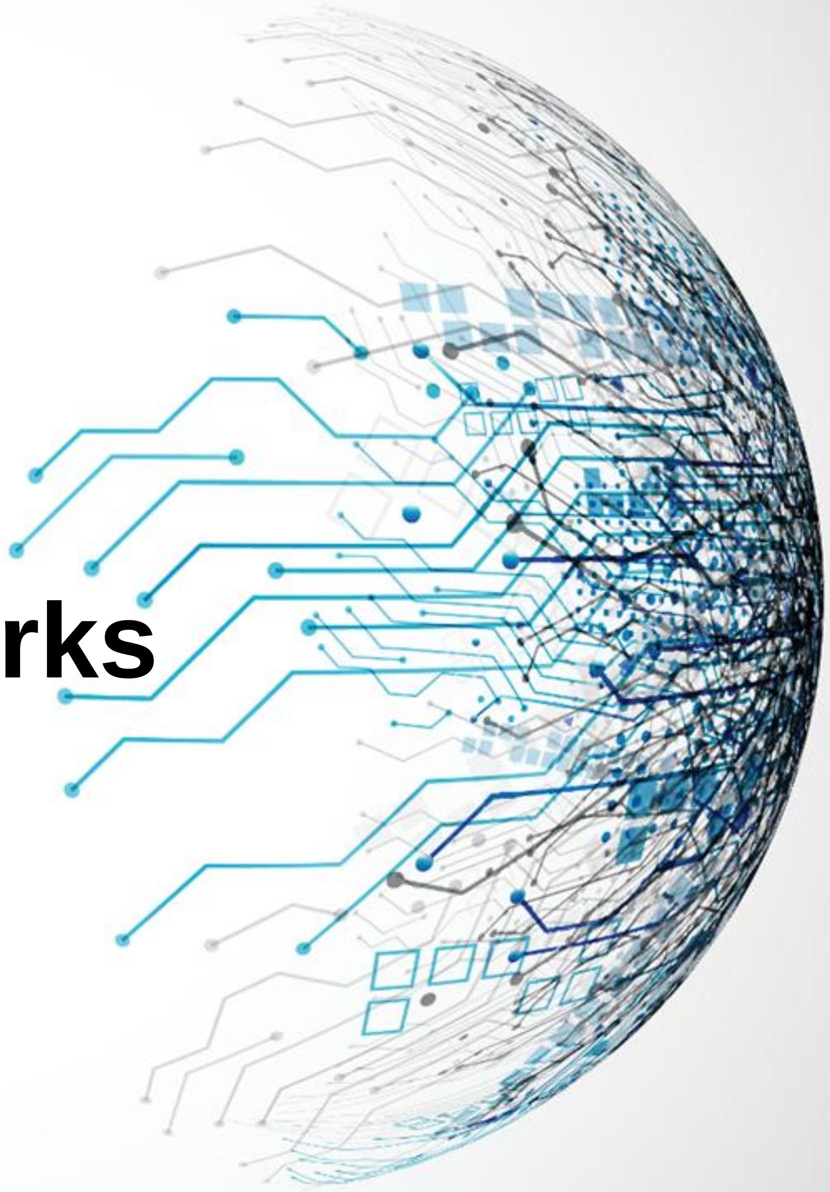
Deep Learning

Transfer Learning
&
Generative Adversarial Networks



Dr. Mohammed Salah Al-Radhi

(slides by: Dr. Bálint Gyires-Tóth)

BME-VIK-TMIT

22 October 2024

Copyright

Outline

1. Transfer Learning & Pretrained Networks

2. Generative Adversarial Networks

Transfer Learning & Pretrained Networks

Motivation

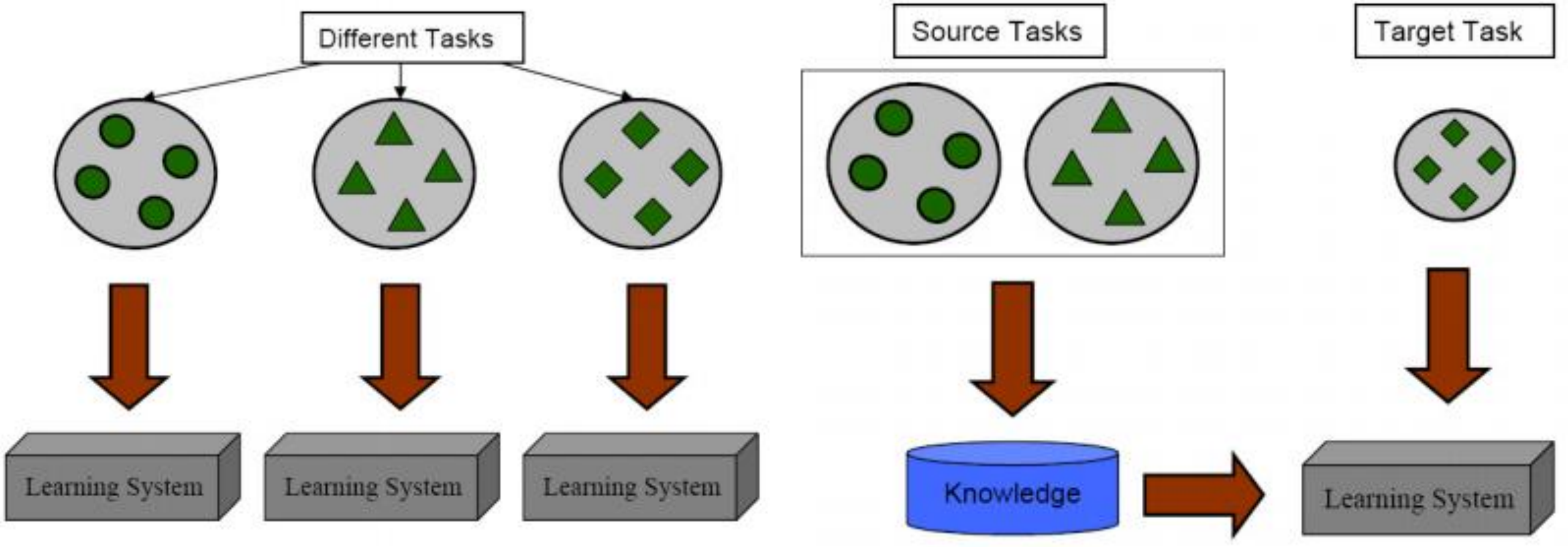
- Lots of data, time, resources needed to train and tune a neural network from scratch
  - An ImageNet deep neural networks can take weeks to train and fine-tune from scratch.
  - Unless you have 256 GPUs, possible to achieve in 1 hour

➢ Cheaper, faster way of adapting a neural network by exploiting their generalization properties!

What is Transfer Learning?

- It is the process of training a model on a large-scale dataset and then using that pretrained model to conduct learning for another downstream task (i.e., target task).
- OR, Transferring the knowledge of one model to perform a new task.

Traditional vs. Transfer Learning



(a) Traditional Machine Learning (b) Transfer Learning

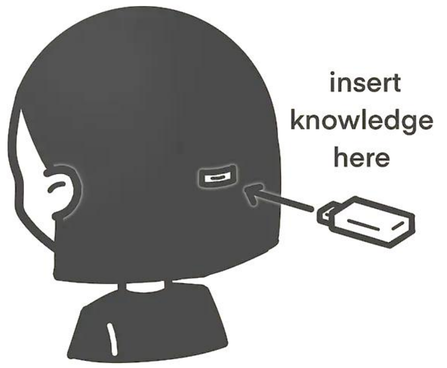
https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.147.9185&rep=rep1&type=pdf

Transfer Learning

In TL, we are typically working with two datasets:

- ❑ **<u>Source dataset</u>**, that typically contains a large amount of data

- ❑ **<u>Target dataset</u>**, that is typically smaller, and contains classes that do not appear in the source dataset

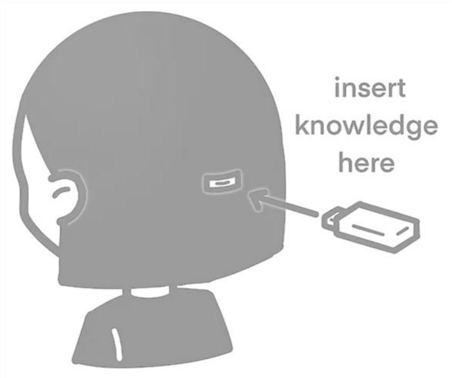


Transfer Learning

In TL, we are typically working with two datasets:

- **Source dataset**, that typically contains a large amount of data
- **Target dataset**, that is typically smaller, and contains classes that do not appear in the source dataset

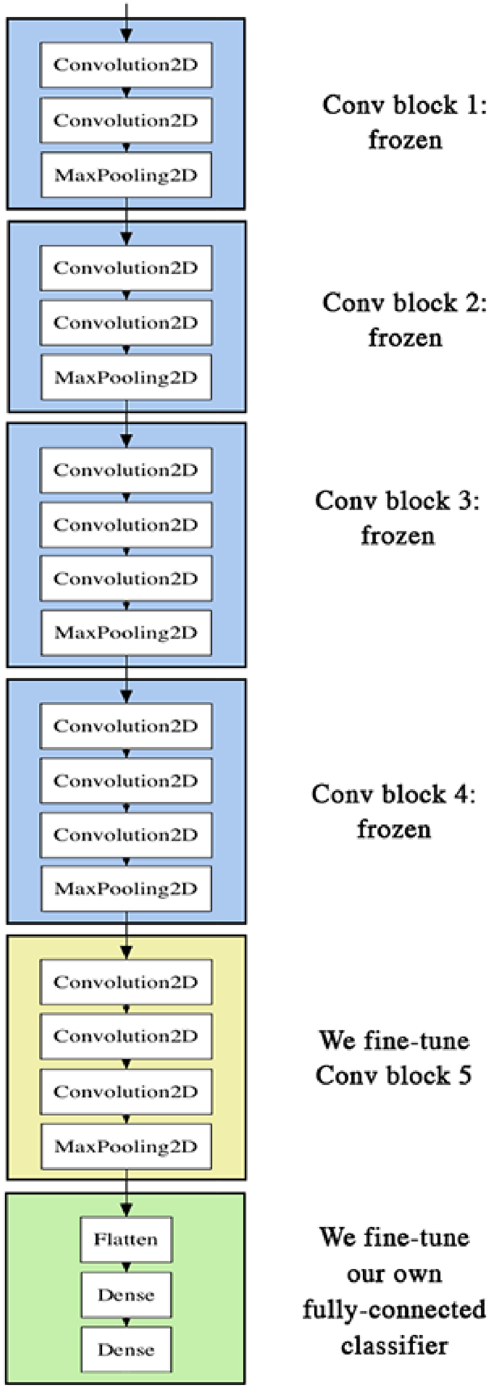
Can we use the information in the source dataset to improve classification accuracy on the target dataset?



