DEEP NETWORK DEVELOPMENT

Imre Molnár
PhD student, ELTE, AI Department
imremolnar@inf.elte.hu
curiouspercibal.github.io

Tamás Takács
PhD student, ELTE, AI Department
tamastheactual@inf.elte.hu
tamastheactual.github.io

Deep Network Development

Lecture 12.

Deep Learning Tools For Computer Vision part 2

Budapest, 09th May 2025

1 Generative Modeling

2 Image Inpainting

3 Neural Rendering

Deep Network Development

Lecture 12.

Deep Learning Tools For Computer Vision part 2

Budapest, 09th May 2025

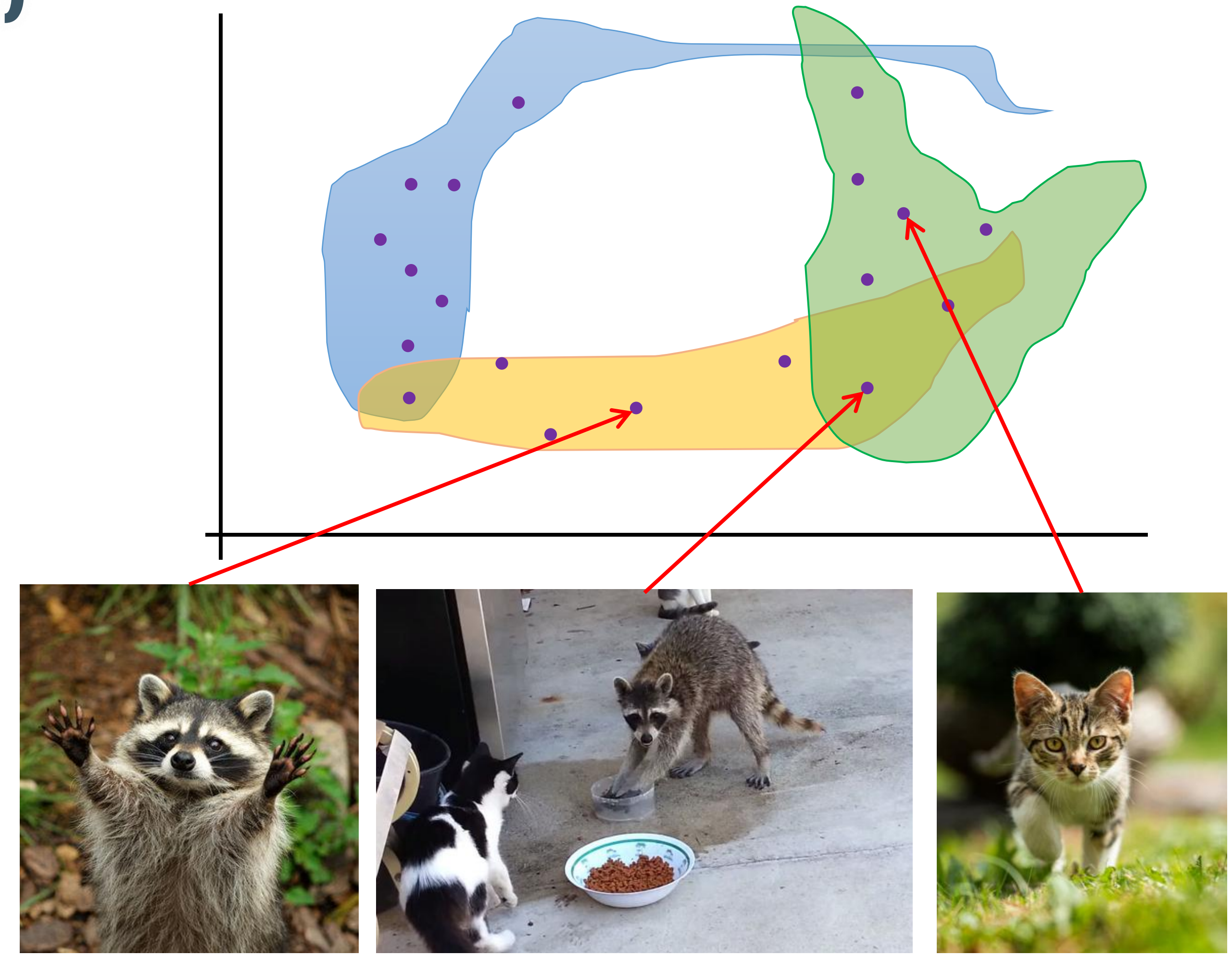
1 Generative Modeling

2 Image Inpainting

3 Neural Rendering

Deep Generative Modelling (DGM)

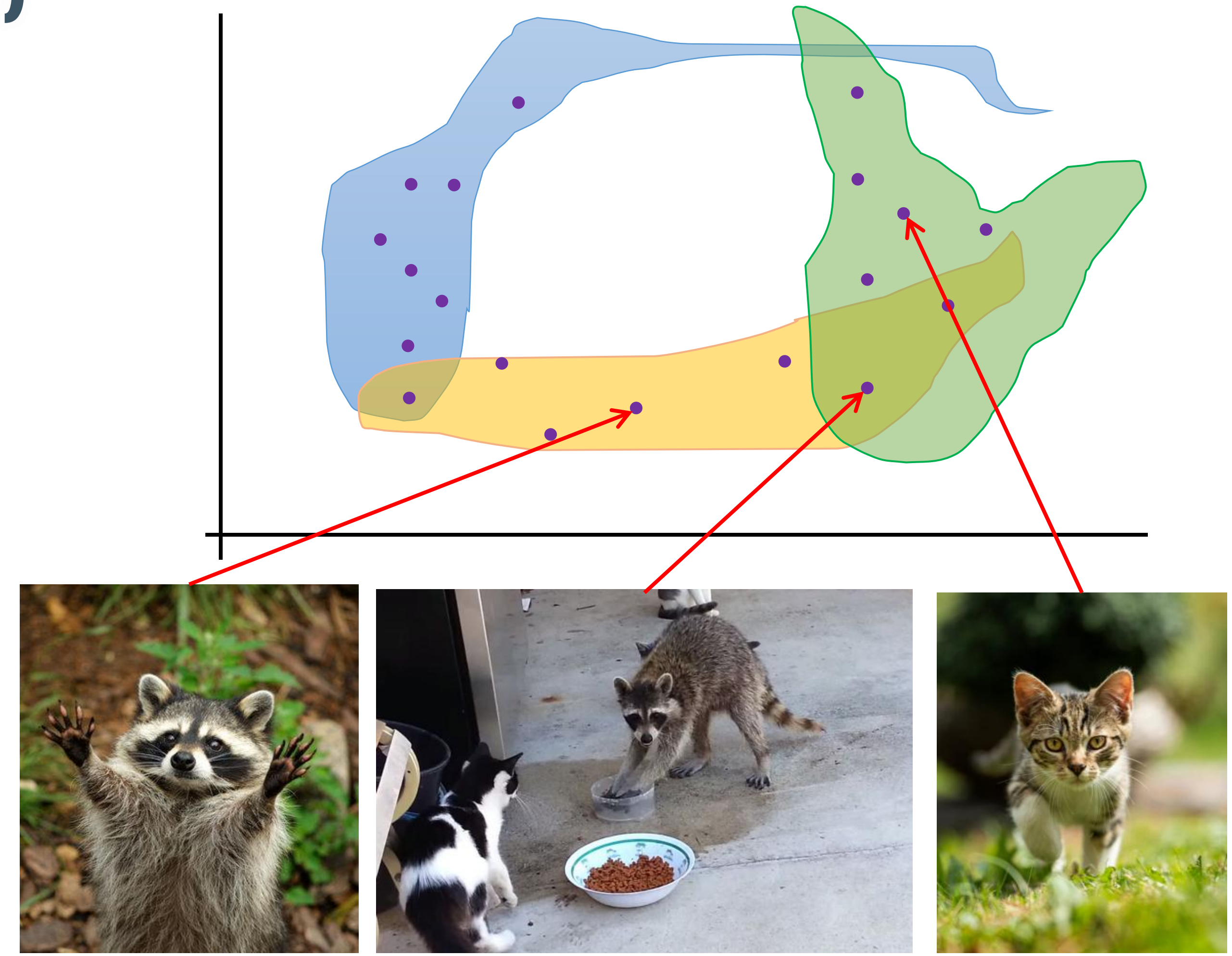
- The ambitious goal in DGM training is to learn an unknown probability distribution from a typically small number of independent and identically distributed samples.
- When trained successfully,
  - we can use the DGM to estimate the likelihood of a given sample
  - create new samples that are similar to the samples from the unknown distribution (generator).



Lars Ruthotto and Eldad Haber - An Introduction to Deep Generative Modeling (2021) - https://arxiv.org/abs/2103.05180

Deep Generative Modelling (DGM)

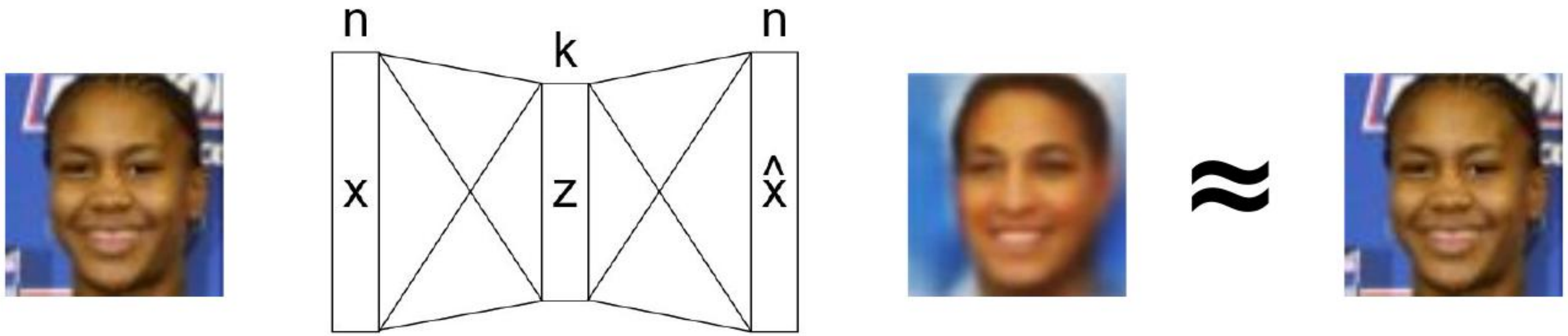
1. Uniquely identifying a probability distribution from a finite number of samples is impossible.
2. Training the generator requires a way to quantify its samples' similarity to those from the intractable distribution.
3. We assume, that we can approximate the intractable distribution by transforming a known and much simpler distribution (Gaussian) in a latent space of known dimension



Lars Ruthotto and Eldad Haber - An Introduction to Deep Generative Modeling (2021) - https://arxiv.org/abs/2103.05180

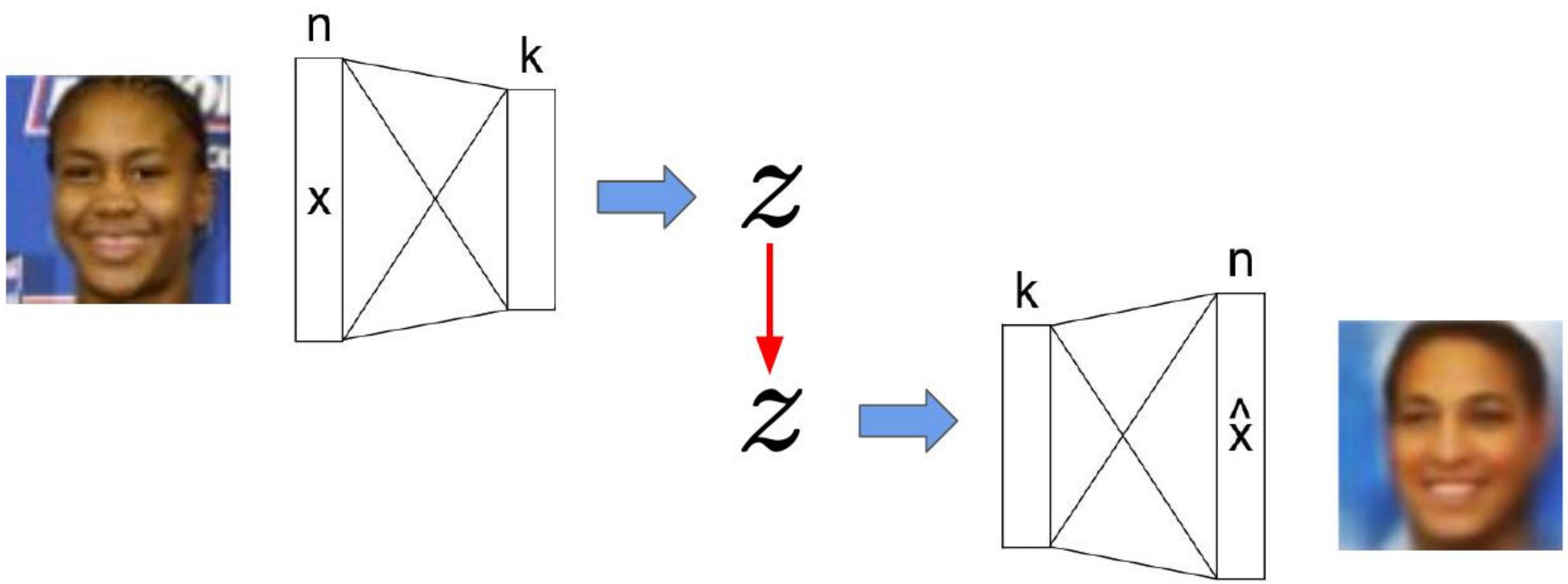
Autoencoders

First, take a pre-trained autoencoder:



