《神经网络理论与应用》第六讲

Neural Network Theory and Applications

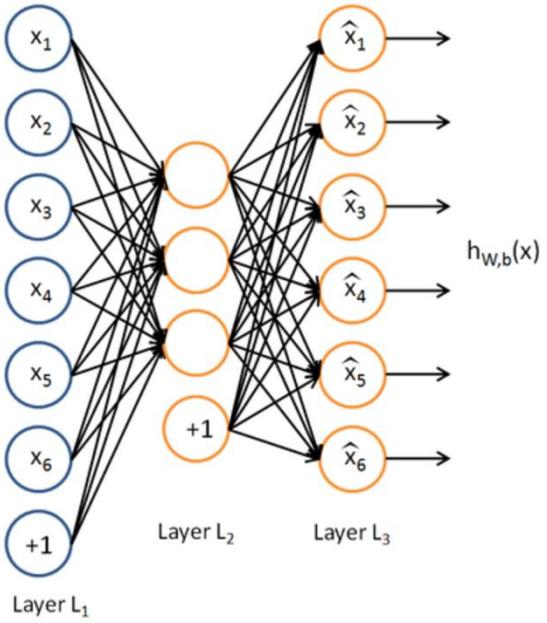
主讲教师：郑伟龙
助教：尹昊龙、史涵雯

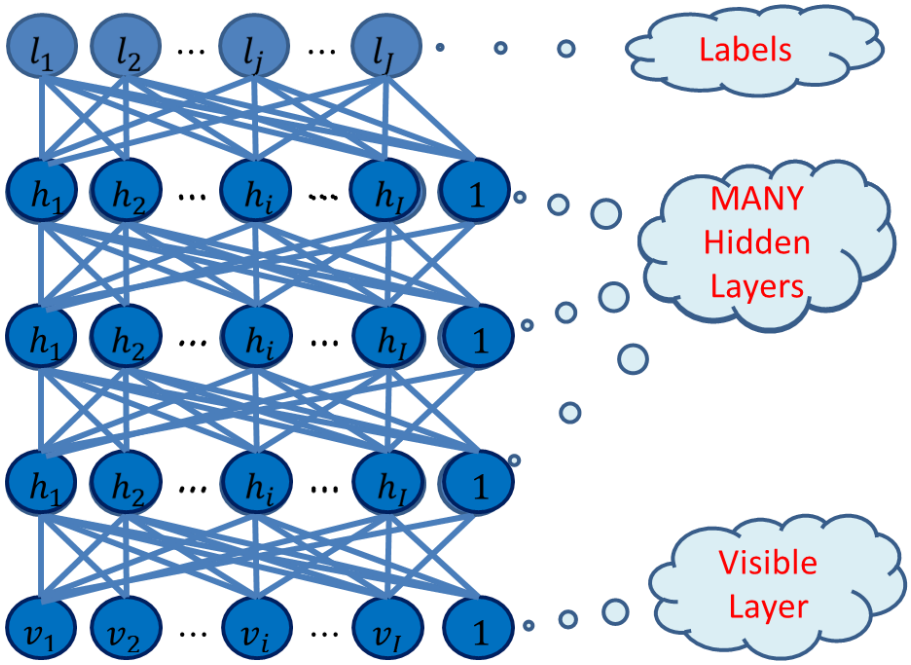
上海交通大学计算机科学与工程系

weilong@sjtu.edu.cn
http://bcmi.sjtu.edu.cn

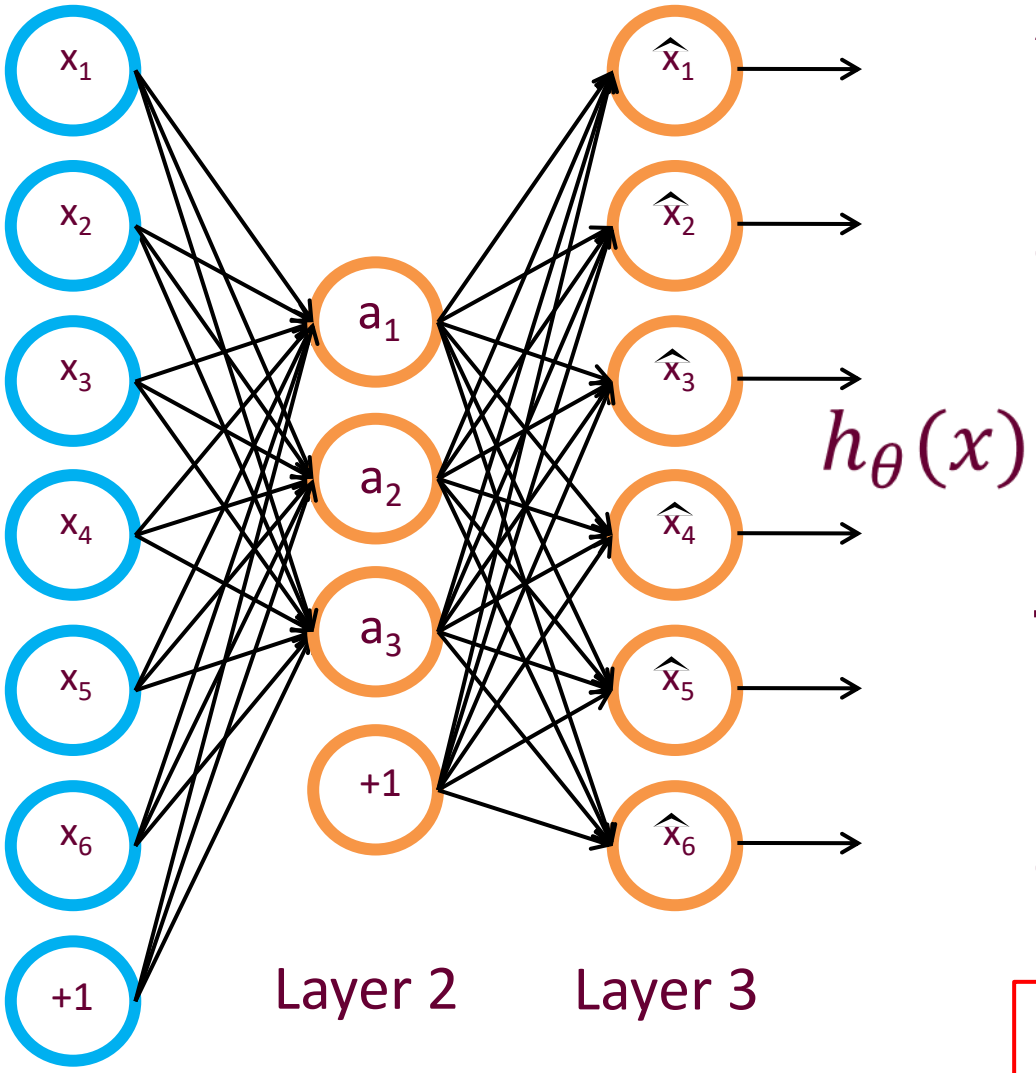
Last Lecture Recap

- Deep Auto-encoder
- Deep Belief Networks





Auto-Encoders



Autoencoder.

Network is trained to output the input (learn identify function).

$$h_\theta(x) \approx x$$

Trivial solution unless:
- Constrain number of units in Layer 2 (learn compressed representation), or
- Constrain Layer 2 to be sparse.

Auto-Encoders

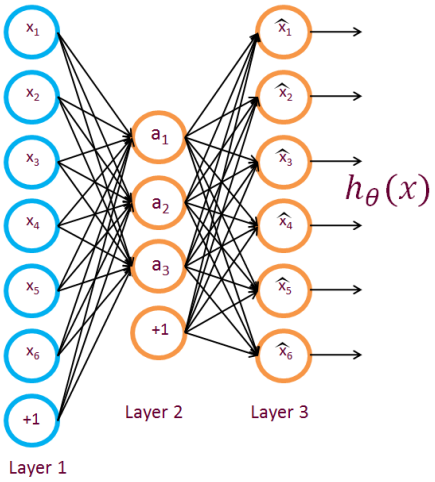
Training a sparse autoencoder.

Given unlabeled training set x_1, x_2, …

$$\min_{\theta} \|h_\theta(x) - x\|^2 + \lambda \sum_i |a_i|$$

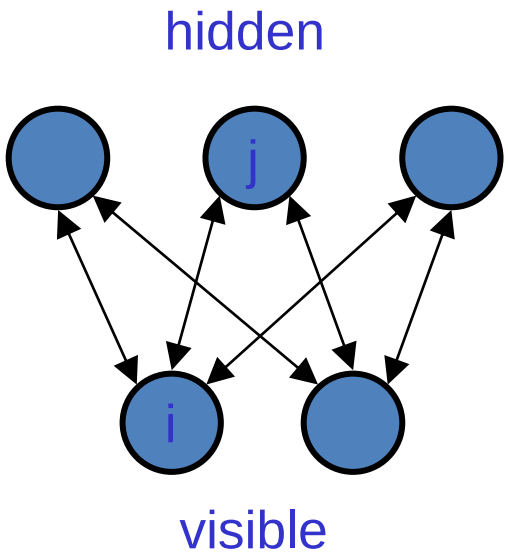
Reconstruction error term

L_1 sparsity term



Restricted Boltzmann Machines (RBM)

- We restrict the connectivity to make learning easier.
  - Only one layer of hidden units.
  - No connections between hidden units.
- In an RBM, the hidden units are conditionally independent given the visible states.
- So we can quickly get an unbiased sample from the posterior distribution when given a data-vector.



RBM: Weights → Energies → Probabilities

- Joint distribution $\mathrm{p}(\mathbf{v}, \mathbf{h}; \theta)$ is defined in terms of an energy function $\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta)$

$$\mathrm{p}(\mathbf{v}, \mathbf{h}; \theta) = \frac{exp\big(-\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta)\big)}{Z}$$

- For a Bernoulli-Bernoulli RBM

$$\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}\, v_i h_j - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} a_j h_j$$

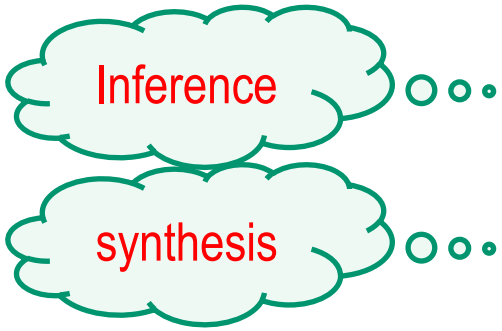
- For a Gaussian-Bernoulli RBM

$$\mathrm{E}(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}\, v_i h_j + \frac{1}{2}\sum_{i=1}^{V} (v_i - b_i)^2 - \sum_{j=1}^{H} a_j h_j$$

- $\mathrm{p}(\mathbf{v}, \mathbf{h}; \theta)$ → generative model!

Restricted Boltzmann Machines (RBM)

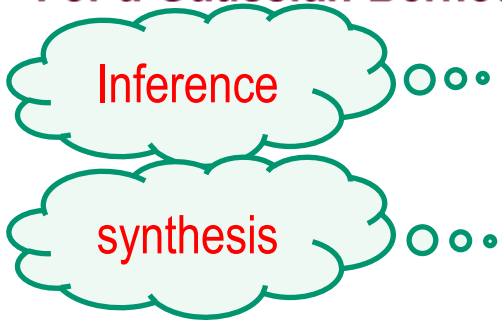
- Conditional probabilities are very easy to calculate
- For a Bernoulli-Bernoulli RBM



$$p(h_j = 1|\mathbf{v};\theta) = \sigma\left(\sum_{i=1}^{V} w_{ij}\, v_i + a_j\right)$$

$$p(v_i = 1|\mathbf{h};\theta) = \sigma\left(\sum_{j=1}^{H} w_{ij}\, h_j + b_i\right)$$

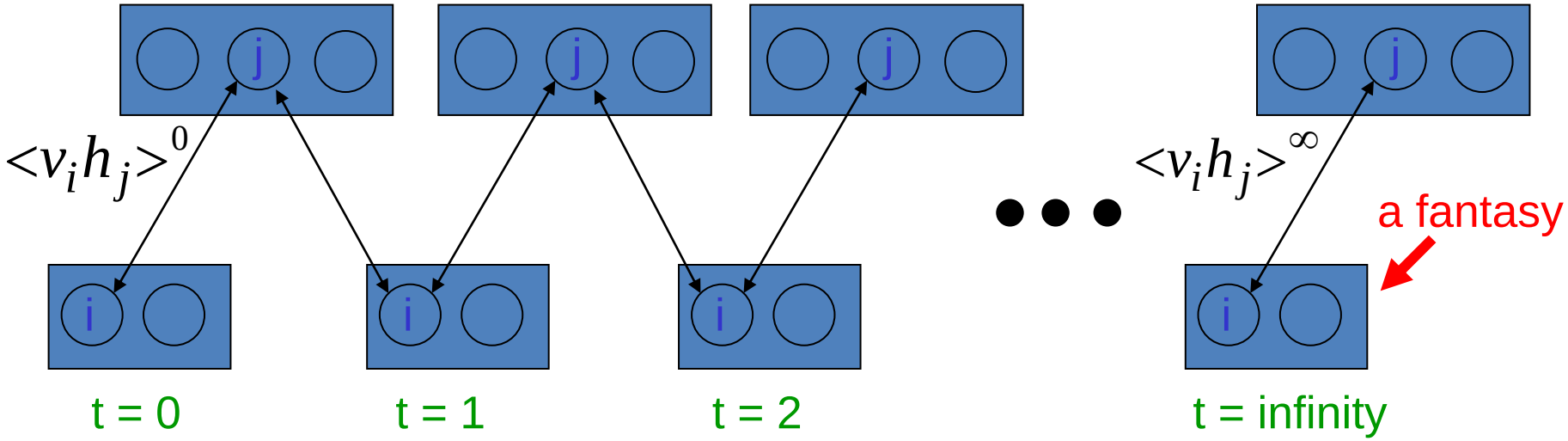
- For a Gaussian-Bernoulli RBM



$$p(h_j = 1|\mathbf{v};\theta) = \sigma\left(\sum_{i=1}^{V} w_{ij}\, v_i + a_j\right)$$

$$p(v_i|\mathbf{h};\theta) = N\left(\sum_{j=1}^{H} w_{ij}\, h_j + b_i, 1\right)$$

Maximum likelihood learning for RBM



Start with a training vector on the visible units.

Then alternate between updating all the hidden units in parallel and updating all the visible units in parallel.

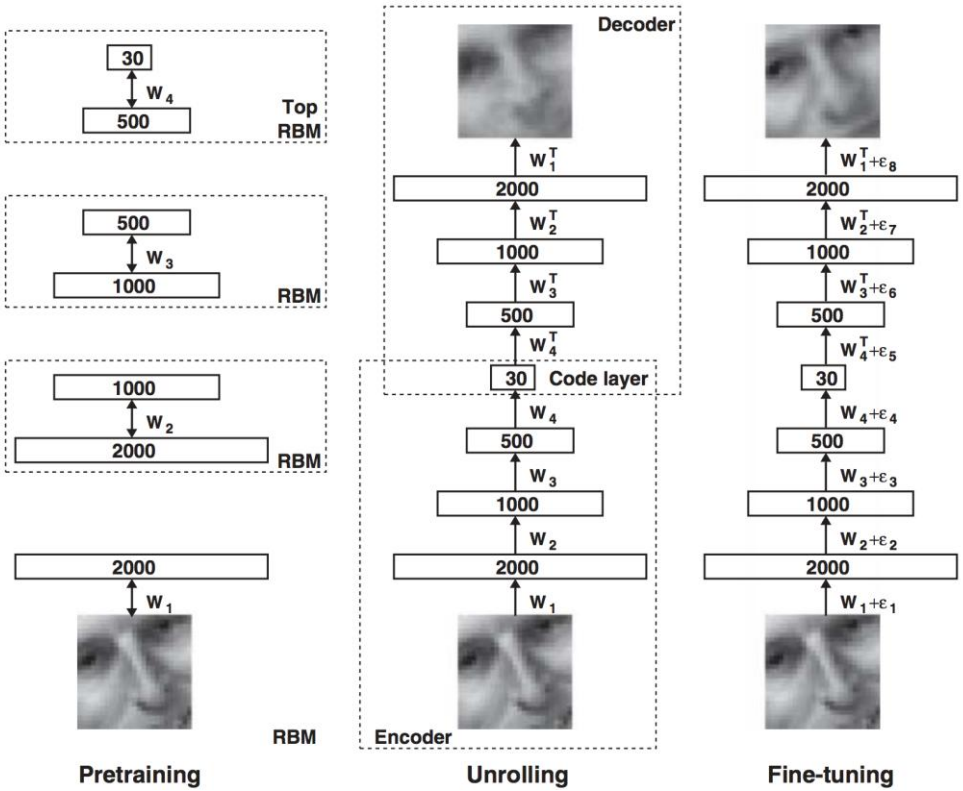
$$\frac{\partial \log p(v)}{\partial w_{ij}} = <v_i h_j>^0 - <v_i h_j>^\infty$$

Training RBMs

- $\Delta w_{ij} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}$
- Approximate $\langle v_i h_j \rangle_{model}$
  i. Initialize $\mathbf{v_0}$ at data
  ii. Sample $\mathbf{h_0} \sim p(\mathbf{h}|\mathbf{v_0})$
  iii. Sample $\mathbf{v_1} \sim p(\mathbf{v}|\mathbf{h_0})$
  iv. Sample $\mathbf{h_1} \sim p(\mathbf{h}|\mathbf{v_1})$
  v. Call $(\mathbf{v_1}, \mathbf{h_1})$ a sample from the model.
- $(\mathbf{v}_\infty, \mathbf{h}_\infty)$ is a true sample from the model. $(\mathbf{v_1}, \mathbf{h_1})$ is a very rough estimate but worked
- Contrastive divergence algorithm (CD)

深度自编码器 - DAE

- Deep Auto-Encoder一般指利用受限玻尔兹曼机(RBM)进行预训练，最后进行参数微调的深度自编码器。
- 该结构最早由Hinton于2006年提出。

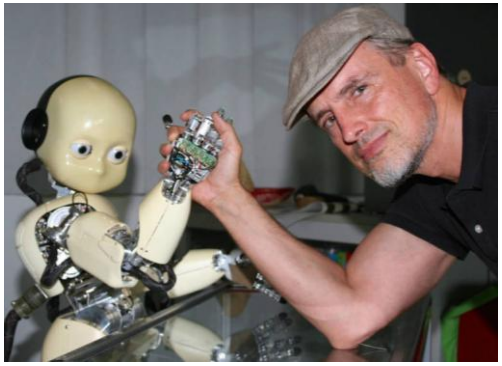


Outline of Lecture Eight

- Deep Auto-encoder
- Deep Belief Networks
- Generative adversarial networks (GAN)
- Self-supervised Learning

History of Generative Adversarial Network

- Unsupervised adversarial networks were proposed by Jürgen Schmidhuber in 1990.
- The basic idea of generative adversarial networks was published in a 2010 blog post by Olli Niemitalo
- The name “GAN” was introduced by Ian Goodfellow et al. in 2014. Their paper popularized the concept and influenced subsequent work.



