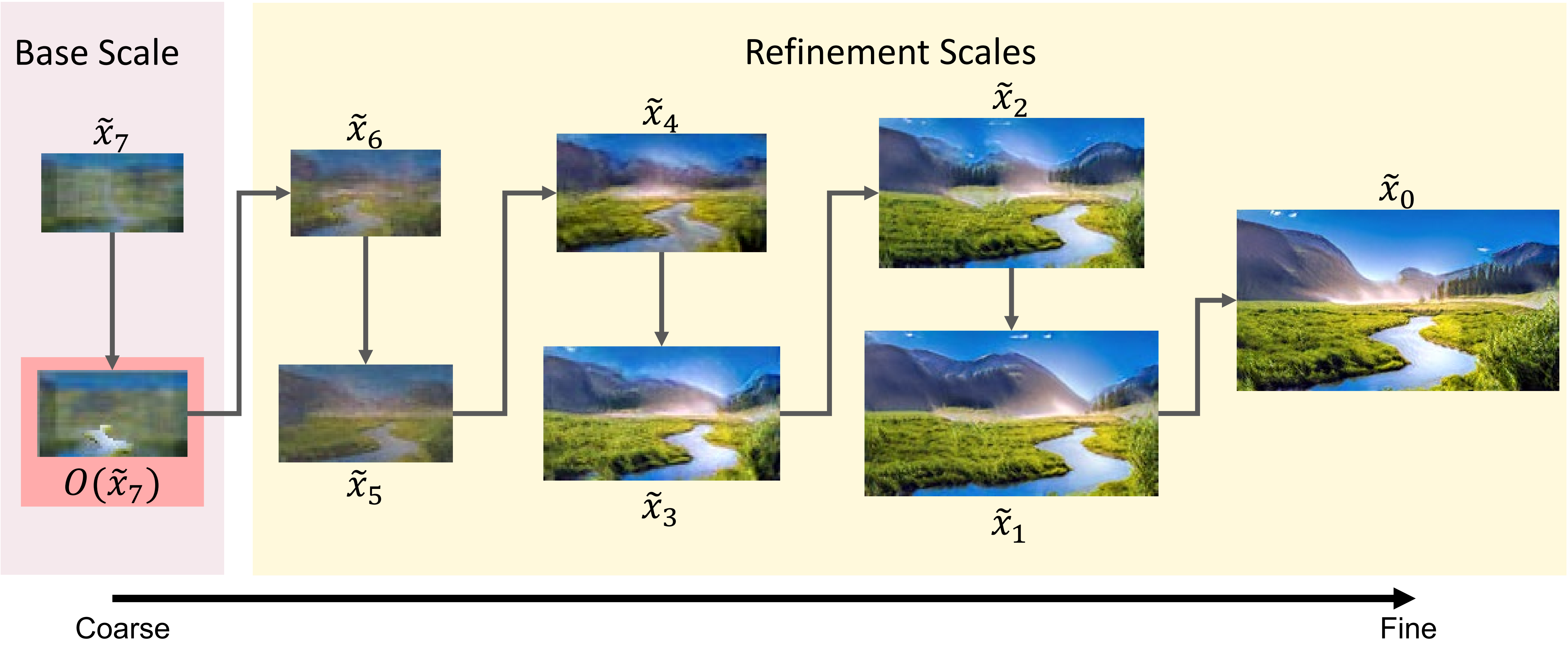




$\tilde{x}_n$ - Generated image at scale n

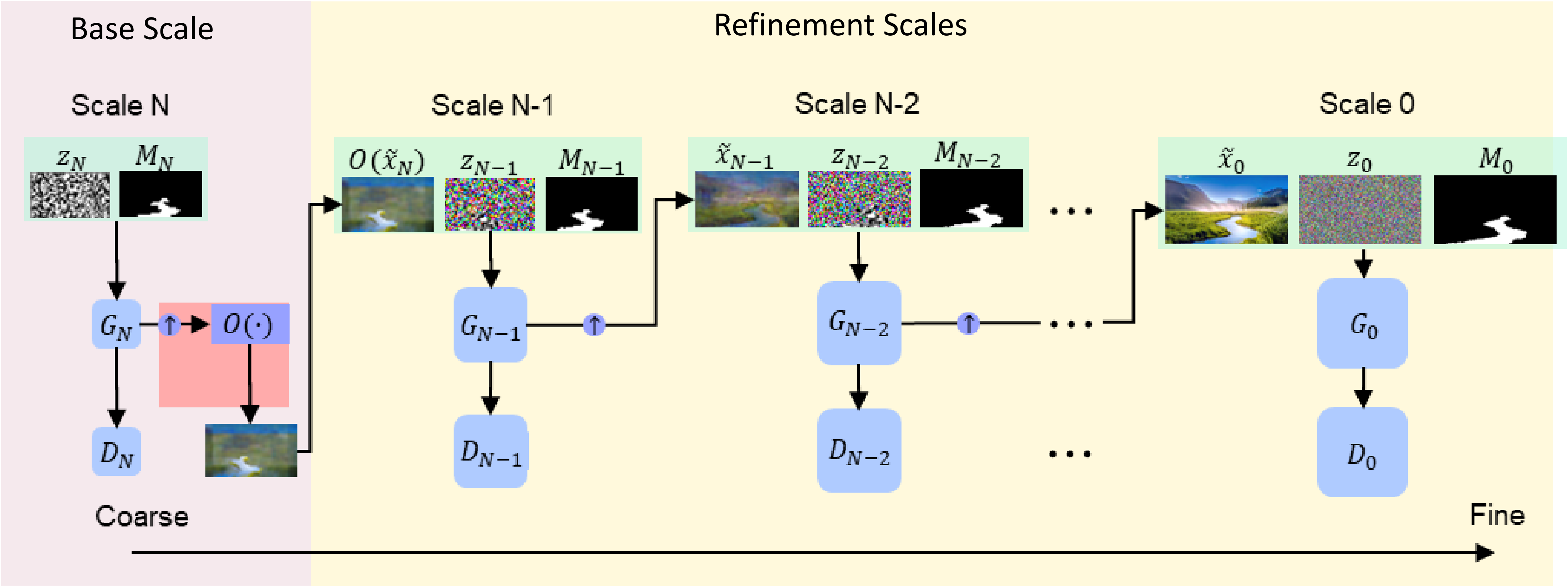
$N = 7$

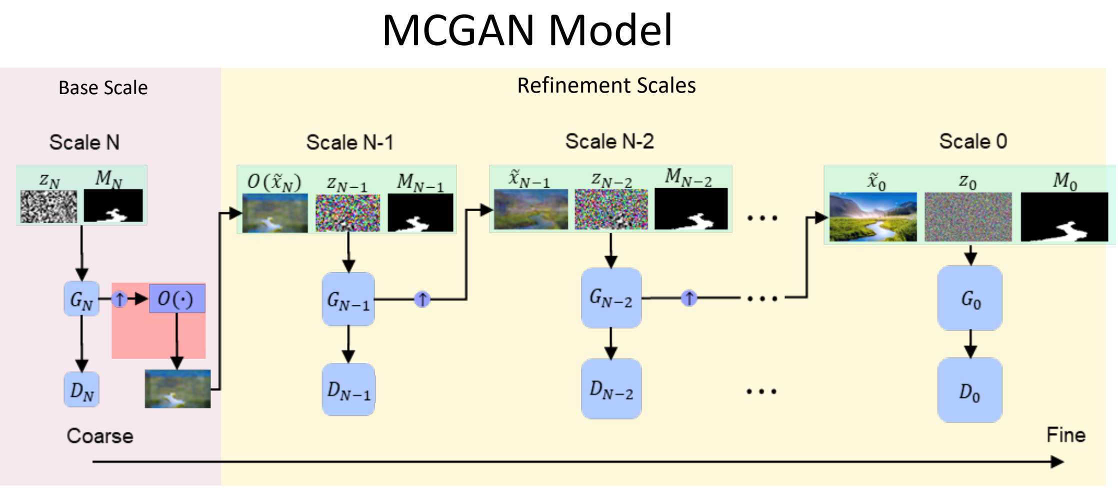
MCGAN Model



MCGAN Model

- ↑ - Upsample
- $O(\cdot)$- Overlay Function
- G_n - Generator
- D_n - Discriminator

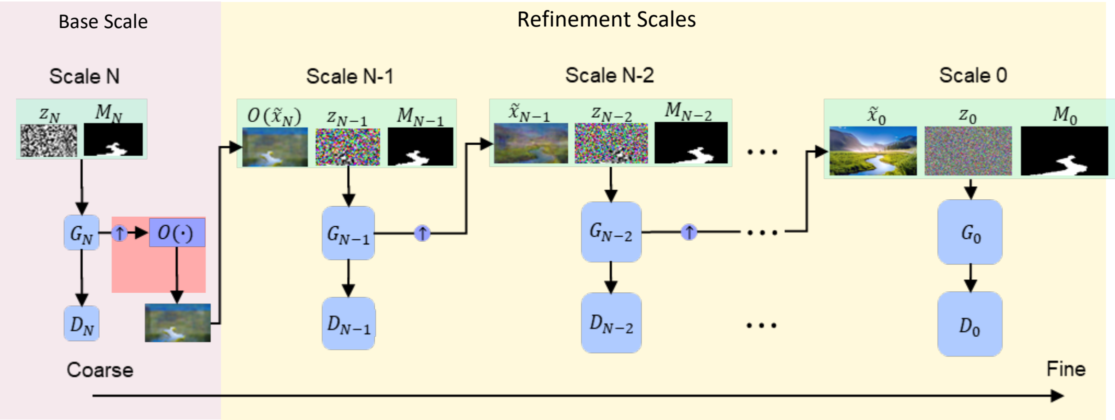




$$\min_{G_n} \max_{D_n} \mathcal{L}_{adv}(G_n, D_n) + \alpha\mathcal{L}_{rec}(G_n) + \lambda\mathcal{L}_{DA}(G_n)$$

- $\mathcal{L}_{adv}$ - Adversarial Loss
- $\mathcal{L}_{rec}$ - Reconstruction Loss
- $\mathcal{L}_{DA}$ - De-association Loss
- G_n - Generator at scale n
- D_n - Discriminator at scale n
- α, λ - Hyperparameters