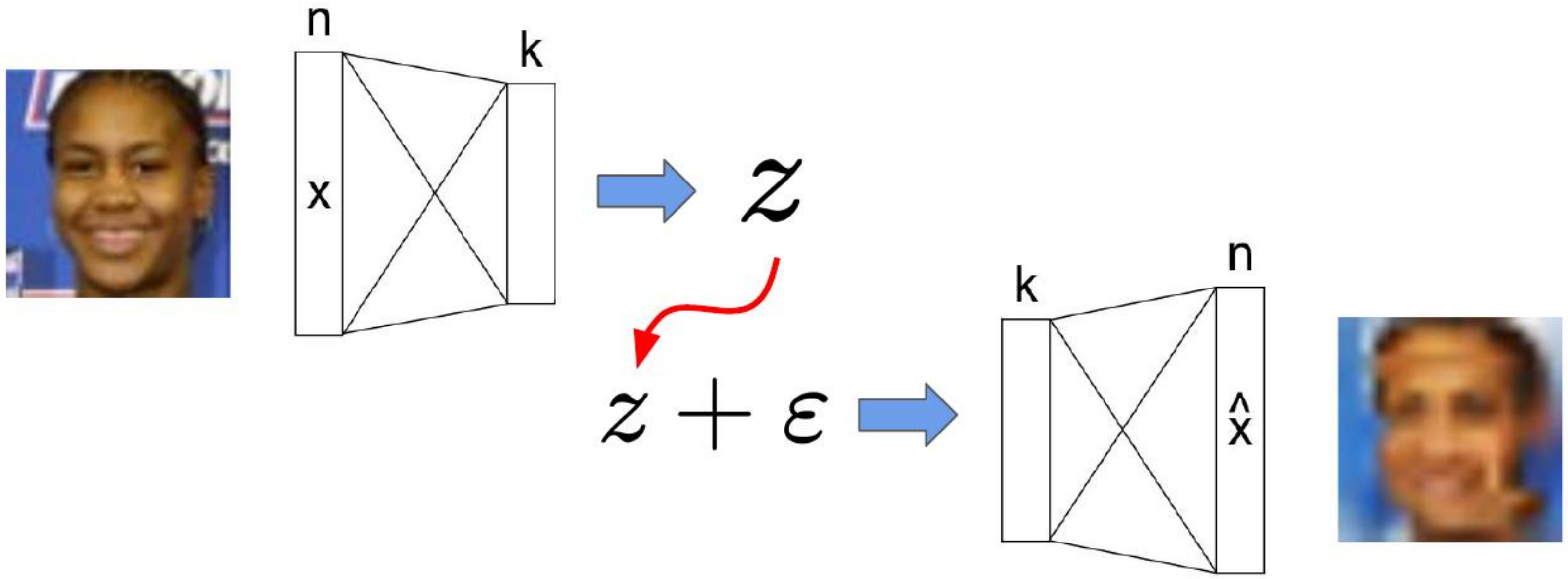
Autoencoders

Split into encoder and decoder



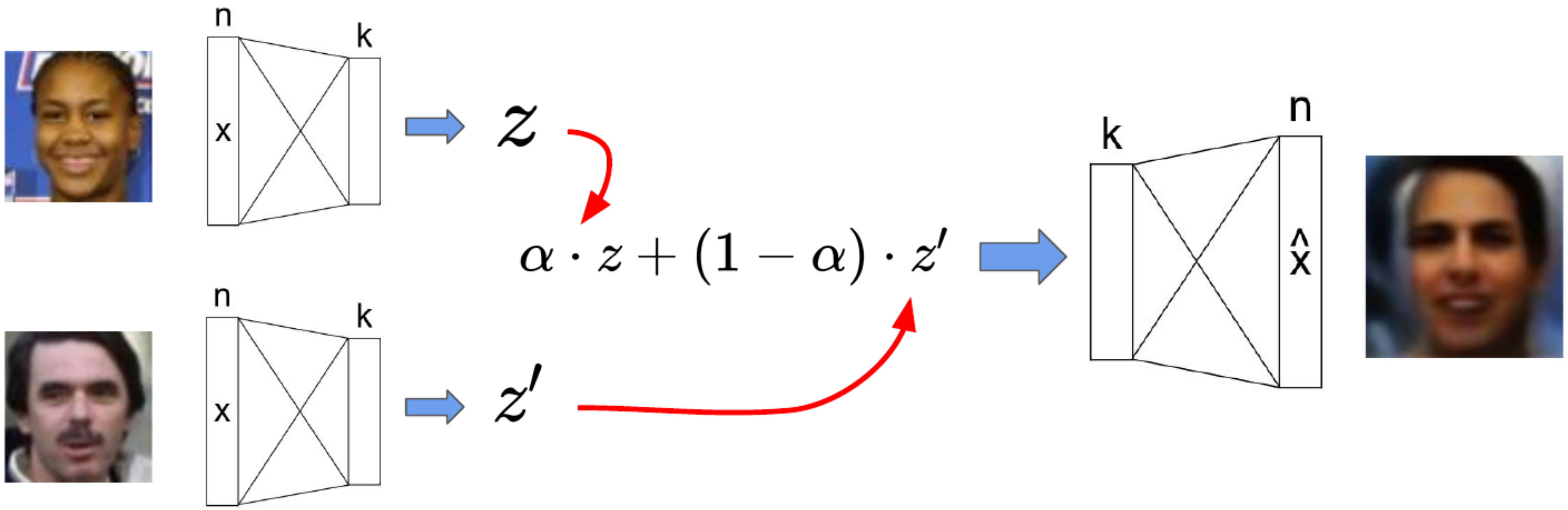
Autoencoders

Then modify the latent space representation and feed it into the decoder



Autoencoders

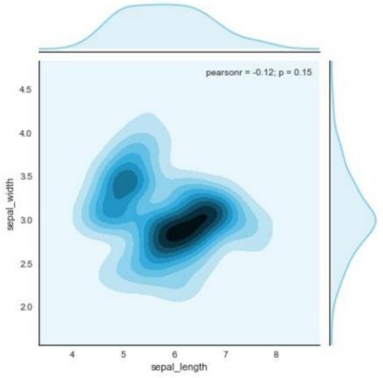
Similarly, try to blend two faces

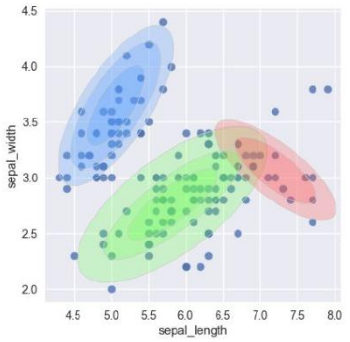


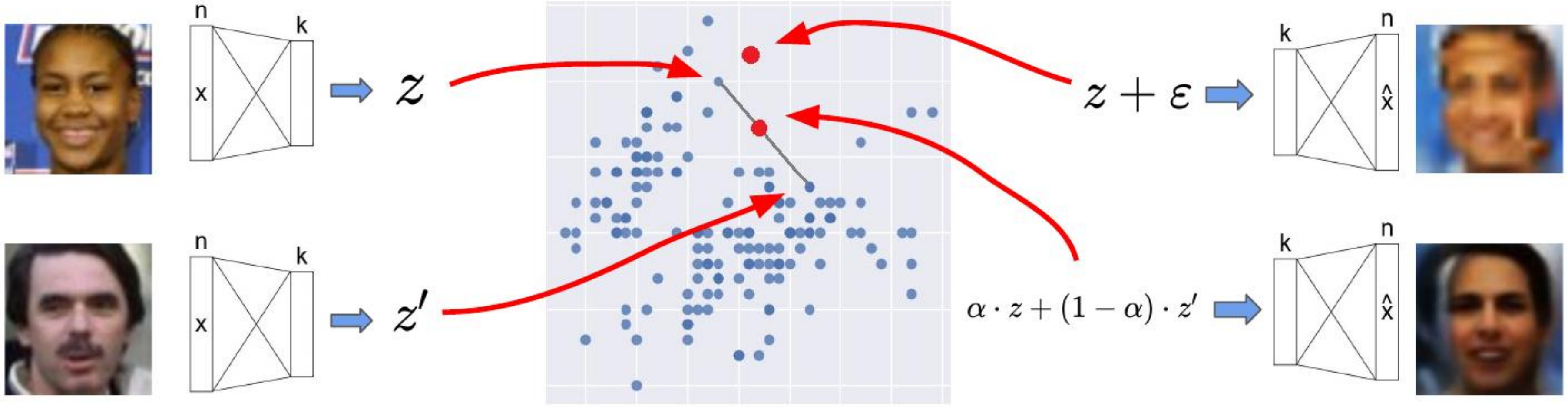
$$\alpha \cdot z + (1 - \alpha) \cdot z'$$

Autoencoders

- The autoencoder didn't see the modified represent so it struggles with reconstruction
- The blended representation might not be valid

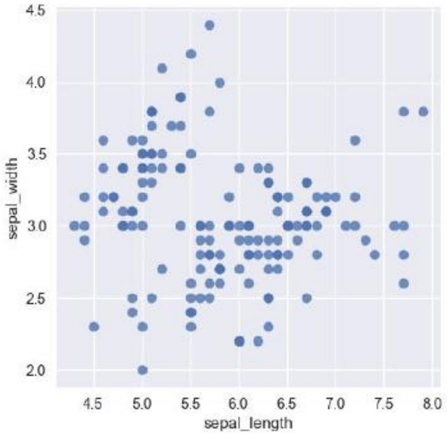


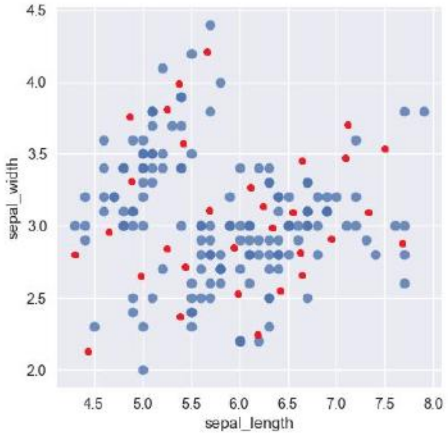




Autoencoders

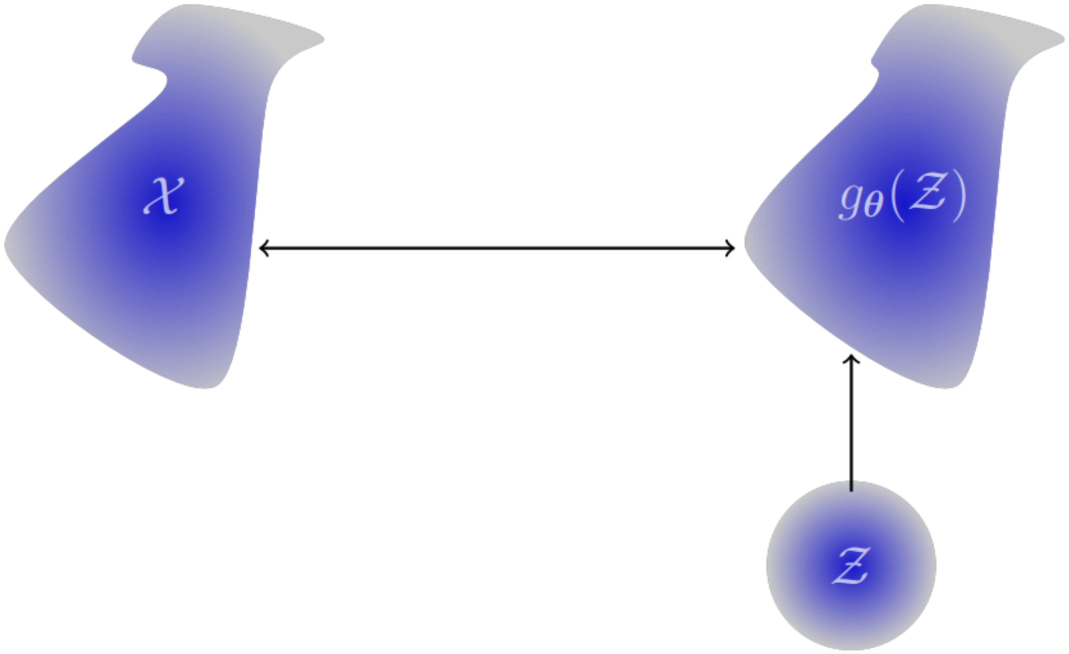
- For smaller dimensions we can use noising, but in higher dimensions we will likely generate invalid instances
- The distribution of the latent variable **z** that is producing the reconstructed data sample is generally intractable.