Freezing and Fine-tuning

Process

1. Start with pre-trained network
2. Partition network into:
    - Featurizers: identify which layers to keep
    - Classifiers: identify which layers to replace
3. Re-train classifier layers with new data
4. Unfreeze weights and fine-tune whole network with smaller learning rate

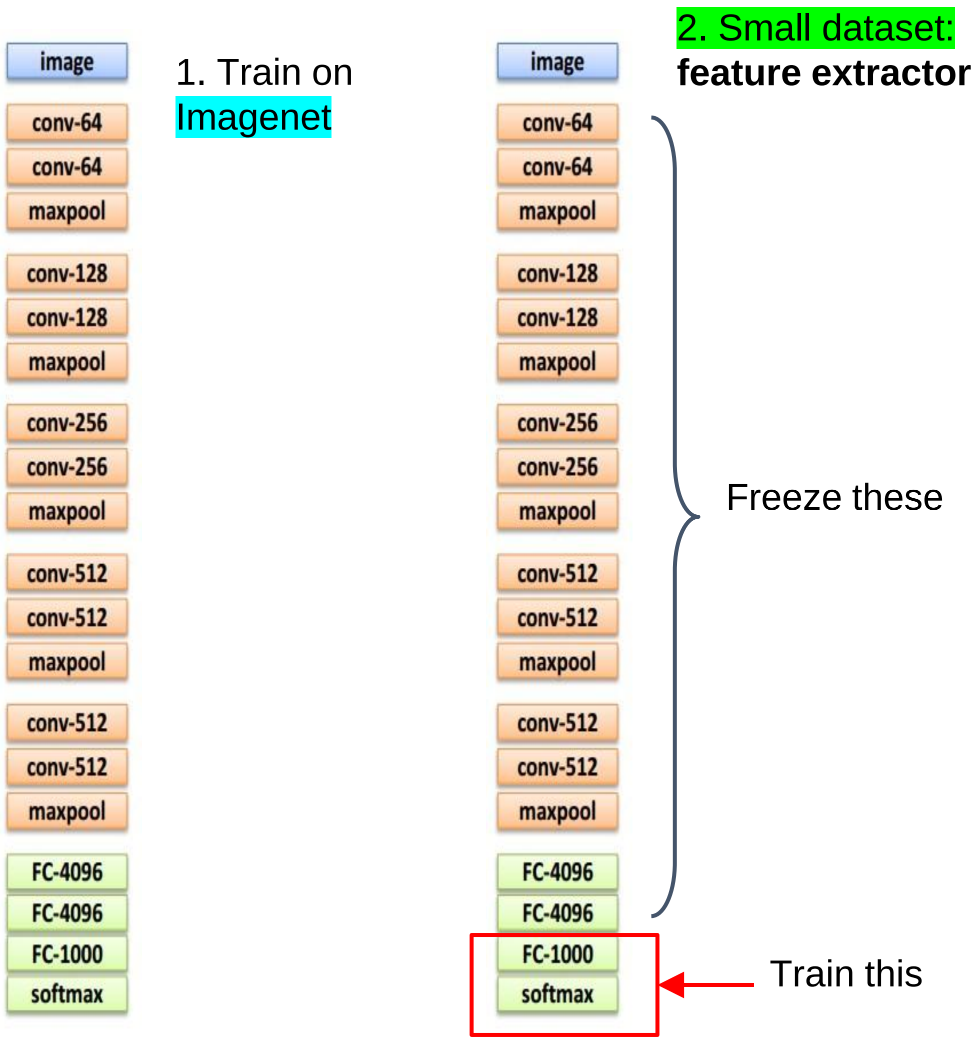


https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

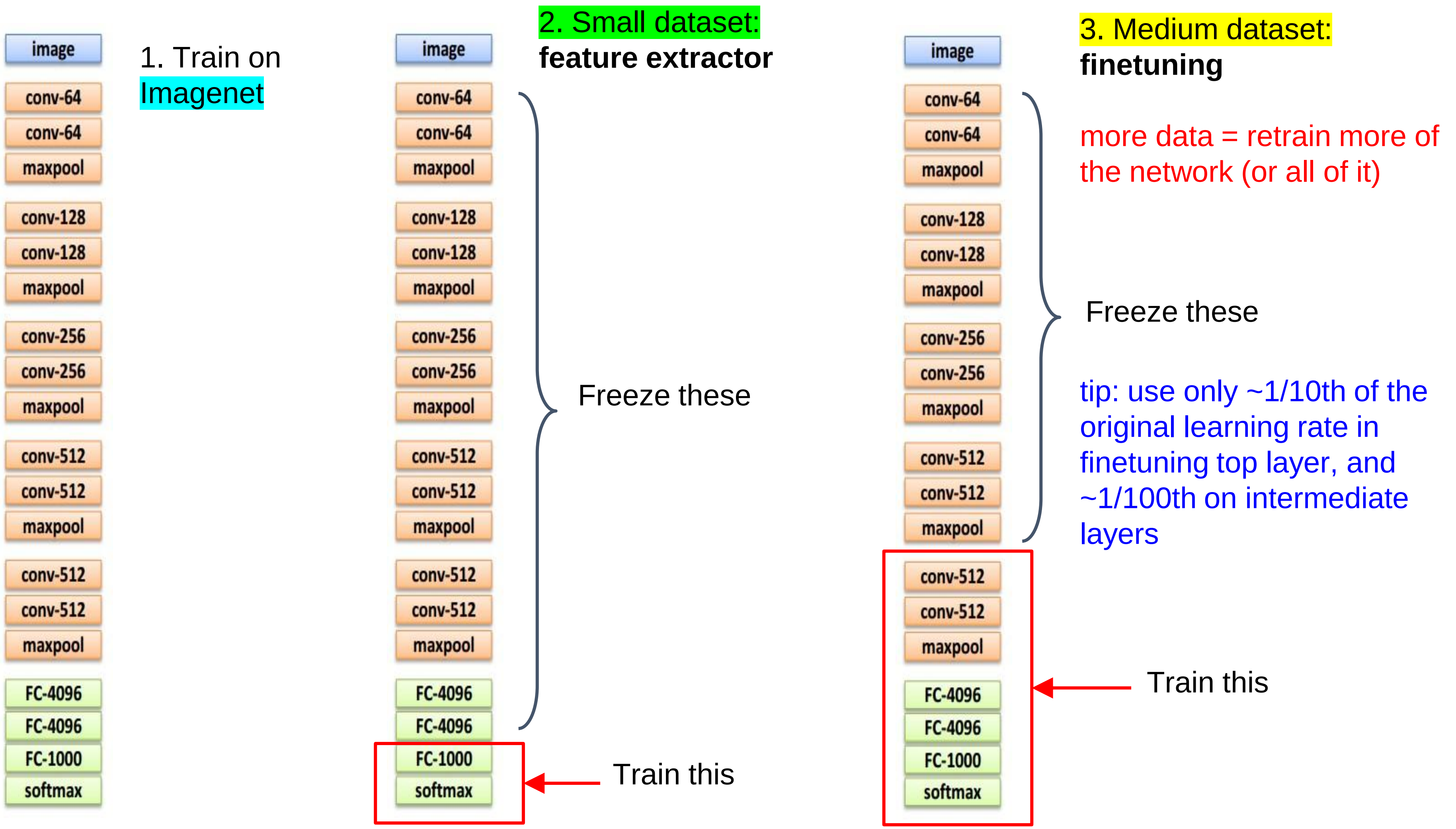
Transfer Learning with CNNs

1. Train on Imagenet

Transfer Learning with CNNs



Transfer Learning with CNNs



When and how to fine-tune?

Suppose we have model A, trained on dataset A

Q: How do we apply transfer learning to dataset B to create model B?

When and how to fine-tune?

Suppose we have model A, trained on dataset A

Q: How do we apply transfer learning to dataset B to create model B?

Dataset size	Dataset similarity	Recommendation
Large	Very different	Train model B from scratch, initialize weights from model A
Large	Similar	OK to fine-tune (less likely to overfit)
Small	Very different	Train classifier using the earlier layers (later layers won't help much)
Small	Similar	Don't fine-tune (overfitting). Train a linear classifier

https://cs231n.github.io/transfer-learning/

Why use pretrained models?

- Cost
  - Development
    - OpenAI GPT-4: $100 million[1] (~36 milliád HUF)
    - Meta Llama 2: $20 million[2] (~8 milliárd HUF)
  - Training 1 model
    - Stable Diffusion: $50k[3] (18 millió HUF)
    - DINOv2: $50k12k (~4 millió HUF)
      - ViT-L/14 on ImageNet-22k
      - 96 A100 GPUs, 3.3 days[4]
  - Data
- Adding new tasks
- Lifelong learning

[1] https://www.wired.com/story/openai-ceo-sam-altman-the-age-of-giant-ai-models-is-already-over/
[2] https://www.promptengineering.org/how-does-llama-2-compare-to-gpt-and-other-ai-language-models/
[3] https://www.mosaicml.com/blog/stable-diffusion-2
[4] https://github.com/facebookresearch/dinov2
Kép: Freepik AI Image Generator
https://www.mosaicml.com/blog/mosaicbert

Which is the Fastest Image Pretrained Model

- Depth refers to the topological depth of the network. This includes activation layers, batch normalization layers etc.
- Depth counts the number of layers with parameters.

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	99 MB	0.749	0.921	25,636,712	168
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

The top-1 and top-5 accuracy refers to the model's performance on the ImageNet validation dataset