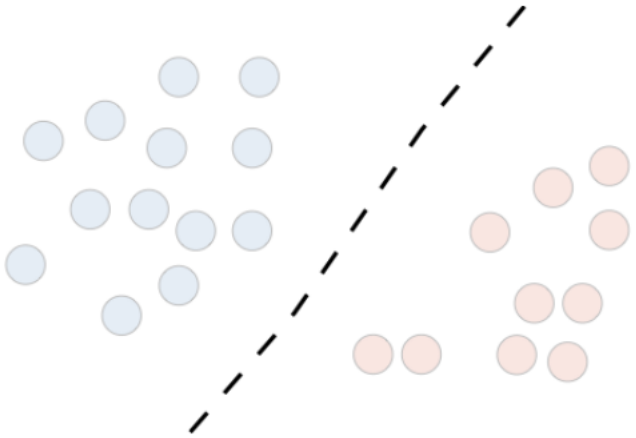
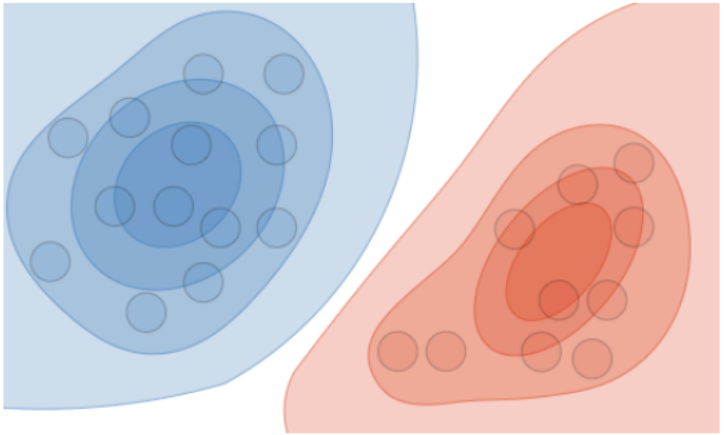
Jürgen Schmidhuber

Ian Goodfellow

Generative Model vs. Discriminative Model

- In machine learning, two main approaches are called the generative approach and the discriminative approach.
- Given an observable variable X and a target variable Y, a generative model is a statistical model of the joint probability distribution on X × Y, P(X,Y). For example, RBM, GAN
- A discriminative model is a model of the conditional probability of the target Y, given an observation X, $P(Y|X)$. For example, MLP, SVM

Generative Model vs. Discriminative Model

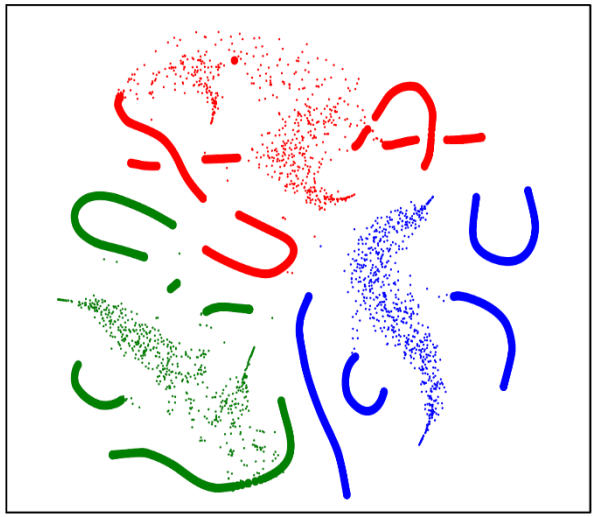
	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

Generative Model

- A generative model is a model for generating all values for a phenomenon, both those that can be observed in the world and "target" variables that can only be computed from those observed.
- Types of generative models are:
  - Gaussian mixture model
  - Hidden Markov model
  - Probabilistic context-free grammar
  - Naive Bayes
  - Averaged one-dependence estimators
  - Latent Dirichlet allocation
  - Restricted Boltzmann machine
  - Generative adversarial networks

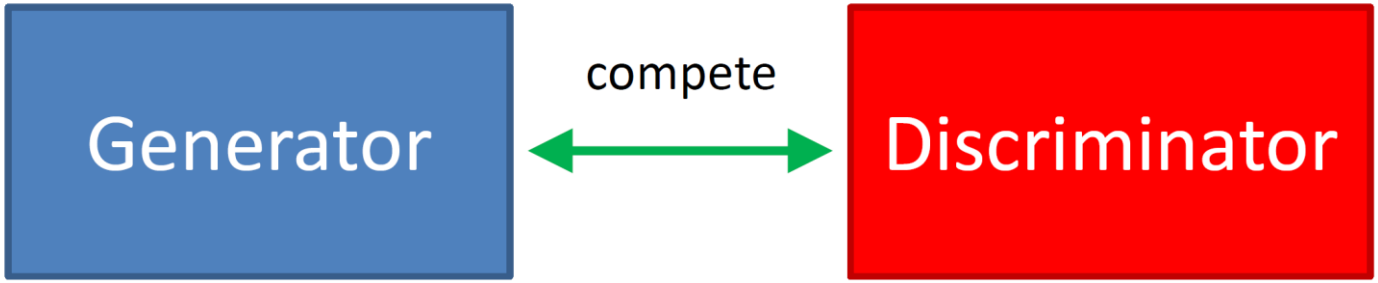
Generative Adversarial Network

- Generative adversarial networks (GANs) were introduced by Ian Goodfellow et al. in 2014. GANs are unsupervised deep generative networks, which are implemented by a system of two neural networks contesting with each other in a zero-sum game framework.
- GANs are used to generate realistic-like signals such as images, voice, text, EEG and so on.



GANs Model

- Normal GANs consist of two components: generator and discriminator.
- Generator G: creates (fake) samples that the discriminator cannot distinguish.
- Discriminator D: determine whether samples are fake or real.



Generator

- **Generator:**

  A differentiable function which is modeled as a neural network.

- **Input:**

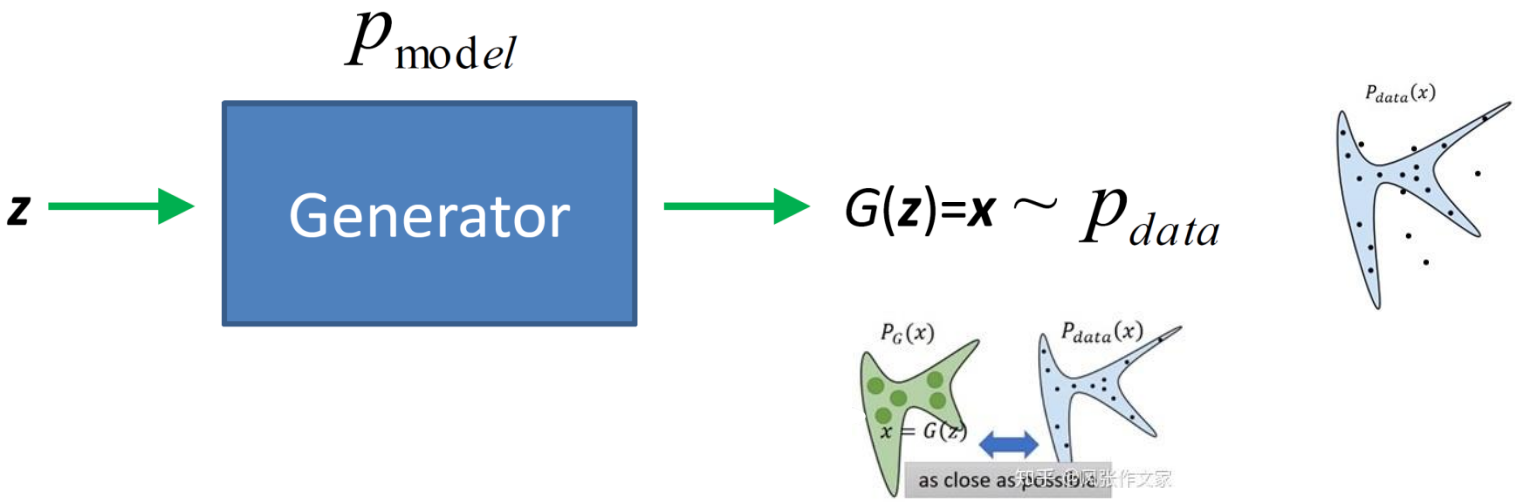
  z: Random noise vector from some simple prior distribution.

- **Output:**

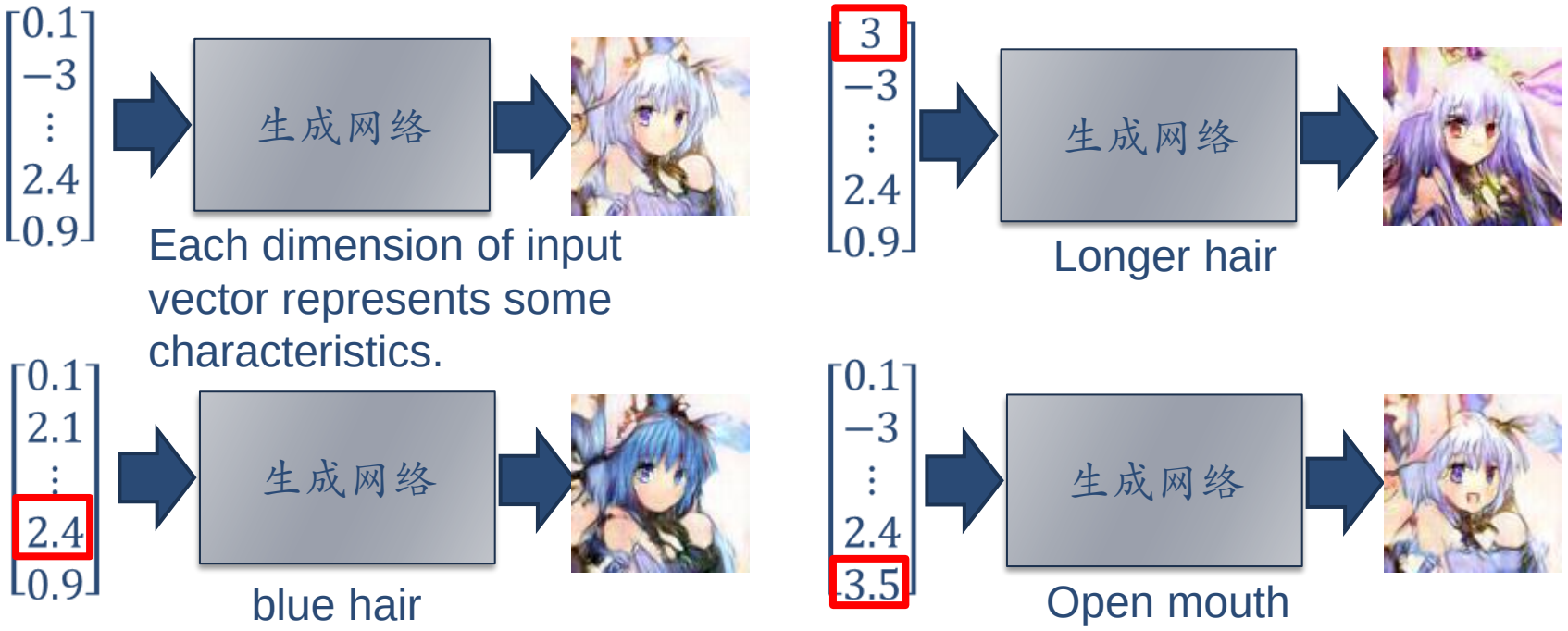
  x = G(z): Generated samples.

Generator

- Generator learns the underlying distribution of real data and the generated data G(z) confuse discriminator.
- The dimension of z should be at least as large as that of x.



Generator Examples

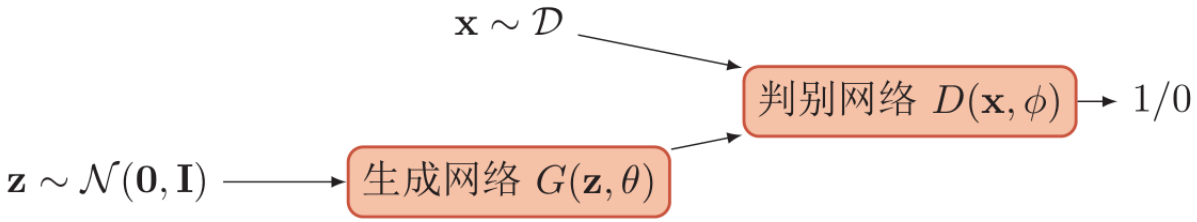


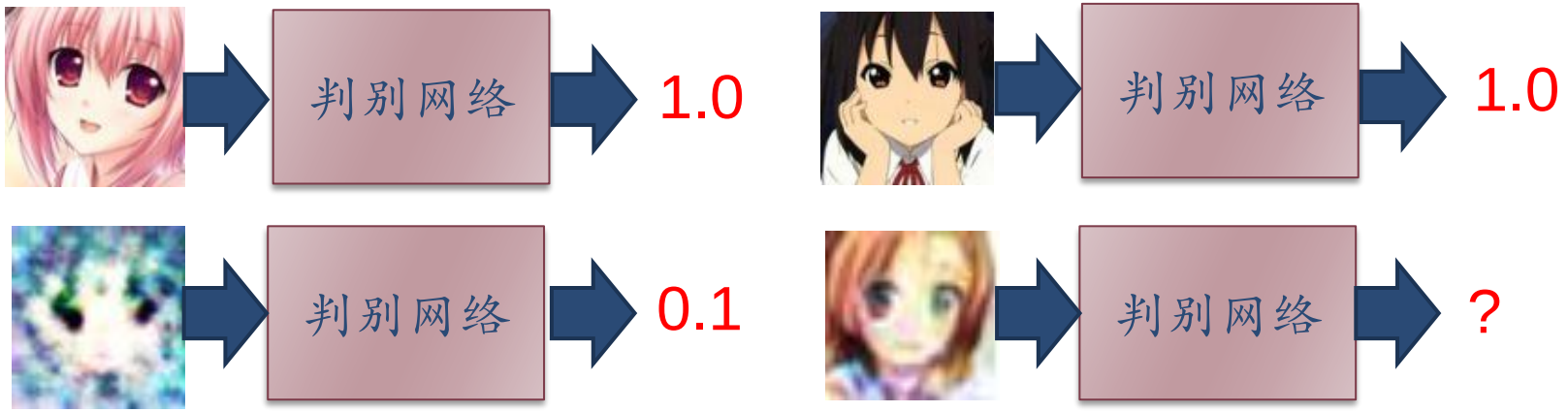
Discriminator

- Discriminator is also modeled as a neural network.
- Input: Real sample and generated sample x.
- Output:
- –1 for real samples
- –0 for fake samples
- Discriminator tries to distinguish whether a sample comes from real distribution or generated distribution.

Discriminator

- 判别网络的输入则为真实样本或生成网络的输出，其目的是将生成网络的输出从真实样本中尽可能分辨出来。





MinMax Game

- 对抗训练
  - 生成网络要尽可能地欺骗判别网络。
  - 判别网络将生成网络生成的样本与真实样本中尽可能区分出来。

- 两个网络相互对抗、不断调整参数，最终目的是使判别网络无法判断生成网络的输出结果是否真实。

MinMax Game

生成网络
(student)

判别网络
(teacher)

生成网络
v1

No eyes

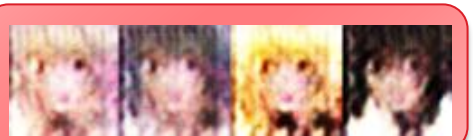
判别网络
v1

生成网络
v2

No mouth

判别网络
v2

生成网络
v3



GANs Formulation

- The adversarial training is formulated as a minimax game, where:
  - The discriminator is trying to maximize its reward V(D, G)
  - The generator is trying to minimize Discriminator's reward

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- Maximize discriminator:

$$\max_D V(D,G) = E_{x\sim p_{data}\,(x)}[log(D(x))] + E_{z\sim p_z\,(z)}[log(1 - D(G(z)))]$$

- Minimize generator

$$\min_G V(D,G) = E_{z\sim p_z\,(z)}[log(1 - D(G(z)))]$$

Zero-sum Game (零和博弈)

- In game theory and economic theory, a zero-sum game is a mathematical representation of a situation in which each participant's gain or loss of utility is exactly balanced by the losses or gains of the utility of the other participants. If the total gains of the participants are added up and the total losses are subtracted, they will sum to zero.
- Thus, cutting a cake, where taking a larger piece reduces the amount of cake available for others, is a zero-sum game if all participants value each unit of cake equally.
- In psychology, zero-sum thinking refers to the perception that a situation is like a zero-sum game, where one person's gain is another's loss.

Training Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

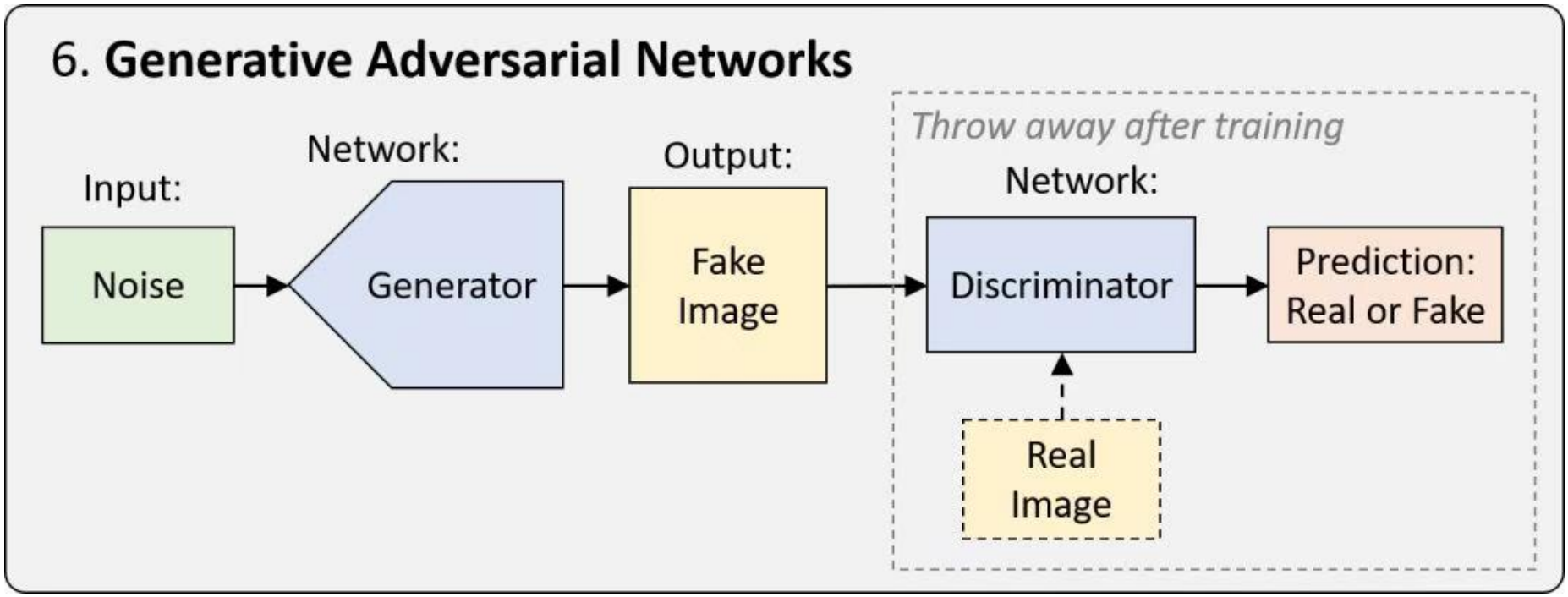
end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Training Algorithm (2)

- The Algorithm takes *k* steps to optimize for the Discriminative net, D, and then one step of optimizing G, the generative net based on the outcome from D after above *k* steps.
- While optimizing for D, we work on the entire value function. For G, we only use the second term of the value function, as the gradient w.r.t. for the first term is zero.
- It is important to note that we ascend the gradient when optimizing for D, as it is a maximization problem, and descend the gradient when optimizing for G as it is a minimization problem.
- Initially when G is far from optimal, the gradients in second optimization might be very small. Instead, we can ascend the gradient for log(D(G(z))) initially.