Variational Autoencoders

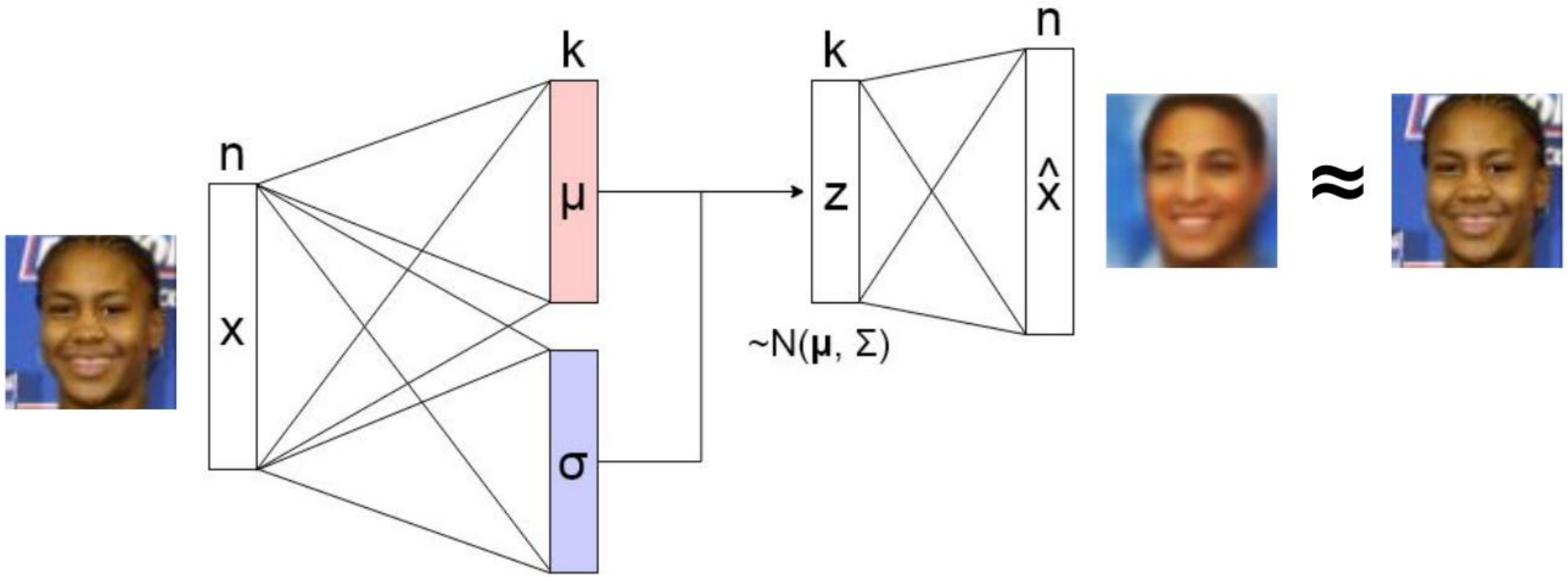
- We approximate the distribution of the latent variable to be in a parameterized probability distribution (Normal distribution)
- The normal distribution is tractable – we can sample from the distribution and compute probabilities efficiently
- This network takes **x** as an input and generates the parameter of the parameters of the distribution – in case of Normal distribution the **mean** and **covariance**



Variational Autoencoders

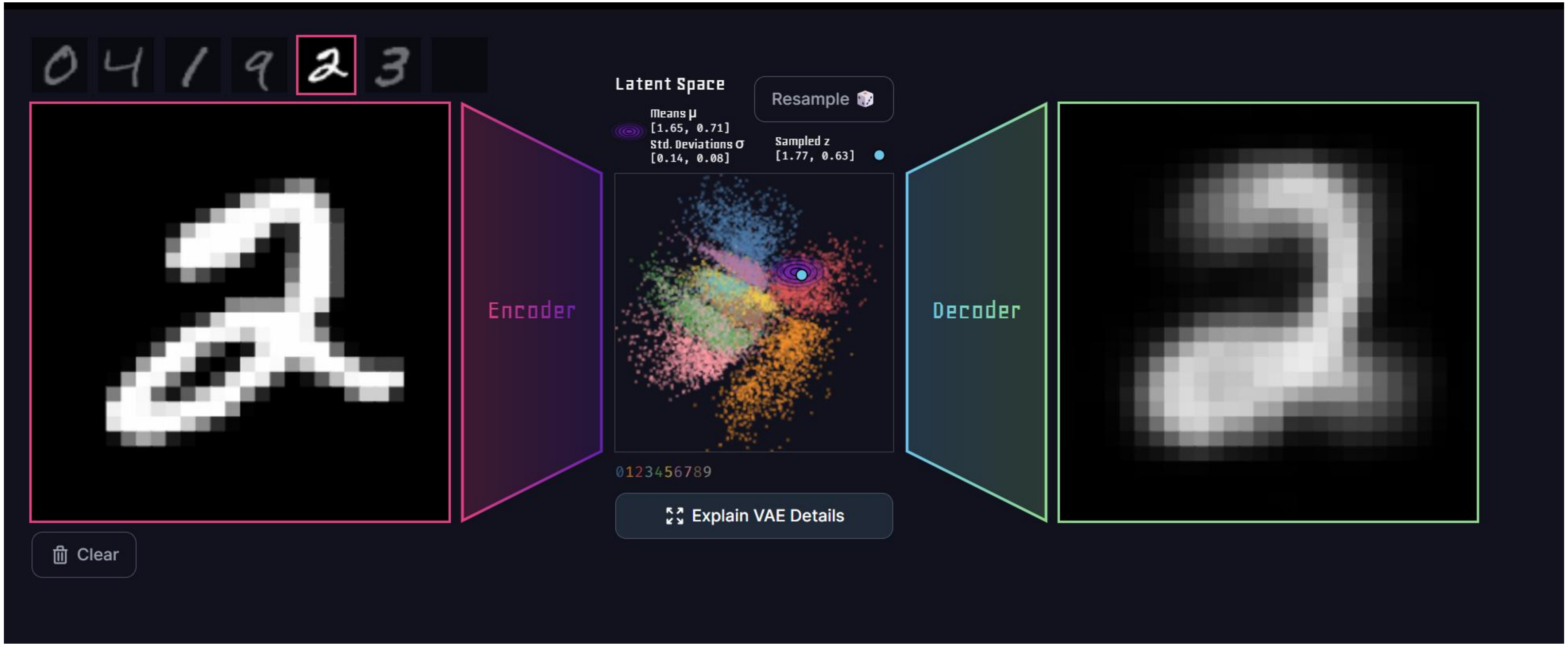
Let's constrain the possible representations into a know distribution (Normal distribution)

- If we would know the distribution, the generation wouldn't be an issue
- The results are blurry because of the minimization of the MSE loss



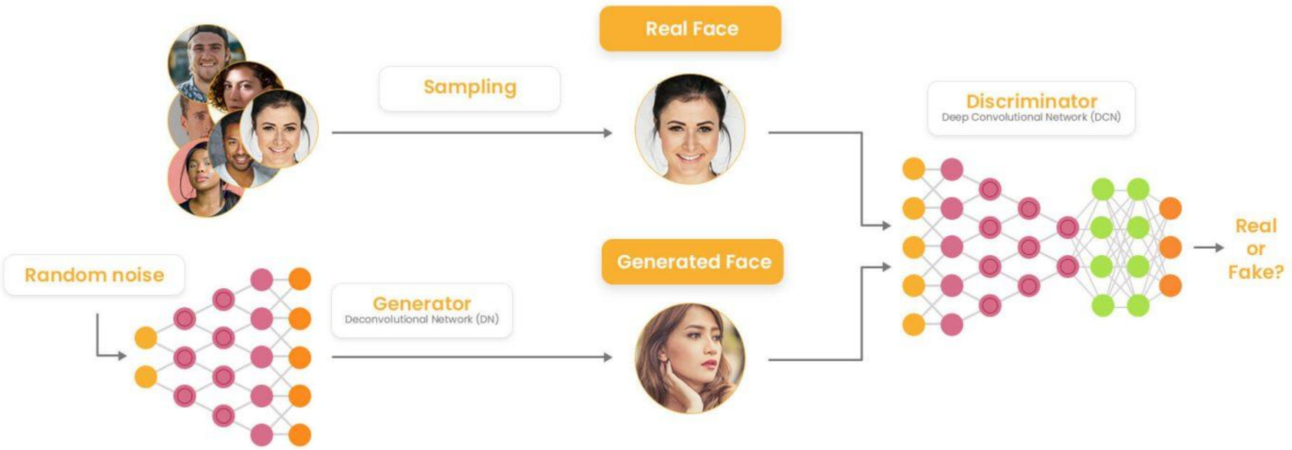
Variational Autoencoders

https://xnought.github.io/vae-explainer/



Generative Adversarial Networks

- We train the network by minimizing a loss function that measures the distance between the generated and the sampled image.
- Compare the distribution in data space
- It does not try to infer the latent variable

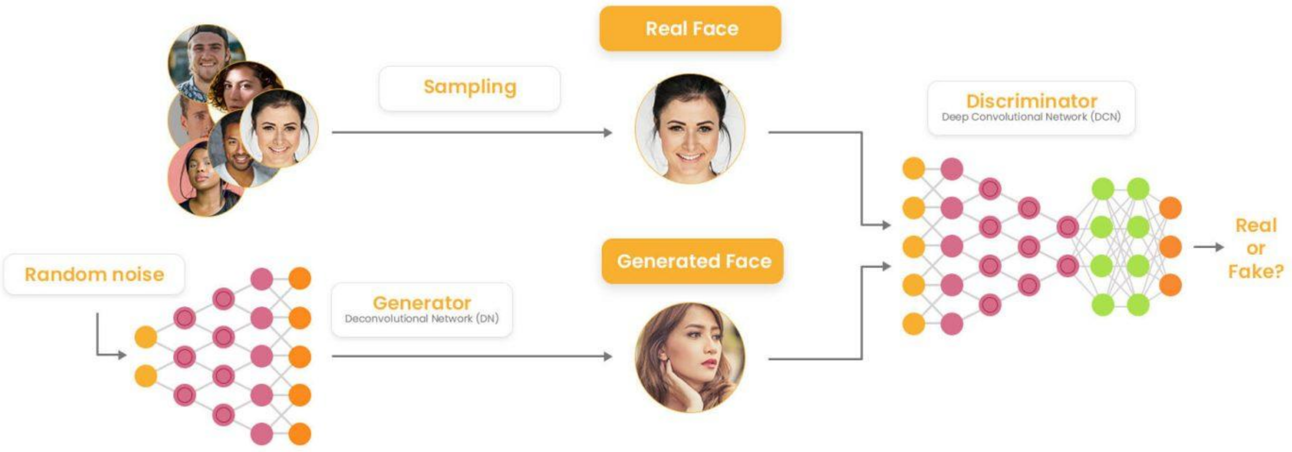


Generative Adversarial Networks

A Generator and a Discriminator are competing

- **Generator:** Generates images.
- **Discriminator**: Tries to predict if the input image is real or fake (binary classification).