MCGAN Model



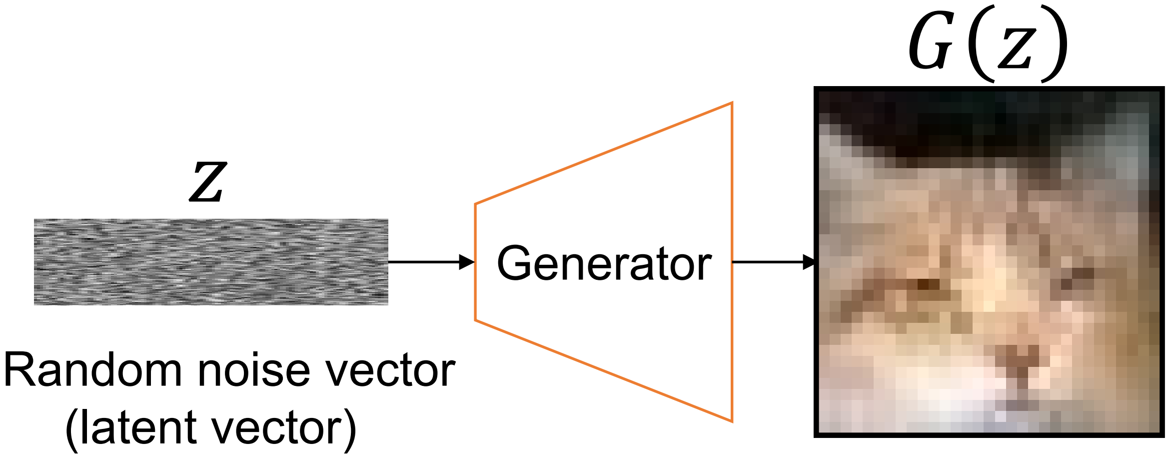
$$\min_{G_n} \max_{D_n} \mathcal{L}_{adv}(G_n, D_n) + \alpha\mathcal{L}_{rec}(G_n) + \lambda\mathcal{L}_{DA}(G_n)$$

- $\mathcal{L}_{adv}$ - Adversarial Loss
- $\mathcal{L}_{rec}$ - Reconstruction Loss
- $\mathcal{L}_{DA}$ - De-association Loss
- G_n - Generator at scale n
- D_n - Discriminator at scale n
- α, λ - Hyperparameters

- Discriminator: distinguish real and fake
- Generator: fool discriminator into thinking generated image is real

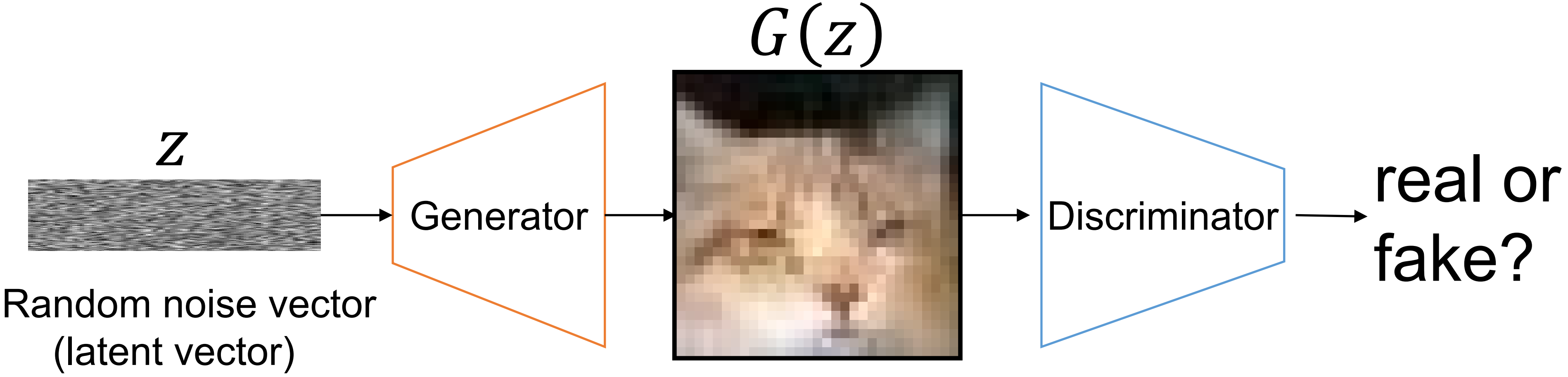
We used the Wasserstein GAN with gradient penalty (WGAN-GP)

$$\mathbb{E}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathbb{E}_{x\sim P_r}[D(x)] + \lambda_{GP}\mathbb{E}_{\hat{x}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]$$



G: generate fake samples

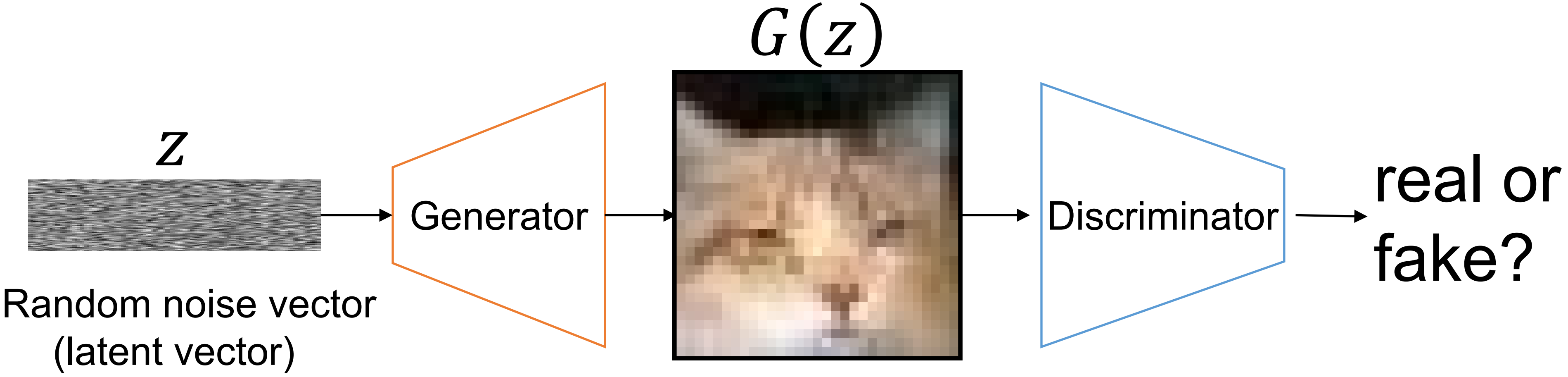
[Goodfellow et al. 2014



G: generate fake samples that can fool D

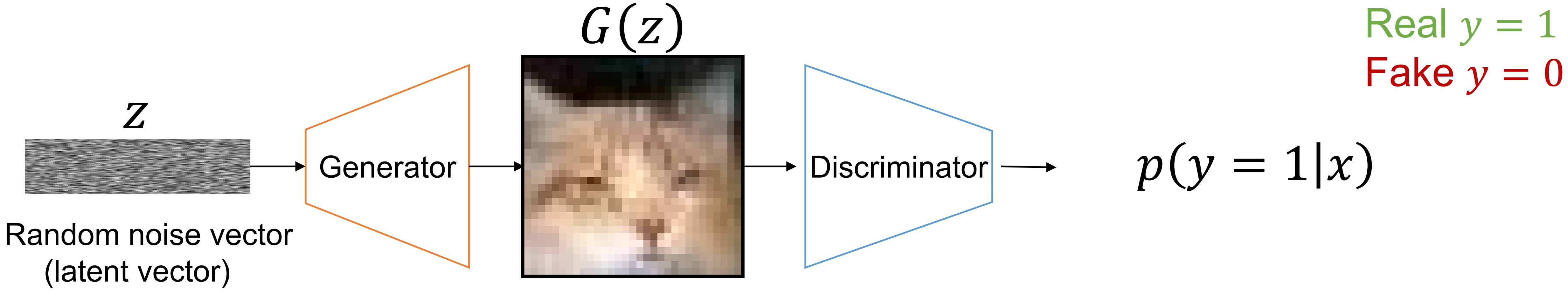
D: classify fake samples vs. real images

[Goodfellow et al. 2014]



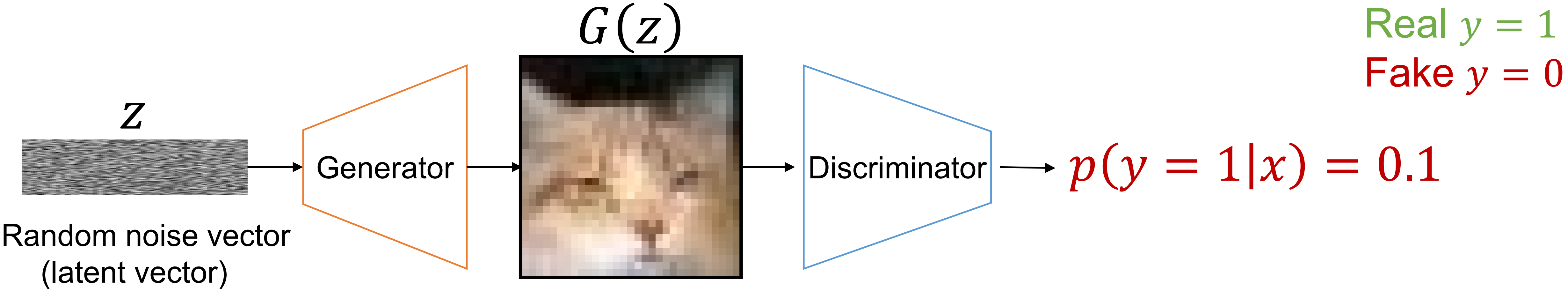
$$\min_G \max_D \mathbb{E}_{z,x}[\log D(x) + \log(1 - D(G(z)))]$$

[Goodfellow et al. 2014]