Generative Adversarial Network



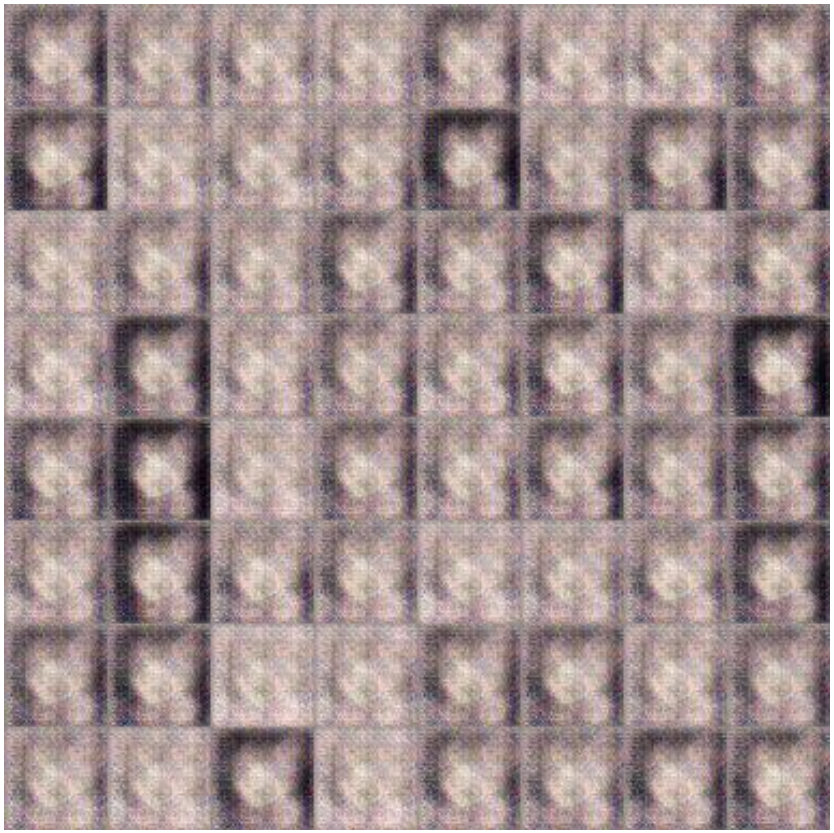
Advantages of GANs

- **Plenty of existing work on Deep Generative Models**
  - **Boltzmann Machine**
  - **Deep Belief Nets**
  - **Variational AutoEncoders (VAE) (变分自编码器）**
  - **Stable Diffusion Model**
- **Advantages of 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.**

Problems of GANs

- Non-convergence: It's difficult to converge for GANs.
- Finding equilibrium is a game of two players
- Exploiting convexity in function space, GAN training is theoretically guaranteed to converge if we can modify the density functions directly, but:
  - Instead, we modify G (sample generation function) and D (density ratio), not densities.
  - We represent G and D as highly non-convex parametric functions.
- 'Oscillation': can train for a very long time, generating very many different categories of samples, without clearly generating better samples.

Anime Face Generation



100 updates

1000 updates

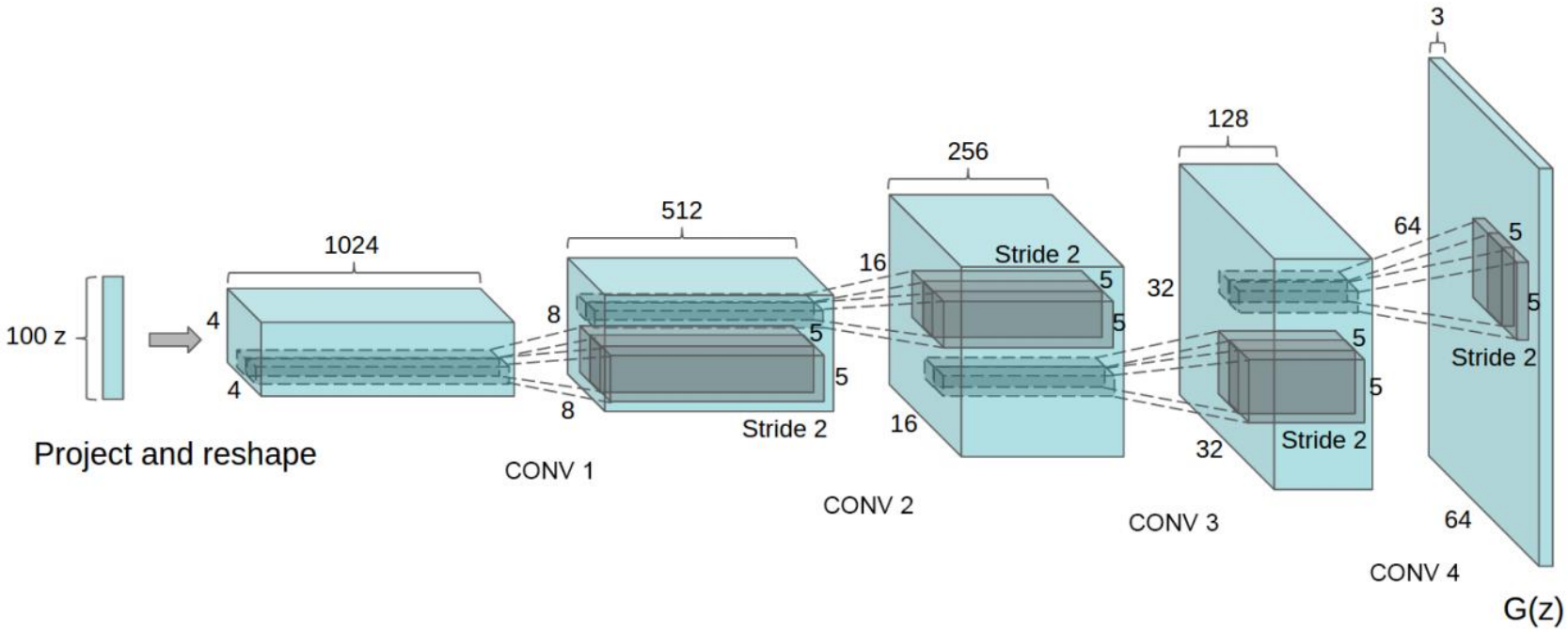
Anime Face Generation

10,000 updates

50,000 updates

DCGANs

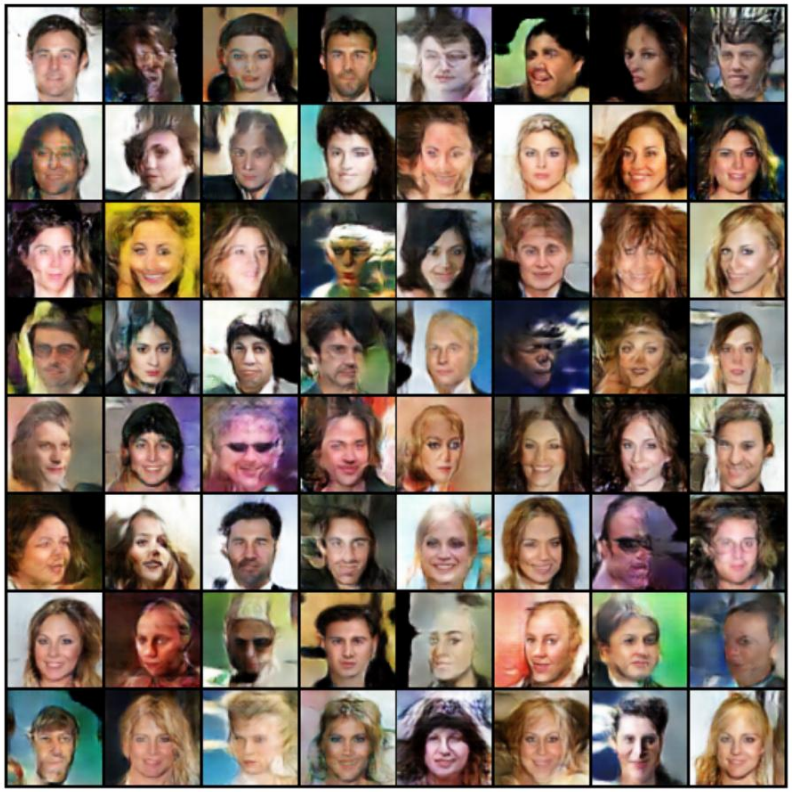
- 判别网络是一个传统的深度卷积网络，但使用了带步长的卷积来实现下采样操作，不用最大池化（pooling）操作。
- 生成网络使用一个特殊的深度卷积网络来实现使用微步卷积来生成64×63大小的图像。



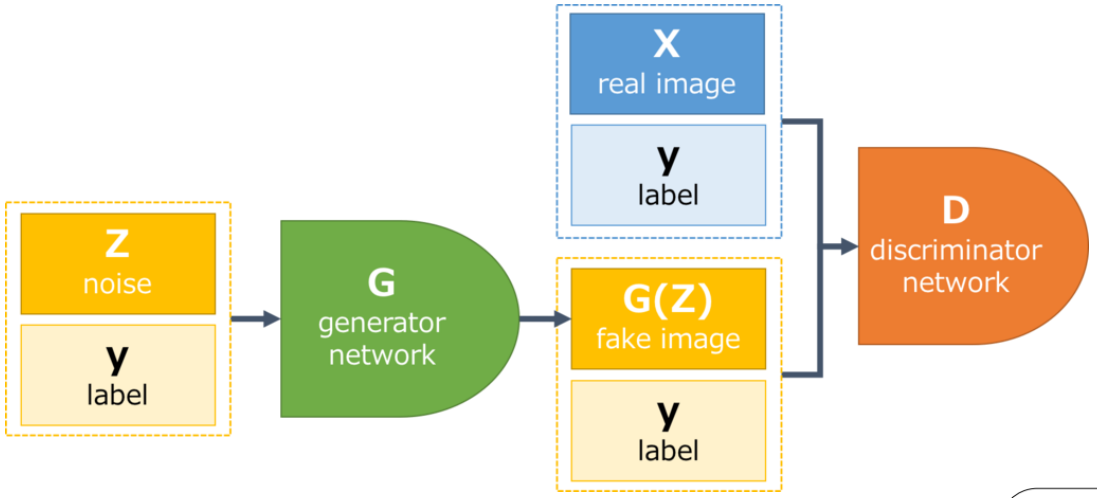
DCGANs

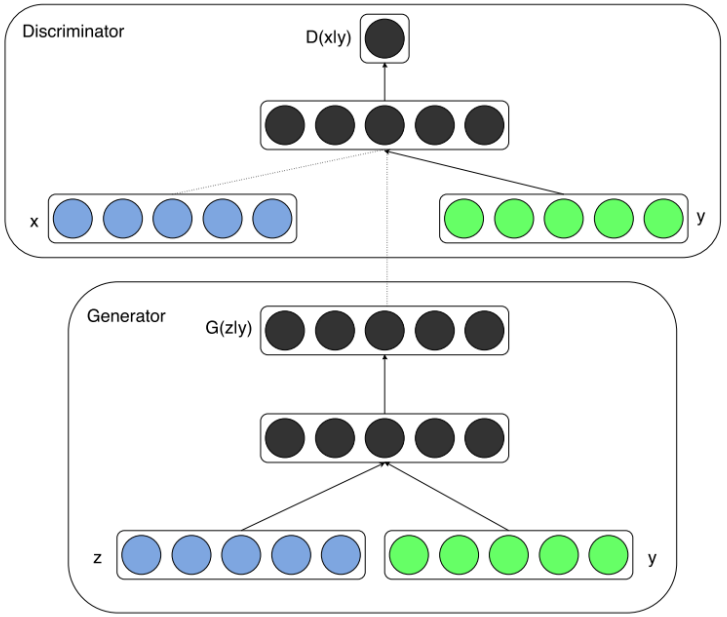
Real Images

Fake Images

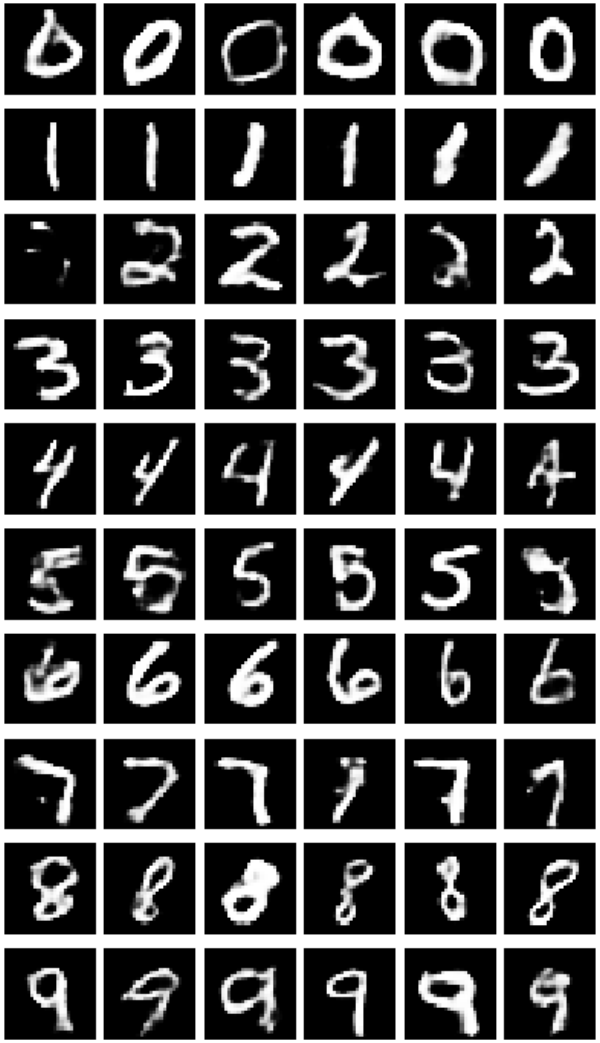


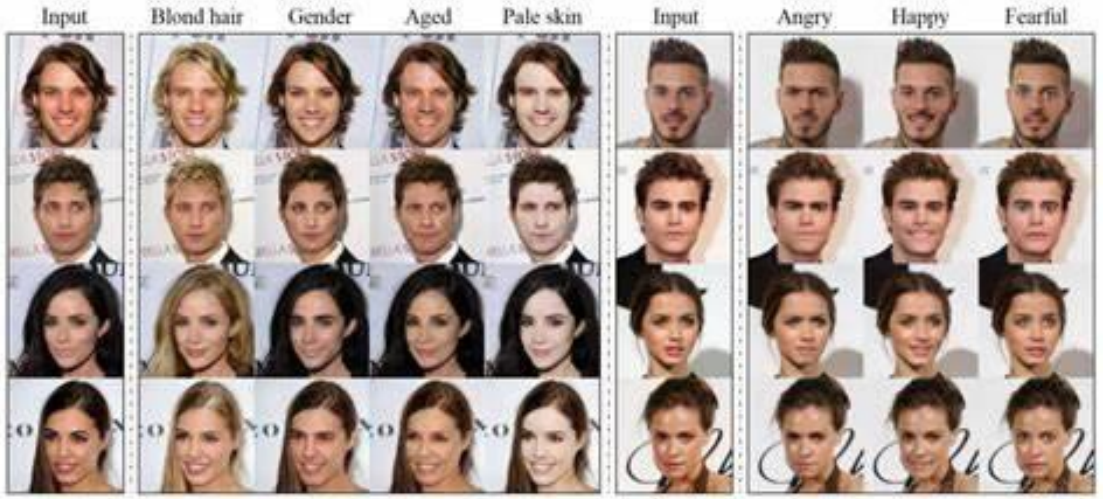
Conditional GAN



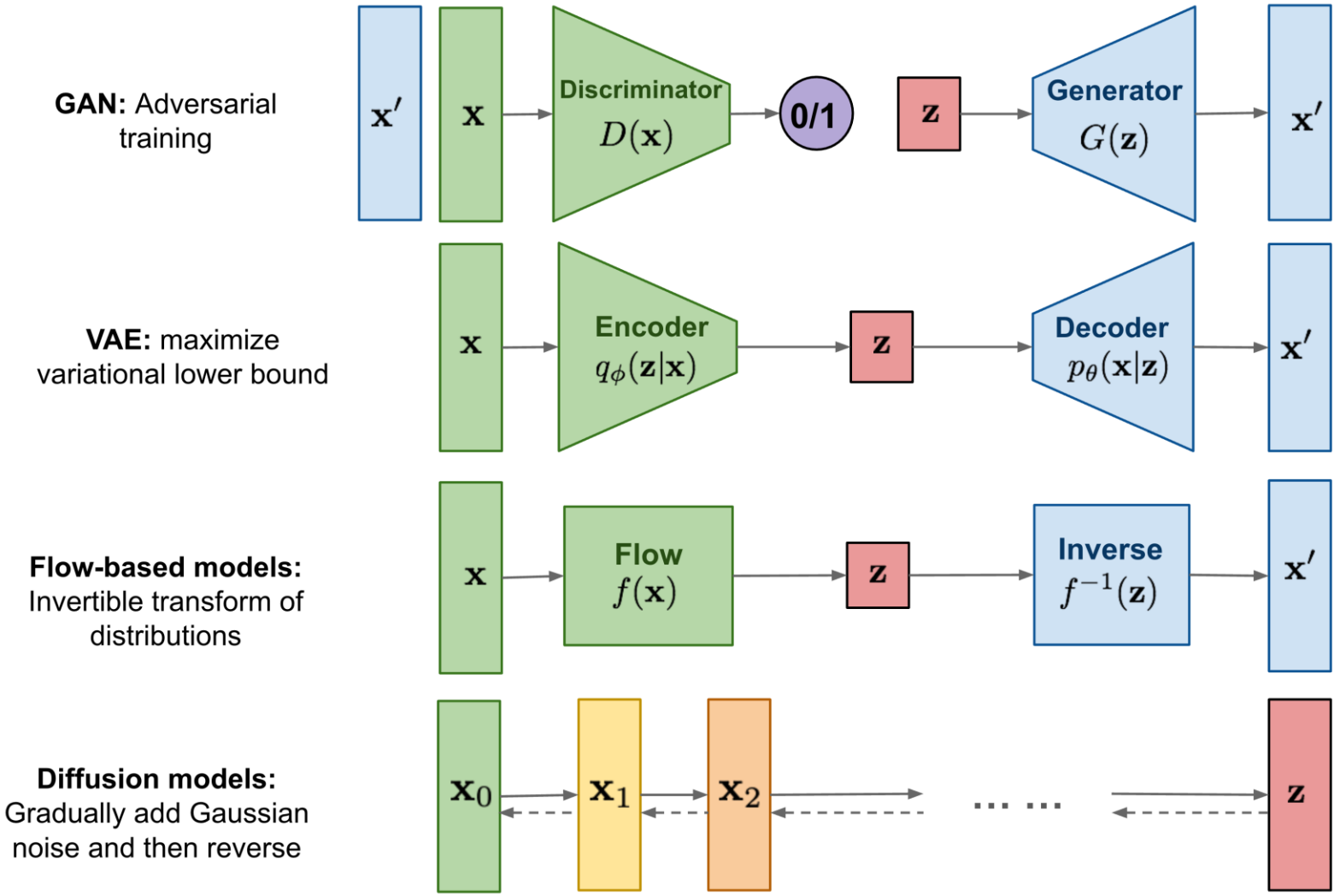


Conditional GAN



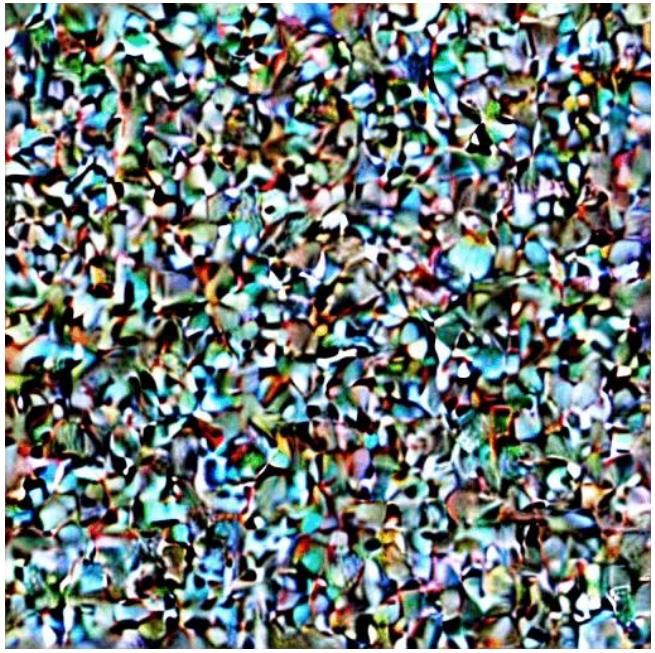


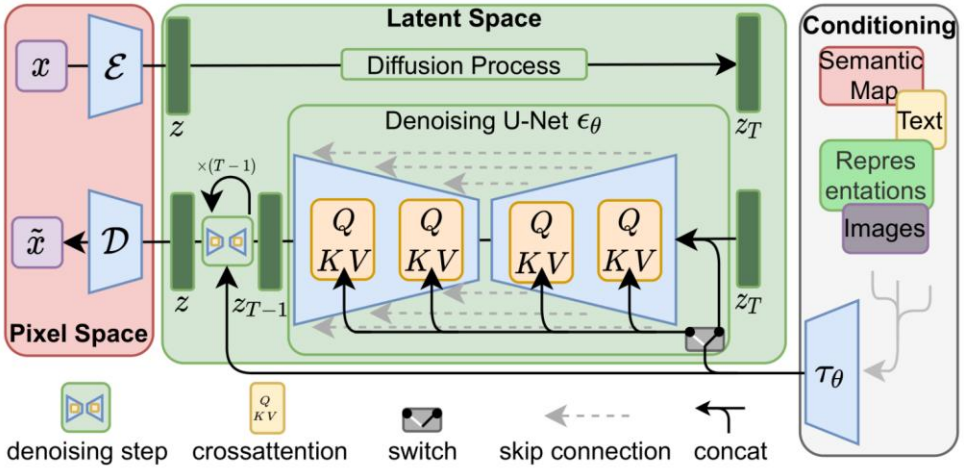
Comparing Generative Models



https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Stable Diffusion Model





https://www.reddit.com/r/StableDiffusion/comments/xcjj7u/sd_img2img_after_effects_i_generated_2_images_and/