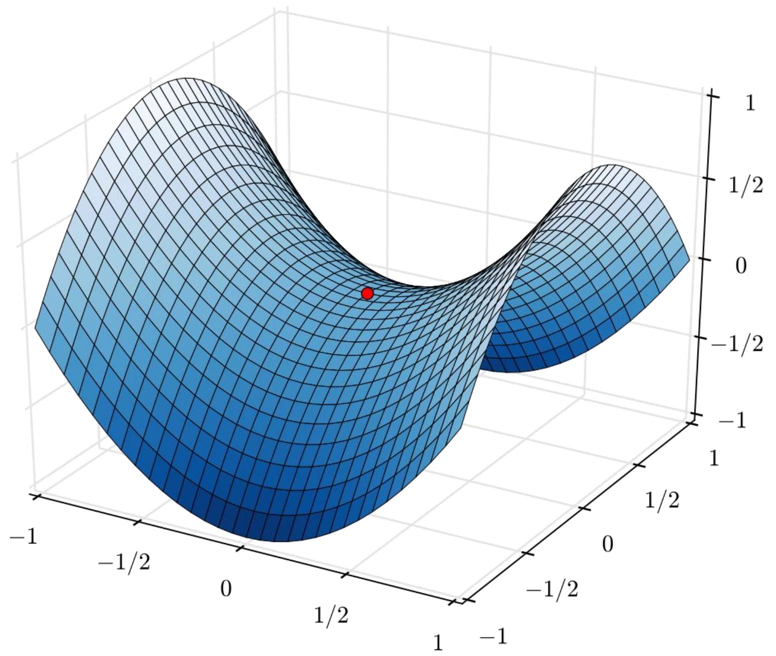
- If the generator generates an image that the discriminator considers a valid image, we punish the discriminator
- If the discriminator predicts that the generated image is fake we punish the generator



Generative Adversarial Networks

- Due to the binary classification the GAN is trained with the cross-entropy loss

$$J_{GAN}(\theta,\phi) = \mathbb{E}_{x\sim X}\left[\log\left(d_\phi(x)\right)\right] + \mathbb{E}_{z\sim Z}\left[\log\left(1 - d_\phi(g_\theta(z))\right)\right]$$



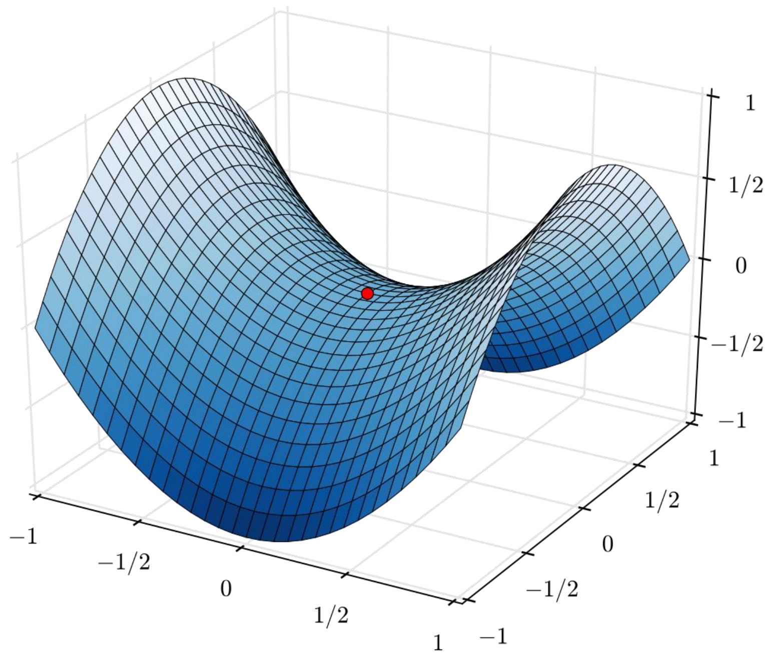
- **d** – the discriminator with parameters ϕ
- **g** – the generator with parameters θ

- The discriminator is trying to maximize the loss (it can effectively detect fakes from real images)
- The generator is trying to minimize the loss (it can generate deceiving images)
- Thus the generator and the discriminator play a zero-sum game

Generative Adversarial Networks

- Finding this saddle point is hard
- The loss does not tell exactly whether the generator or the discriminator is good

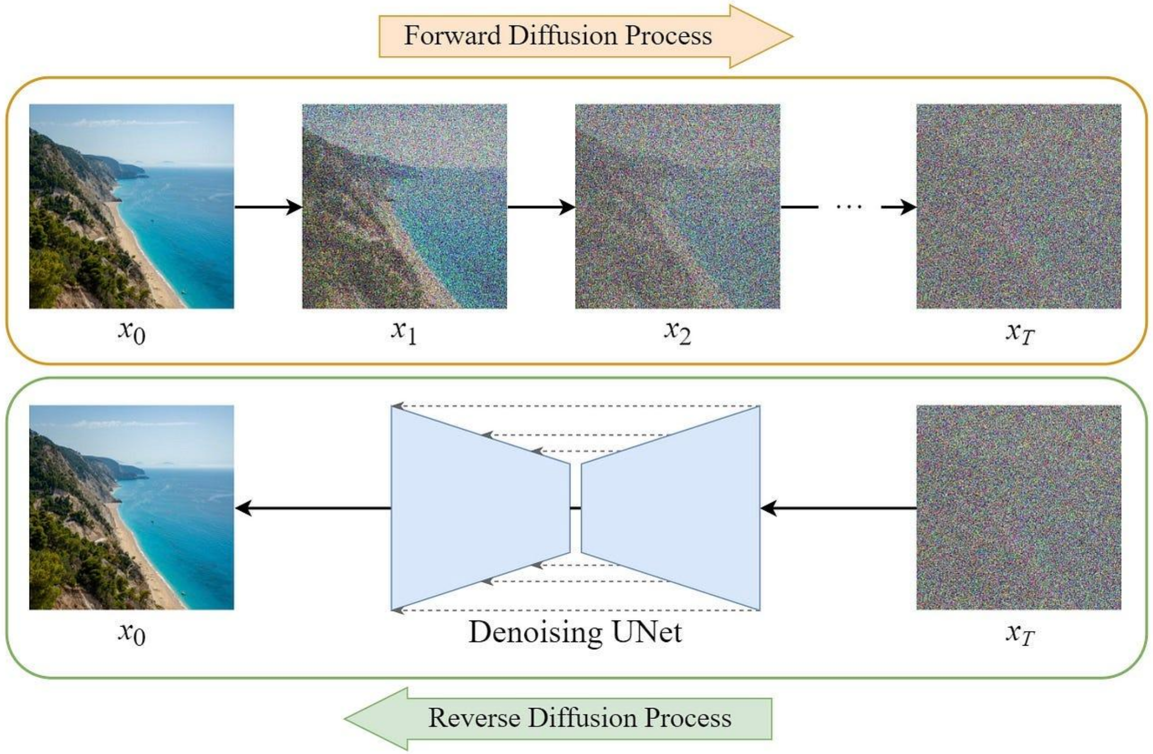
- Let's assume that we have an optimal generator – the discriminator can only random guess if the image is generated or sampled from the dataset
- If we have a slightly off generator, then the discriminator can greatly increase the objective

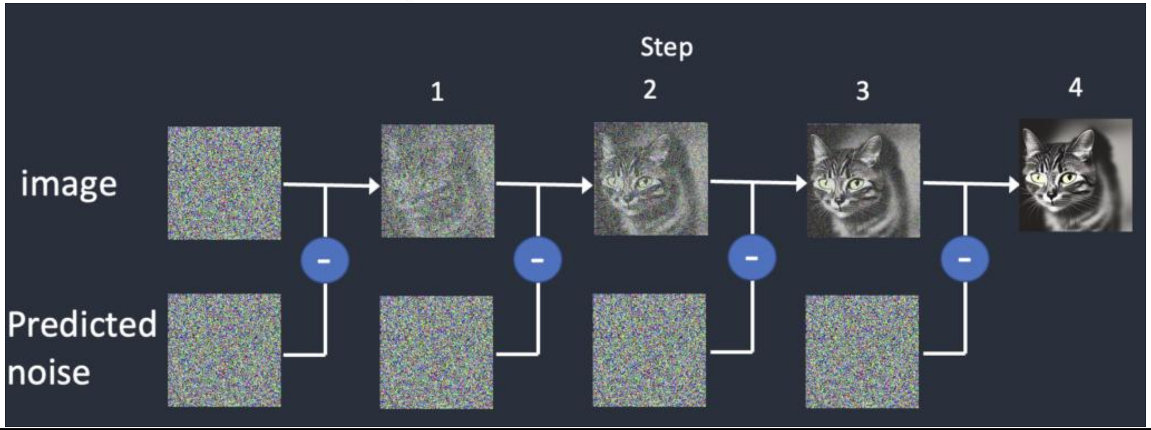


Stable Diffusion

The main idea is to train a denoising network.

- In the forward diffusion process we gradually add Gaussian noise to the image in each step
- The model estimates the added noise and subtracts it from the image (recreating the original image)
- In the reverse diffusion process we start from gaussian noise and the network gradually removes the noise (in multiple passes), thus generating an image





Deep Network Development

Lecture 12.

Deep Learning Tools For Computer Vision part 2

Budapest, 09th May 2025

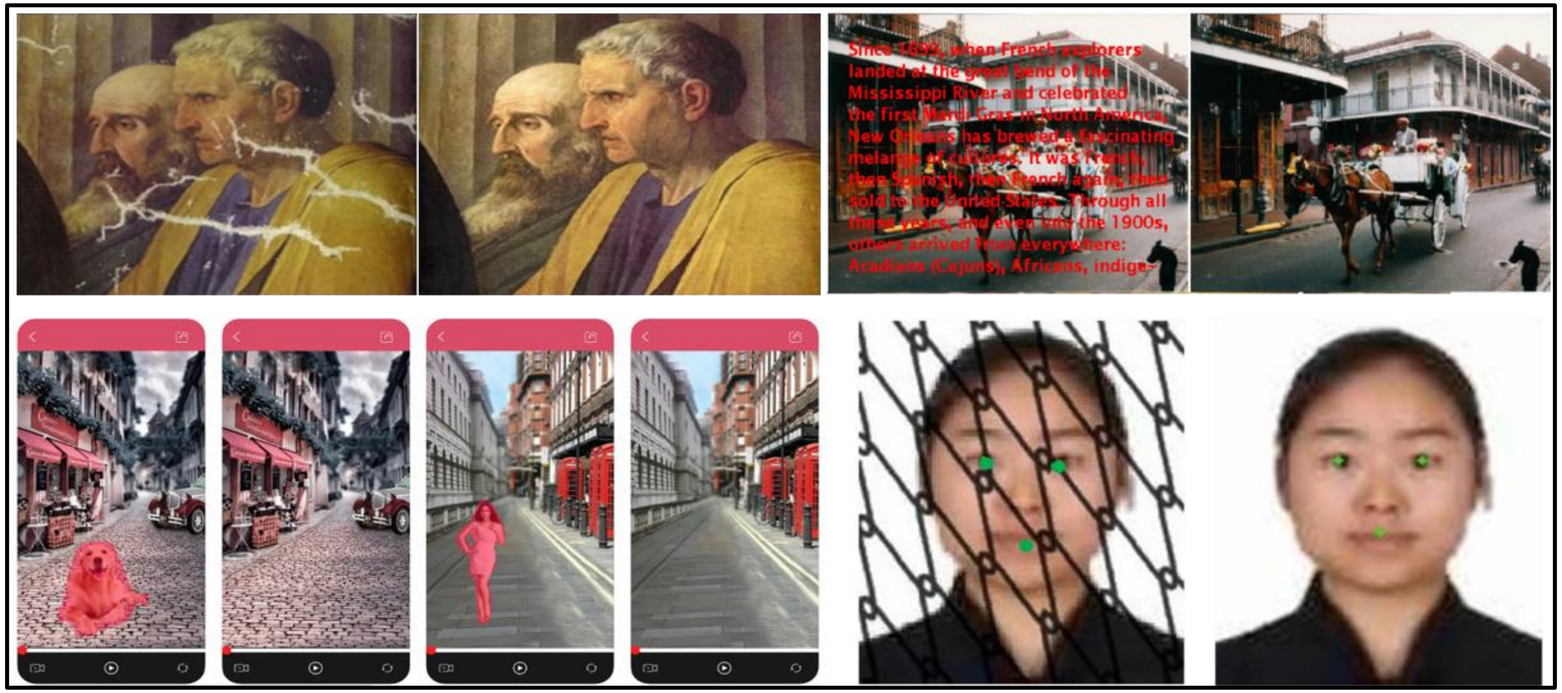
1 Generative Modeling

2 Image Inpainting

3 Neural Rendering

Image Inpainting

Image inpainting is about filling in missing or occluded regions in digital images, aiming to restore plausible, realistic content. Its applications range from cultural relic restoration to virtual scene editing and film production.