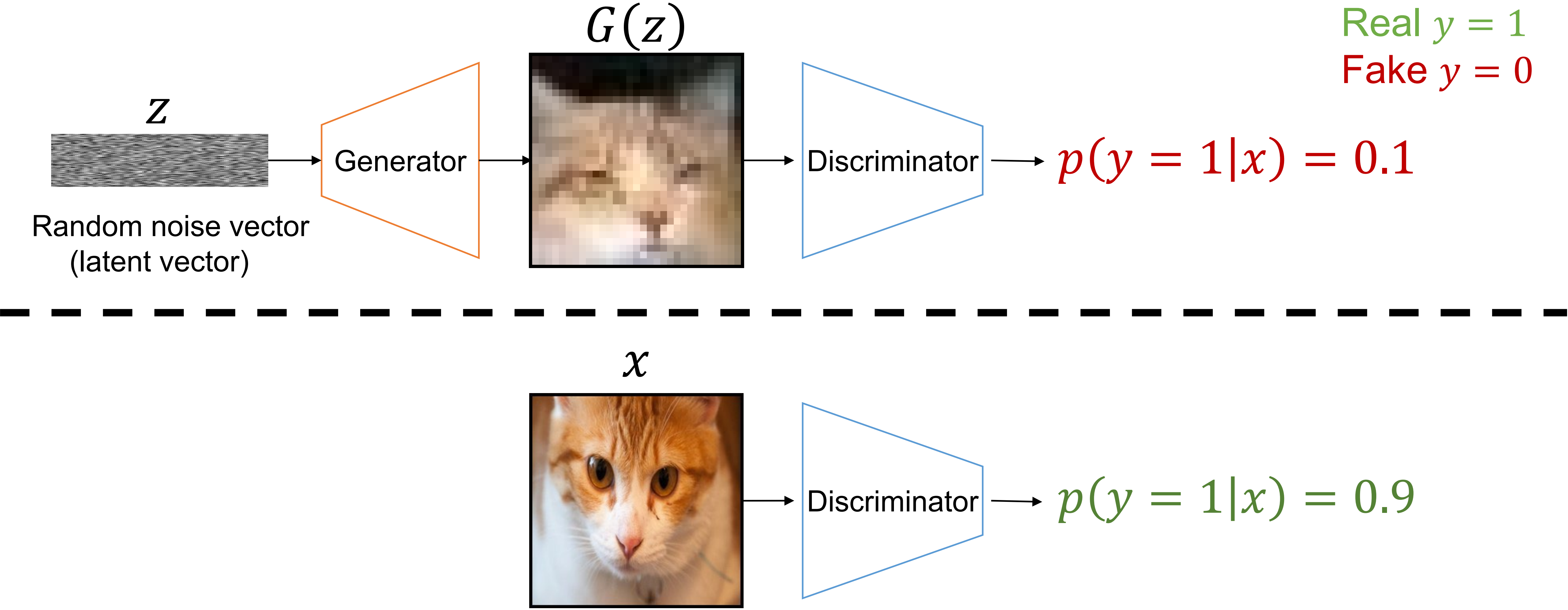
$$\min_G \max_D \mathbb{E}_{z,x}[\log D(x) + \log(1 - D(G(z))]$$

[Goodfellow et al. 2014



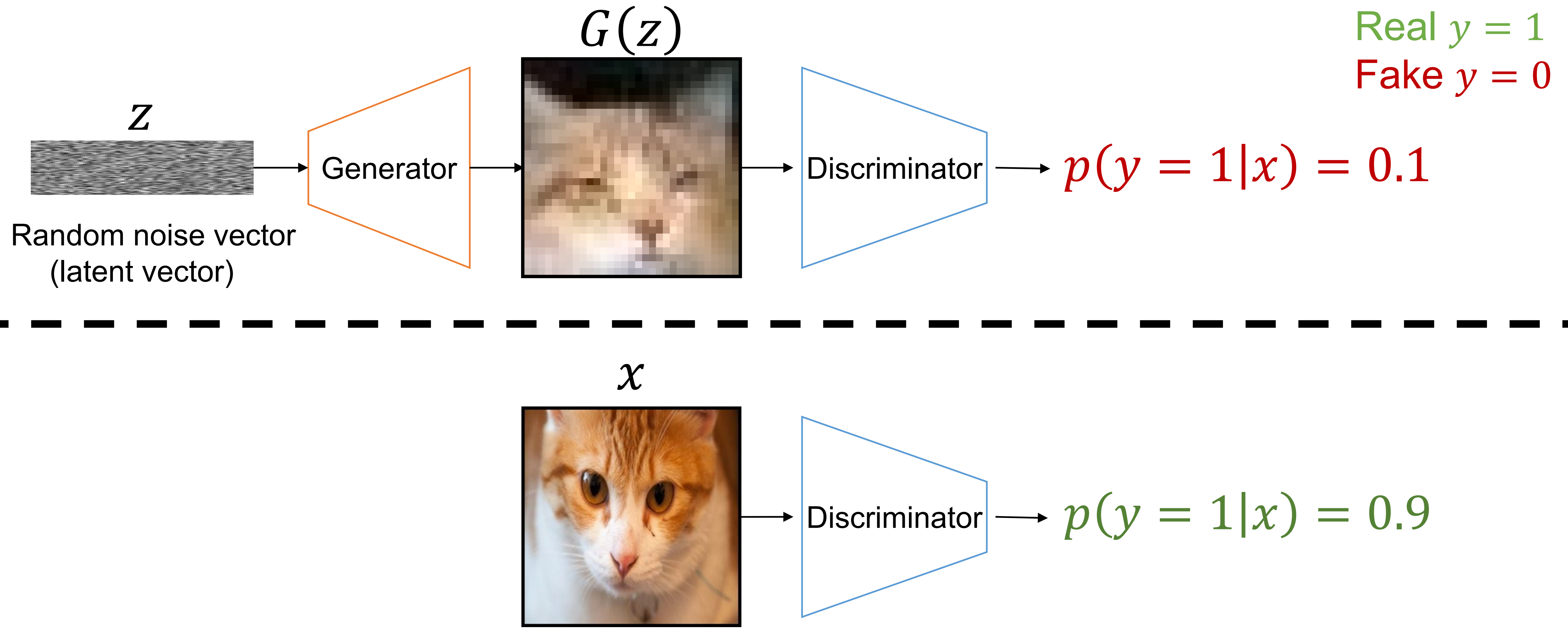
$$\min_G \max_D \mathbb{E}_{z,x}[\log D(x) + \log(1 - \underbrace{D(G(z))}_{\textbf{fake}})]$$

[Goodfellow et al. 2014



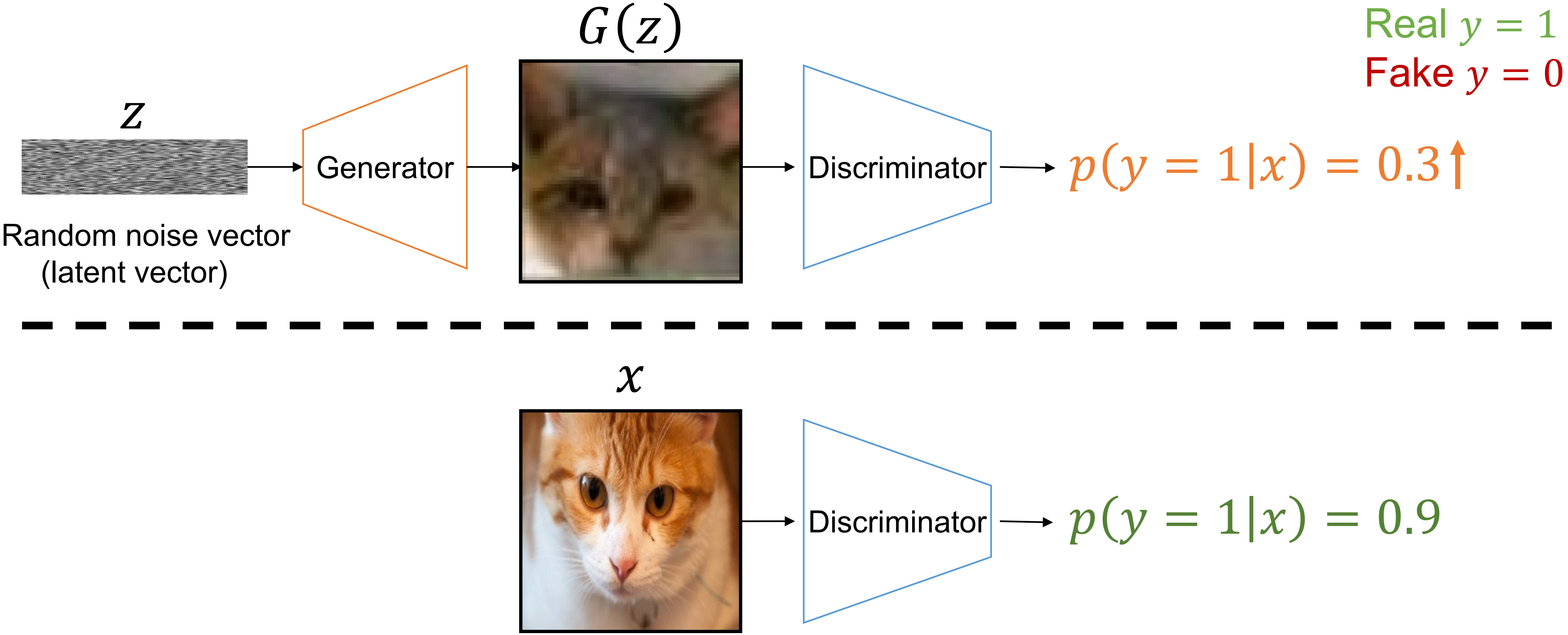
$$\min_G \max_D \mathbb{E}_{z,x}[\log \underbrace{D(x)}_{\text{real}} + \log(1 - D(G(z)))]$$

[Goodfellow et al. 2014



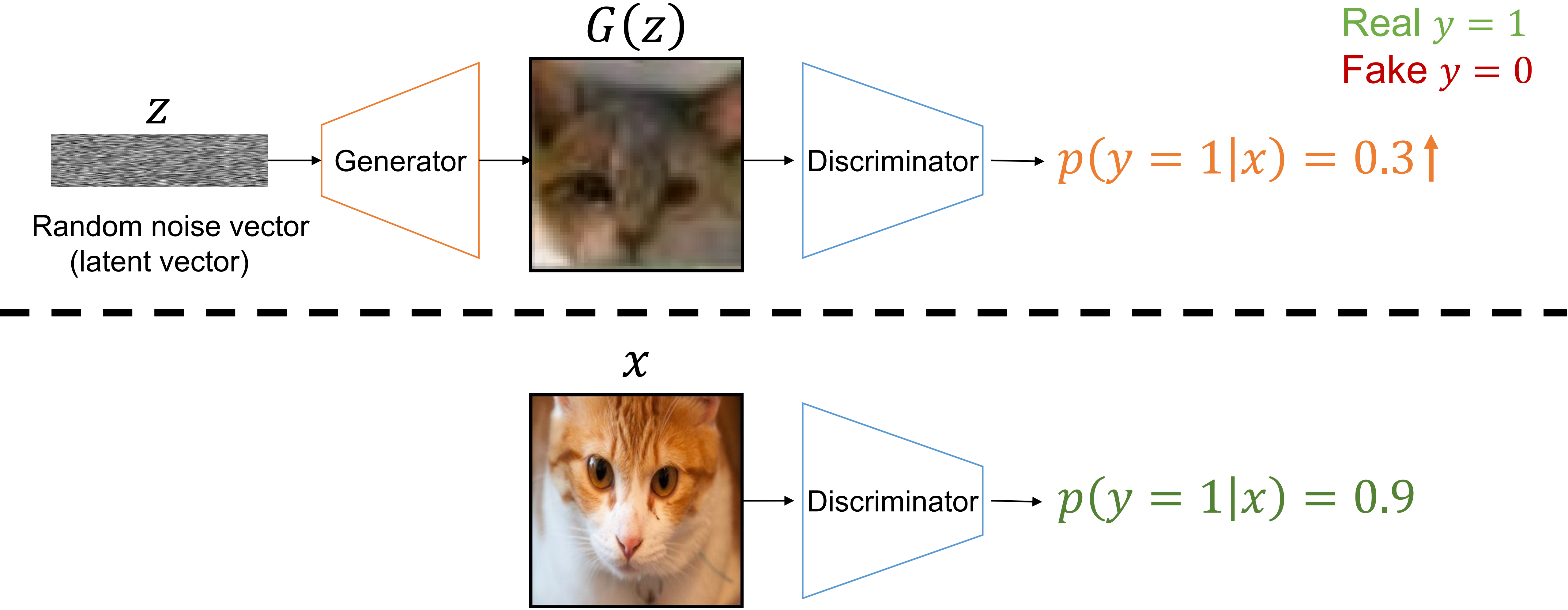
$$\min_{G} \max_{D} \mathbb{E}_{z,x}[\underbrace{\log D(x)}_{\textbf{real}} + \underbrace{\log(1 - D(G(z)))}_{\textbf{fake}}]$$