Some Resources

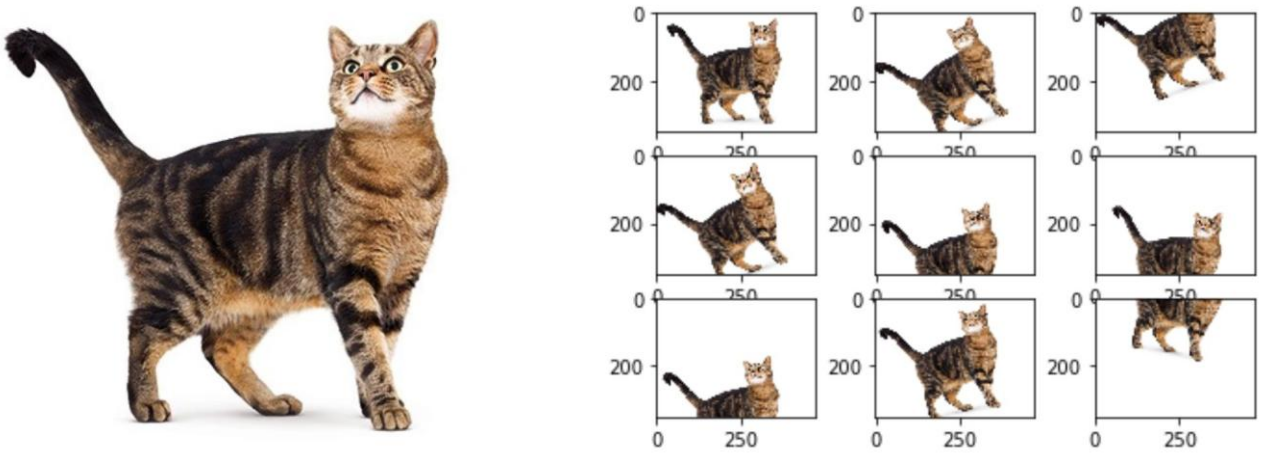
- **Diffusion model in general**
  - What are Diffusion Models? | Lil'Log
  - Generative Modeling by Estimating Gradients of the Data Distribution | Yang Song
- **Stable diffusion**
  - **Annotated & simplified code:** U-Net for Stable Diffusion (labml.ai)
  - **Illustrations:** The Illustrated Stable Diffusion – Jay Alammar

视频生成模型Sora

Prompt: A flock of paper airplanes flutters through a dense jungle, weaving around trees as if they were migrating birds.

Data Augmentation

- It is common to generate artificial images through geometric transformation (translation, rotation, scaling, horizontal shearing) in the field of computer vision.
- Similar with CV, geometric transformation methods have been applied to generate EEG signals.



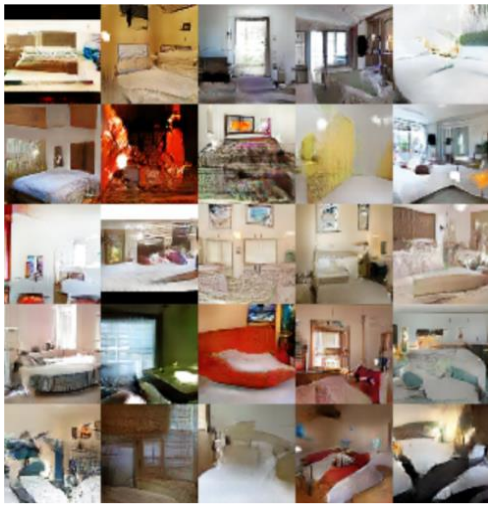
Enlarge dataset by translation

Data Augmentation

- These data augmentation methods only lead to a signal-level transformation through distortions, which is not helpful for dividing a clear boundary between data manifolds.
- Recently, GANs have revealed their potential in generating realistic-like data such as images by adopting an adversarial training.
- In the filed of computer vision, researchers have adopted GANs-based methods for data augmentation and made significant progresses.

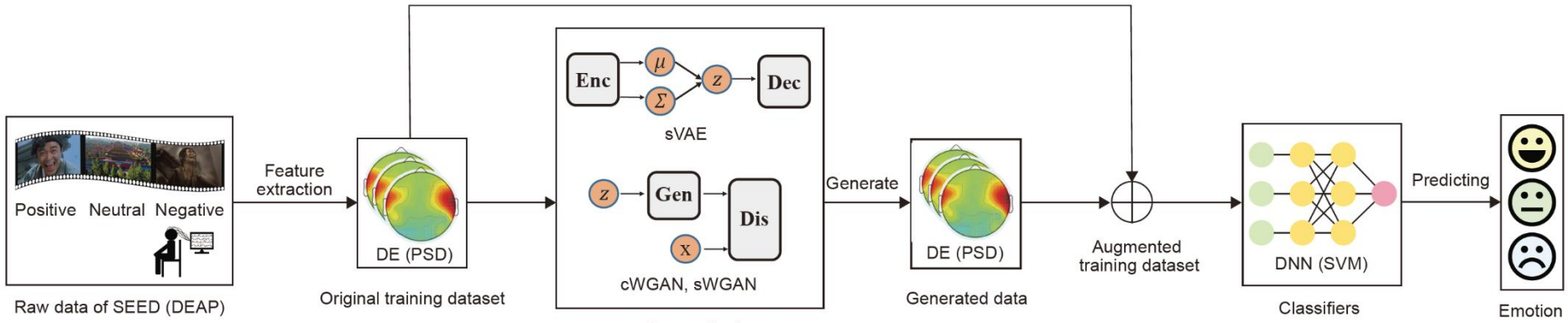
Data Augmentation and GANs

- Unlike geometric transformation, GANs capture the latent distribution of the data by unsupervised learning.
- Data generated by GANs enlarge the data manifold, which leads to a better margin for the classifier.



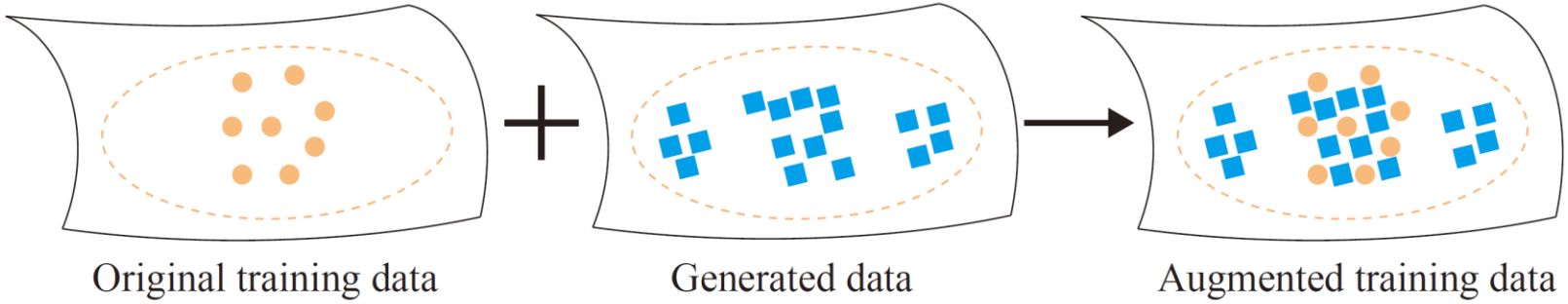
Pictures generated by GANs

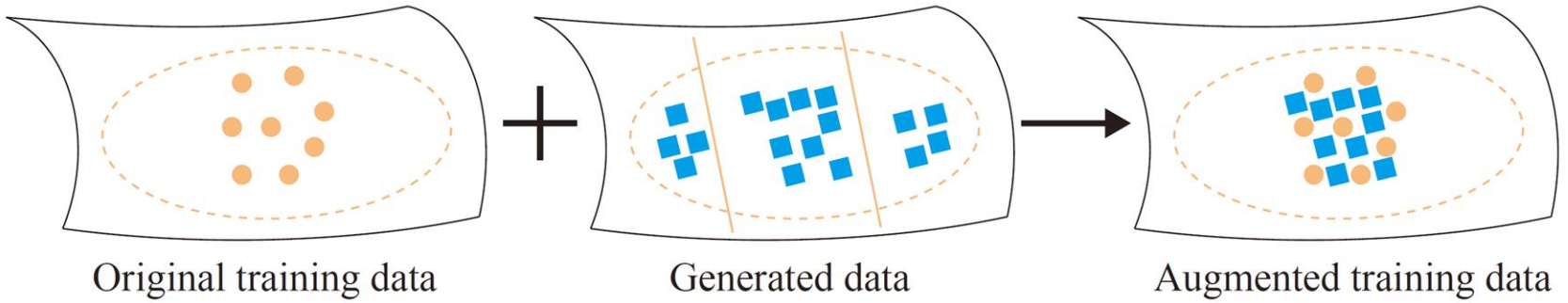
利用生成式模型自动生成脑电数据



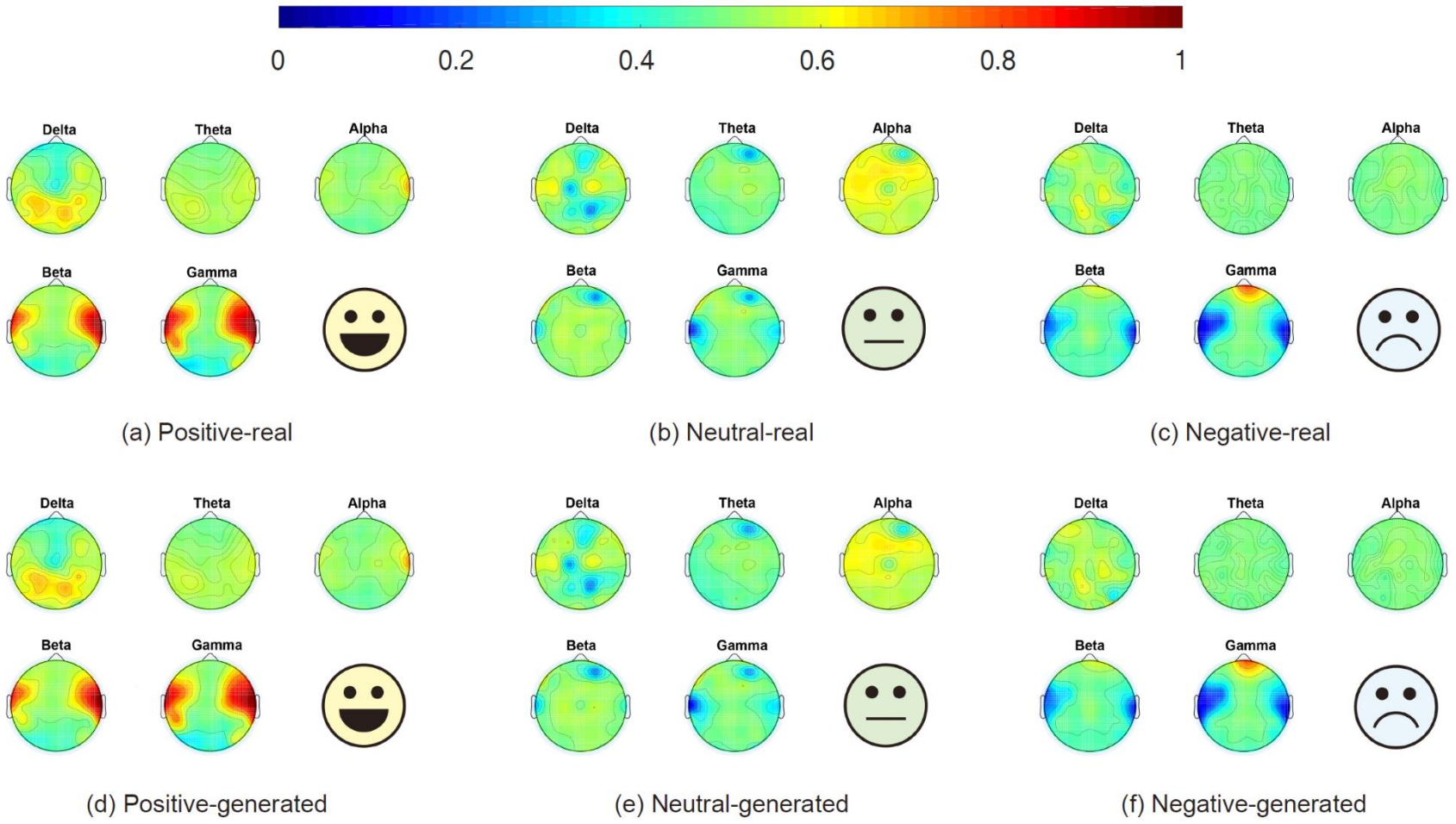
Yun Luo, Li-Zhen Zhu, Zi-Yu Wan, Bao-Liang Lu, Data Augmentation for Enhancing EEG-based Emotion Recognition with Deep Generative Models, ***Journal of Neural Engineering***，2020