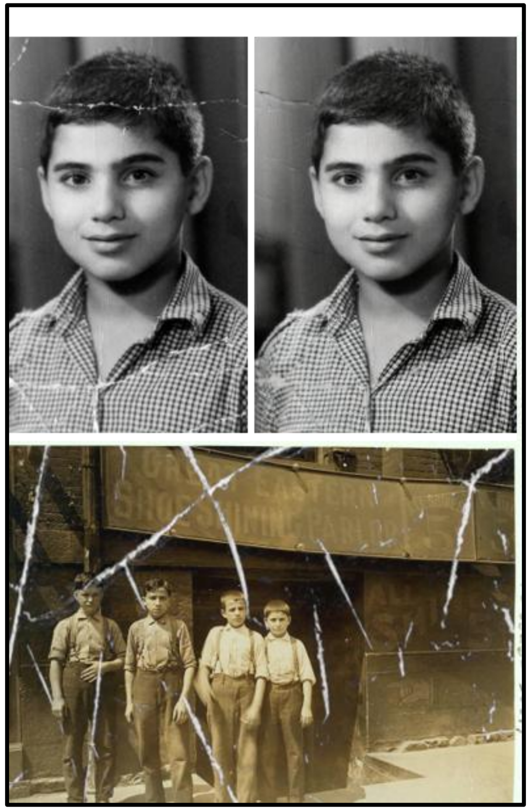


Image Inpainting

Image inpainting is about filling in missing or occluded regions in digital images, aiming to restore plausible, realistic content. Its applications range from cultural relic restoration to virtual scene editing and film production.

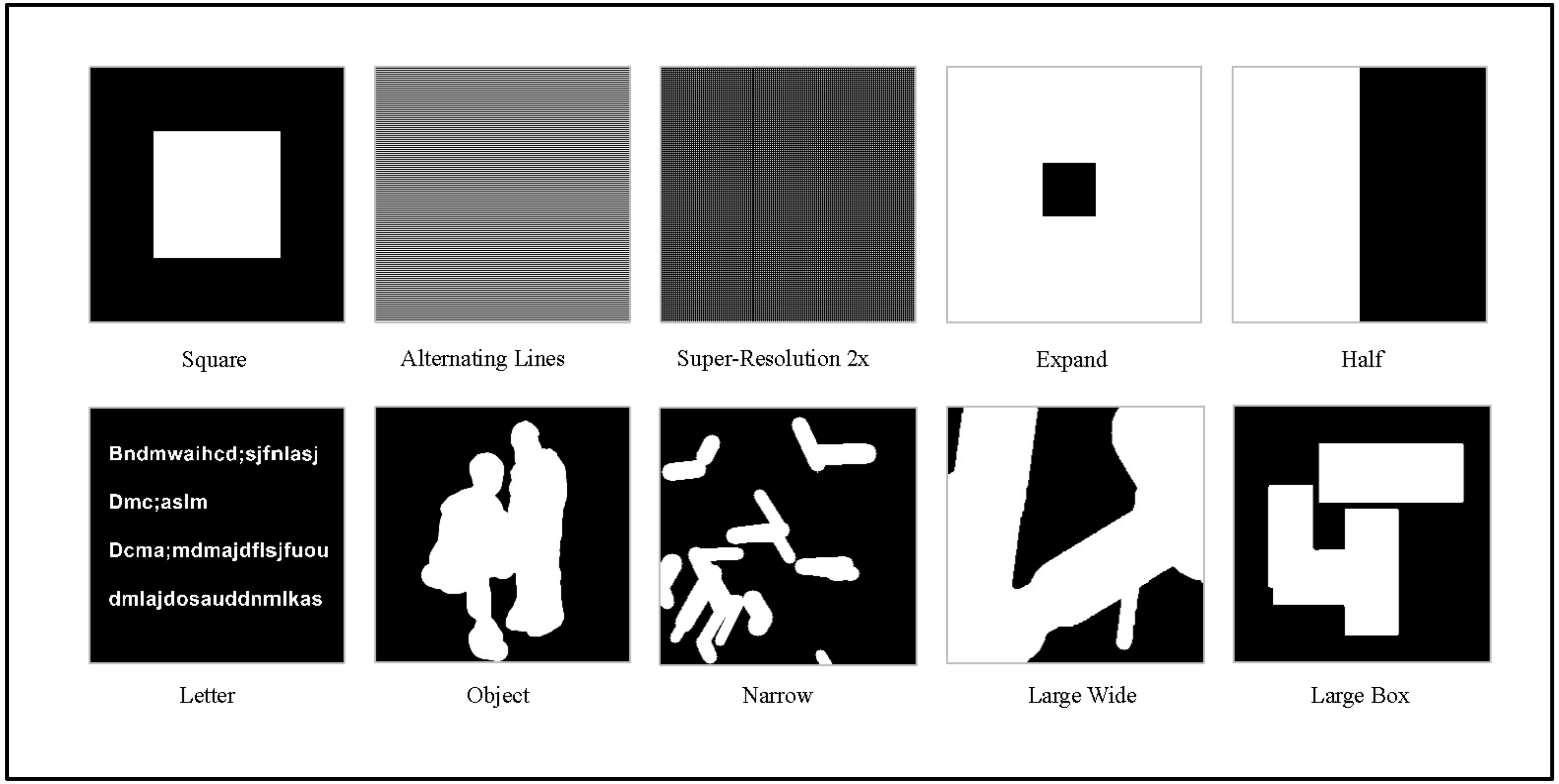
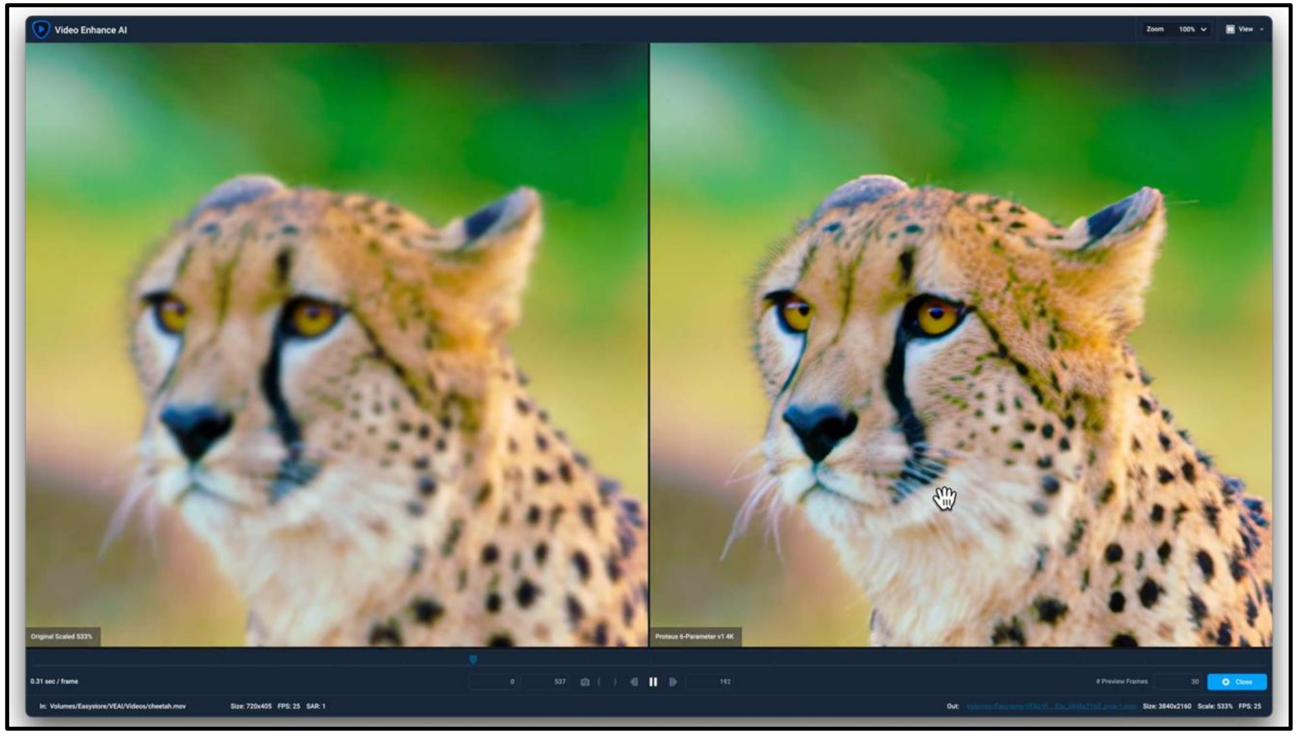


Image Inpainting

Adobe Photoshop image expanding

Image Inpainting

Topaz Video enhance AI (super resolution)



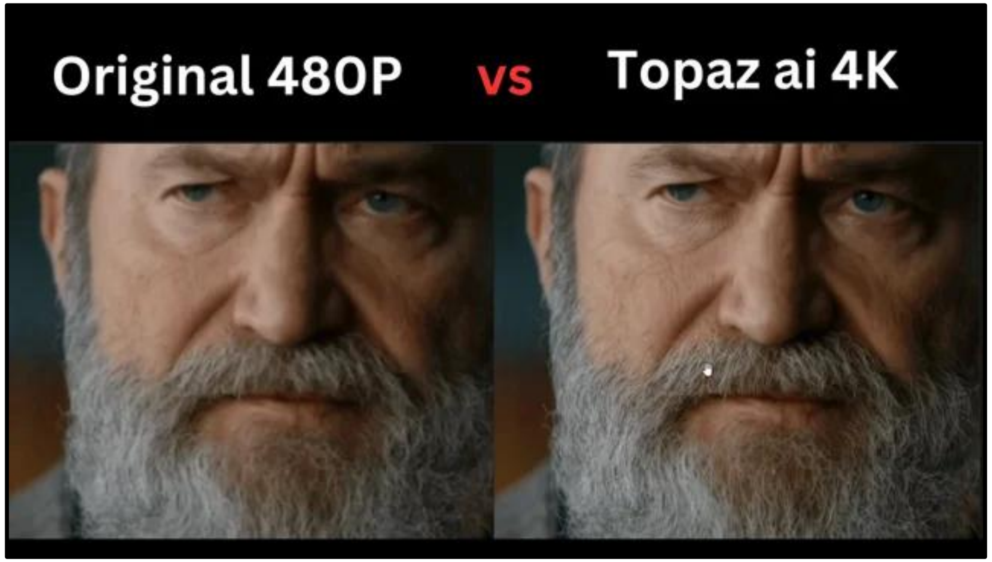


Image Inpainting

Object removal and Text Image editing

Traditional Methods

Diffusion Based methods

Pros: Good at handling small gaps, extending smooth areas, and completing continuous structures like edges and curves.