[Goodfellow et al. 2014



$$\min_G \max_D \mathbb{E}_{z,x}\left[\log D(x) + \underbrace{\log(1 - D(G(z)))}_{\textbf{Update } G}\right]$$

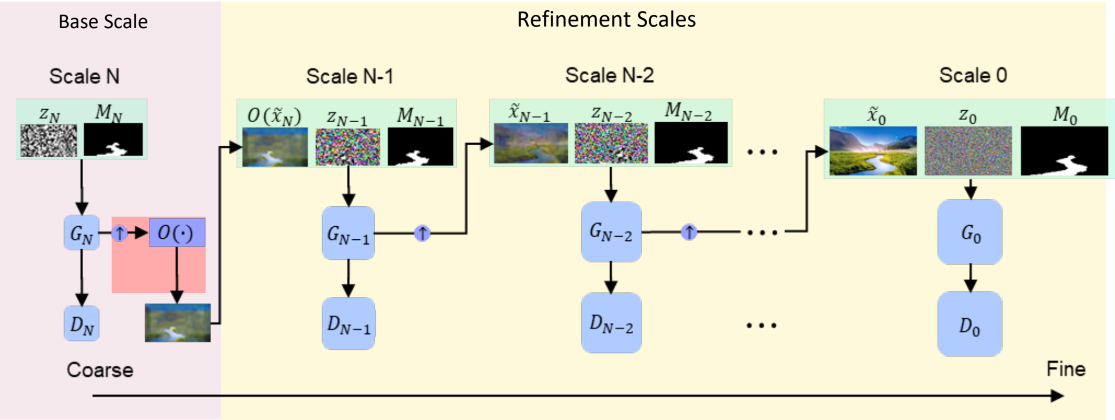
[Goodfellow et al. 2014



Discriminator implicitly learns a measure of realism
from examples of real and fake

[Goodfellow et al. 2014

MCGAN Model



$$\min_{G_n} \max_{D_n} \mathcal{L}_{adv}(G_n, D_n) + \alpha\mathcal{L}_{rec}(G_n) + \lambda\mathcal{L}_{DA}(G_n)$$

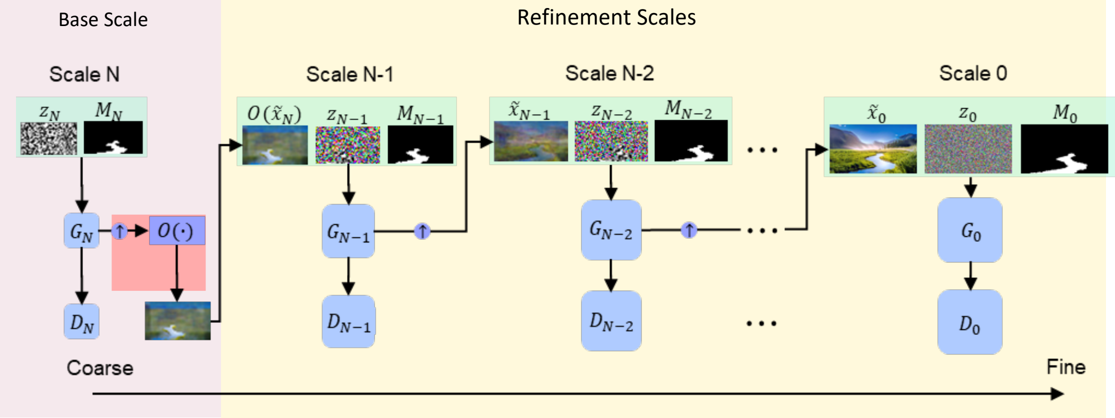
- $\mathcal{L}_{adv}$ - Adversarial Loss
- $\mathcal{L}_{rec}$ - Reconstruction Loss
- $\mathcal{L}_{DA}$ - De-association Loss
- G_n - Generator at scale n
- D_n - Discriminator at scale n
- α, λ - Hyperparameters

- Discriminator: distinguish real and fake
- Generator: fool discriminator into thinking generated image is real

We used the Wasserstein GAN with gradient penalty (WGAN-GP)

$$\mathbb{E}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathbb{E}_{x\sim P_r}[D(x)] + \lambda_{GP}\mathbb{E}_{\hat{x}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]$$

MCGAN Model



$$\min_{G_n} \max_{D_n} \mathcal{L}_{adv}(G_n, D_n) + \alpha\mathcal{L}_{rec}(G_n) + \lambda\mathcal{L}_{DA}(G_n)$$

- $\mathcal{L}_{adv}$ - Adversarial Loss
- $\mathcal{L}_{rec}$ - Reconstruction Loss
- $\mathcal{L}_{DA}$ - De-association Loss
- G_n - Generator at scale n
- D_n - Discriminator at scale n
- α, λ - Hyperparameters

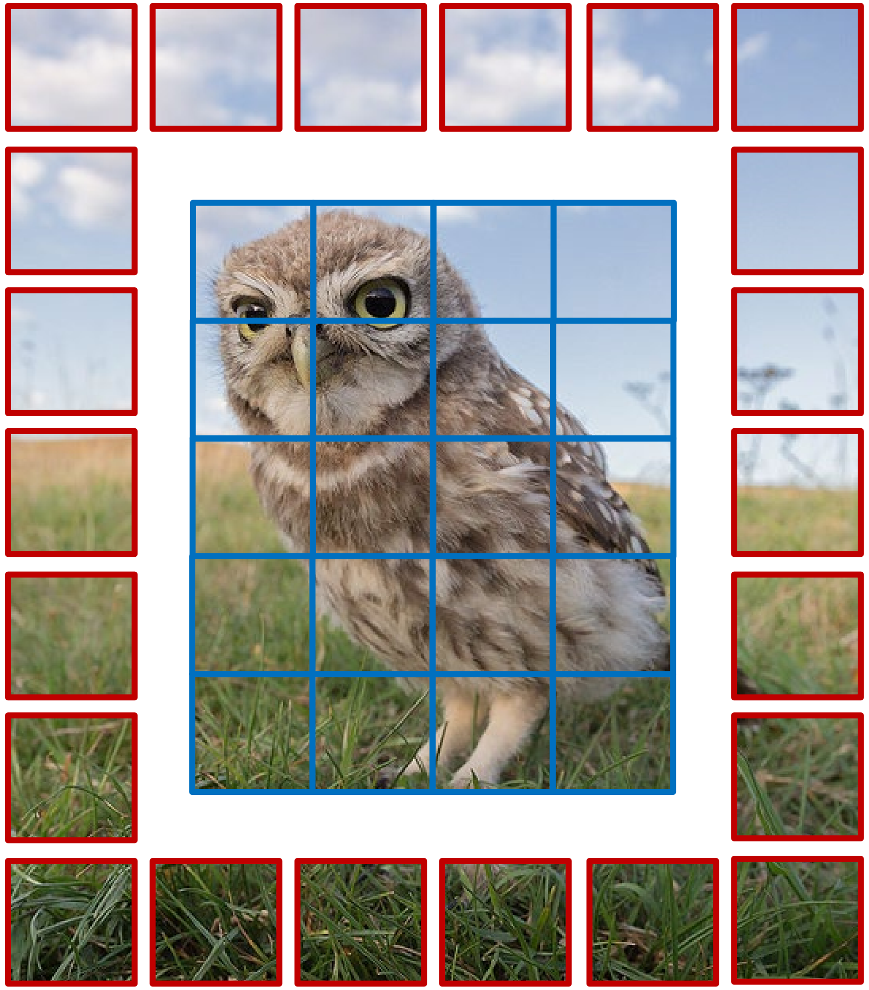
Ensures that we can reconstruct the original image

$$||G_n(z_n^{rec}, (\tilde{x}_{n+1}^{rec}) \uparrow^r) - x_n||_2$$

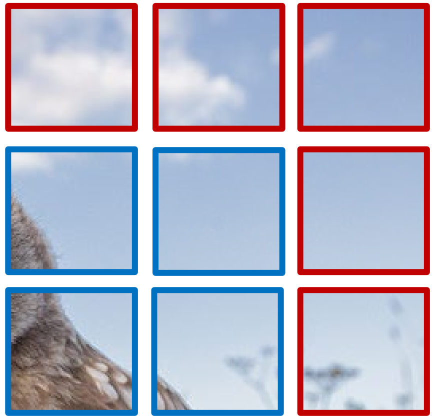
- G_n - Generator at scale n
- z_n^{rec} - upsampled noise map
- $(\tilde{x}_{n+1}^{rec}) \uparrow^r$ - upsampled generated image from previous scale
- x_n - downsampled original image for scale n