Approaches to using pretrained networks

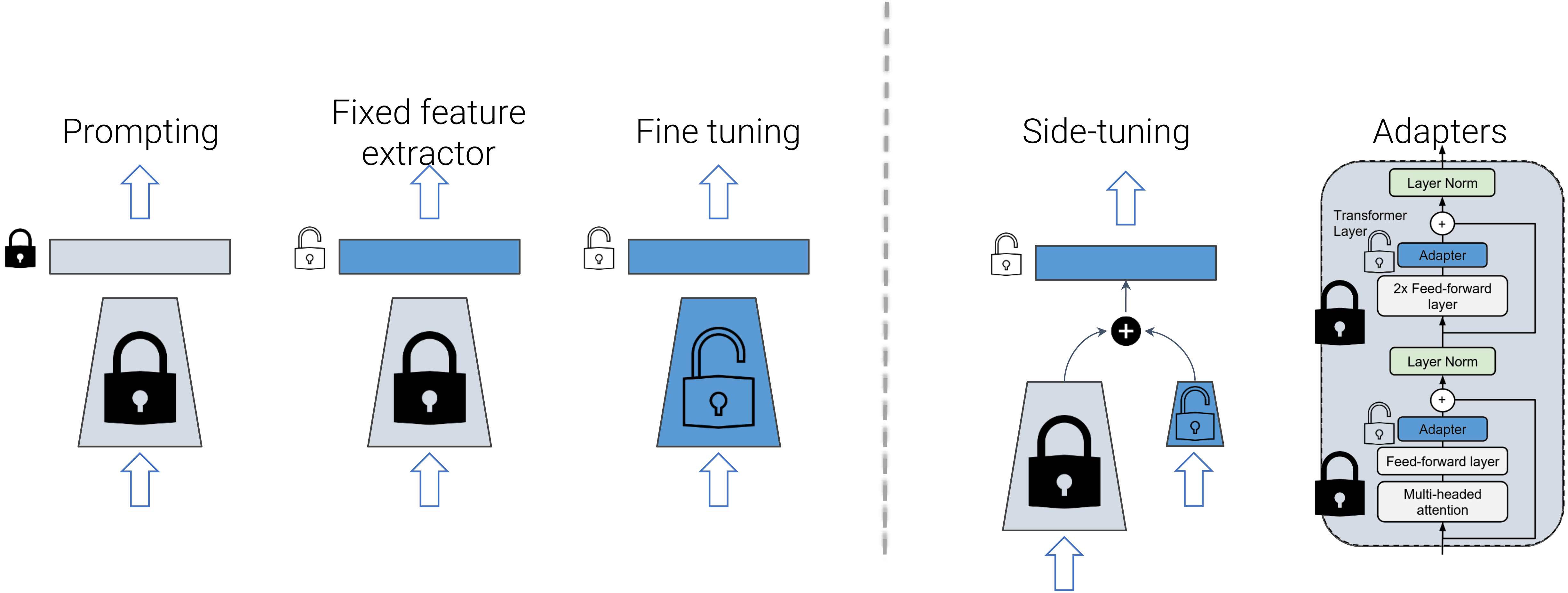
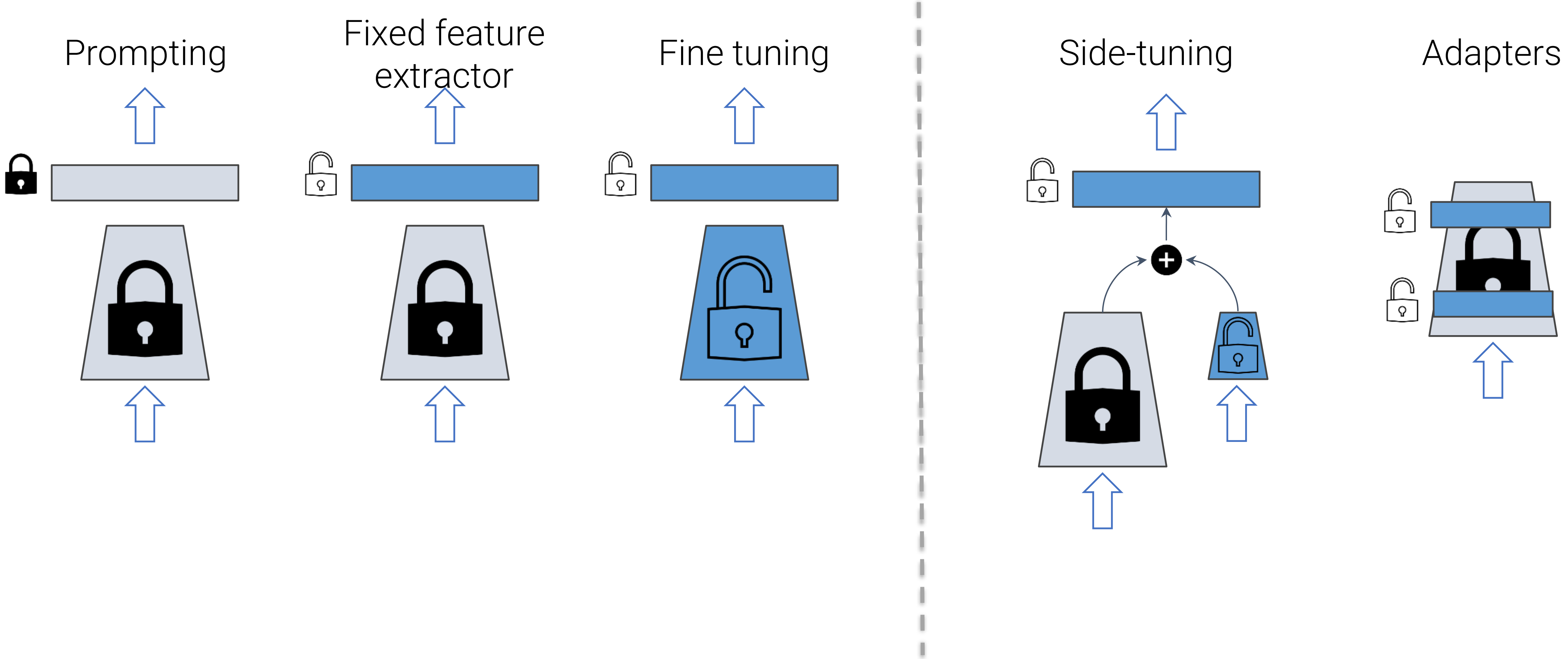


Figure from: Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). Parameter-Efficient Transfer Learning for NLP. In Proceedings of the 36th International Conference on Machine Learning (pp. 2790–2799). PMLR.
More reading on side tuning: http://sidetuning.berkeley.edu/

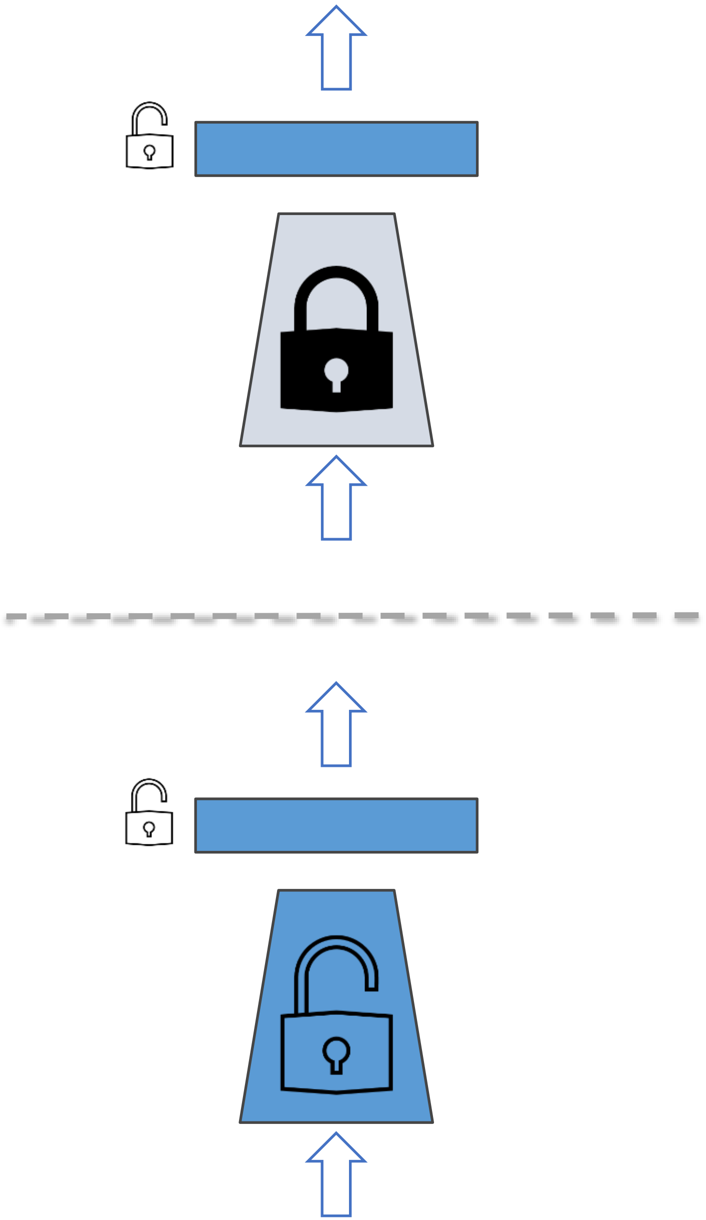
Approaches to using pretrained networks



Approaches to using pretrained networks

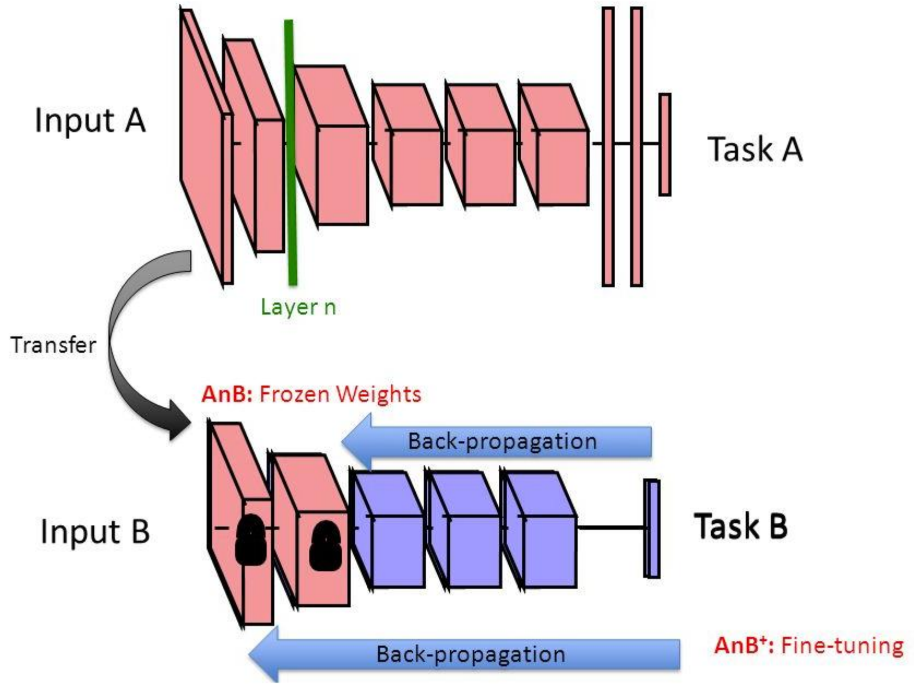
- Fixed feature extractor
  - \+ Memory, compute efficient
  - \+ No forgetting
  - \+ Multi-use feature extractor
  - \- Lack of flexibiltiy

- Fine tuning
  - \+ Flexible
  - \+ Usually better accuracy
  - \- Forgetting, catastrophic forgetting



Transfer Learning Applications

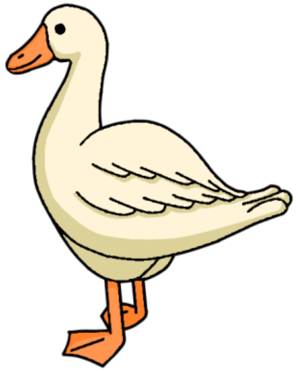
- Image classification (most common): learn new image classes
- Text sentiment classification
- Text translation to new languages
- Speaker adaptation in speech recognition
- Question answering



Generative Adversarial Networks

What is GANs ..?