Cons: Struggles with large texture areas, as it tends to blur the regions being filled, making it unsuitable for complex textures

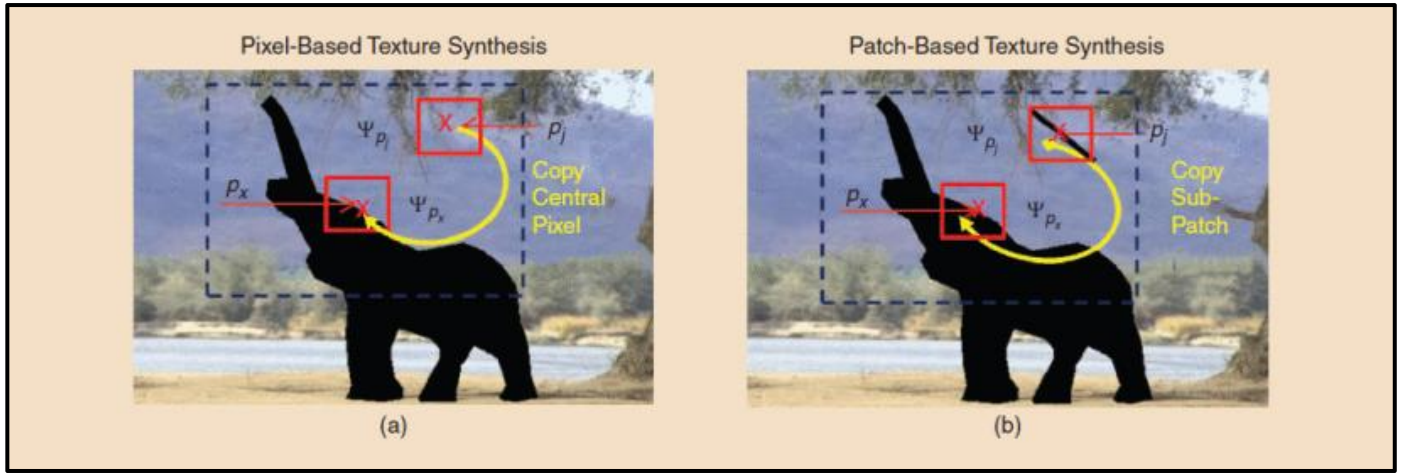
Christine Guillemot; Olivier Le Meur - Image Inpainting : Overview and Recent Advances (2014) - https://ieeexplore.ieee.org/document/6678248

Traditional Methods

Exemplar-Based Inpainting

Pros: Capable of filling large textured areas by copying similar patches from known regions. It leverages self-similarity to reproduce textures accurately.

Cons: Computationally intensive and may produce repetitive patterns or "texture garbage" when handling stochastic textures.



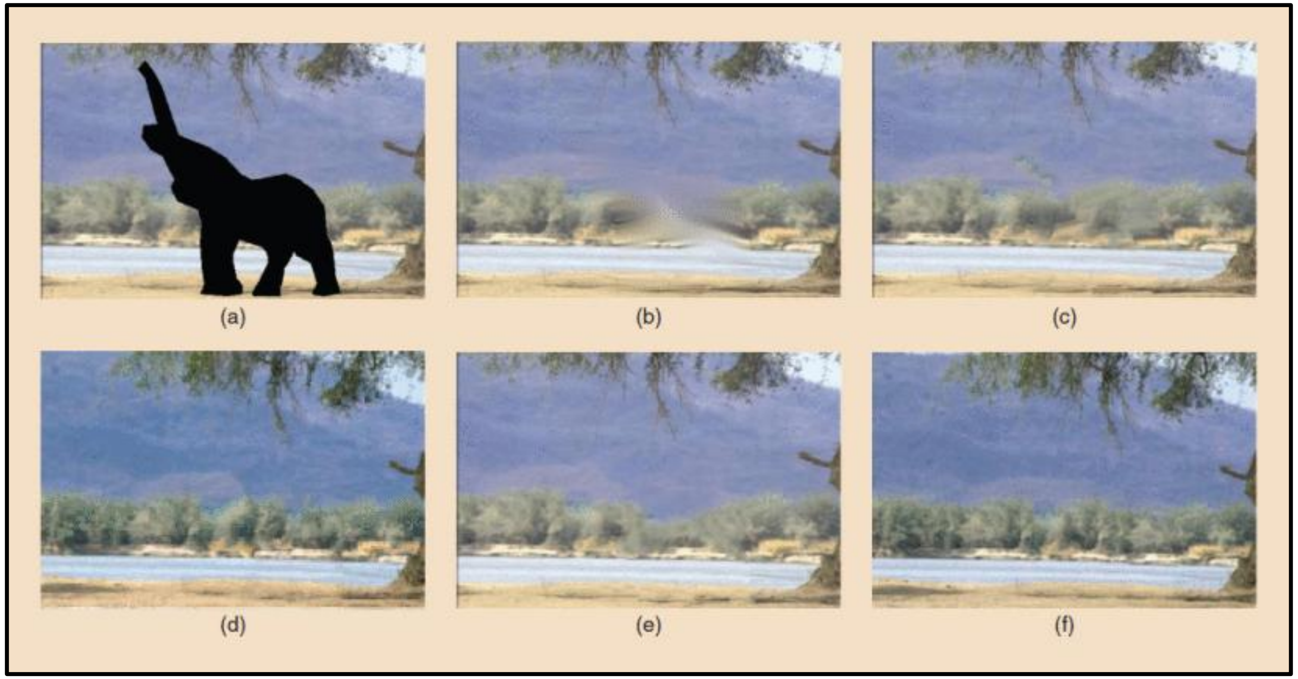
Christine Guillemot; Olivier Le Meur - Image Inpainting : Overview and Recent Advances (2014) - https://ieeexplore.ieee.org/document/6678248

Traditional Methods

Hybrid methods

Pros: Combine diffusion for structures and exemplar-based methods for textures, resulting in more visually consistent inpainted images. They offer better handling of complex textures and structures.

Cons: More complex and computationally intensive due to the combination of techniques



(a) original, (b) diffusion based (c) exemplar based, (e) hybrid method

Christine Guillemot; Olivier Le Meur - Image Inpainting : Overview and Recent Advances (2014) - https://ieeexplore.ieee.org/document/6678248

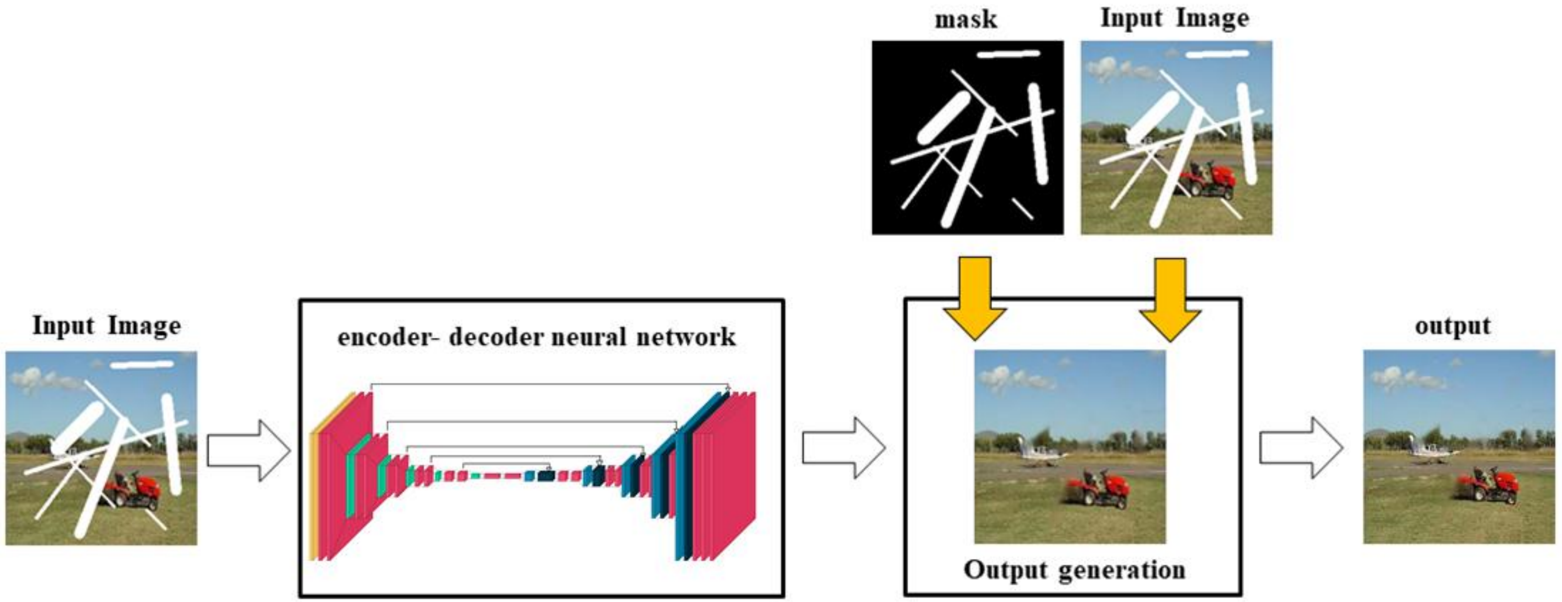
Deep Learning Based Approaches

Autoencoders:

Pros:
- Efficient feature extraction
- Simplicity and Stability
- Denoising Capabilition

Cons:
- Blurred output
- Lack of diversity
- Limited realms for large holes

Similarly GANs can also be used:
- Detail oriented results
- Diversity
- High Realms
- But more tricky to train
- Higher computational demand
- Artifacts and inconsistencies

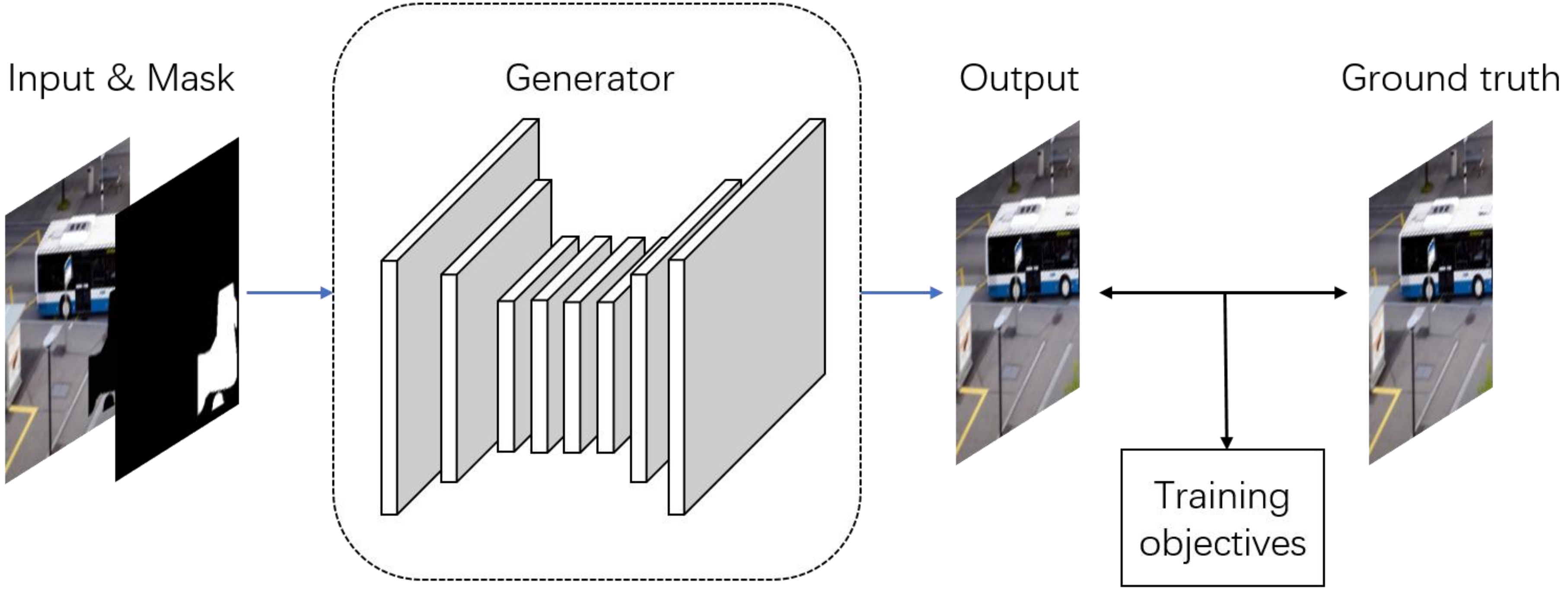


Single-Shot

Uses one generator to directly produce the inpainted image.