- Discriminator views the image as patches

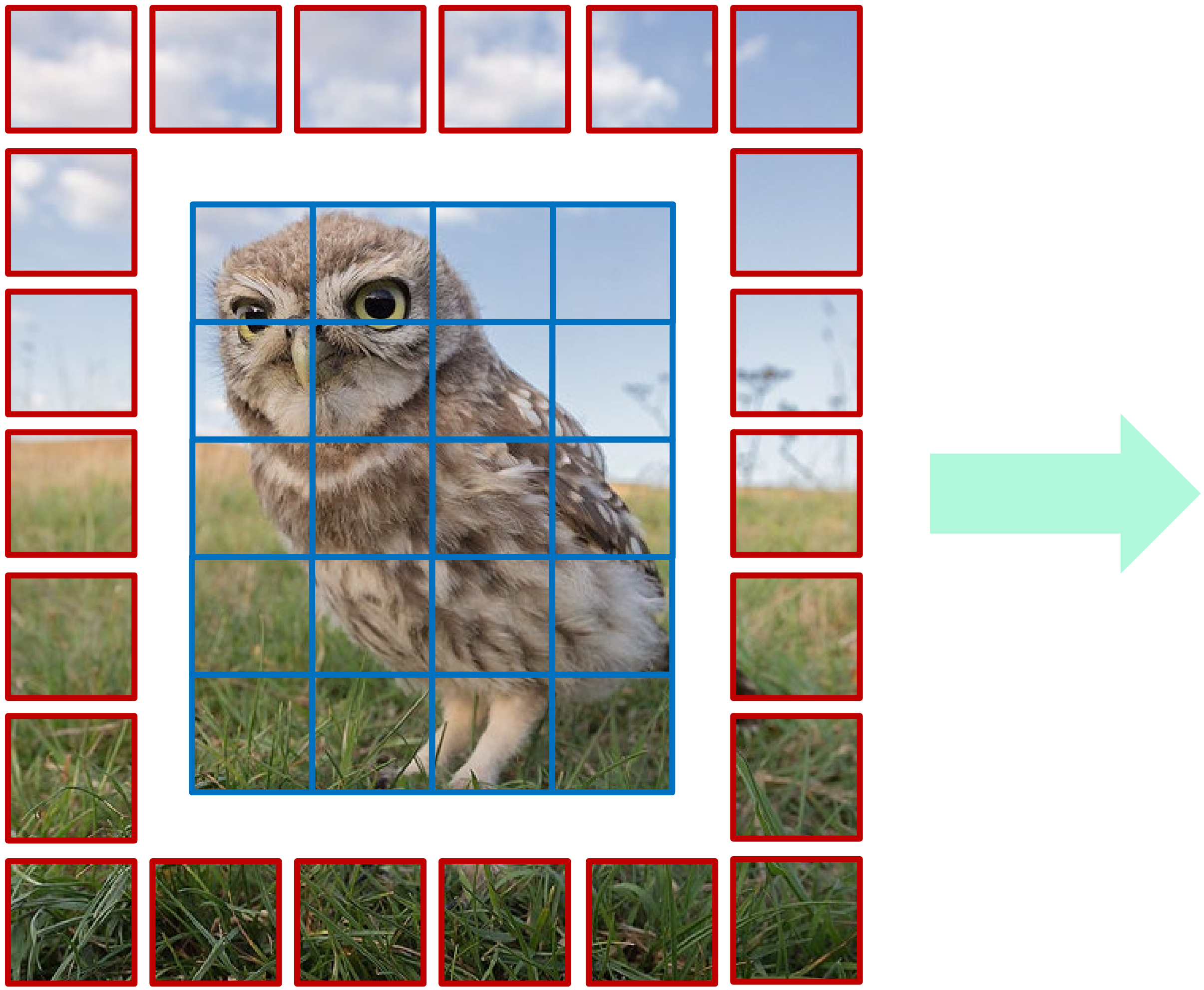
- The discriminator associates the patches with each other
- Image is fake if for example patch a is not beside patch b

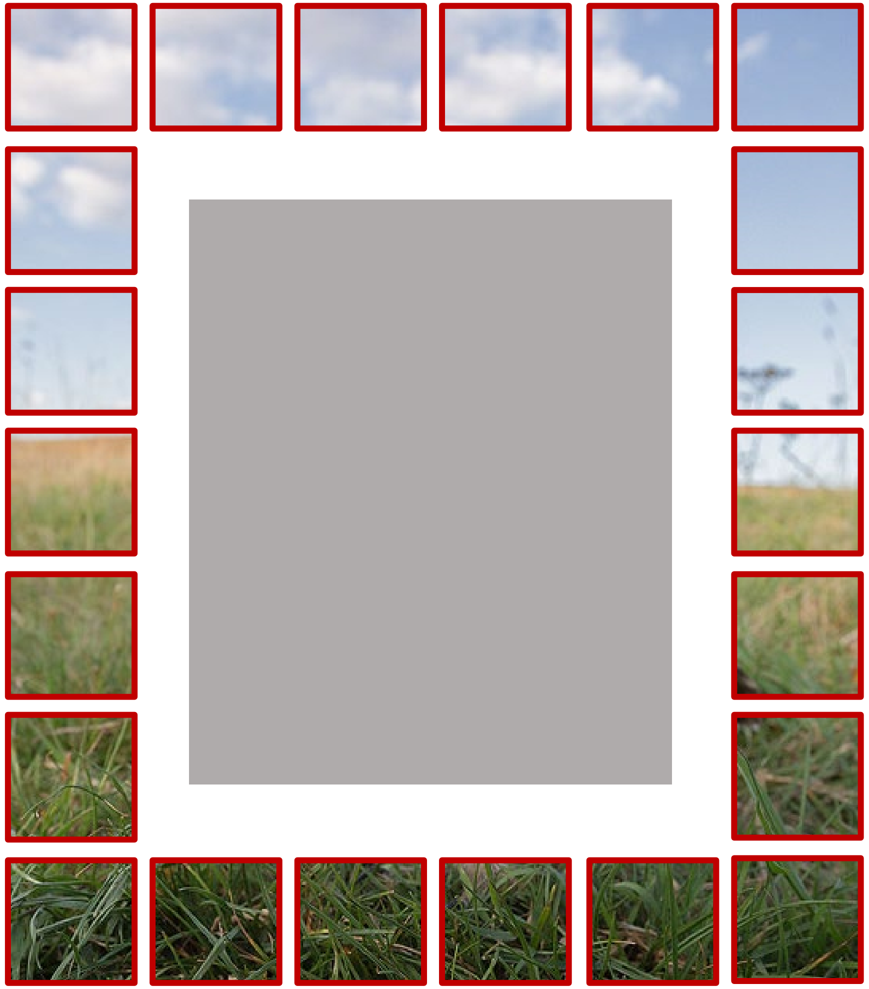


- Similarly, discriminator associates the blue patches (object of interest) with the red patches (its surroundings)