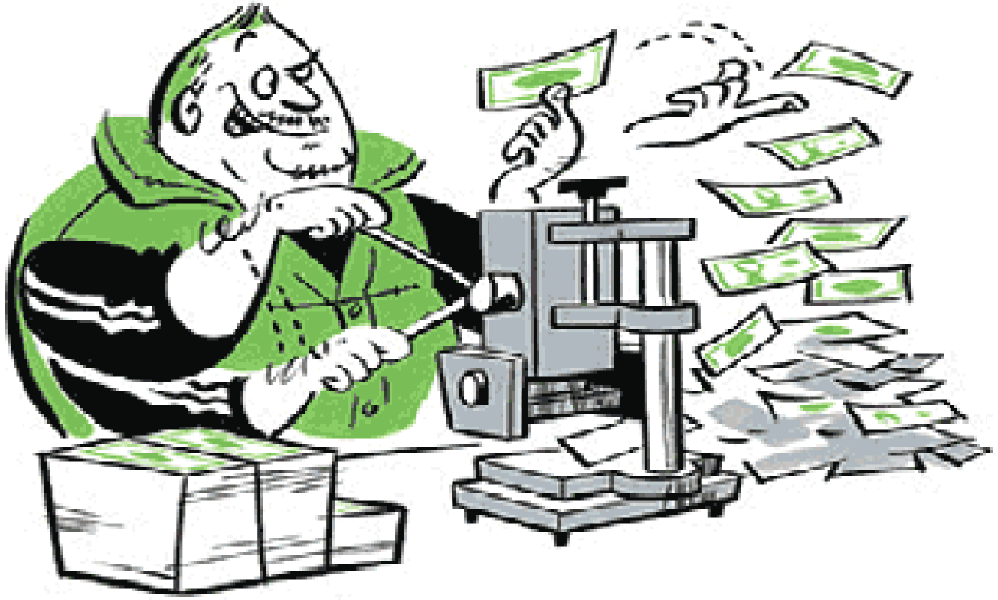
- **Generative**
  - Learn a generative model
- **Adversarial**
  - Trained in an adversarial setting
- **Networks**
  - Use Deep Neural Networks



Generator v.s. Discriminator

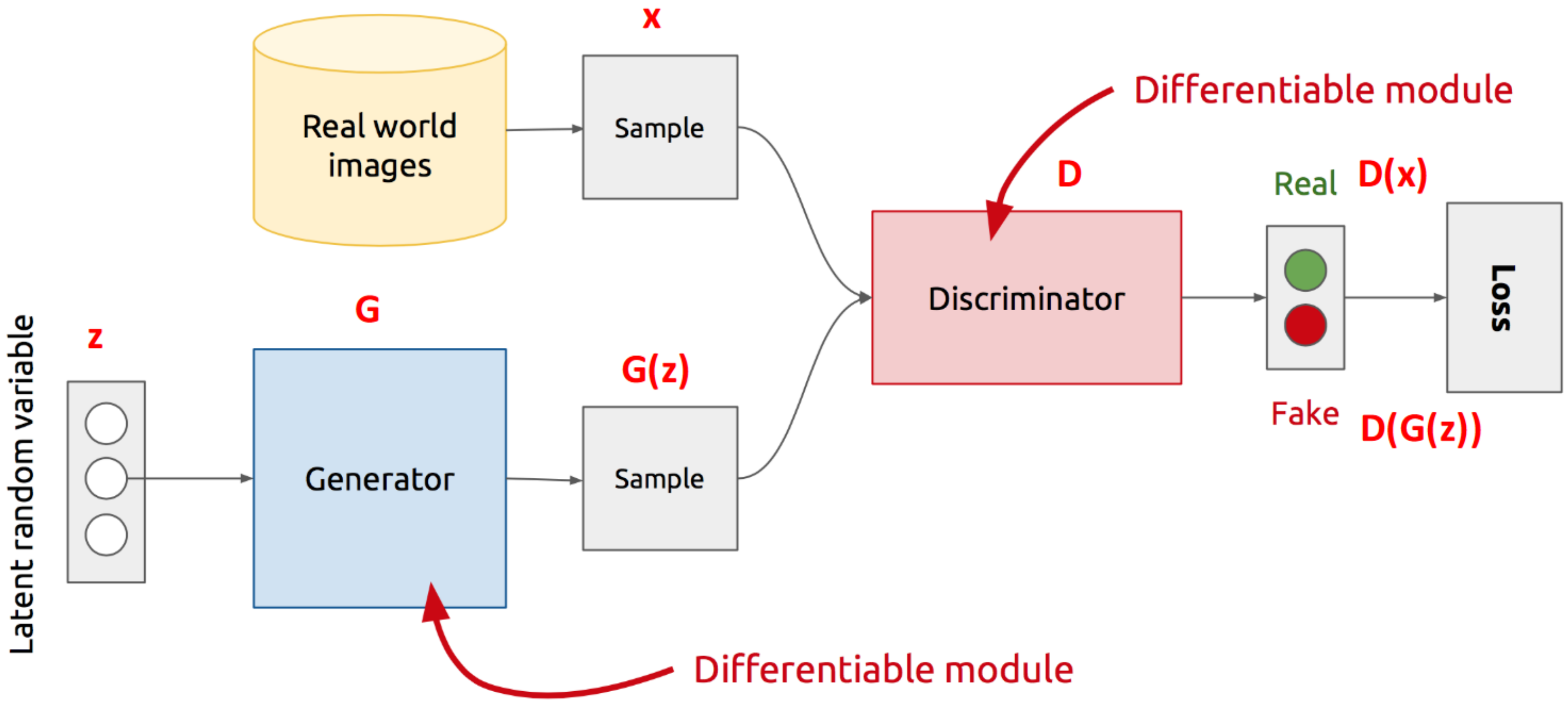
discriminator

generator



Forrás: Goodfellow (2016), https://arxiv.org/abs/1701.00160

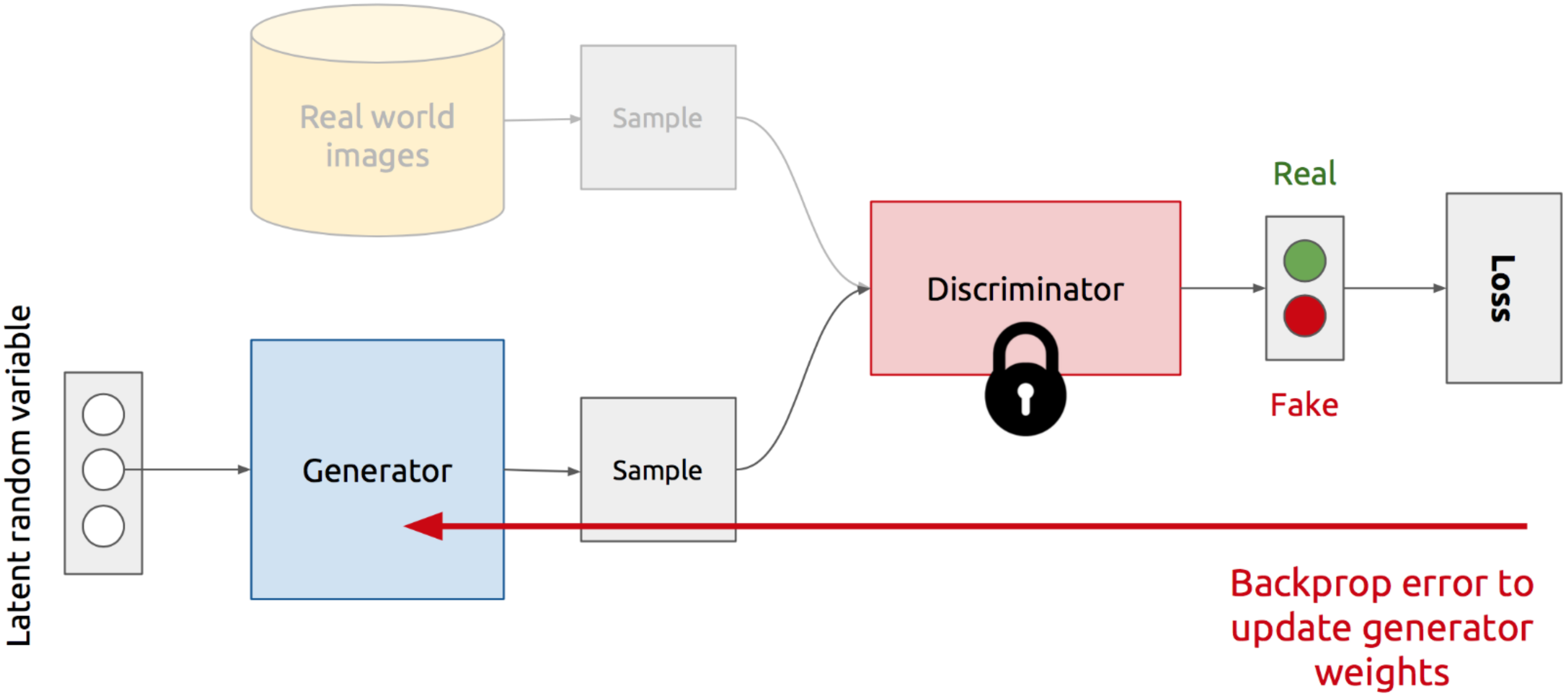
GAN’s Architecture



- **Z** is some random noise (Gaussian/Uniform).
- **Z** can be thought as the latent representation of the image.