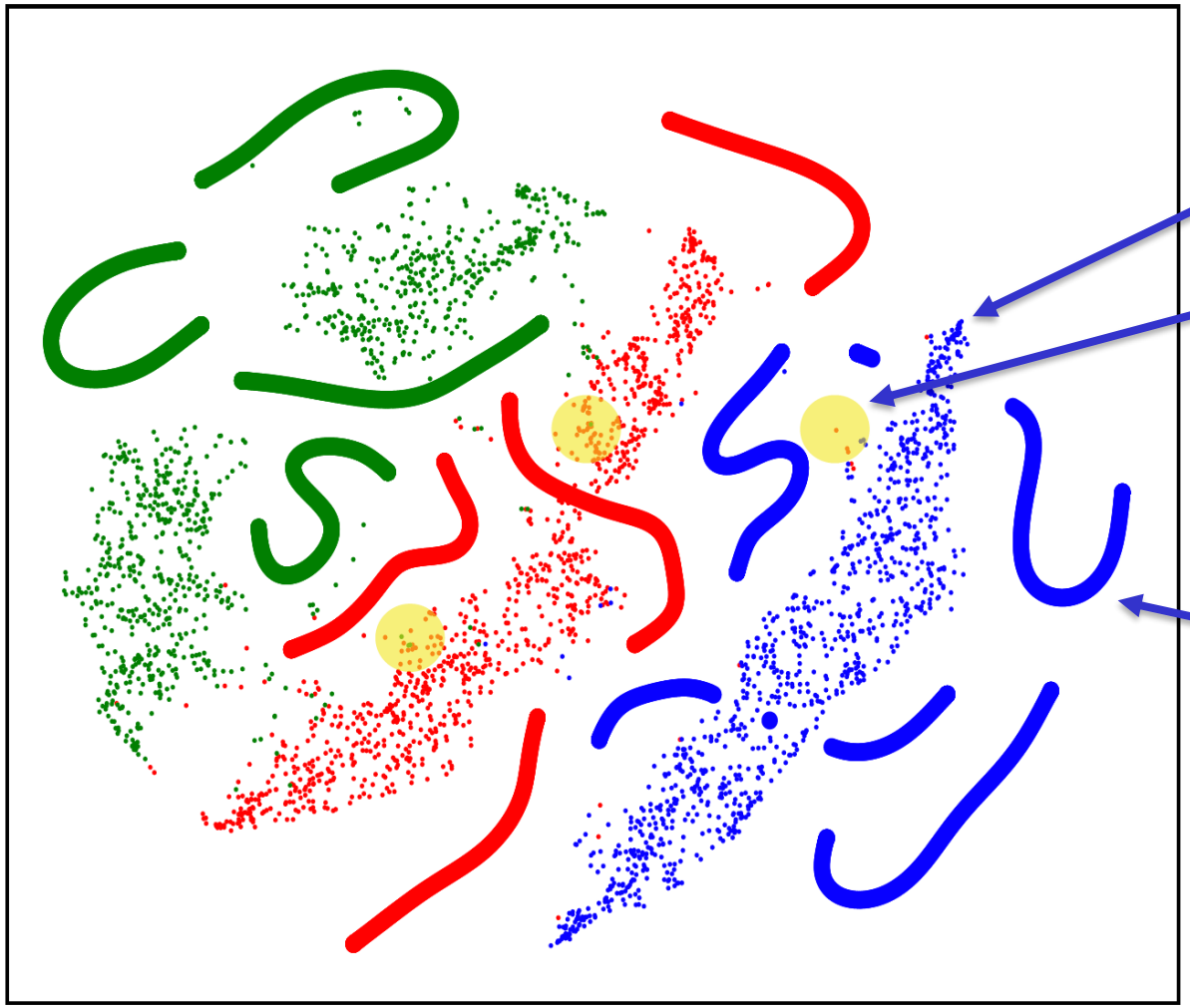
两种数据扩增策略：全部使用与部分使用





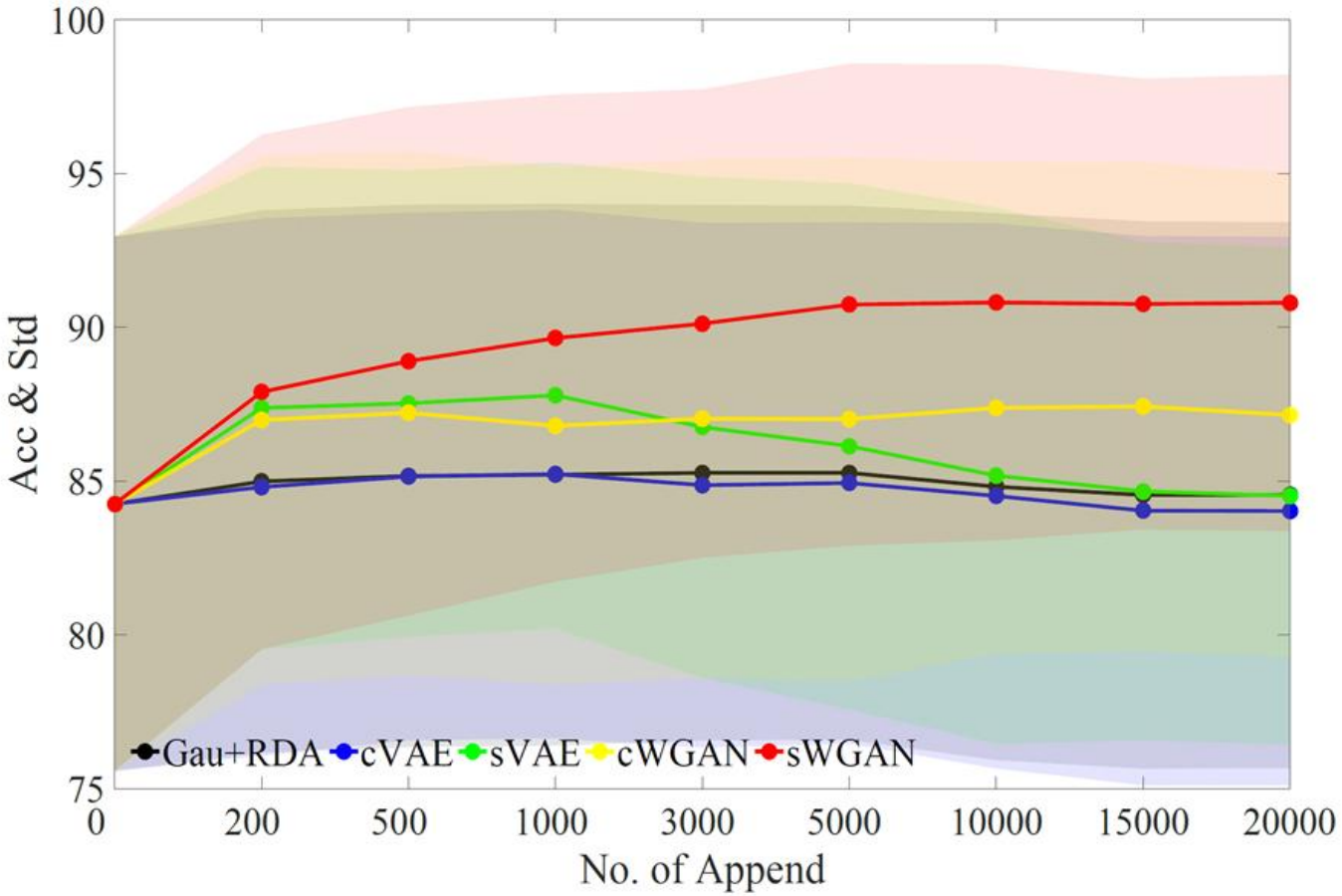
真实数据与生成数据的微分熵特征脑壳图





真实数据和生成数据的二维表示

数据增强的数量与模型性能提升的关系



- 增加的数据在10倍以内会达到性能提升的峰值；
- 对SEED和DEAP数据集，分别取得了10.2%和5.4%的识别准确率的提升

References

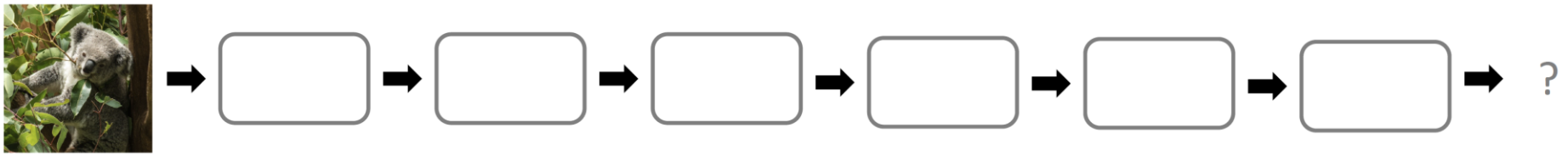
1. Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets, International Conference on Neural Information Processing Systems. MIT Press, 2014:2672-2680.
2. Radford A, Metz L, Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. Computer Science, 2015.
3. Mirza M, Osindero S. Conditional Generative Adversarial Nets. Computer Science, 2014:2672-2680.
4. Ledig C, Wang Z, Shi W, et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. 2016:105-114.
5. Zhang H, Xu T, Li H, et al. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. 2016:5908-5916.
6. Antipov G, Baccouche M, Dugelay J L. Face Aging With Conditional Generative Adversarial Networks. 2017.
7. Liu M Y, Tuzel O. Coupled Generative Adversarial Networks. 2016.

Outline of Lecture Eight

- Deep Auto-encoder
- Deep Belief Networks
- Generative adversarial networks (GAN)
- **Self-supervised Learning**

How to Represent Images w/ Deep Learning?

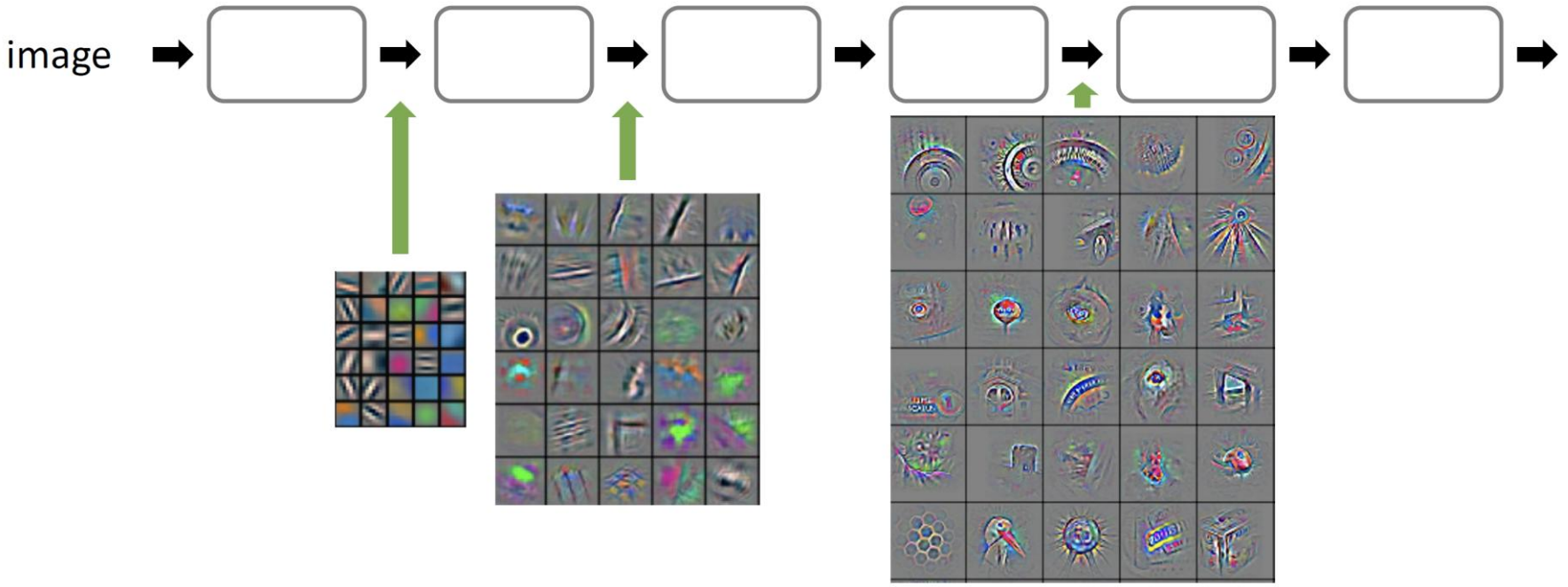
general modules (instead of specialized features)



compose simple modules into complex functions

- build multiple levels of abstractions
- learn by back-prop
- learn from data
- reduce domain knowledge and feature engineering

Multiple Levels of Representations



Deeper layers have “higher-level” features.

Deep Representations are Transferrable!

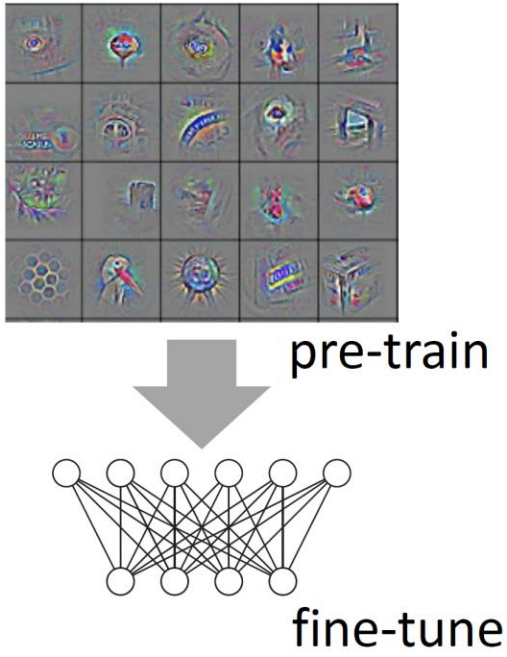
The single most important discovery in DL revolution

Transfer learning:

- pre-train on large-scale data
- fine-tune on small-scale data

- enable DL for small datasets
- revolutionize computer vision

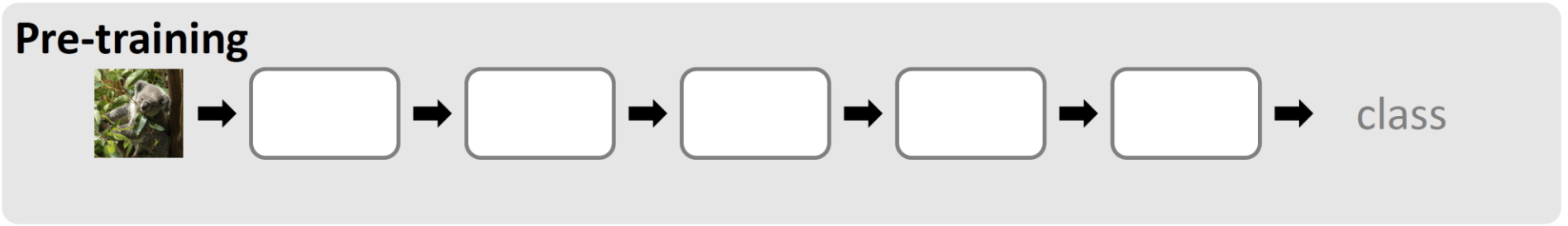
- data: engine for general representation
- GPT: a similar principle



"DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition", Donahue et al. arXiv 2013
"Visualizing and Understanding Convolutional Networks", Zeiler & Fergus. arXiv 2013
"CNN Features off-the-shelf: an Astounding Baseline for Recognition", Razavian. arXiv 2014

Transfer learning

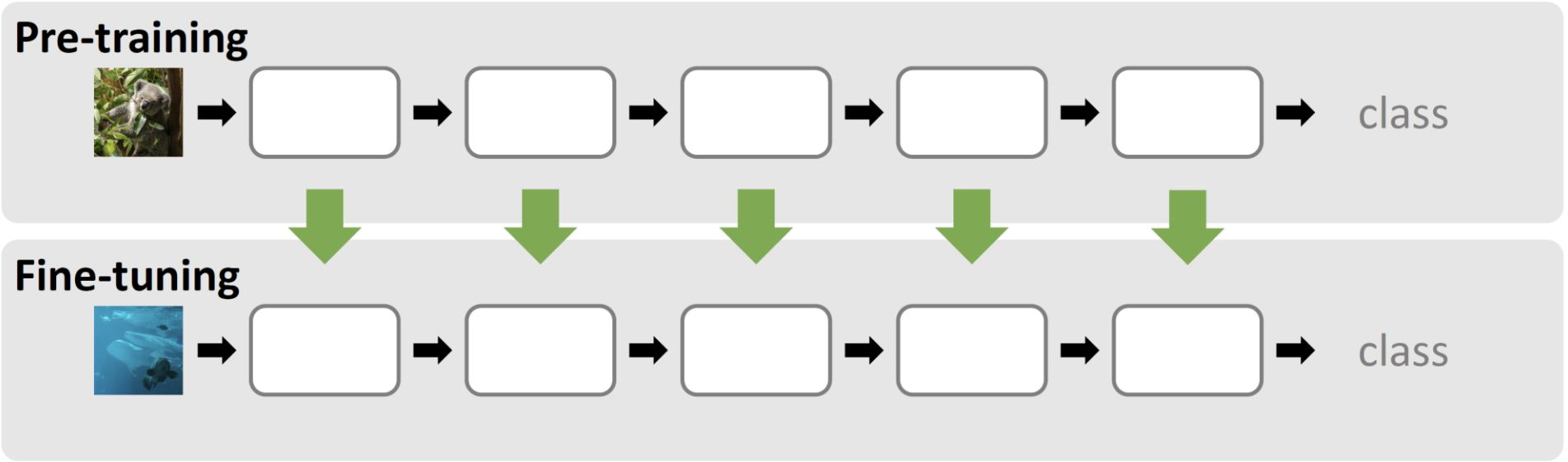
Transfer learning: Pre-training & Fine-tuning



Pre-training:

- to learn **general** representations
- on **large-scale** data
- train for a **long** time
- with **large** models

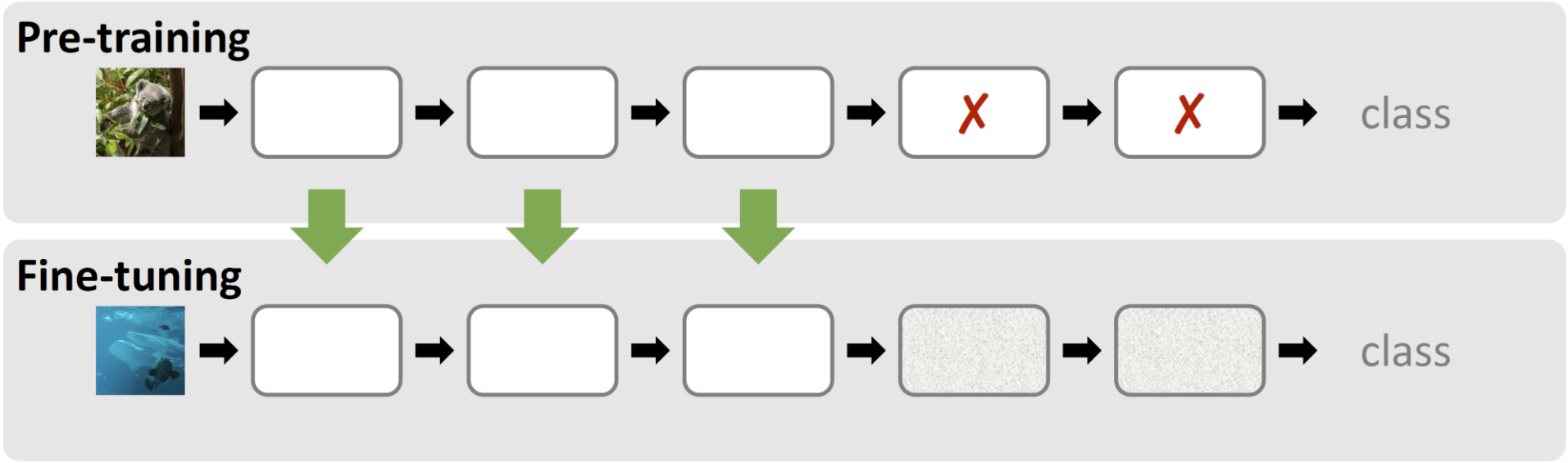
Transfer learning: Pre-training & Fine-tuning



Fine-tuning:

- transfer weights to **specific** tasks
- on **small-scale** data
- train for a **short** time, **lower** learning rate
- enable **large** models with lower risk of overfitting

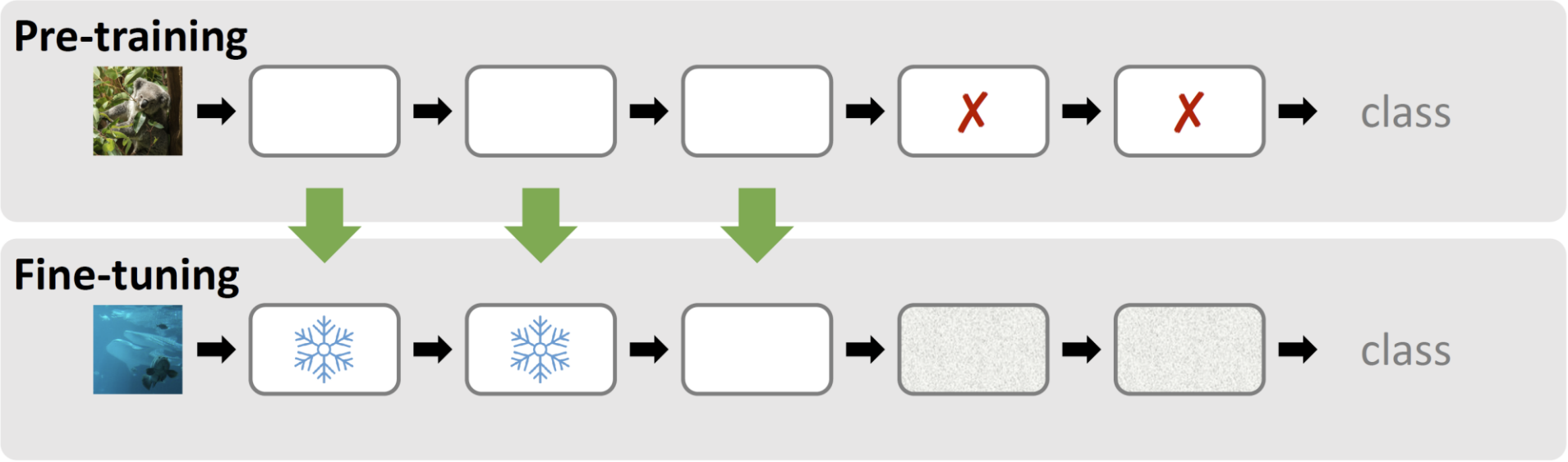
Transfer learning: Pre-training & Fine-tuning



Partial transfer

- pre-train and target domains may differ
- highest-level features are too adapted to pre-training
- randomly initialize new layers

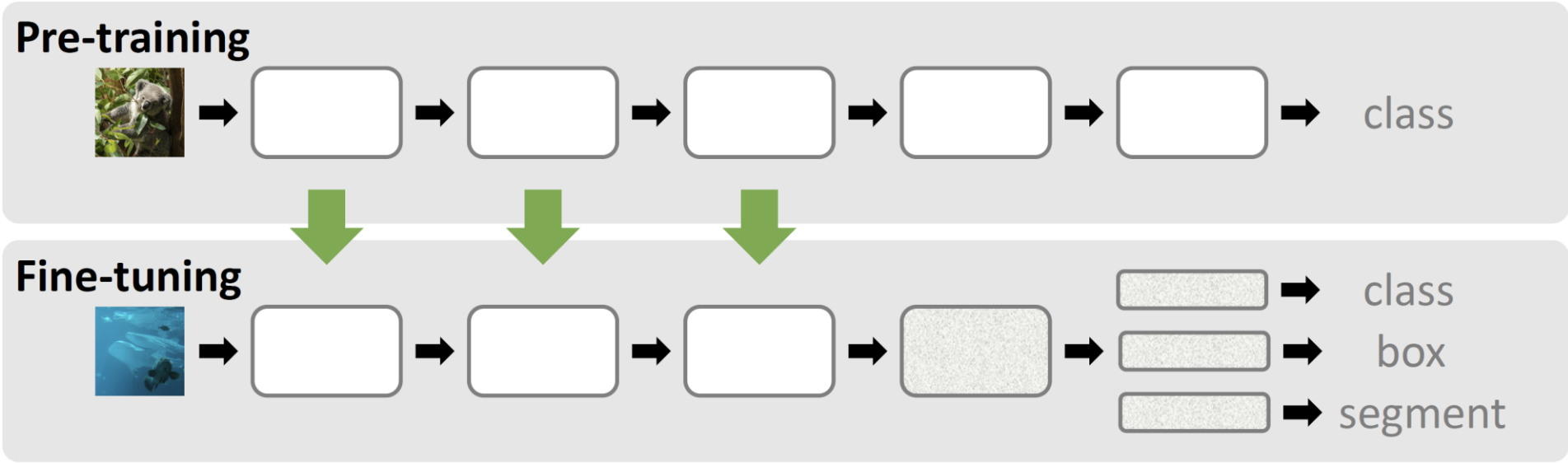
Transfer learning: Pre-training & Fine-tuning