De-Association Loss





$$\mathcal{L}_{DA} = ||(M_n \times \tilde{x}_n) - 0||_2$$

Ablation Study

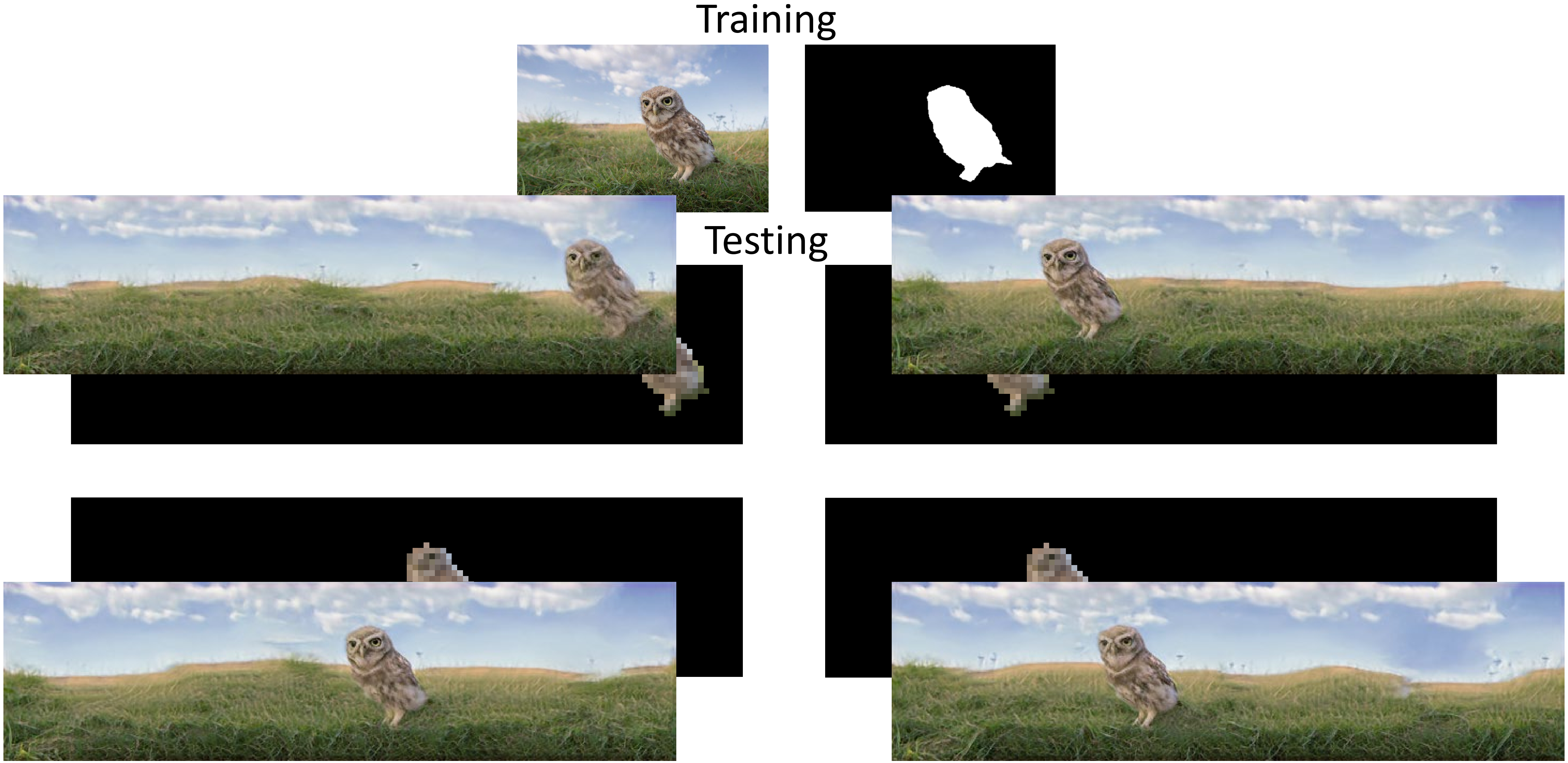
Input

Remove $\mathcal{L}_{DA}$

Remove Overlay

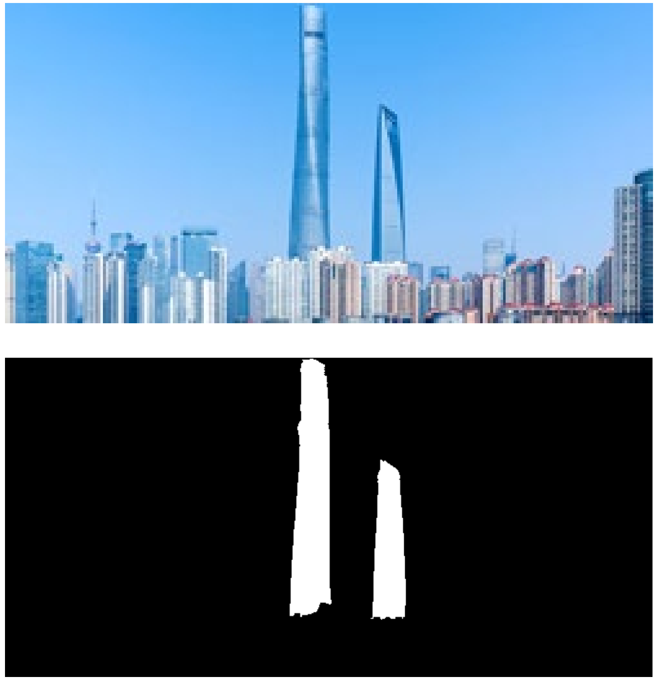
MCGAN

Setting Object Location

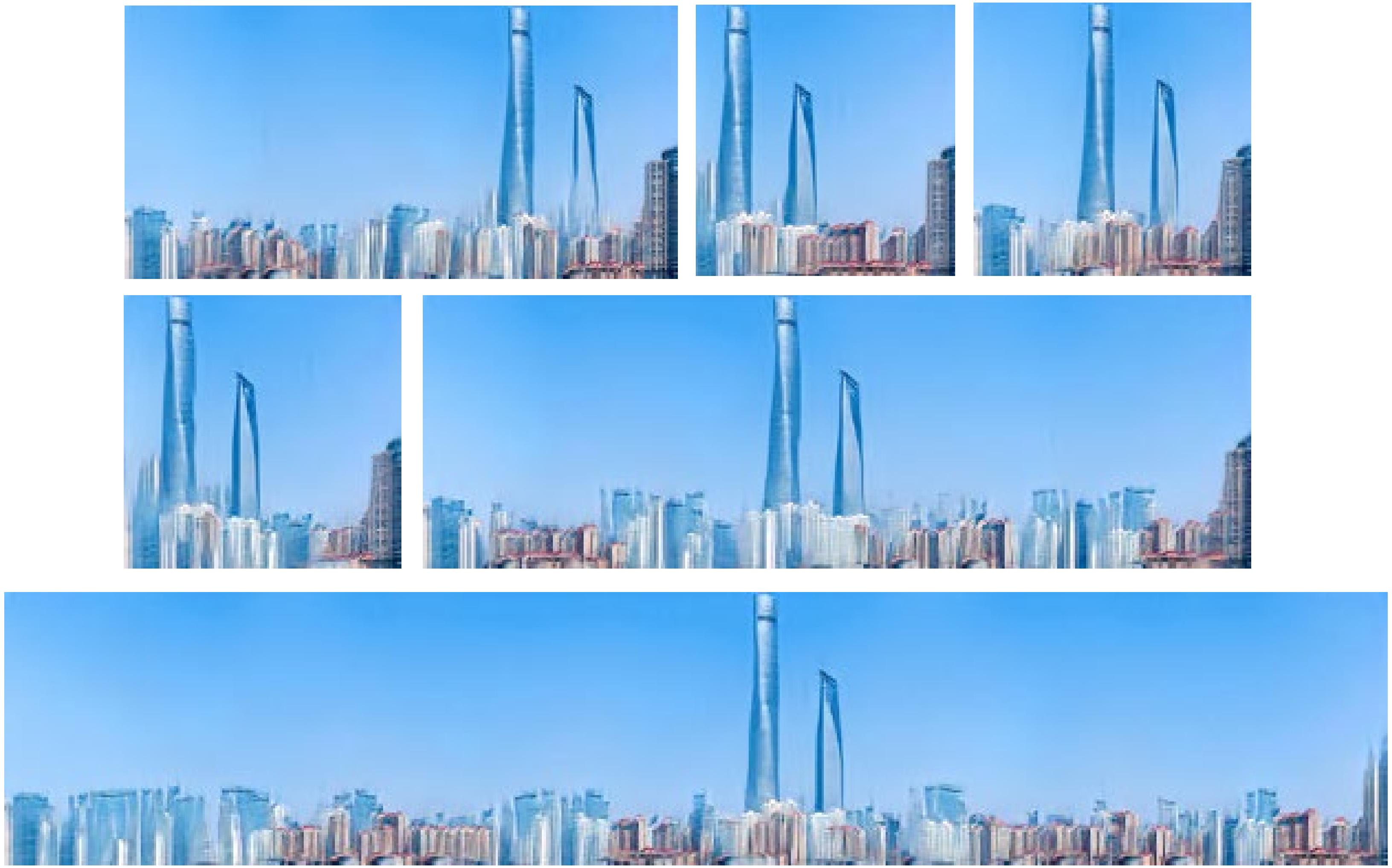


Setting Image Size and Object Location

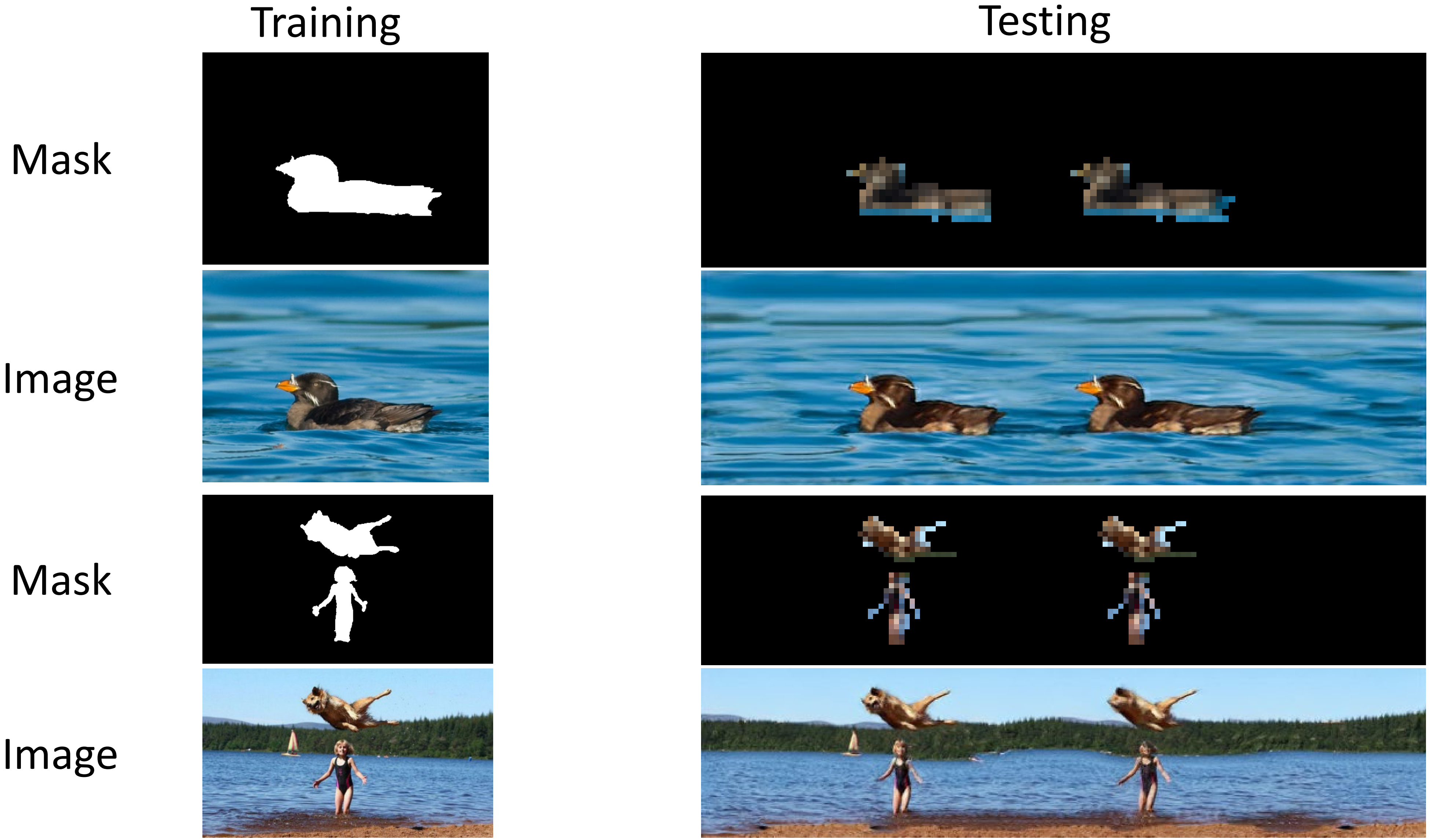