Input: corrupted image and mask (concatenation)

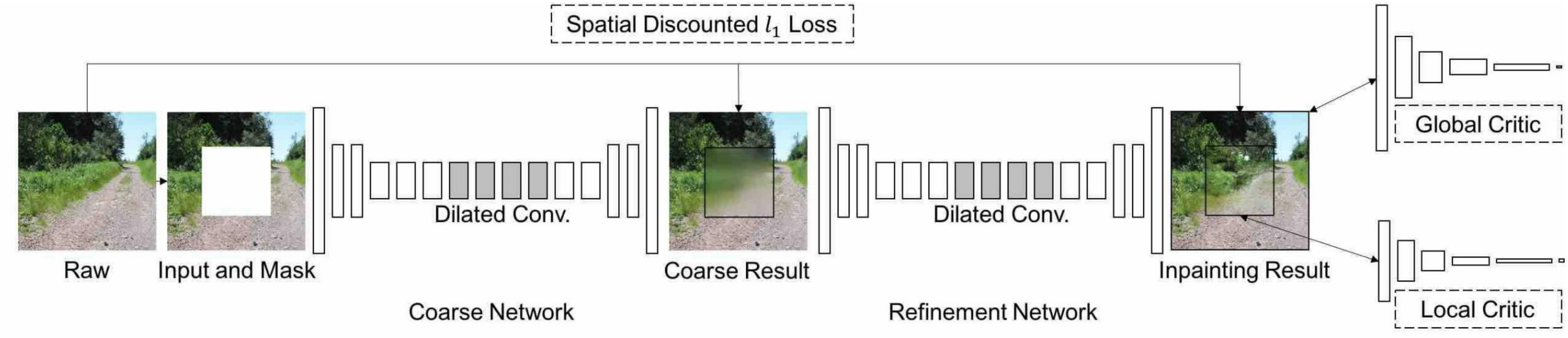
Output: the completed image



Two-stage

Utilizes two generators

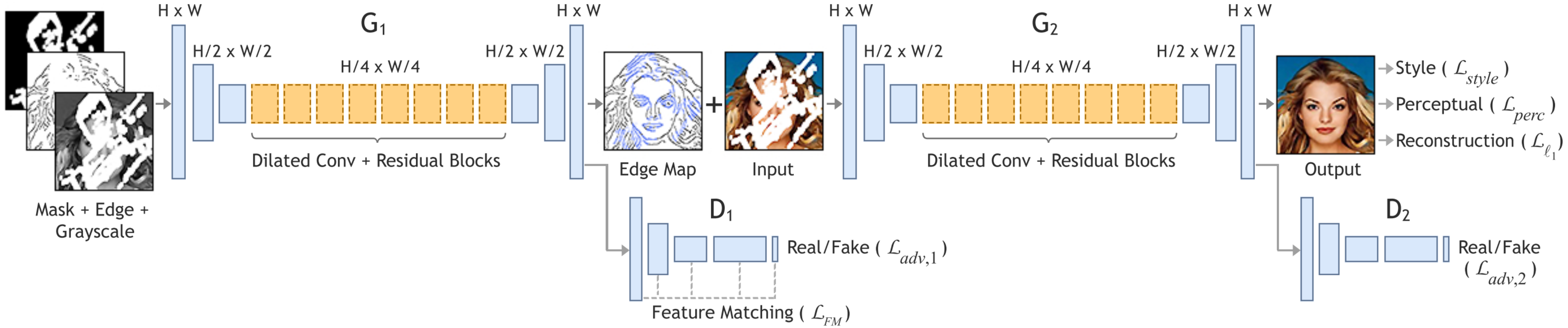
Coarse-to-fine:
- First creates a coarse filling on the missing area
- Second refines the filled area



Two-stage

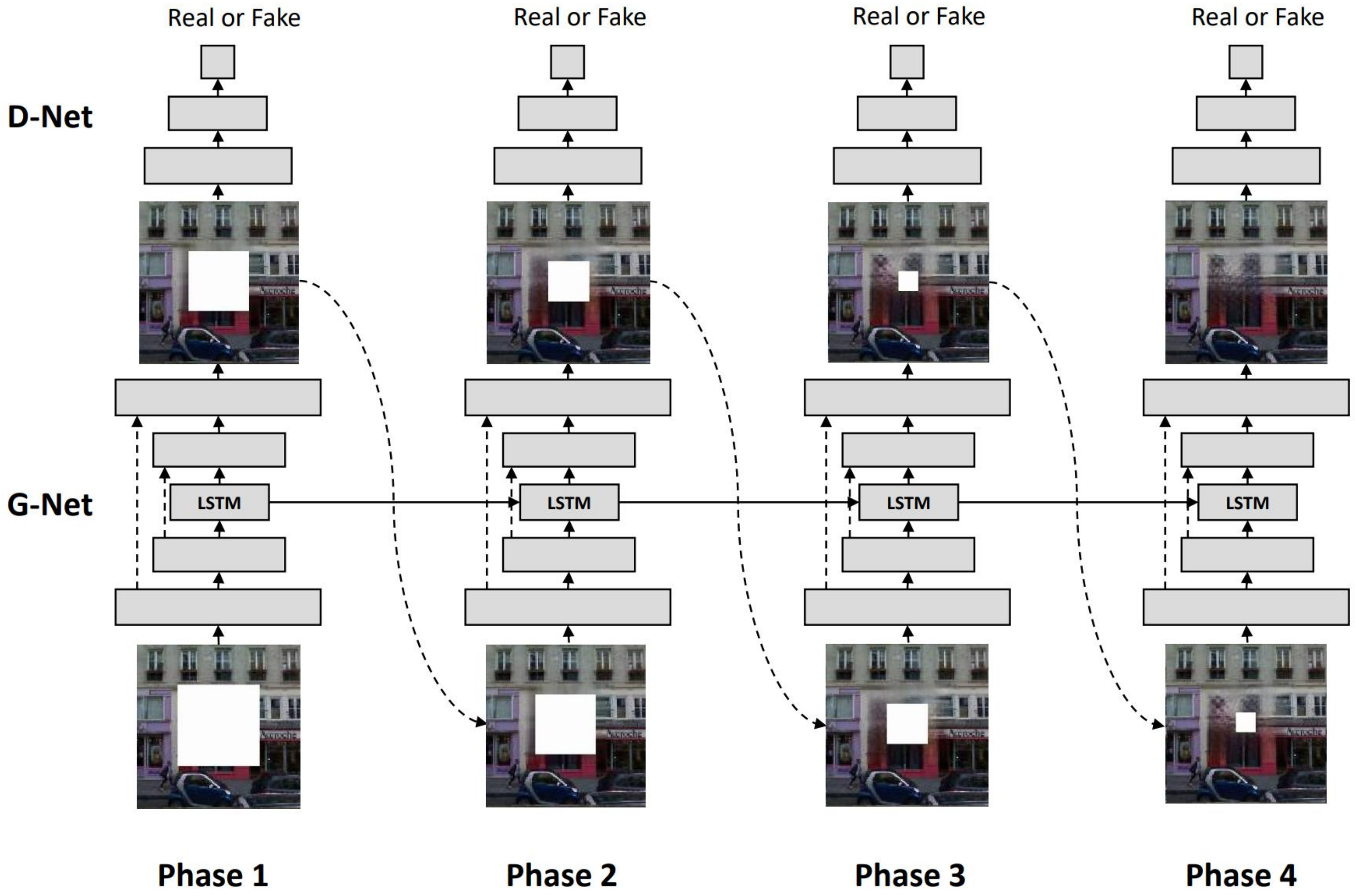
Utilizes two generators

Structure-then-texture:
- First creates a structure map of the image
- Second predicts the complete image



Progressive

Progressive: Iteratively fills the missing area from boundary to center, useful for handling larger missing areas



Lecture 12.

Deep Learning Tools For Computer Vision part 2

Budapest, 09th May 2025

1 Generative Modeling

2 Image Inpainting

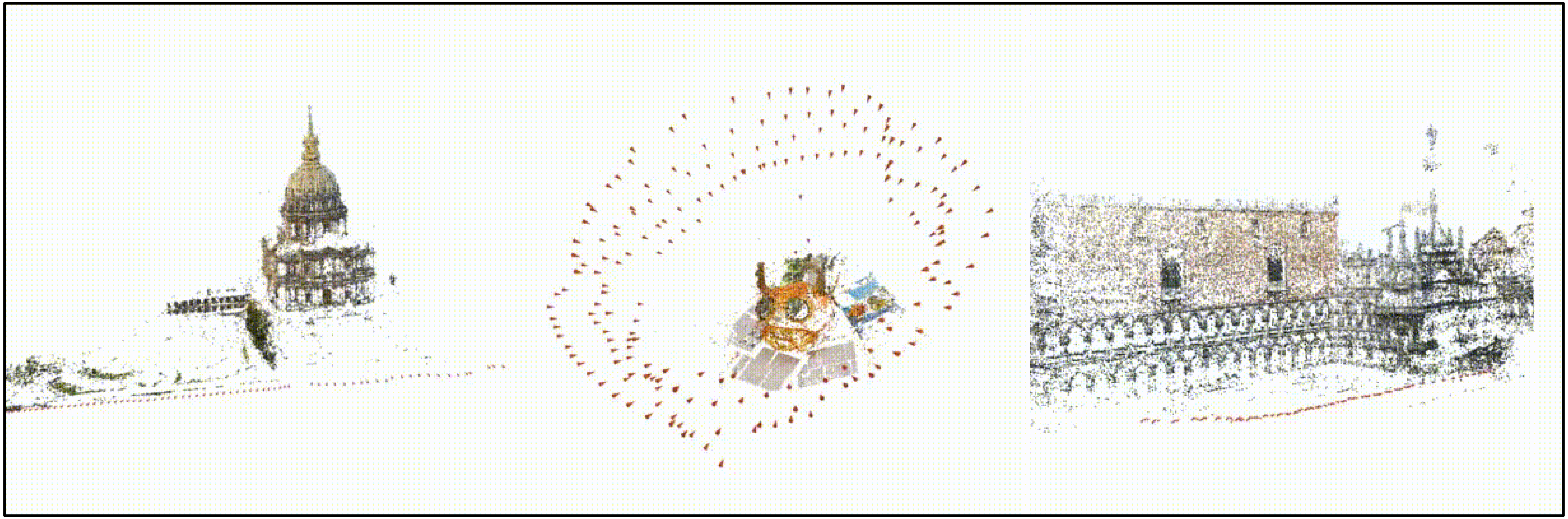
3 Neural Rendering

Neural Rendering

It combines the deep learning model with the physical knowledge of computer graphics, to obtain a controllable and realistic scene model, and realize the control of scene attributes such as lighting, camera parameters, posture and so on.

Structure from Motion

Structure from Motion (SfM) is a field within computer vision that seeks to reconstruct a three-dimensional structure of the environment from a sequence of two-dimensional images.

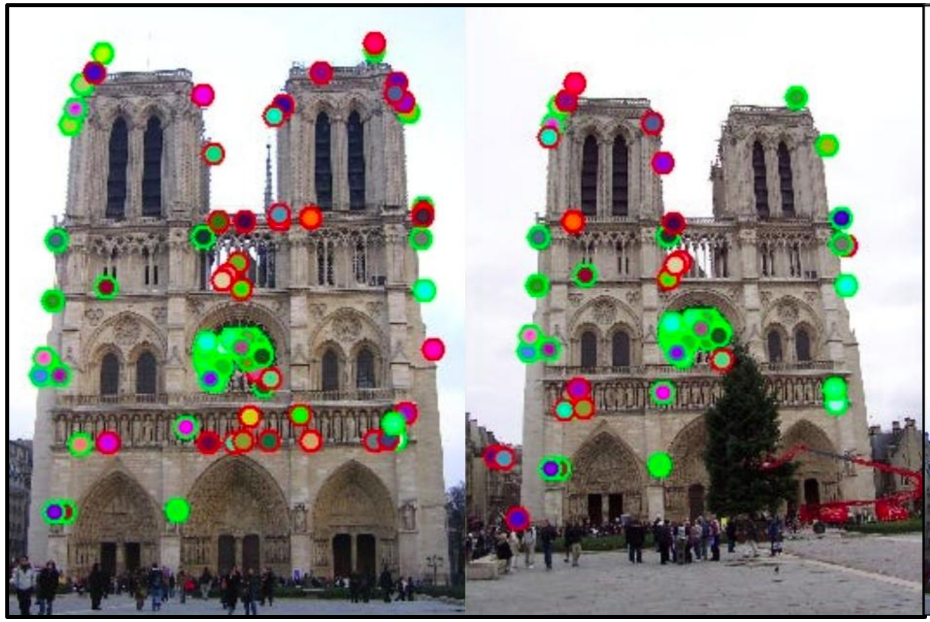


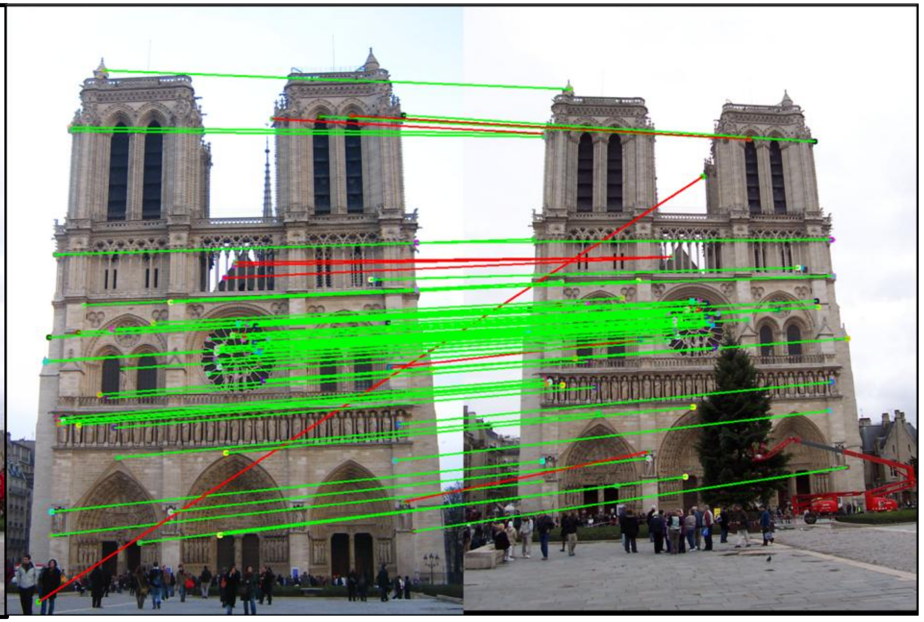
Structure from Motion

Obtaining the geometry of 3D scenes from 2D images is a challenging task because the image formation process (3D world → 2D image captured) is generally not invertible and additional information is needed to solve the reconstruction problem.