Frozen weights

- freeze some/all pre-trained weights
- reduce overfitting if data is too little
- save memory, speed up training

Transfer learning: Pre-training & Fine-tuning



Network surgery

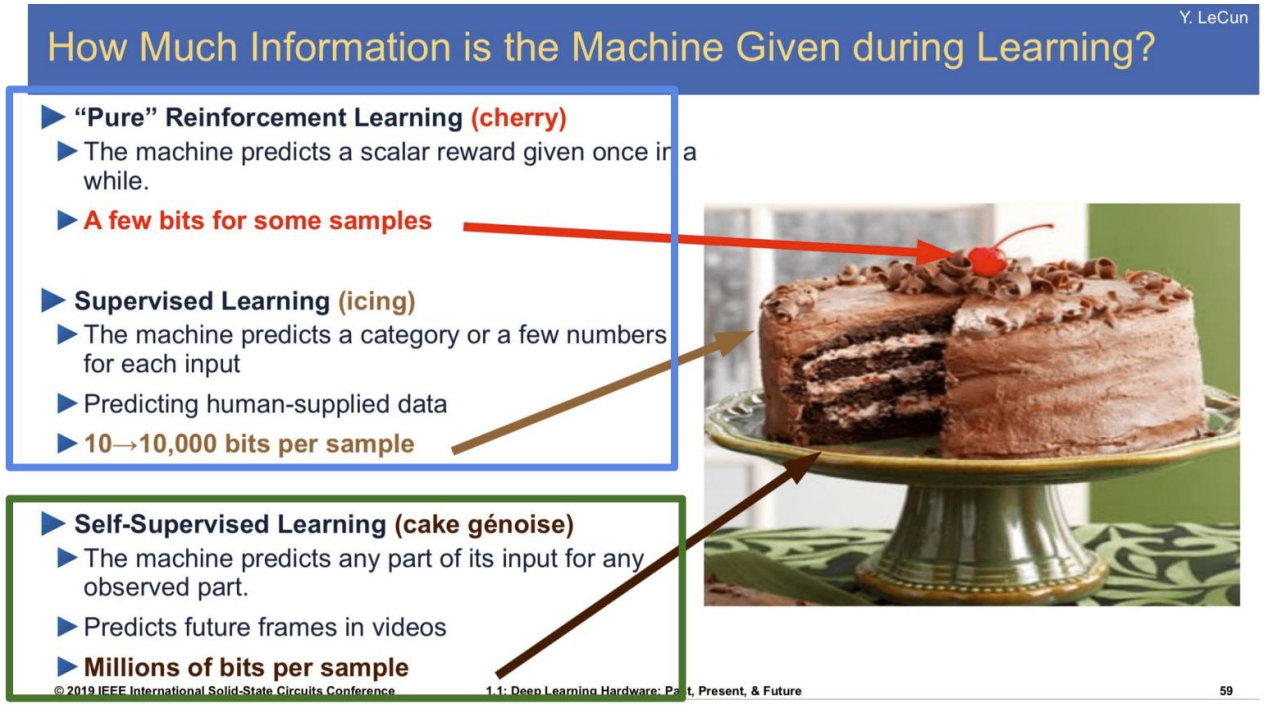
- re-purpose the model for other tasks (detect, segment)
- general features + task-specific predictions

“The Cake of Learning”

downstream tasks

feature extractor

Learn good features through self-supervision

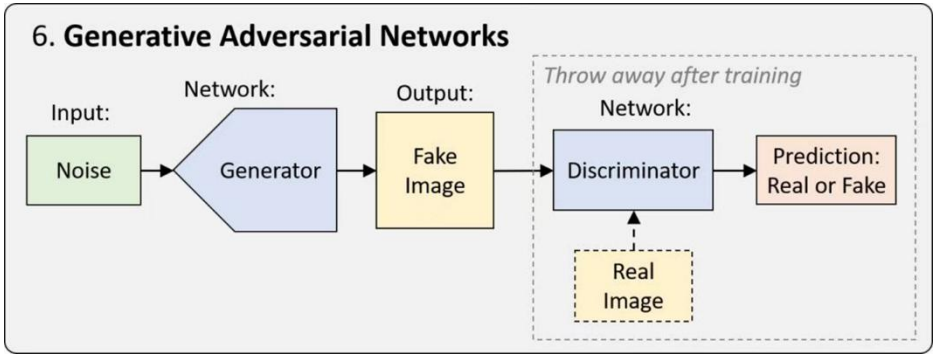


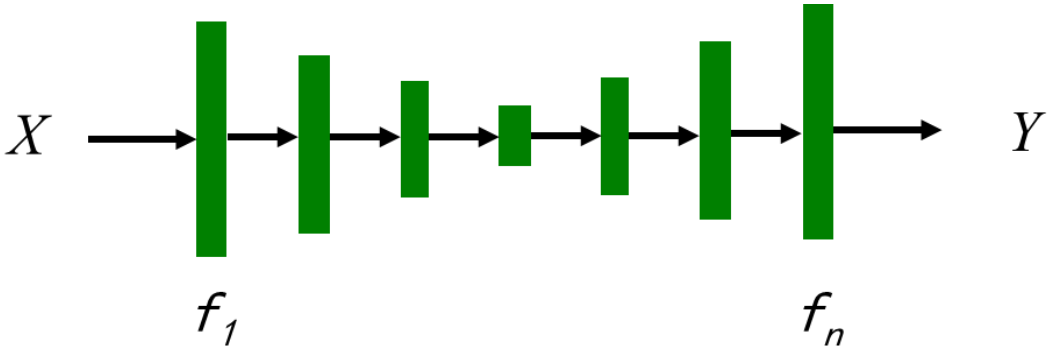
Source: Lecun 2019 Kevnote at ISSCC

Two Important Deep Learning Models

- GAN
- Deep Auto-encoder

} Unsupervised Learning





Self-supervised Learning

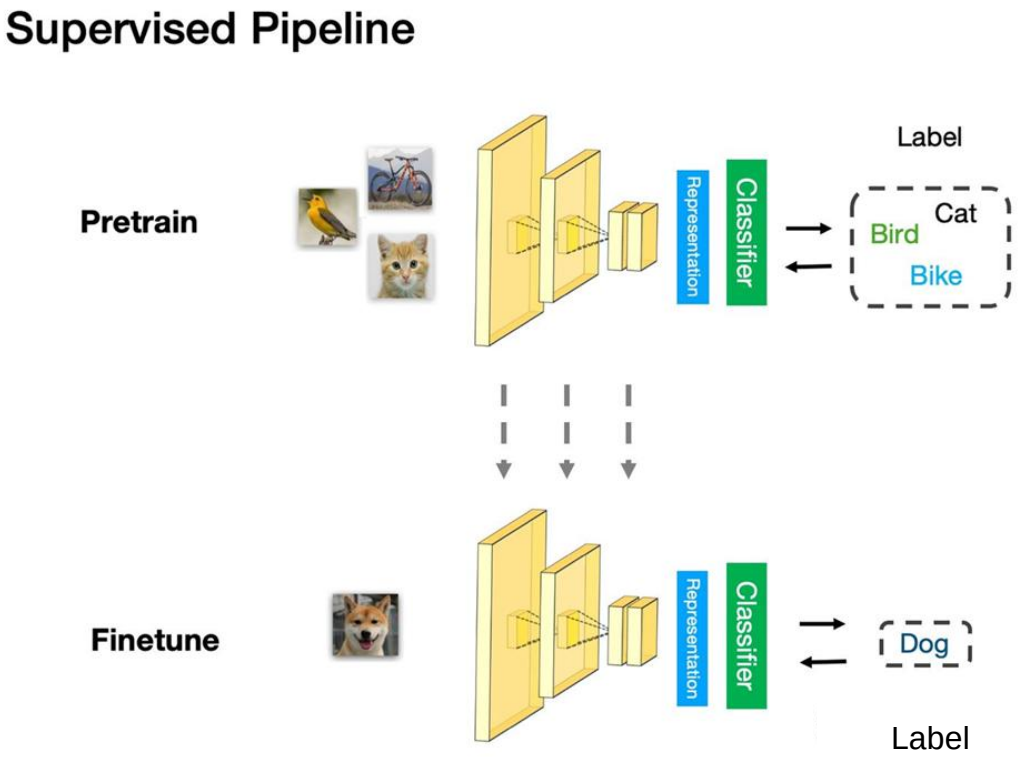
- Self-supervised learning: Using the data itself as supervision
- Self-supervised learning is a form of “unsupervised learning”
- In general, the goal is to:
  - Learn representations from large-scale unlabeled data
  - Learn high-level semantic representations

Self-supervised Learning

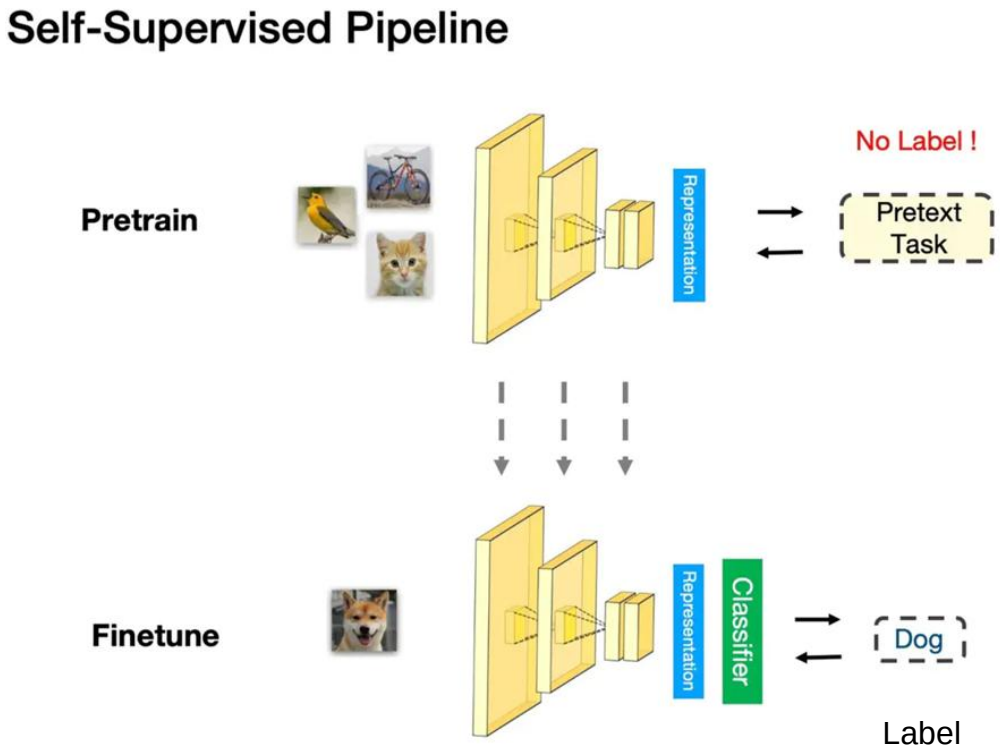
自监督学习的定义：

自监督学习是利用辅助任务（pretext task）或代理任务（proxy task）从大规模的无监督数据中挖掘自身的监督信息，通过这种构造的监督信息对网络进行训练，从而可以学习到对下游任务有价值的表征。

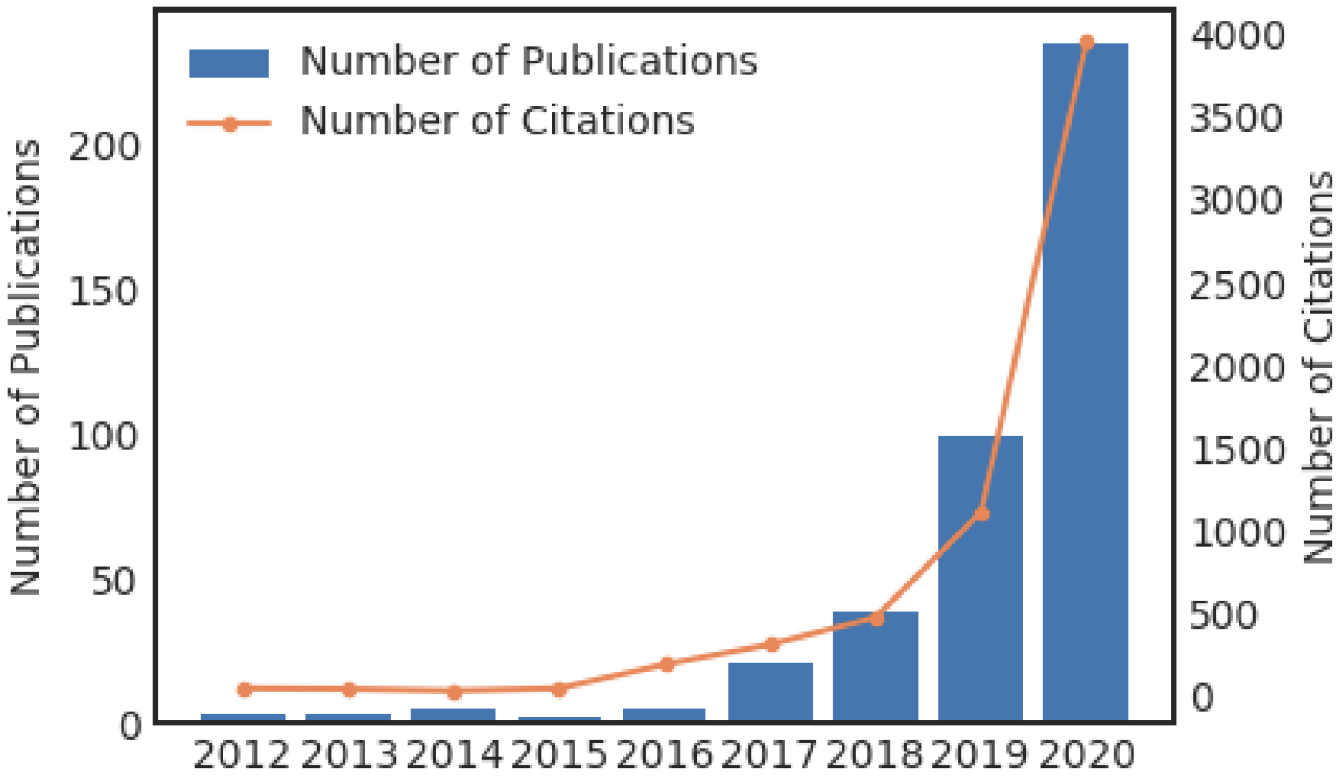
Supervised Learning



Self-supervised Learning



Self-supervised Learning



Term Definition

- **Pretext task (辅助任务）（Proxy task, 代理任务）: Pre-designed tasks for networks to solve, in order to learn features as a pre-trained model.**
- **Downstream Task（下游任务）: A final task for evaluating the quality of features learned through self-supervised learning.**
- **Human-annotated label: Labels of data that are manually annotated by human workers.**
- **Pseudo label: Automatically generated labels based on data attributes for pretext tasks.**

Self-supervised Learning

- What if we could automatically generate labels for unlabeled data with some rules and train the unsupervised dataset in a supervised way?
- In this way, all the information needed, both inputs and labels, has been provided. This is known as self-supervised learning.
- The main purpose of self-supervised learning is to pre-train representations that can be transferred to downstream tasks by fine-tuning.

自编码器的结构、数学模型、性质

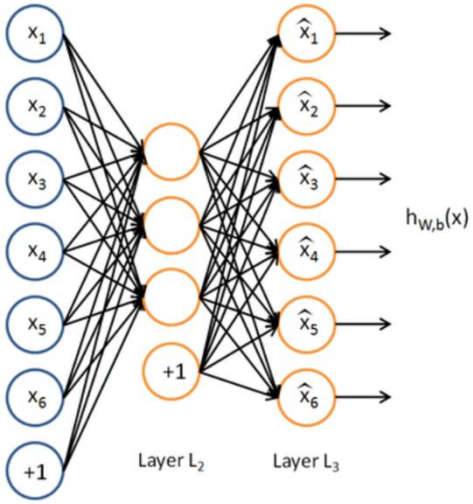
- 以只有一个隐层的自编码器为例，随机初始化权重W和偏执b，使用编码阶段和解码阶段的激活函数均为sigmoid函数，误差函数为平方误差函数。
- 编码阶段：$\sigma(W_{encode}X + b_{encode})$
- 解码阶段：

$$\hat{X} = \sigma(\mathrm{W}_{decode}\sigma(W_{encode}X + b_{encode}) + b_{decode})$$

- 损失函数为

$$L(\hat{X}, X) = \|\hat{X} - X\|_2^2$$

- 利用BP算法可以对网络进行优化求解

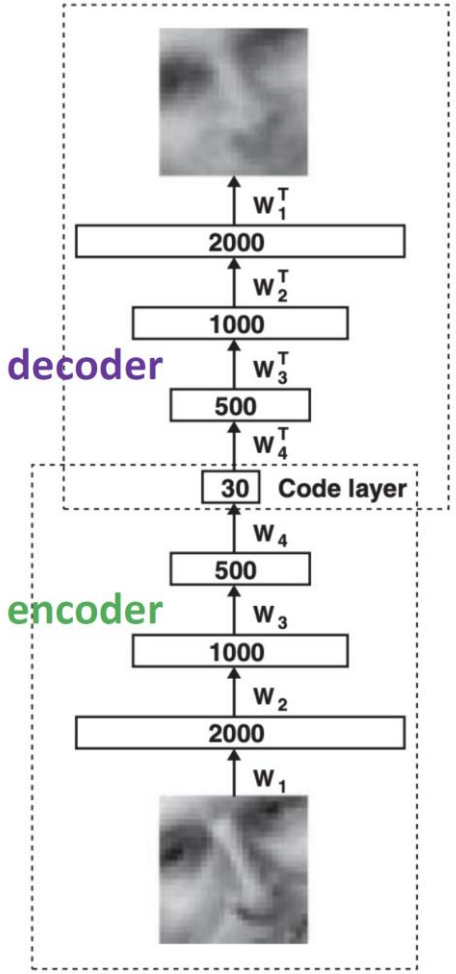


自监督学习与自编码器的区别

- Autoencoding is “self-supervised learning”
- “auto-” == “self-”
- “encoding” ≈ “learning”