Input



Varying Location and Size



Object Replication



Object Removal

Training

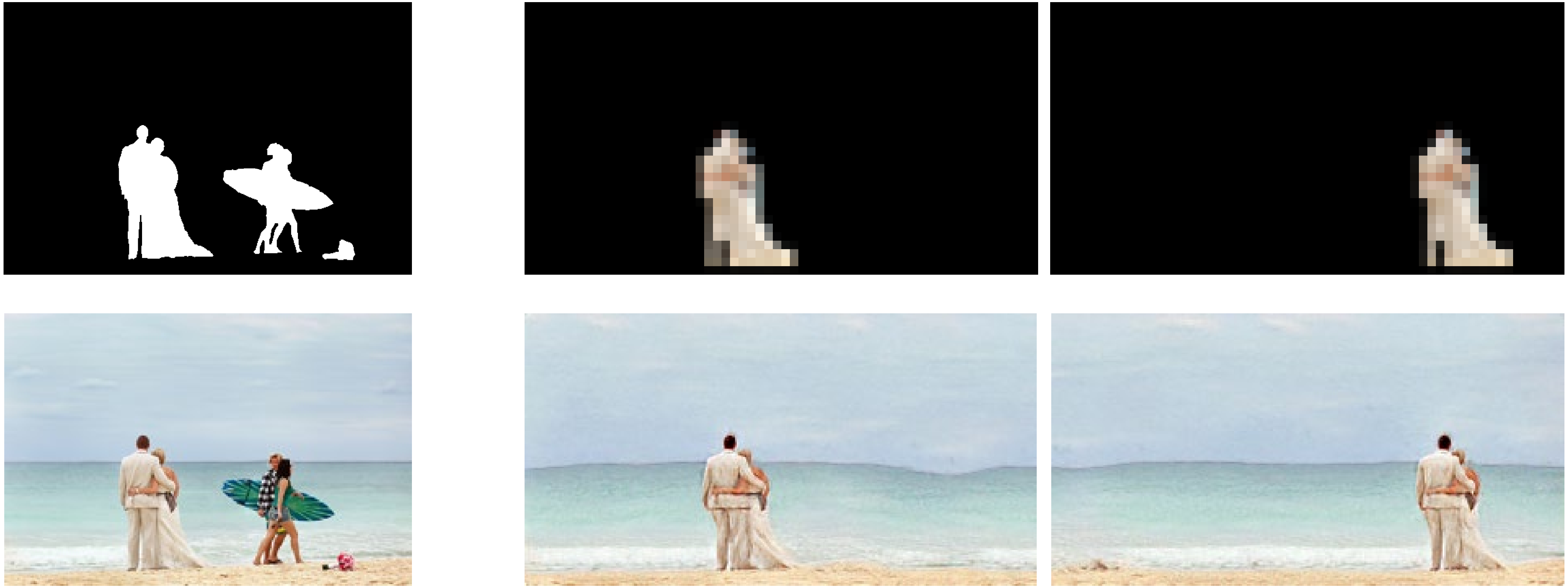
Image Mask Result

Object Removal

Training Testing

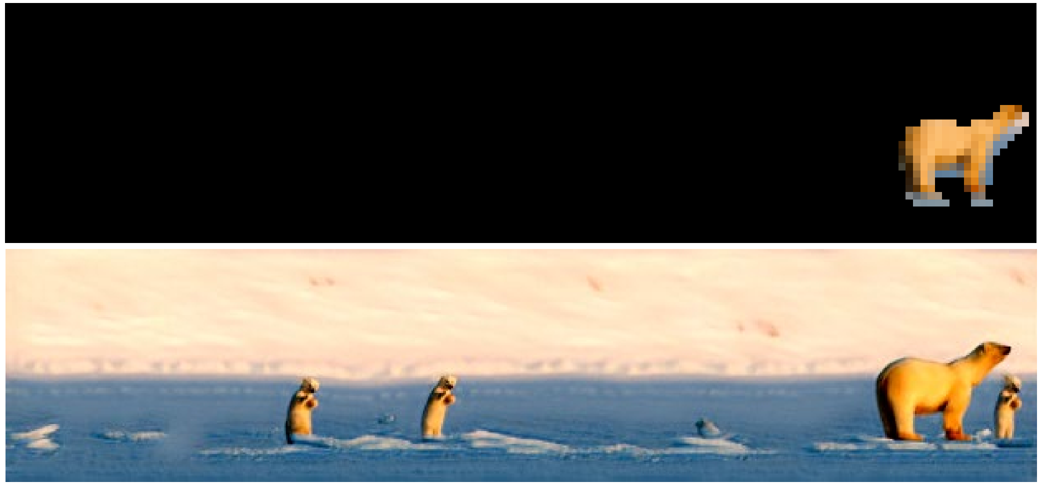
Mask

Image

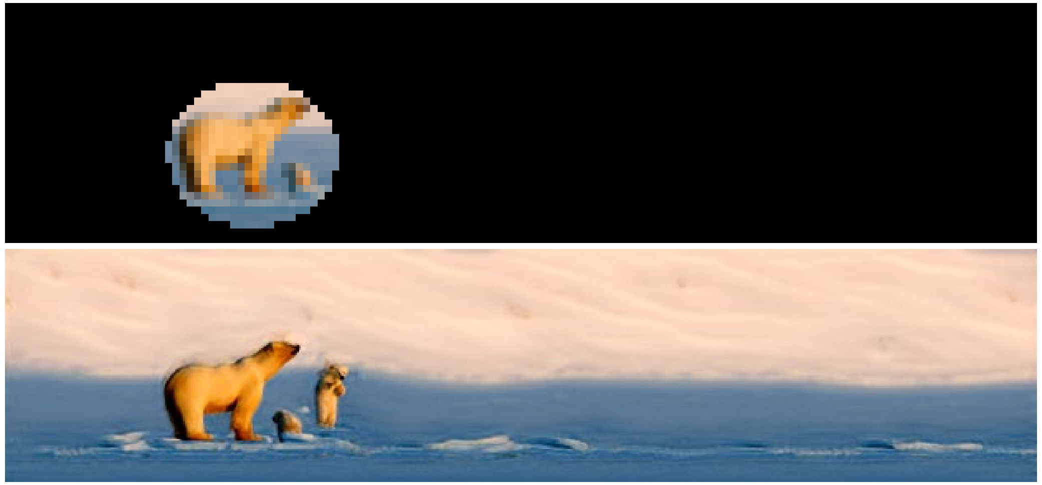


Loose Masks

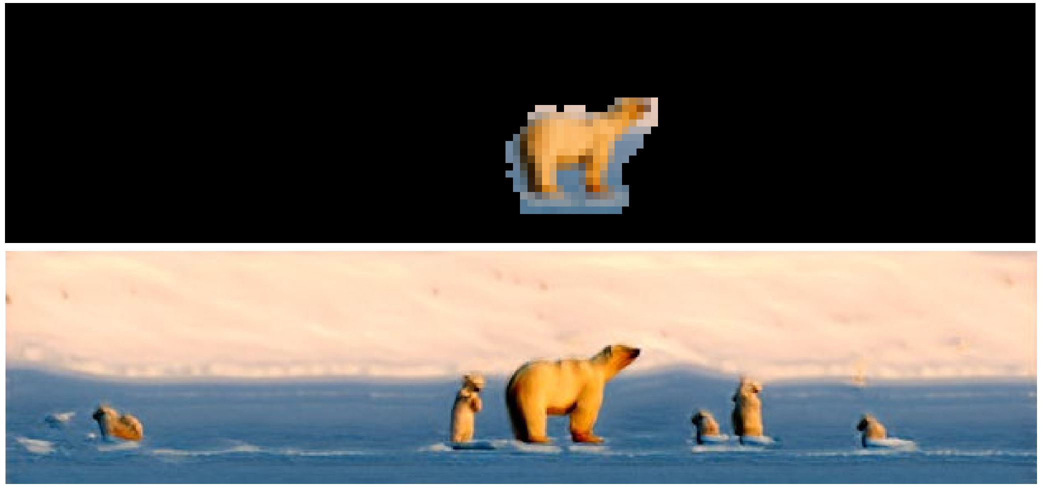
Fitted (Original)