Given its image in two or more views, a 3D point can be reconstructed by triangulation.

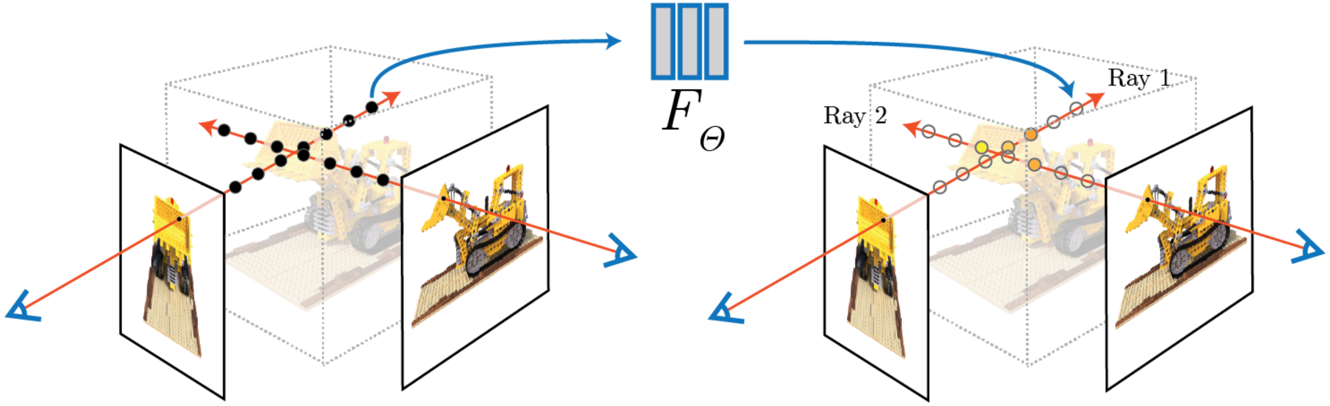
1. Feature extraction
2. Feature matching
3. 3D reconstruction





Neural Radiance Fields (NeRF)

- Used to represent a continuous scene with a neural network
- Treats the scene as a continuous function
- Uses differentiable rendering (volumetric rendering) to recreate the scene
- The differentiable rendering allows us to train the network with backpropagation



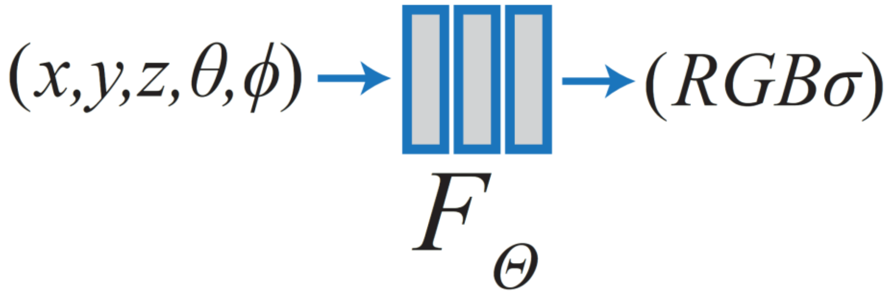
Neural Radiance Fields (NeRF)

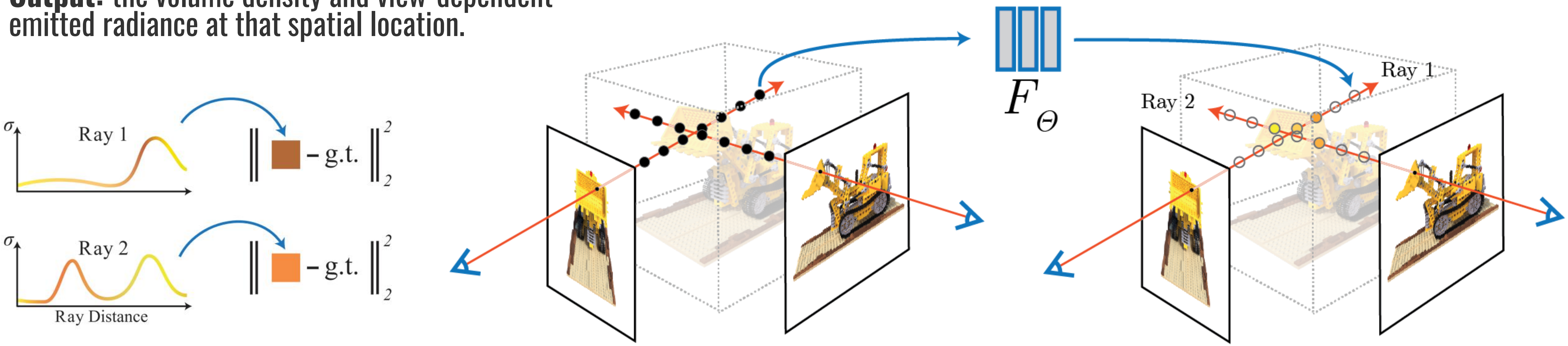
NeRF represents a scene using a fully-connected (non-convolutional) deep network

Input: 5D coordinate

- spatial location (x, y, z)
- viewing direction (θ, φ)

Output: the volume density and view-dependent emitted radiance at that spatial location.





Gaussian Splatting (GS)

3D Gaussian Splatting is a rasterization technique described in [3D Gaussian Splatting for Real-Time Radiance Field Rendering] that allows real-time rendering of photorealistic scenes learned from small samples of images.

Instead of triangles let's use gaussian splats to render a 3D scene.

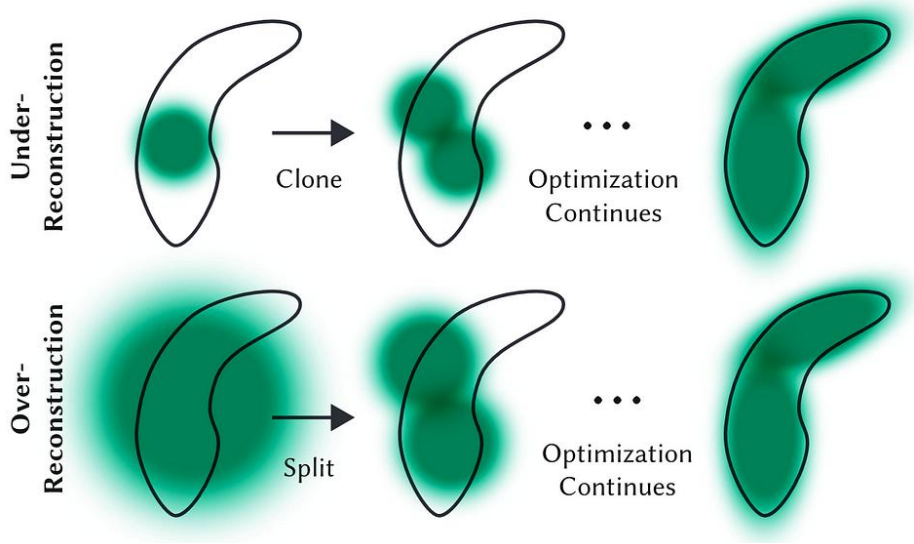
Each splat can be described as:
- **Position**: where it's located (XYZ)
- **Covariance**: how it's stretched/scaled (3x3 matrix)
- **Color**: what color it is (RGB)
- **Alpha**: how transparent it is (α)

https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Gaussian Splatting (GS)

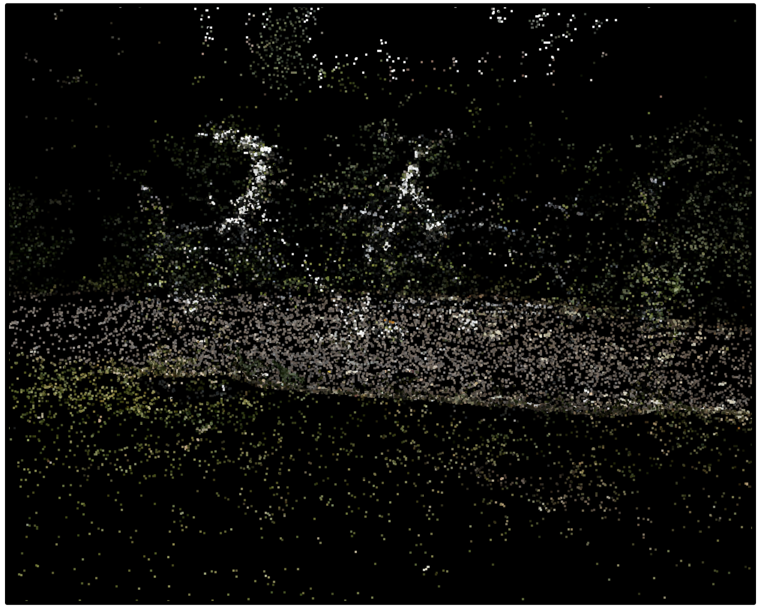
The training procedure uses Stochastic Gradient Descent, similar to a neural network, but without the layers. The training steps are:

1. Rasterize the gaussians to an image
2. Calculate the loss based on the difference between the rasterized image and ground truth image
3. Adjust the gaussian parameters according to the loss
4. Apply automated densification and pruning:
    - If the gradient is large for a given gaussian (i.e. it's too wrong), split/clone it
    - If the gaussian is small, clone it
    - If the gaussian is large, split it
    - If the alpha of a gaussian gets too low, remove it



https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Gaussian Splatting (GS)



Gaussian Splatting (GS)

https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Summary

- In **Deep Generative Modeling** the goal is to learn an unknown probability distribution, then create new samples that are similar to the samples from the unknown distribution.
- **Image Inpainting** is about filling in missing or occluded regions in digital images
- **Neural Rendering** is a technique that uses neural networks to generate, enhance or manipulate visual content in a realistic way.

Resources

Books:

- Courville, Goodfellow, Bengio: Deep Learning
  Freely available: https://www.deeplearningbook.org/
- Zhang, Aston and Lipton, Zachary C. and Li, Mu and Smola, Alexander J.: Dive into Deep Learning
  Freely available: https://d2l.ai/

Courses:

- Deep Learning specialization by Andrew NG
- https://www.coursera.org/specializations/deep-learning

Further Links + Resources

- Deep Learning-based Image and Video Inpainting: A Survey - https://arxiv.org/abs/2401.03395
- An Introduction to Deep Generative Modeling - https://arxiv.org/abs/2103.05180
- Advances in Neural Rendering - https://arxiv.org/abs/2111.05849
- 3D Gaussian Splatting for Real-Time Radiance Field Rendering - https://arxiv.org/abs/2308.04079
- Introduction to 3D Gaussian Splatting - https://huggingface.co/blog/gaussian-splatting

That’s all for today!