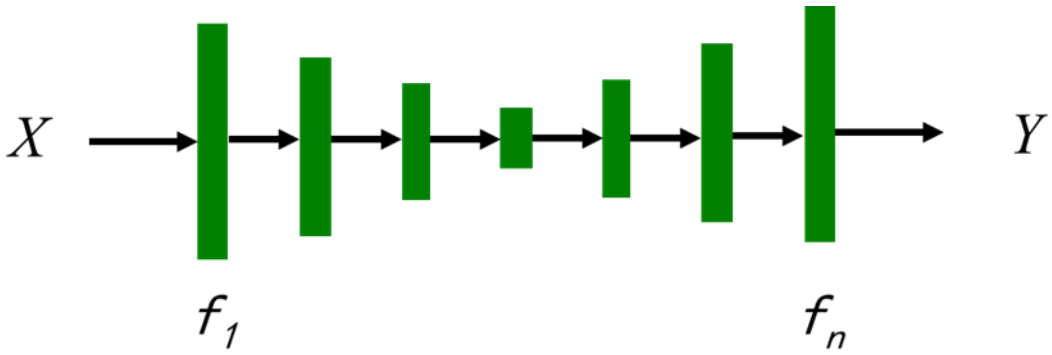
Using data “itself” as the supervision

$$\min_{D,E} \sum_{x} \|D(E(x)) - x\|^2$$



自监督学习与自编码器的区别

- 在深度自编码器中使用 L1 或 L2 损失来衡量输入和输出之间的差距不存在语义信息，这样的度量过多地关注像素级别的细节而忽略了更为重要的语义特征。
- 对于自编码器，往往只是做了维度的降低。但我们希望学习的目的不仅仅是维度更低，还希望包含更多的语义特征，让模型懂的输入究竟是什么，从而帮助下游任务。
- 自监督学习最主要的目的是学习到更丰富的**语义表征**。



Generative vs. Self-supervised Learning

- Both aim to learn from data without manual label annotation.
- Generative learning aims to model data distribution e.g., generating realistic images.
- Self-supervised learning methods solve "pretext" tasks that produce good features for downstream tasks.
  - Learn with supervised learning objectives, e.g., classification, regression.
  - Labels of these pretext tasks are generated *automatically*

Generative vs. Self-supervised Learning

Self-supervised pretext tasks

Example: learn to predict image transformations / complete corrupted images

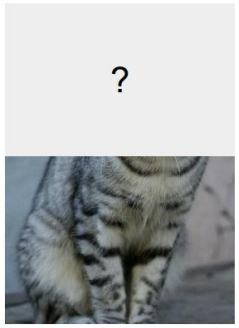
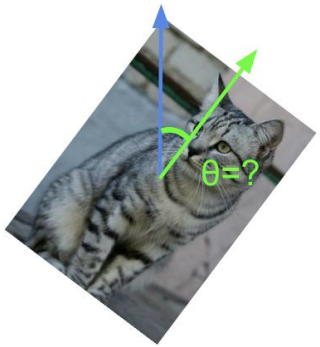


image completion



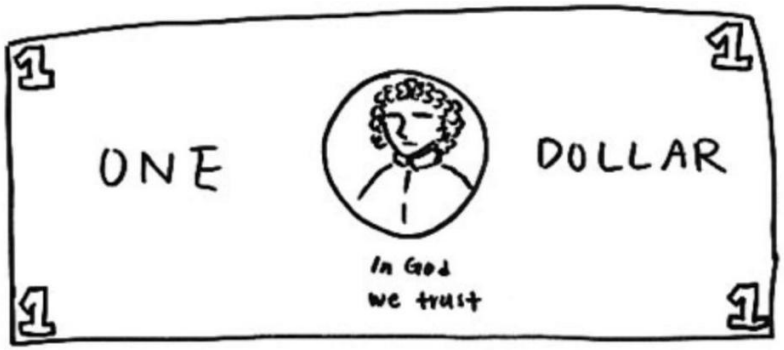
rotation prediction

“jigsaw puzzle”

colorization

1. Solving the pretext tasks allow the model to learn good features.
2. We can automatically generate labels for the pretext tasks.

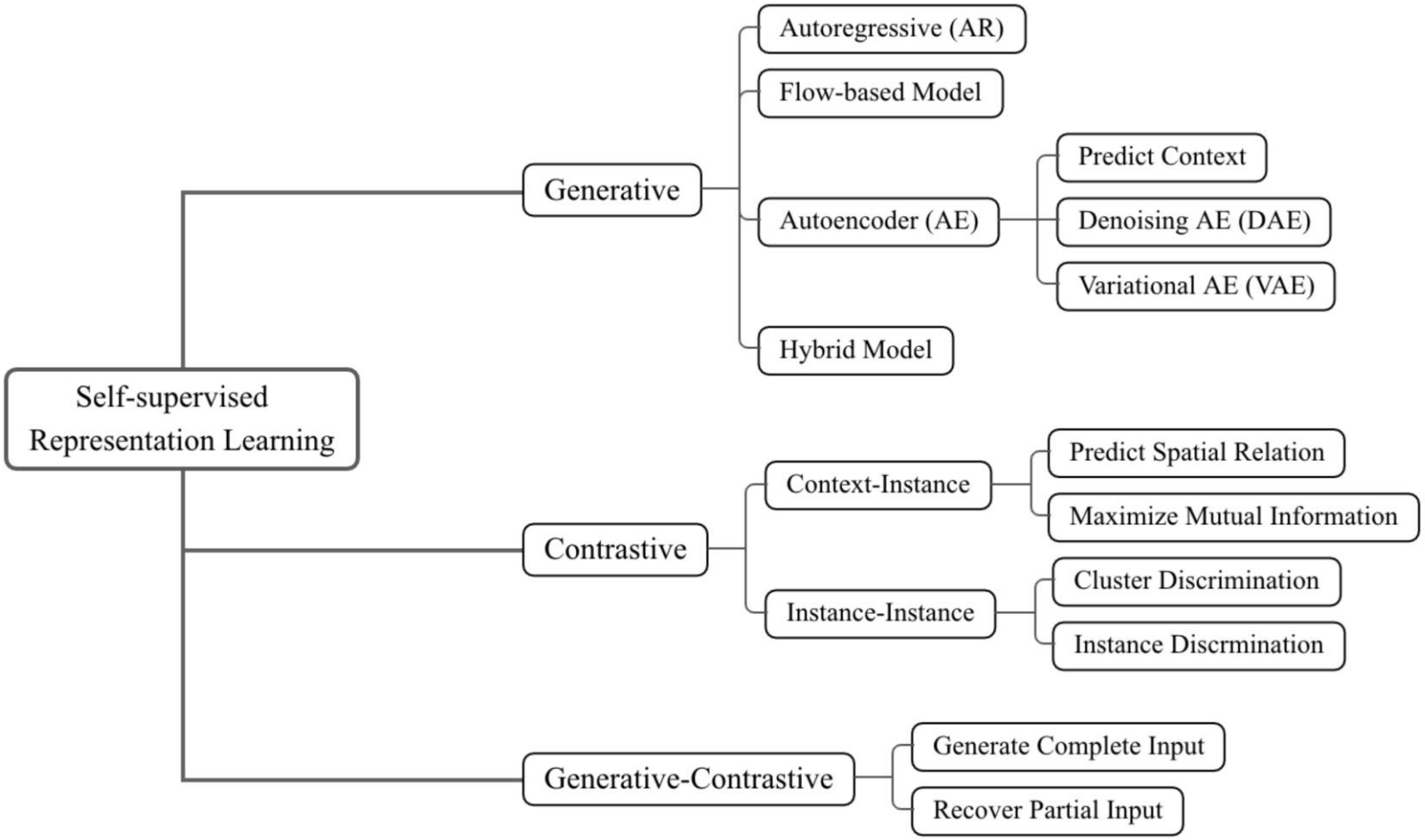
Generative vs. Self-supervised Learning



Left: Drawing of a dollar bill from memory. Right: Drawing subsequently made with a dollar bill present. Image source: Epstein, 2016

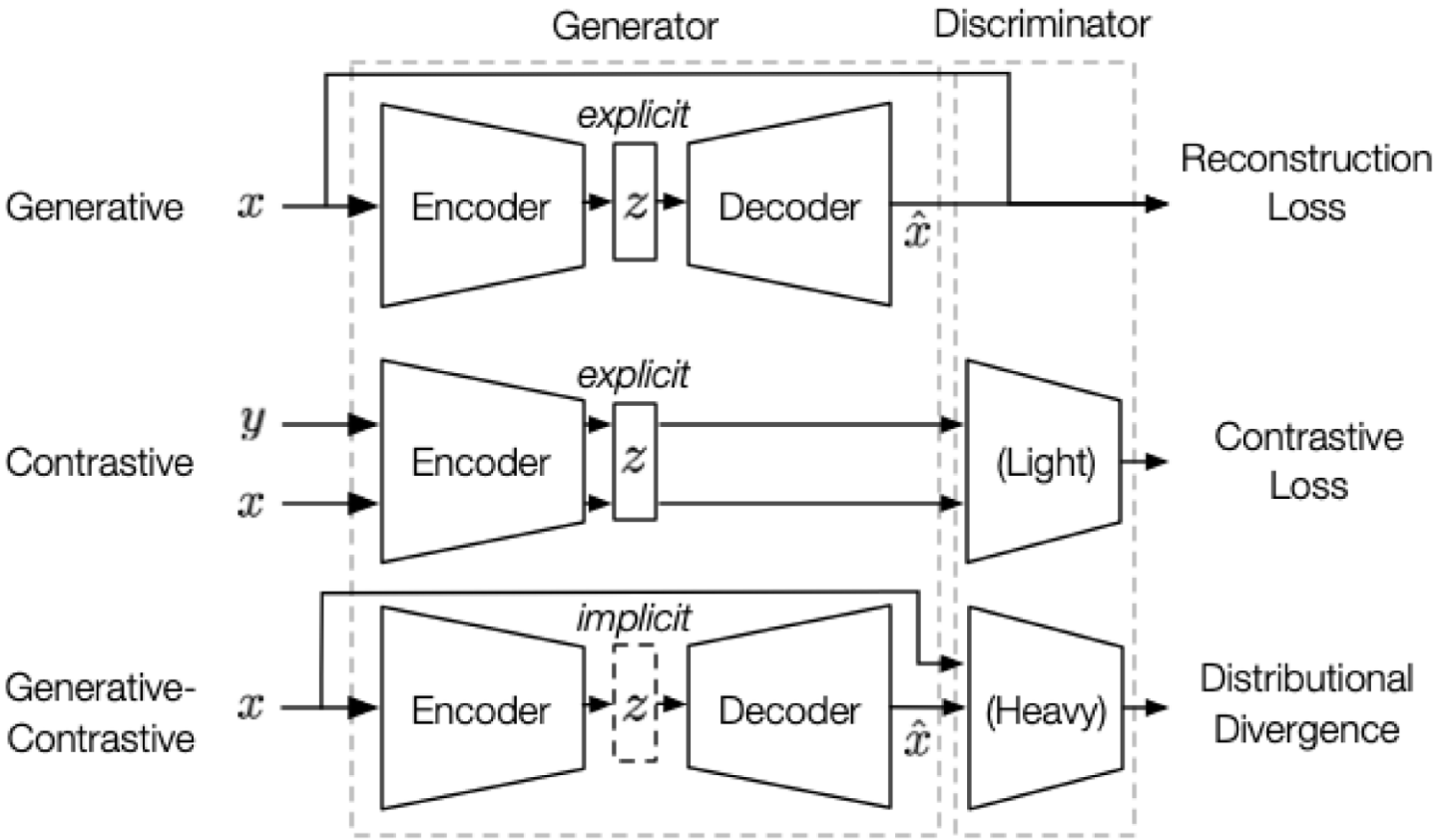
Learning to generate pixel-level details is often unnecessary; learn high-level semantic features with pretext tasks instead

Self-supervised Learning



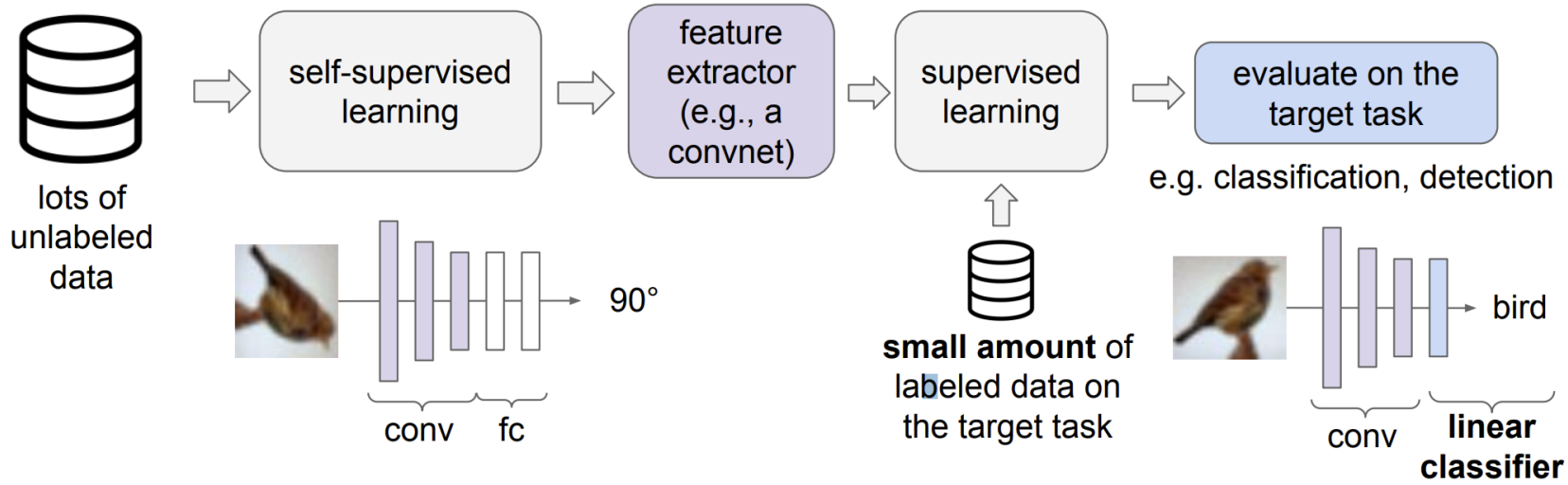
Liu, Xiao, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. "Self-supervised learning: Generative or contrastive." IEEE Transactions on Knowledge and Data Engineering 35, no. 1 (2021): 857-876.

Self-supervised Learning



Liu, Xiao, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. "Self-supervised learning: Generative or contrastive." IEEE Transactions on Knowledge and Data Engineering 35, no. 1 (2021): 857-876.

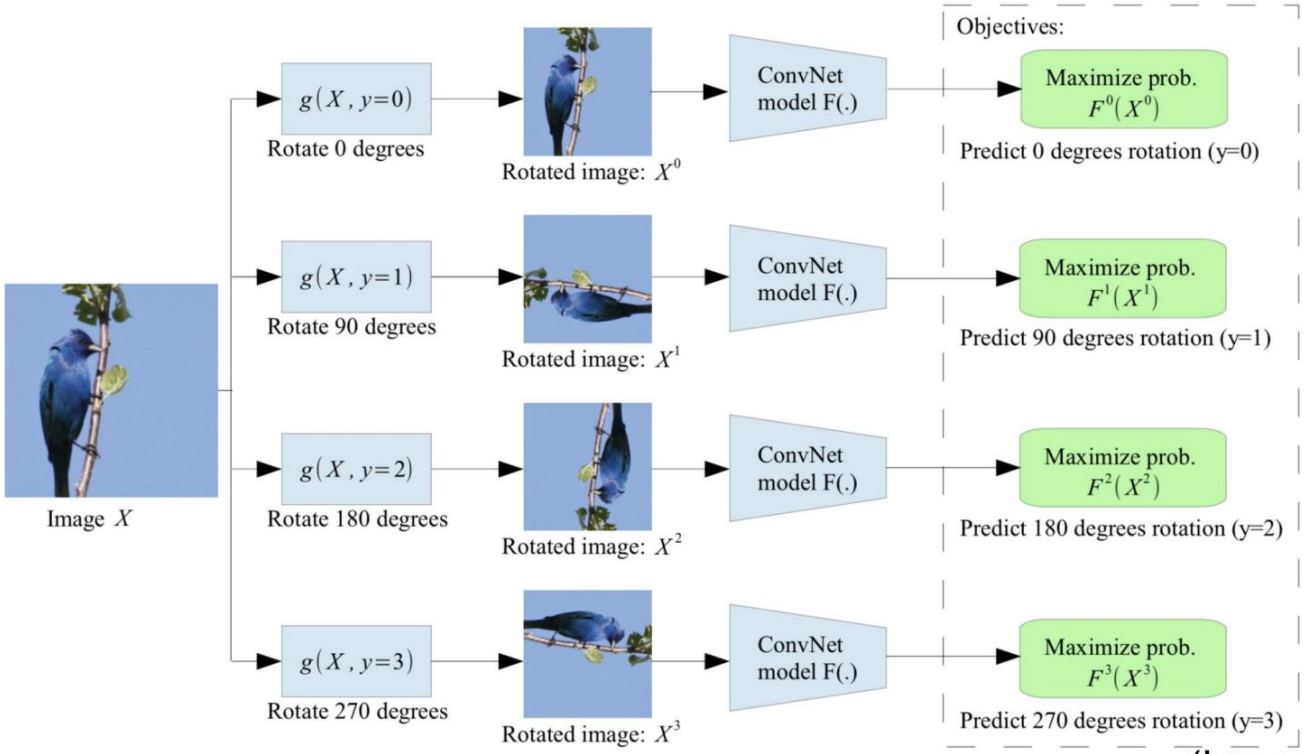
How to evaluate a self-supervised learning method?



1. Learn good feature extractors from self-supervised pretext tasks, e.g., predicting image rotations

2. Attach a shallow network on the feature extractor; train the shallow network on the target task with small amount of labeled data

Pretext task: predict rotations

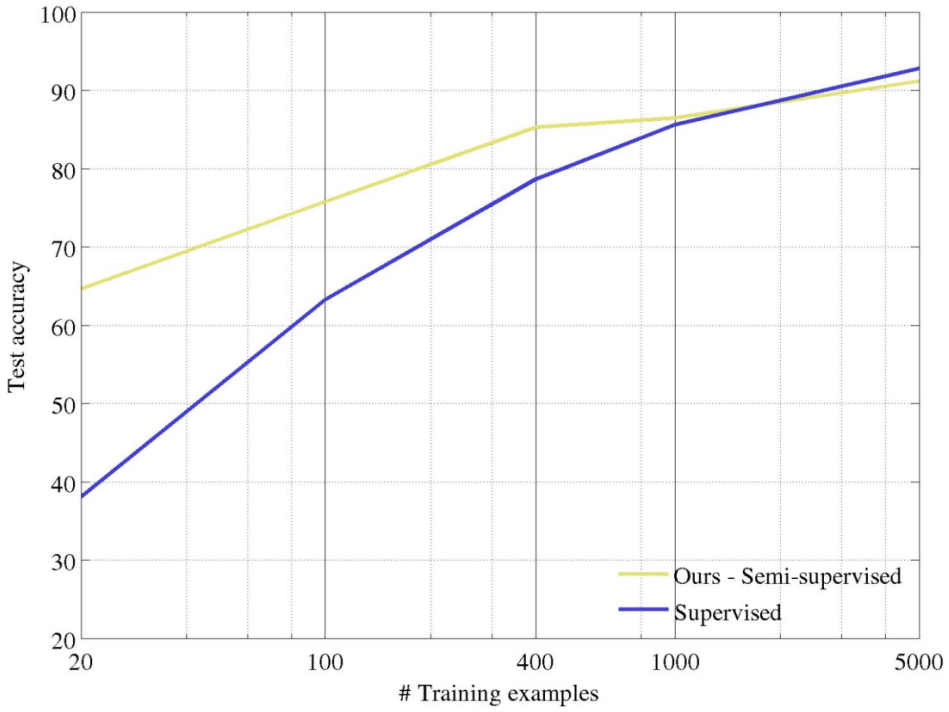


Self-supervised learning by rotating the entire input images.