Circle



Paint Brush

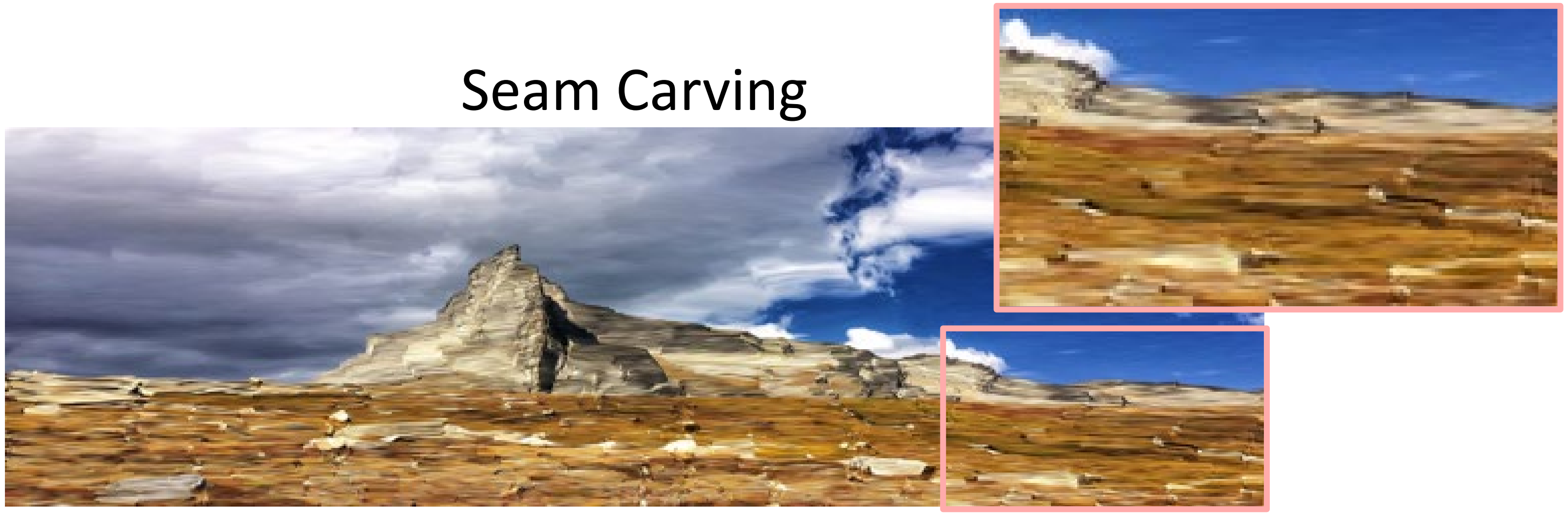


Box

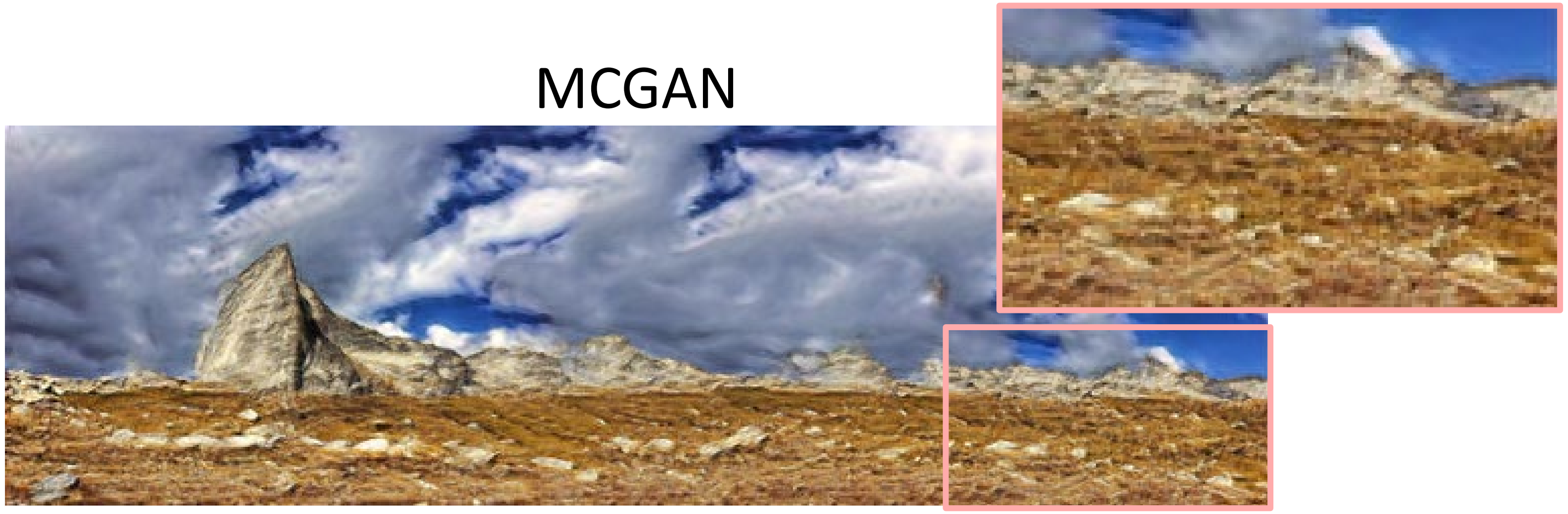
Comparison: Seam Carving

Input

Seam Carving



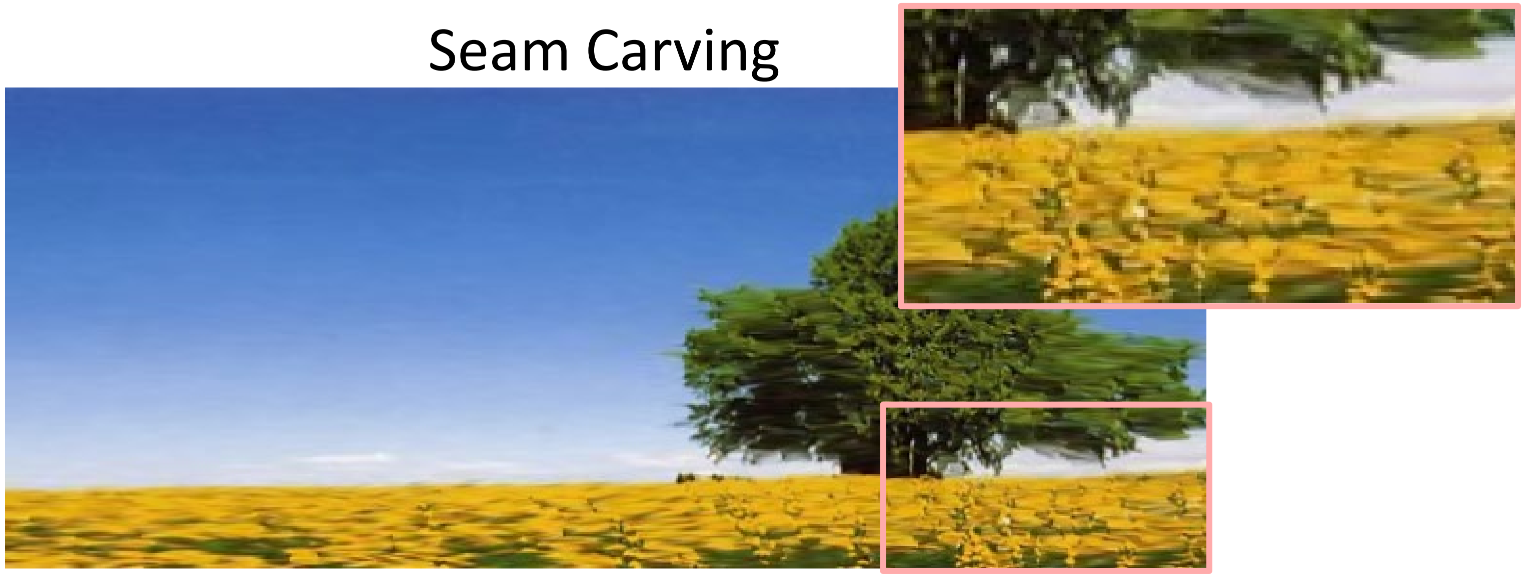
MCGAN



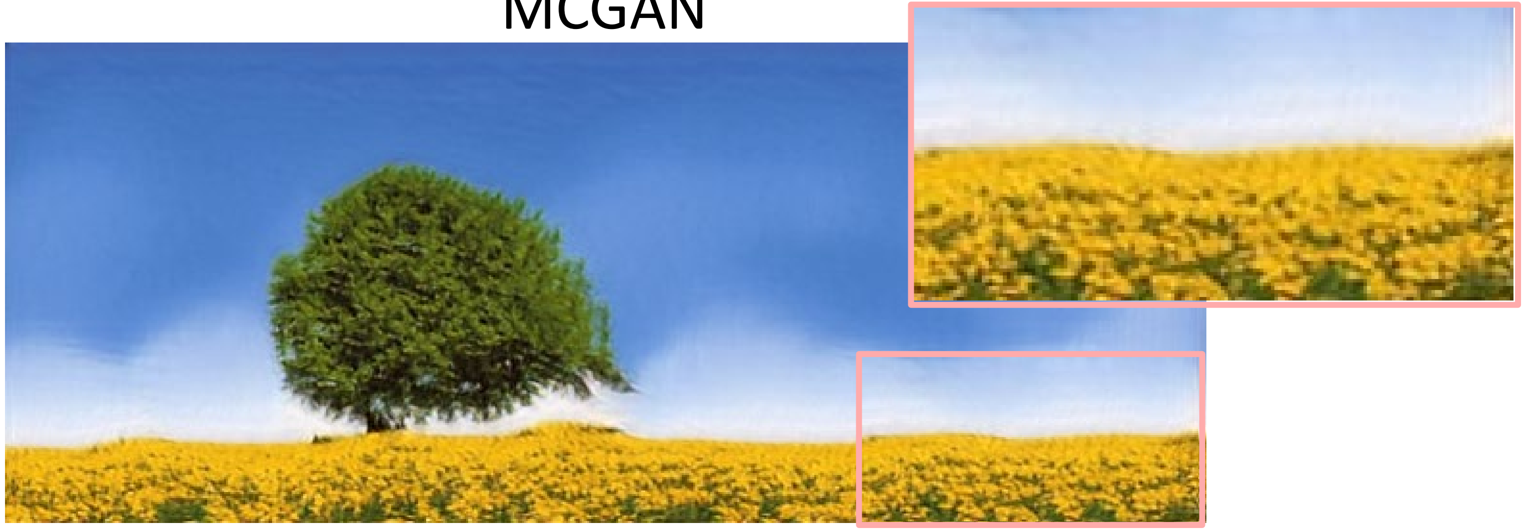
Comparison: Seam Carving

Input

Seam Carving



MCGAN



Comparison: GAN-Based Approach

InGAN (ICCV 2019)

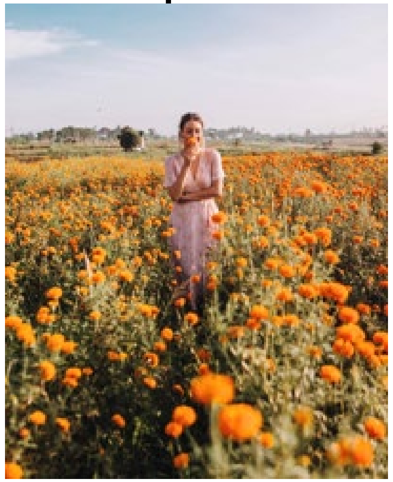
Input

SinGAN (ICCV2019)

MCGAN

Comparison: GAN-Based Approach

Input



SinGAN (ICCV2019)

InGAN (ICCV 2019)

MCGAN

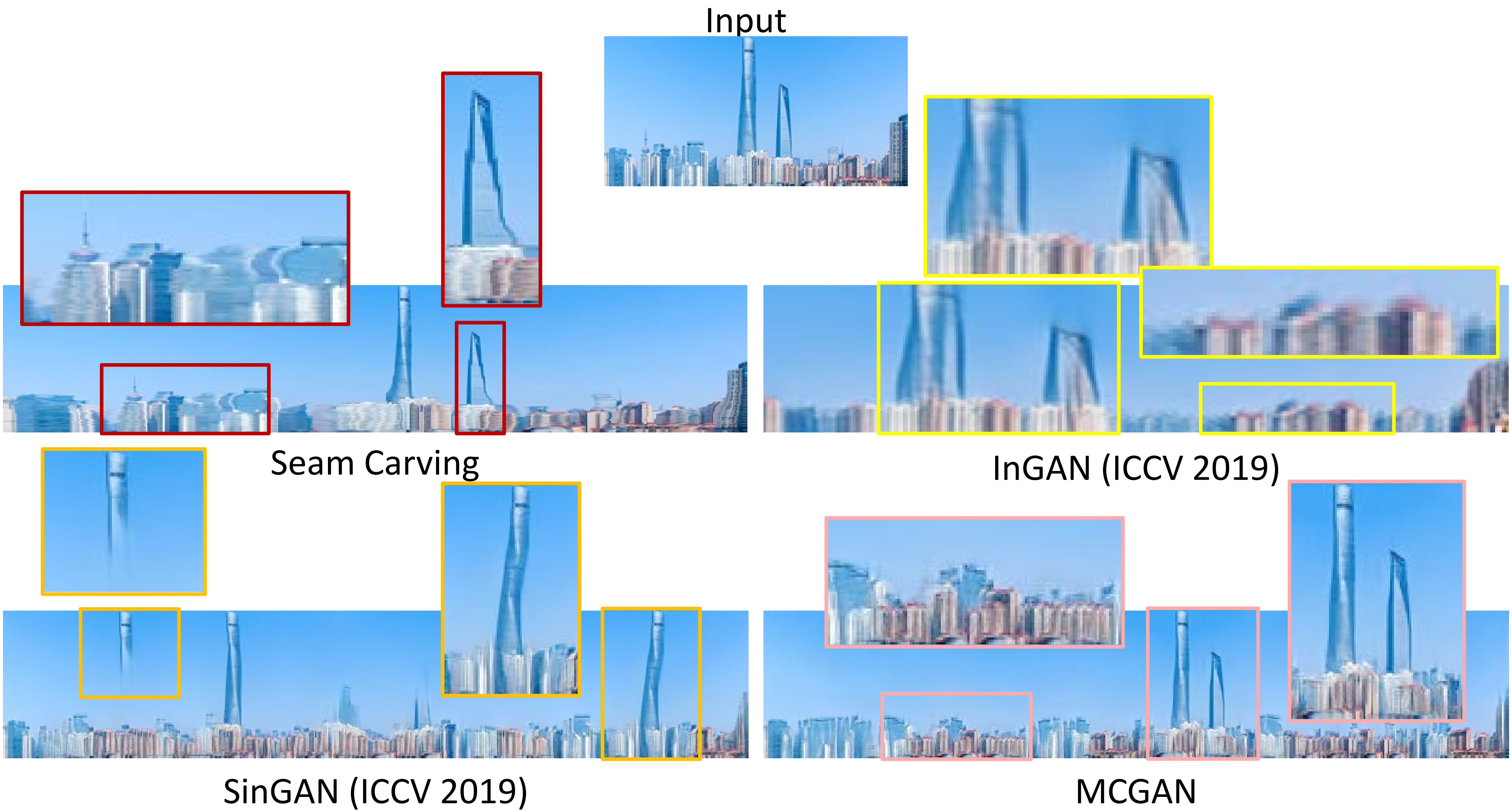
Comparison: GAN-Based Approach

InGAN (ICCV 2019)

Input

SinGAN (ICCV2019)

MCGAN


Input
Seam Carving
InGAN (ICCV 2019)
SinGAN (ICCV 2019)
MCGAN

There's still a lot of room for improvement!

Please contact me (daniel.tan@ou.nl) if you are interested or know someone who is interested in working on this!

Very suitable topic for Master's students!

- Not computationally expensive
  - Deep learning that only uses 1 image.
- Fast to train ⇒ Fast to iterate ideas
- Less competition (not overly hyped)

Thank you! Any questions?

Please contact me (daniel.tan@ou.nl) if you are interested or know someone who is interested in working on this!

Future OUrsi talk?

ConCoNet: Class-Agnostic Counting with Positive and Negative Exemplars

Can count anything!
Only specify what to count at test time