https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Training Generator

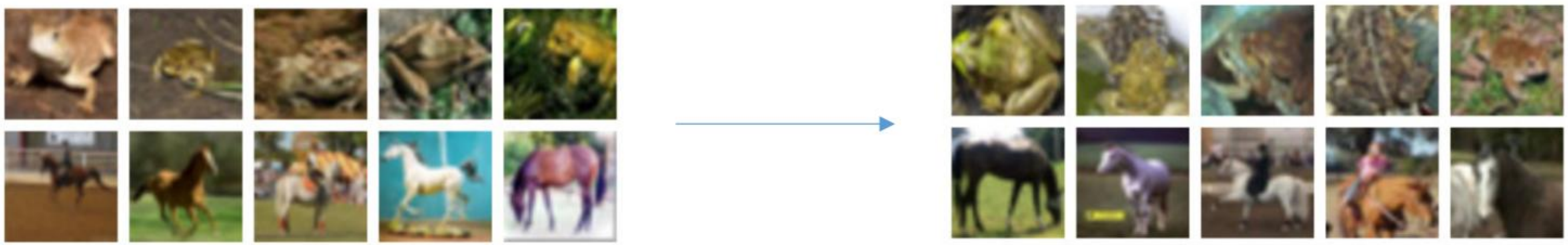


https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

Generative modelling

- Generating samples
  - Training samples

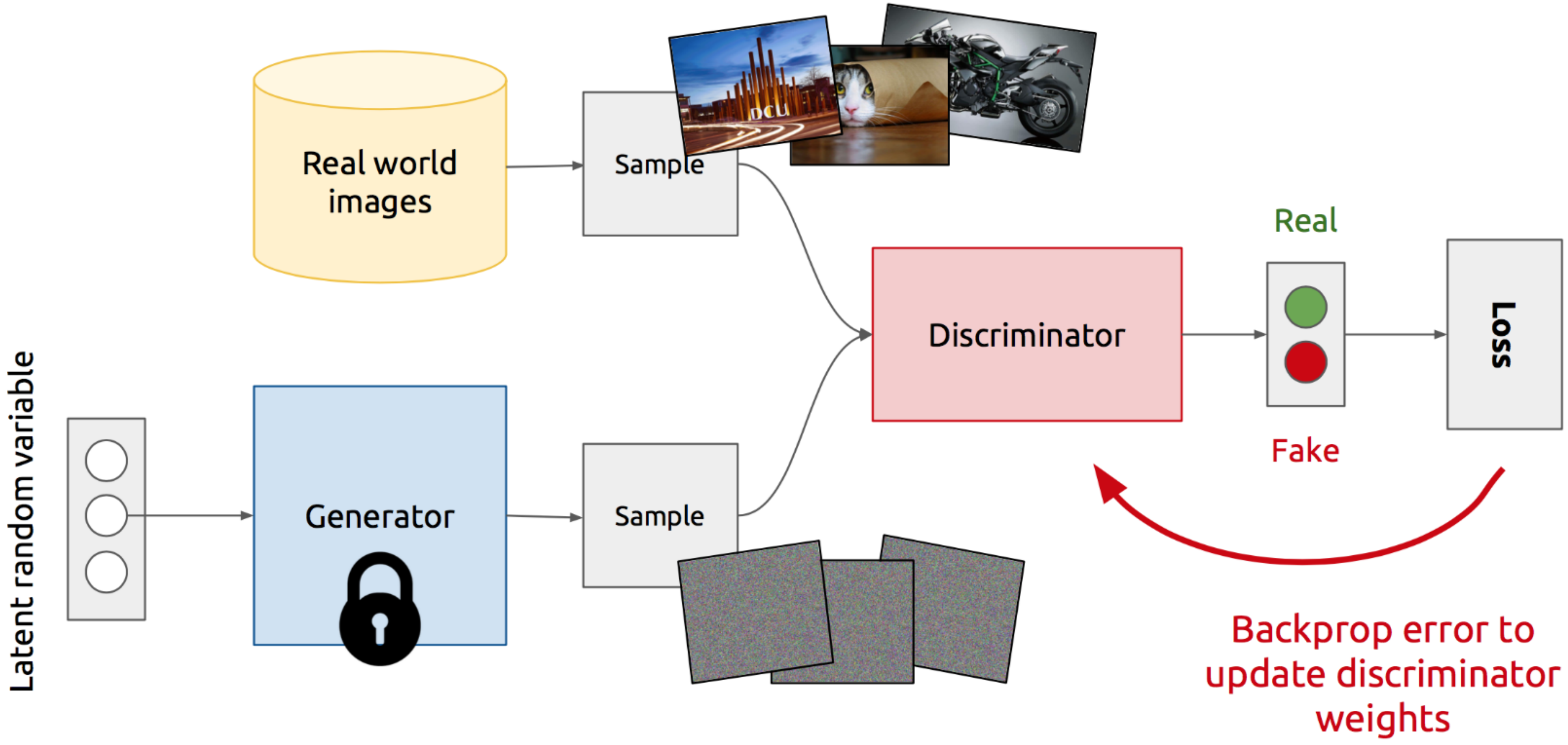
Samples of the model



Why is it worth to deal with generative models?

- Multi-dimensional density distributions
- Simulating the possible future
- Handling missing data
- Multimodal output
- Realistic generation tasks

Training Discriminator

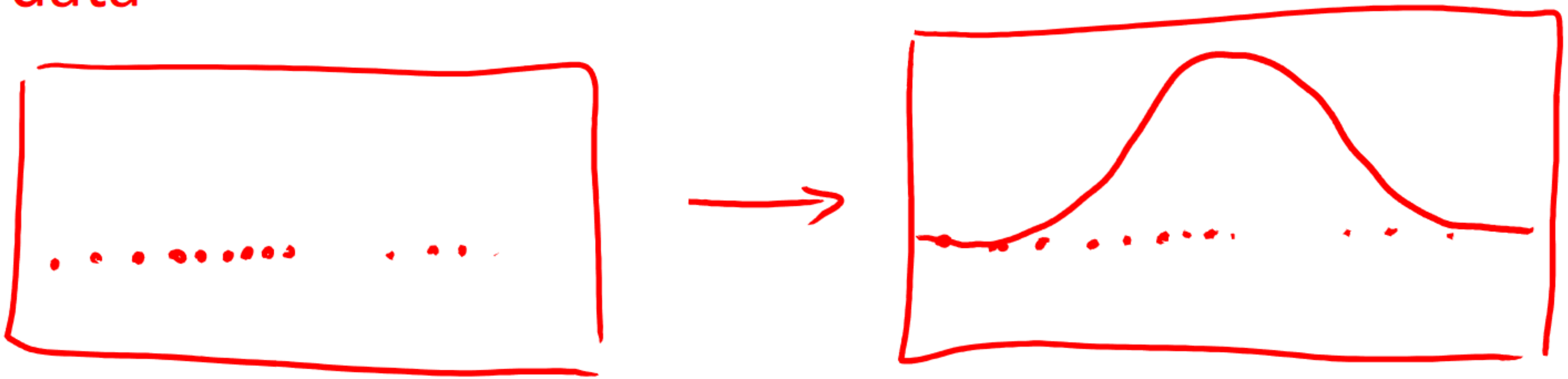


https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

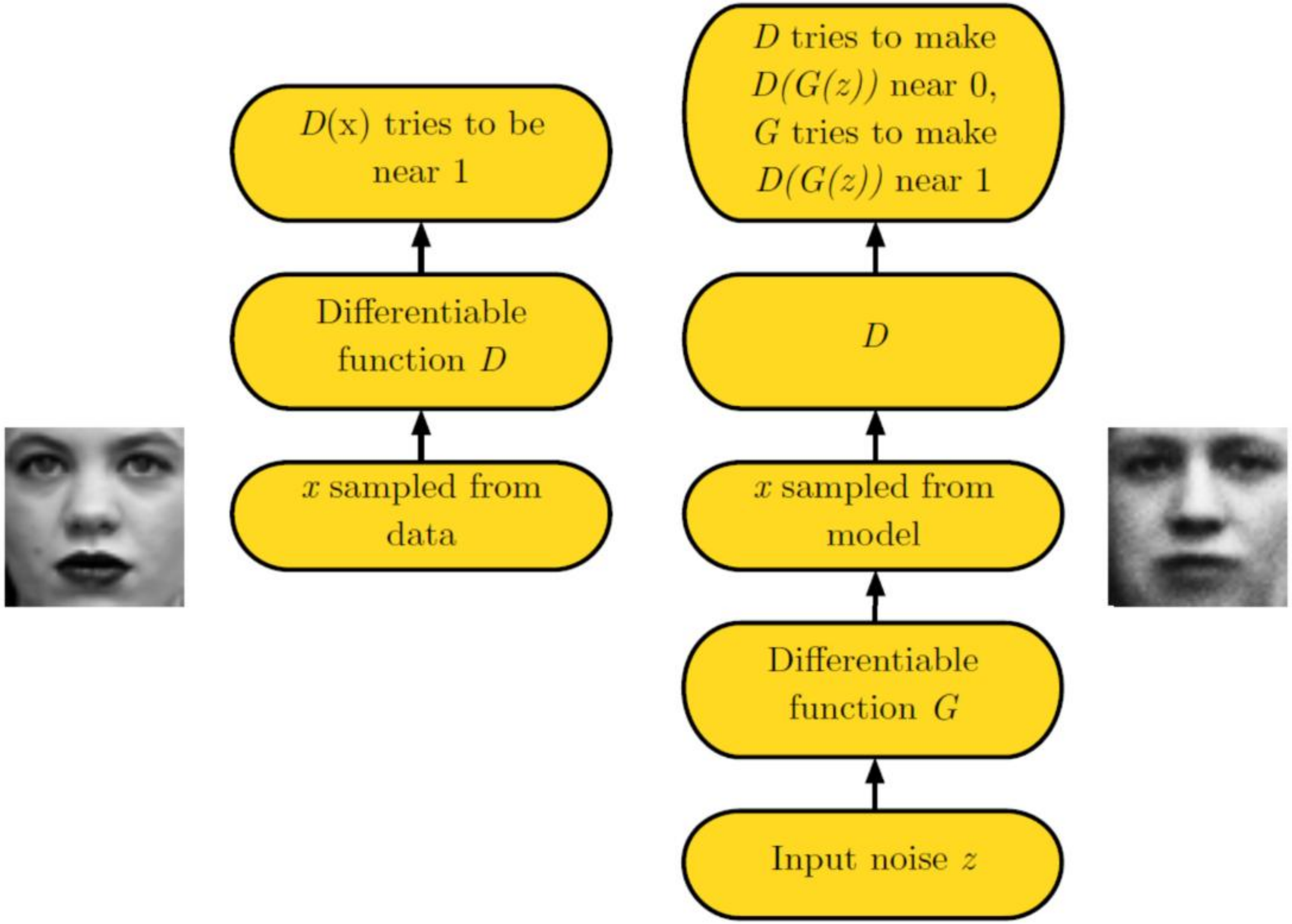
Discriminative modelling

- Estimation of Probability density functions

data



Adversarial Net



Forrás: Goodfellow (2016), https://arxiv.org/abs/1701.00160

Adversarial Training

- We can generate adversarial samples to fool a discriminative model
- We can use those adversarial samples to make models robust
- We then require more effort to generate adversarial samples
- Repeat this and we get better discriminative model

- **GANs extend that idea to generative models:**
  - Generator: generate fake samples, tries to fool the Discriminator
  - Discriminator: tries to distinguish between real and fake samples
  - Train them against each other
  - Repeat this and we get better Generator and Discriminator

Generator v.s. Discriminator

- ***Generator***
- Pros:
  - Easy to generate even with deep model
- Cons:
  - Imitate the appearance
  - Hard to learn the correlation between components

- ***Discriminator***
- Pros:
  - Considering the big picture
- Cons:
  - Generation is not always feasible
    - Especially when your model is deep
  - How to do negative sampling?

Advantages of GANs

- **Plenty of existing work on Deep Generative Models**
  - Boltzmann Machine
  - Deep Belief Nets
  - Variational AutoEncoders (VAE)

- **Why GANs?**
  - Sampling (or generation) is straightforward.
  - Training doesn't involve Maximum Likelihood estimation.
  - Robust to Overfitting since Generator never sees the training data.
  - Empirically, GANs are good at capturing the modes of the distribution.

Goodfellow, Ian. **"NIPS 2016 Tutorial: Generative Adversarial Networks."** arXiv preprint arXiv:1701.00160 (2016).

Why use GANs for Generation?

- Can be trained using back-propagation for Neural Network based Generator/Discriminator functions.
- Sharper images can be generated.
- Faster to sample from the model distribution: *single* forward pass generates a *single* sample.

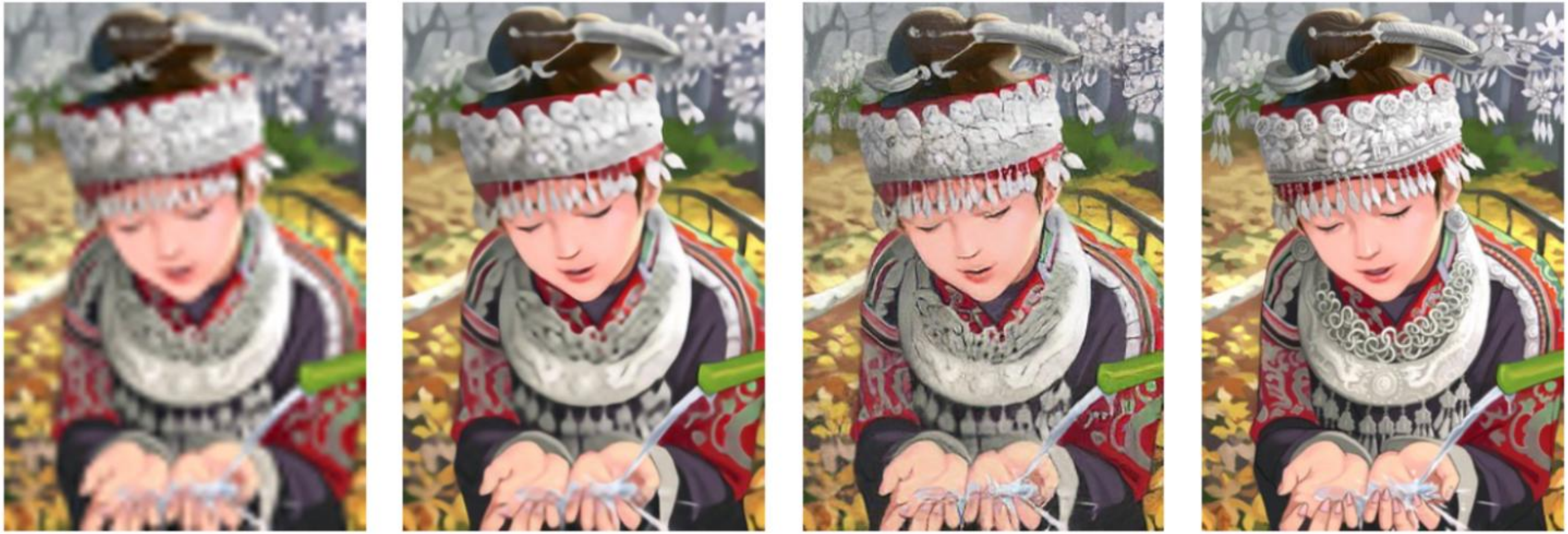
Magic of GANs …

Which one is Computer generated?

Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *arXiv preprint arXiv:1609.04802* (2016).

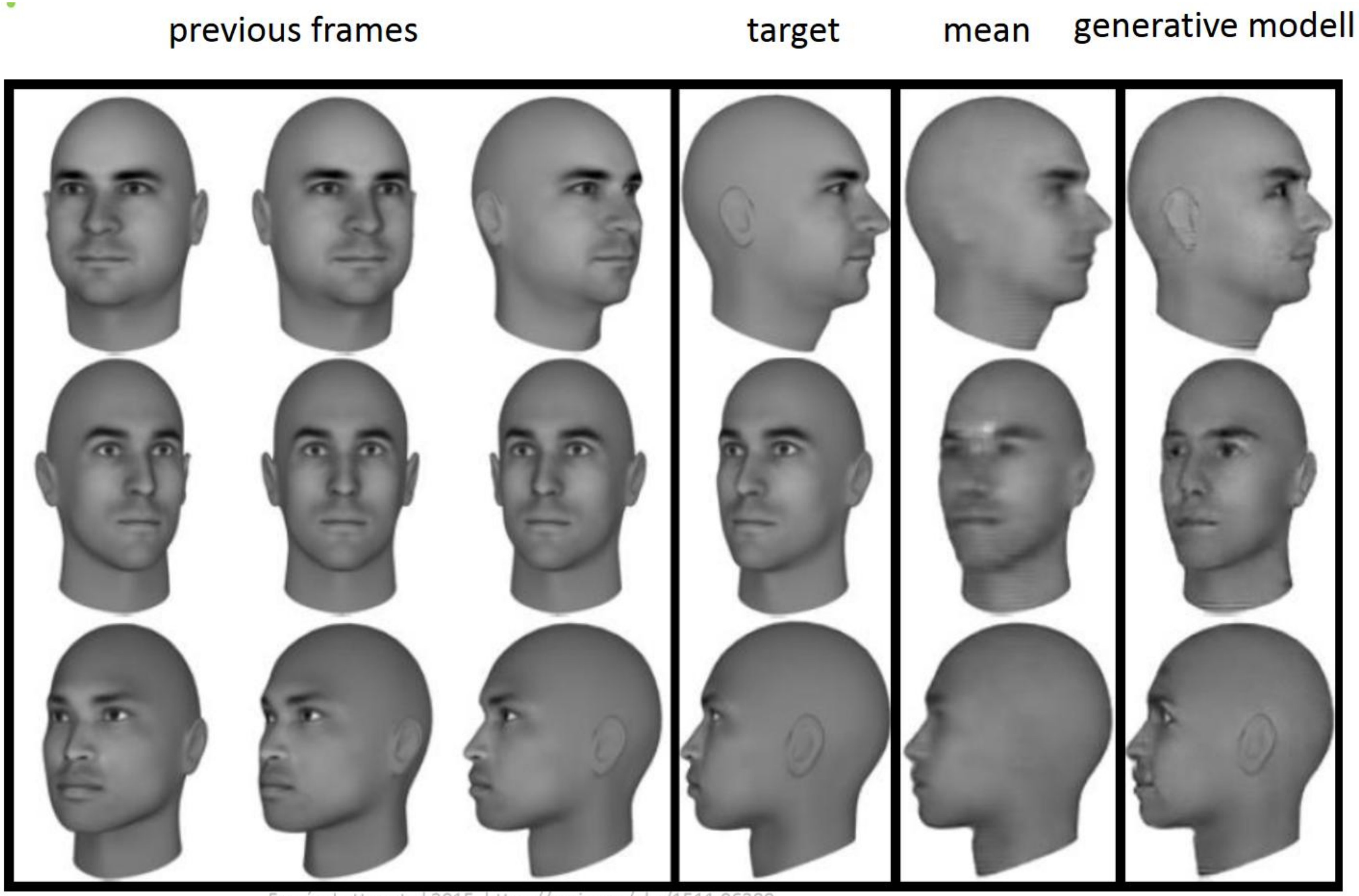
Magic of GANs …

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



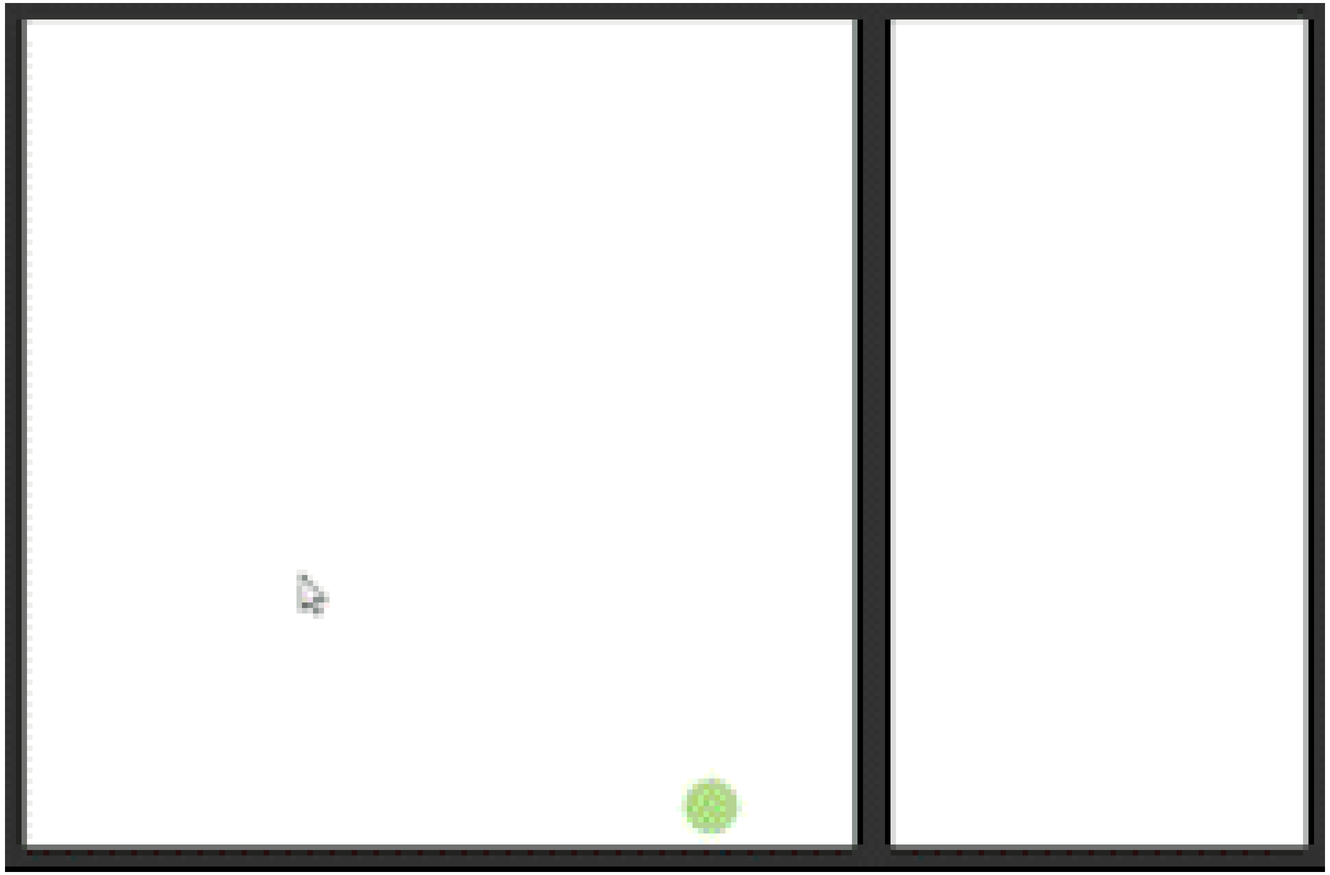
Single Image Super-Resolution

Predicting the next video frame

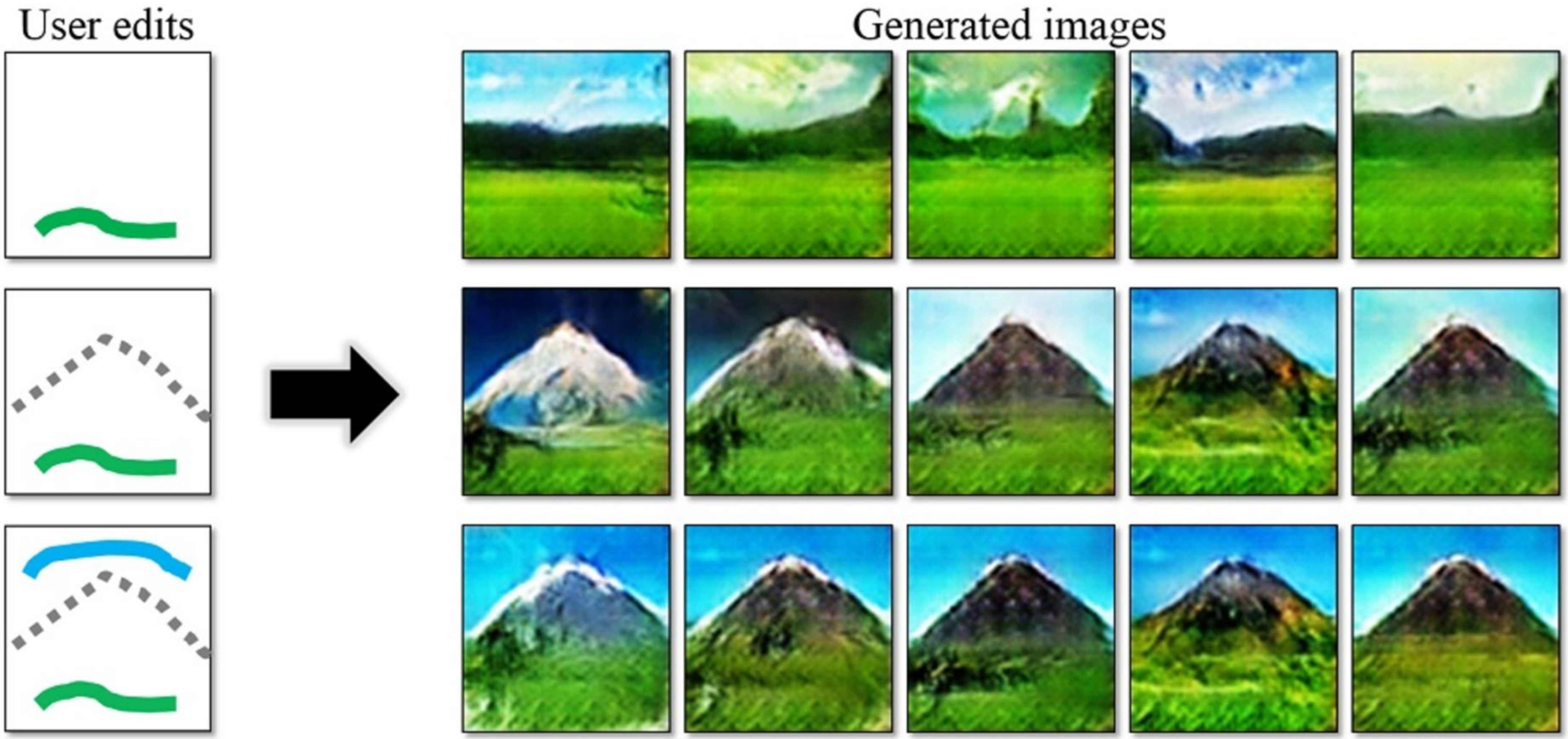


Forrás: Lotter et al 2015, https://arxiv.org/abs/1511.06380

iGAN (interactive)



iGAN (interactive)



http://people.eecs.berkeley.edu/~junyanz/projects/gvm/

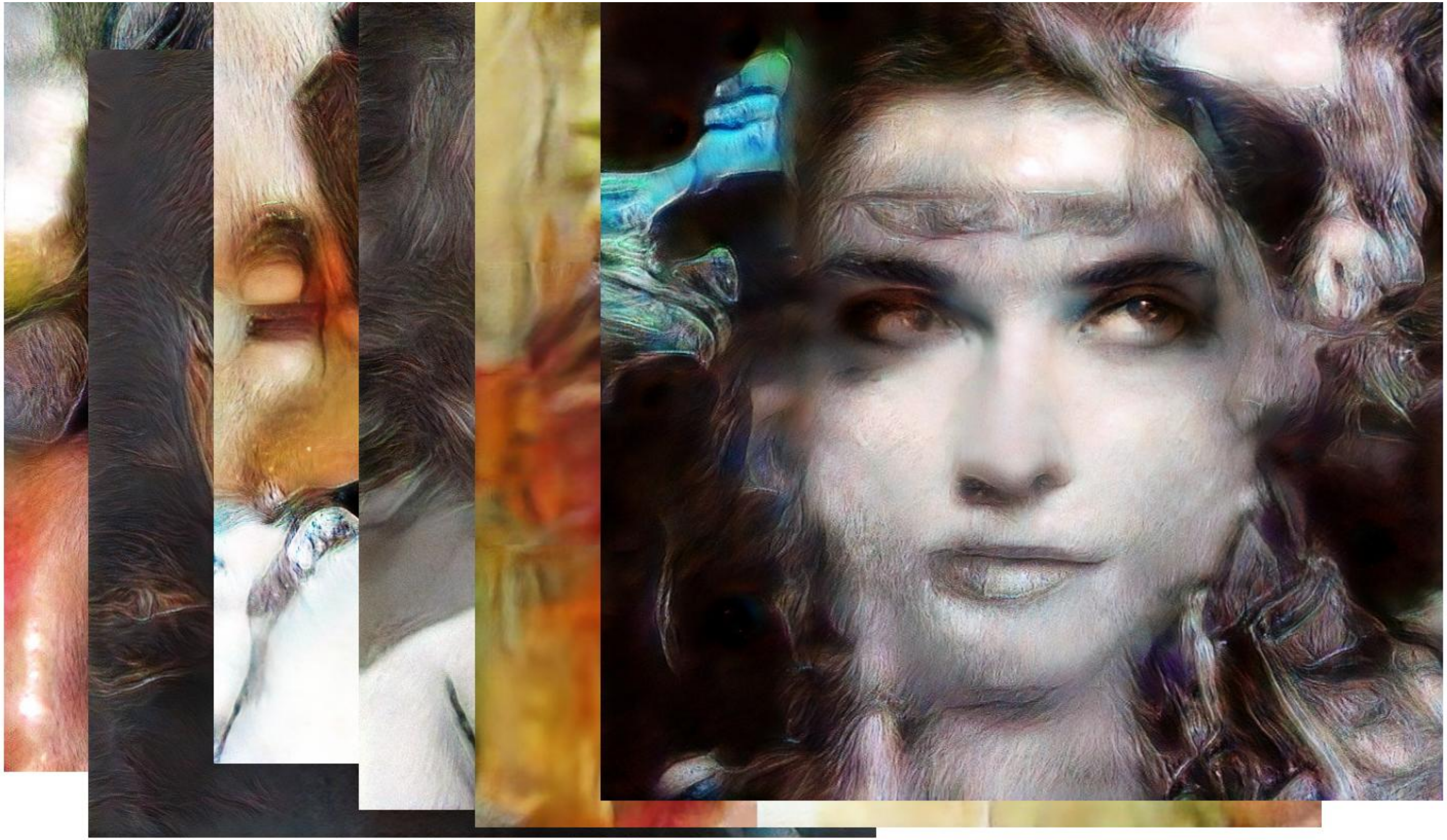
Image-to-image transformation

Forrás: Isola et al 2016, https://arxiv.org/abs/1611.07004

VAE learning to generate images (log time)

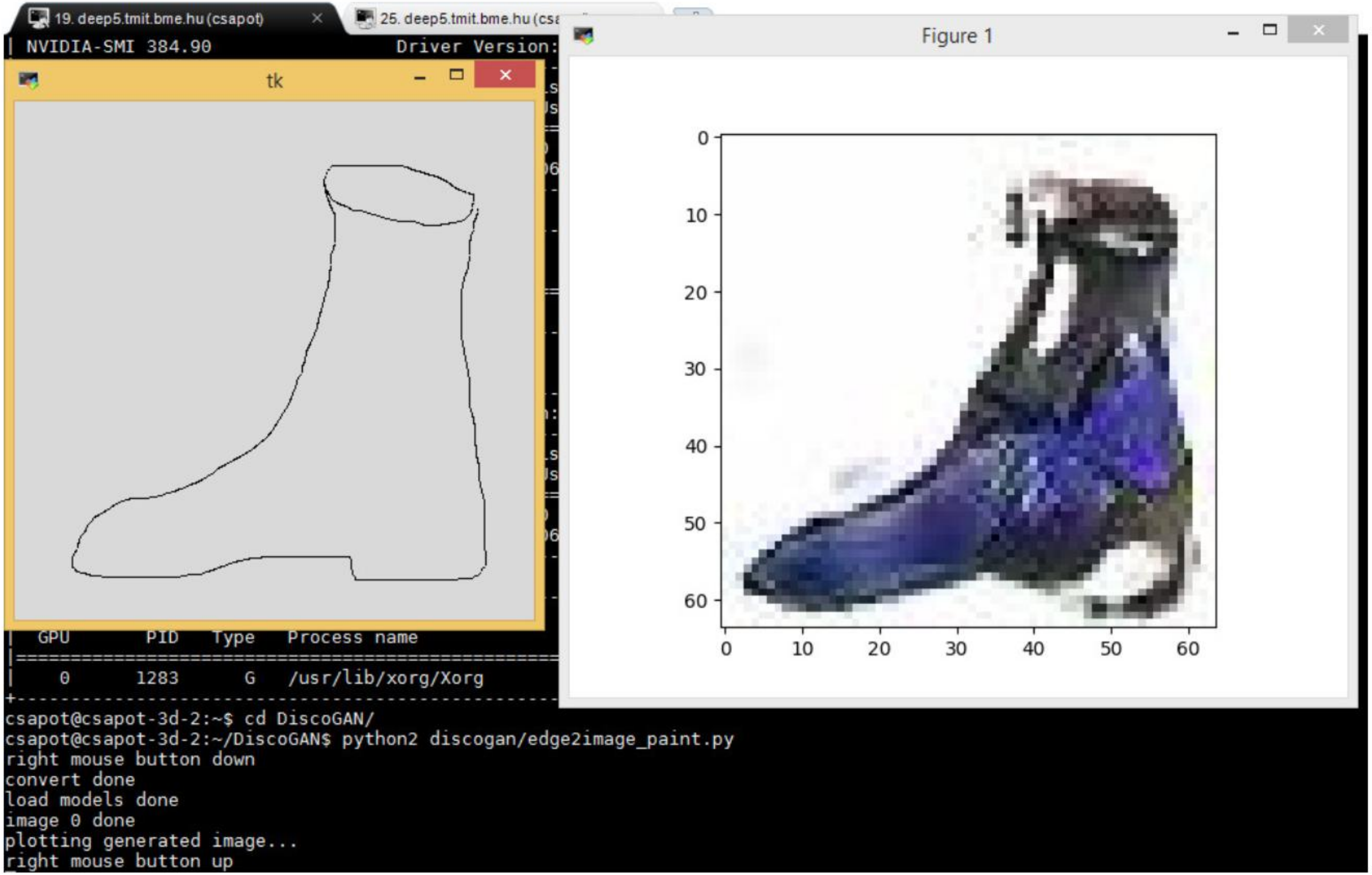
Forrás: https://openai.com/blog/generative-models/

GAN and art



Forrás: https://mtyka.github.io/machine/learning/2017/06/06/highres-gan-faces.html

Drawing to image

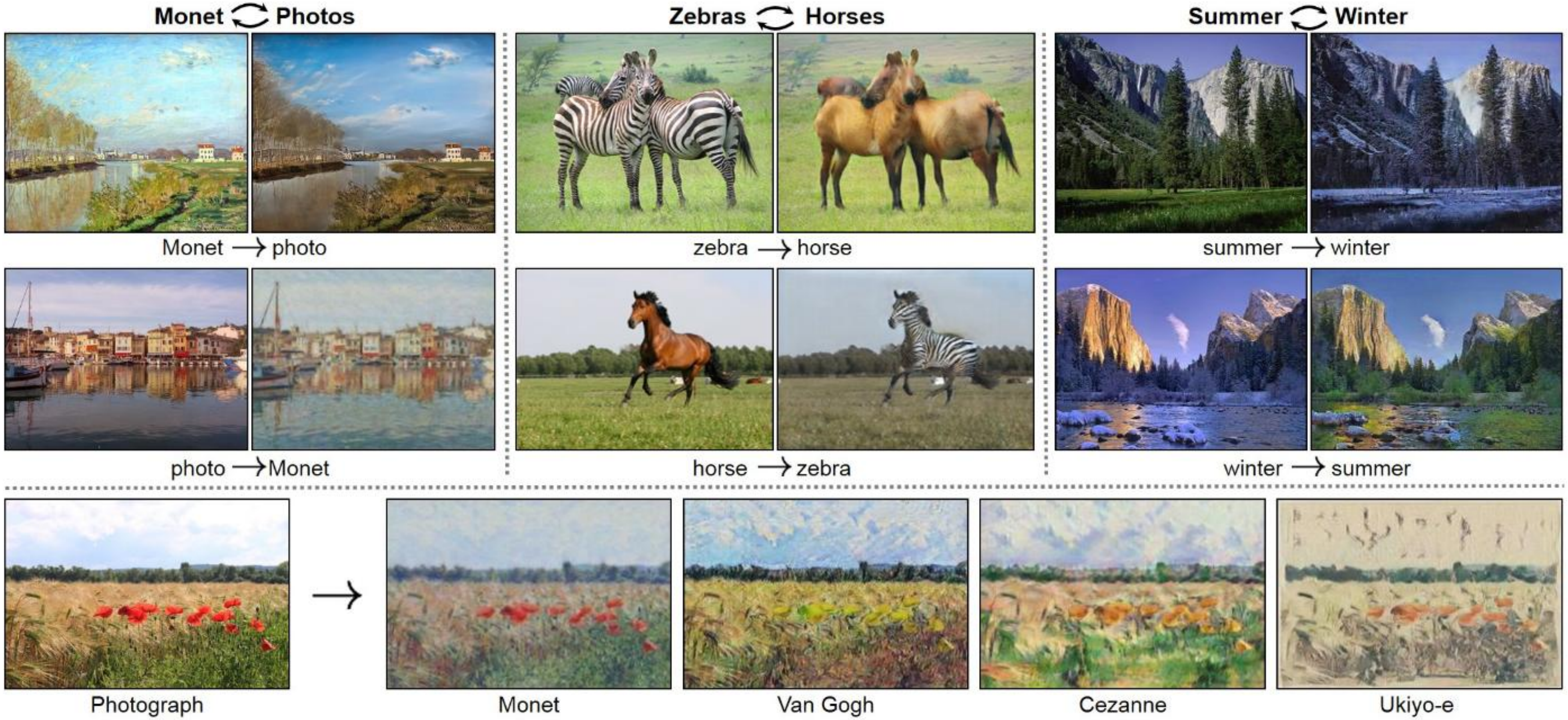


Text-To-Image / GAN: flowers

Caption	Image
this flower has yellow petals and a red stamen	
the center is yellow surrounded by wavy dark red petals	
this flower has lots of small round orange petals	

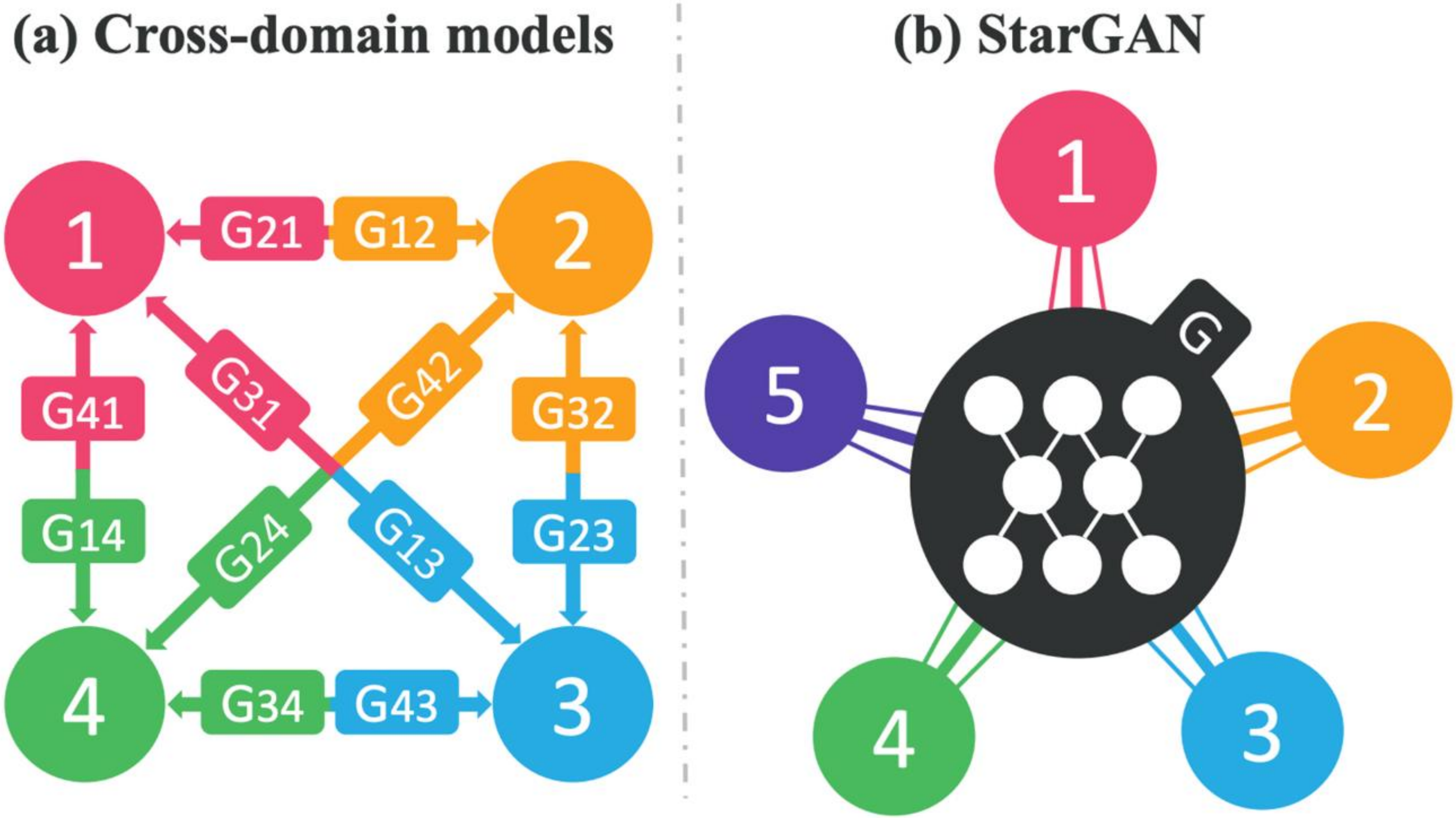
Forrás: https://github.com/reedscot/icml2016

CycleGAN: unpaired image-to-image translation



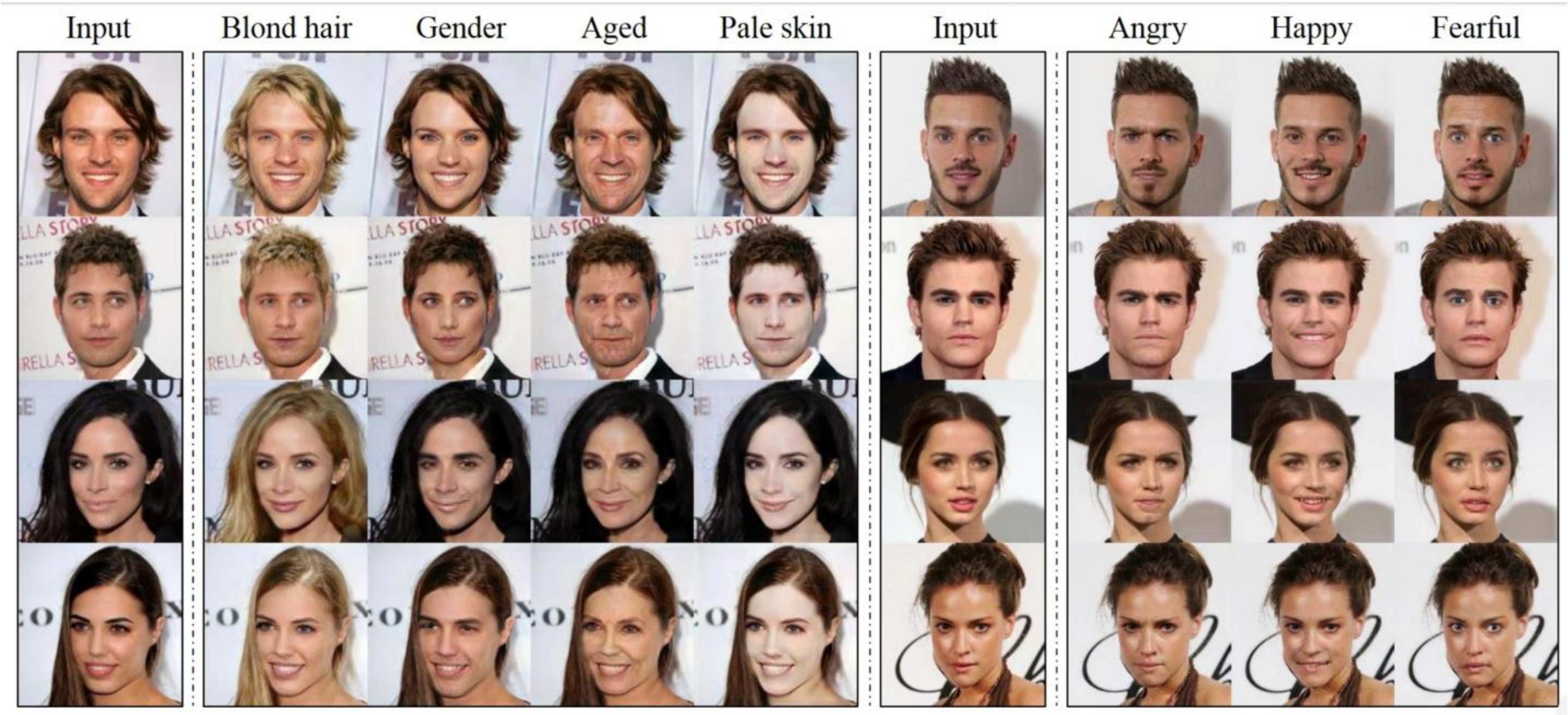
Forrás: https://junyanz.github.io/CycleGAN/

StarGAN: img-to-img translation



Forrás: https://lyusungwon.github.io/studies/2019/01/15/stargan/

StarGAN: img-to-img translation with a single model



Forrás: https://lyusungwon.github.io/studies/2019/01/15/stargan/

Problems with GANs

- **Probability Distribution is Implicit**
  - Not straightforward to compute P(X).
  - Thus **Vanilla GANs** are only good for Sampling/Generation.
- **Training is Hard**
  - Non-Convergence
  - Mode-Collapse

Summary

- GANs are generative models that are implemented using two stochastic neural network modules: **Generator** and **Discriminator**.
- **Generator** tries to generate samples from random noise as input
- **Discriminator** tries to distinguish the samples from Generator and samples from the real data distribution.
- Both networks are trained adversarially (in tandem) to fool the other component. In this process, both models become better at their respective tasks.

Reading List

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. Generative adversarial nets, NIPS (2014).
2. Goodfellow, Ian NIPS 2016 Tutorial: Generative Adversarial Networks, NIPS (2016).
3. Radford, A., Metz, L. and Chintala, S., Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434. (2015).
4. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. Improved techniques for training gans. NIPS (2016).
5. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximization Generative Adversarial Nets, NIPS (2016).
6. Zhao, Junbo, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126 (2016).
7. Mirza, Mehdi, and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014).
8. Liu, Ming-Yu, and Oncel Tuzel. Coupled generative adversarial networks. NIPS (2016).
9. Denton, E.L., Chintala, S. and Fergus, R., 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. NIPS (2015)
10. Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., & Courville, A. Adversarially learned inference. arXiv preprint arXiv:1606.00704 (2016).

Applications:

1. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004. (2016).
2. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. Generative adversarial text to image synthesis. JMLR (2016).
3. Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). Face Aging With Conditional Generative Adversarial Networks. arXiv preprint arXiv:1702.01983.

Please, don't forget to send feedback:

https://bit.ly/bme-dl

Thank you
for your attention

Dr. Mohammed Salah Al-Radhi

malradhi@tmit.bme.hu

(slides by: Dr. Bálint Gyires-Tóth)

22 October 2024