The model learns to predict which rotation is applied (4-way classification)

(Image source: Gidaris et al. 2018)

Evaluation on semi-supervised learning

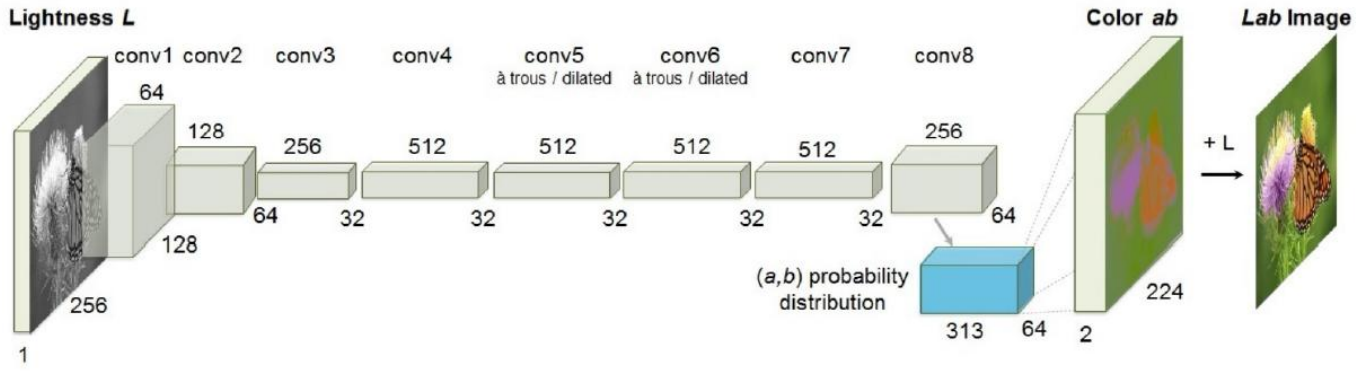


Self-supervised learning on **CIFAR10** (entire training set).

Freeze conv1 + conv2
Learn **conv3 + linear** layers with subset of labeled CIFAR10 data (classification).

(Image source: Gidaris et al. 2018)

Image Generation with Colorization



Input:
Transformed grey level image

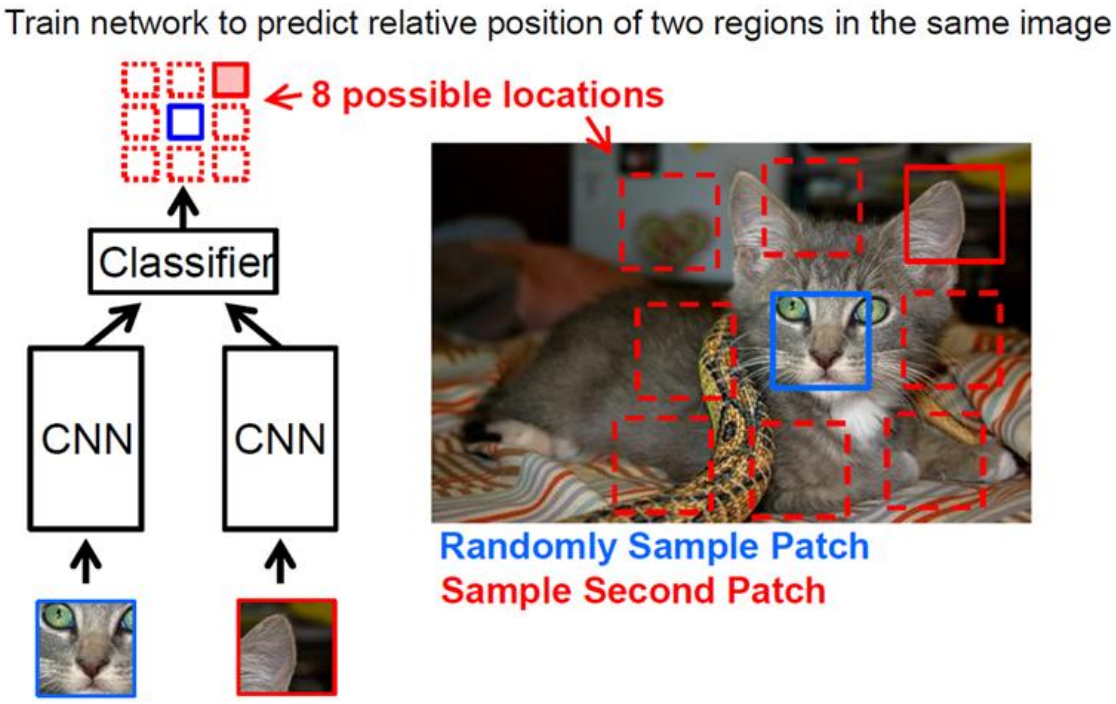
Pseudo label:
Original colorful mage

只有模型可以理解图片中的语义信息才知道哪些部分应该上什么样的颜色。例如，天空是蓝色的，草地是绿色的，只有模型从海量的数据中学习到了这些语义概念，才能得知物体的具体颜色信息。

Zhang R, Isola P, Efros A A. Colorful image colorization, European conference on computer vision. Springer, Cham, 2016: 649-666.

Spatial Context Structure

Relative position prediction



Doersch C, Gupta A, Efros A A. Unsupervised visual representation learning by context prediction, Proceedings of the IEEE international conference on computer vision. 2015: 1422-1430.

Pretext tasks from image transformations

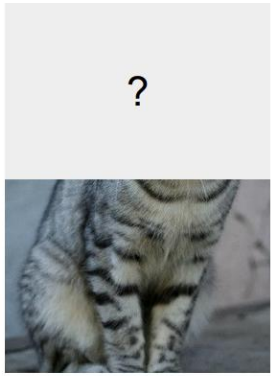
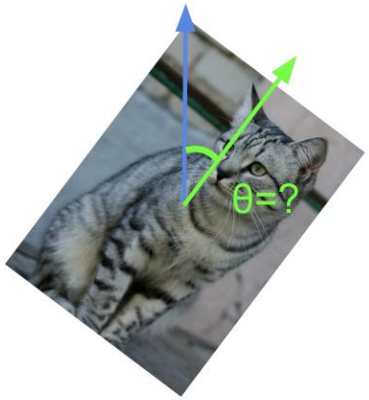


image completion



rotation prediction

“jigsaw puzzle”

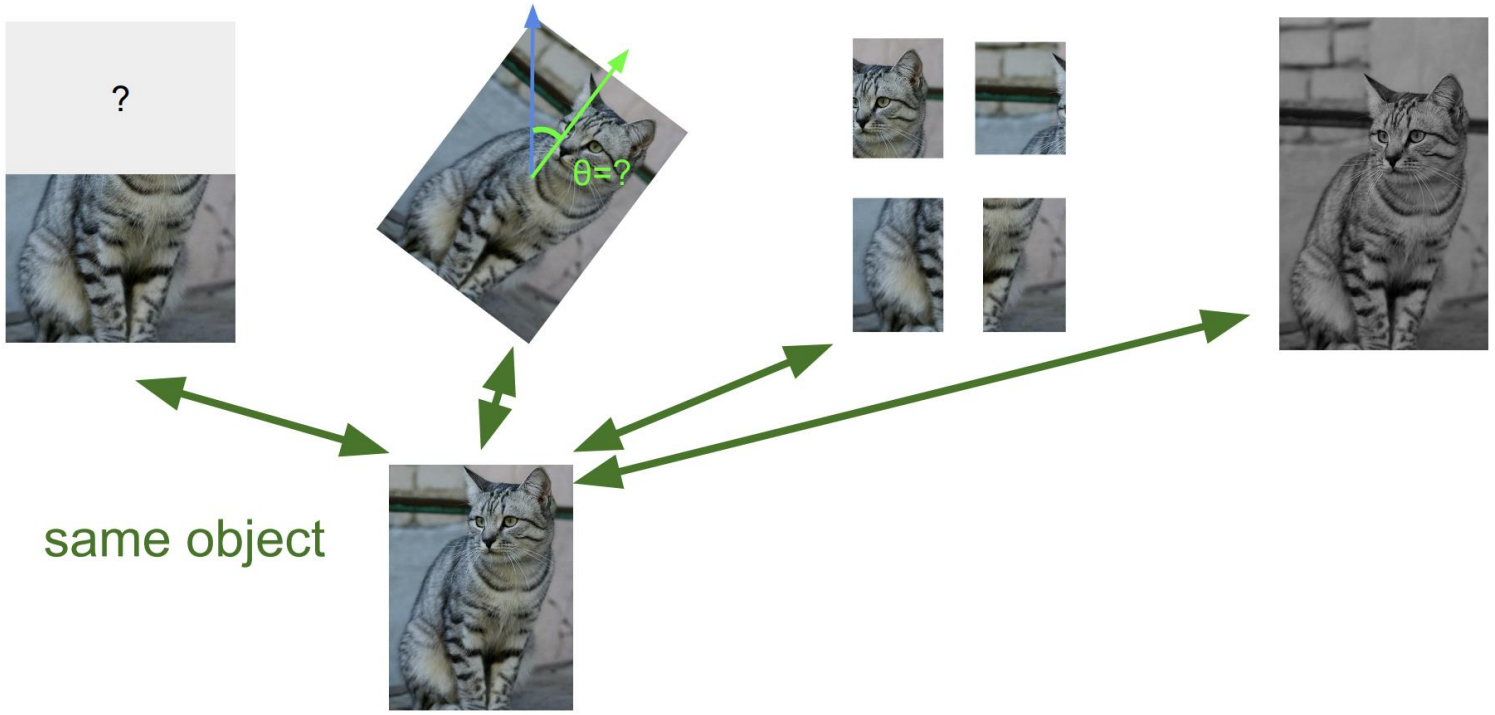
colorization

Learned representations may be tied to a specific pretext task!

Can we come up with a more general pretext task?

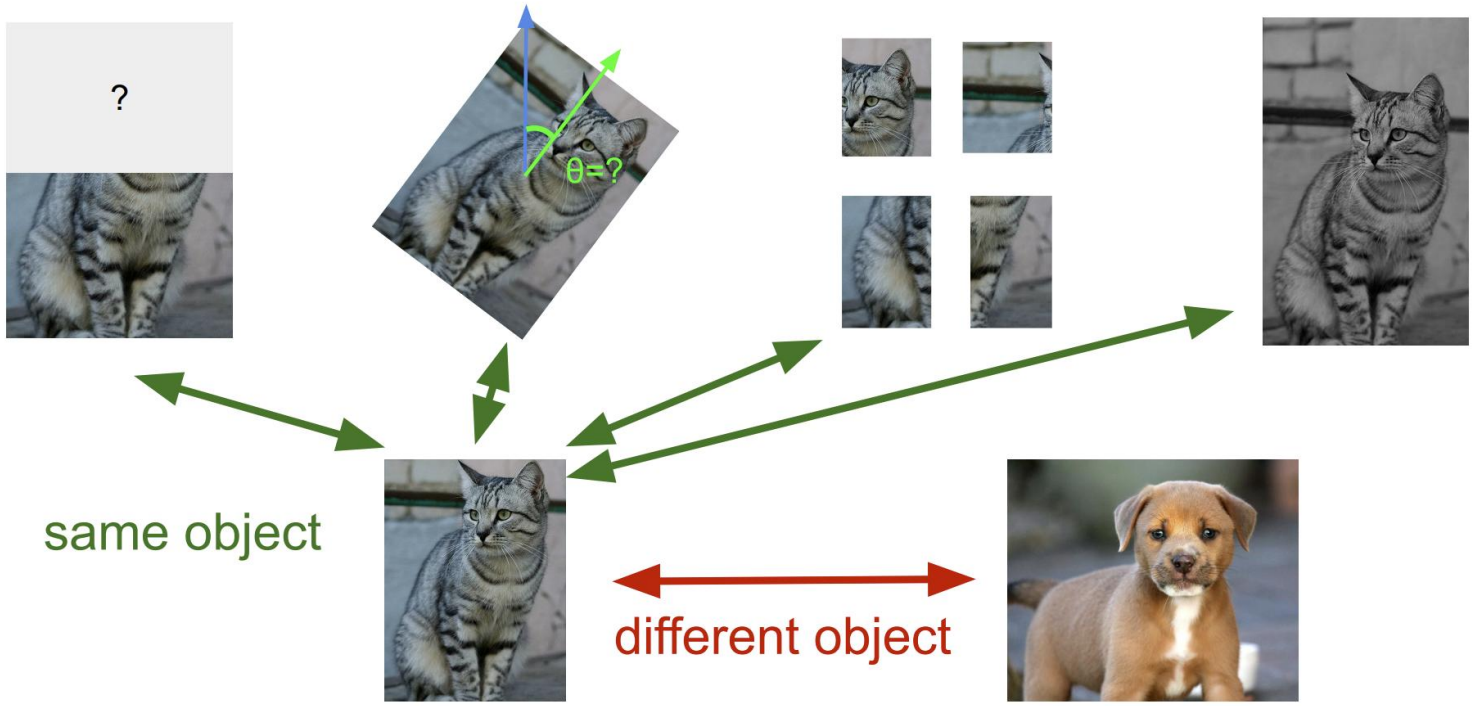
Pretext tasks from image transformations

A more general pretext task?

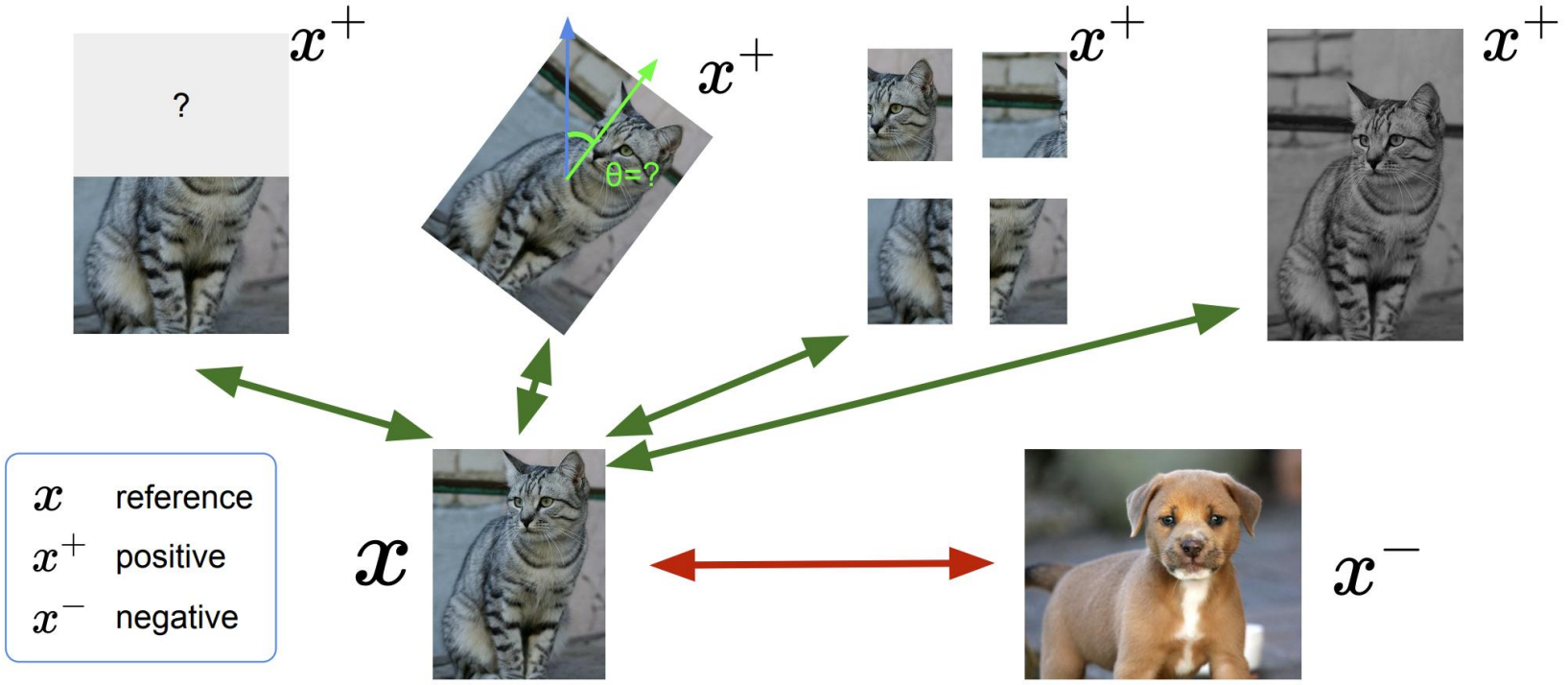


Pretext tasks from image transformations

A more general pretext task?



Contrastive Representation Learning



A formulation of contrastive learning

What we want:

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

x: reference sample; x^+ positive sample; x^- negative sample

Given a chosen score function, we aim to learn an **encoder function** *f* that yields high score for positive pairs (x, x^+) and low scores for negative pairs (x, x^-).

A formulation of contrastive learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

x

x^+

x

x_1^-

x_2^-

x_3^-

...

A formulation of contrastive learning

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

score for the positive pair

score for the N-1 negative pairs

This seems familiar …
Cross entropy loss for a N-way softmax classifier!
I.e., learn to find the positive sample from the N samples

Commonly known as the InfoNCE loss (van den Oord et al., 2018)

Contrastive Methods

Contrastive methods differ from the more traditional generative methods to learn representations, which focus on reconstruction error in the pixel space to learn representations.

- **Using pixel-level losses can lead to such methods being overly focused on pixel-based details, rather than more abstract latent factors.**
- **Pixel-based objectives often assume independence between each pixel, thereby reducing their ability to model correlations or complex structure.**

Contrastive Learning

- **Contrastive methods trained on unlabelled ImageNet data and evaluated with a linear classifier now surpass the accuracy of supervised AlexNet. They also exhibit significant data efficiency when learning from labelled data compared to purely supervised learning (Data-Efficient CPC, Hénaff et al., 2019).**
- **Contrastive pre-training on ImageNet successfully transfers to other downstream tasks and outperforms the supervised pre-training counterparts (MoCo, He et al., 2019).**

Downstream Tasks for Evaluation

- To compare different self-supervised learning methods, there are some commonly used downstream tasks for evaluation. Such as,
  - CV: Semantic segmentation, Object detection, Human action recognition…
  - NLP: Question answering, Sentiment classification…

SimCLR: A Simple Framework for Contrastive Learning

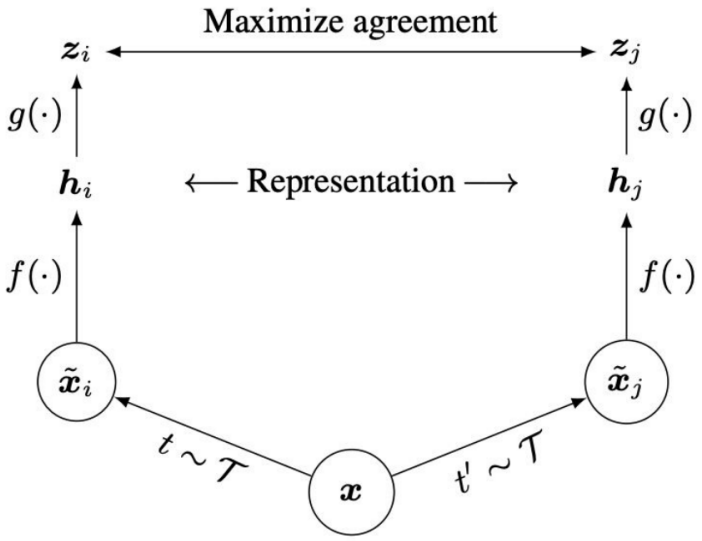
Cosine similarity as the score function:

$$s(u, v) = \frac{u^T v}{||u||||v||}$$

Use a projection network ***g(·)*** to project features to a space where contrastive learning is applied

Generate positive samples through data augmentation:

- random cropping, random color distortion, and random blur.



Source: Chen et al., 2020

SimCLR: A Simple Framework for Contrastive Learning

SimCLR: generating positive samples from data augmentation

(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

Source: Chen et al., 2020

SimCLR: A Simple Framework for Contrastive Learning

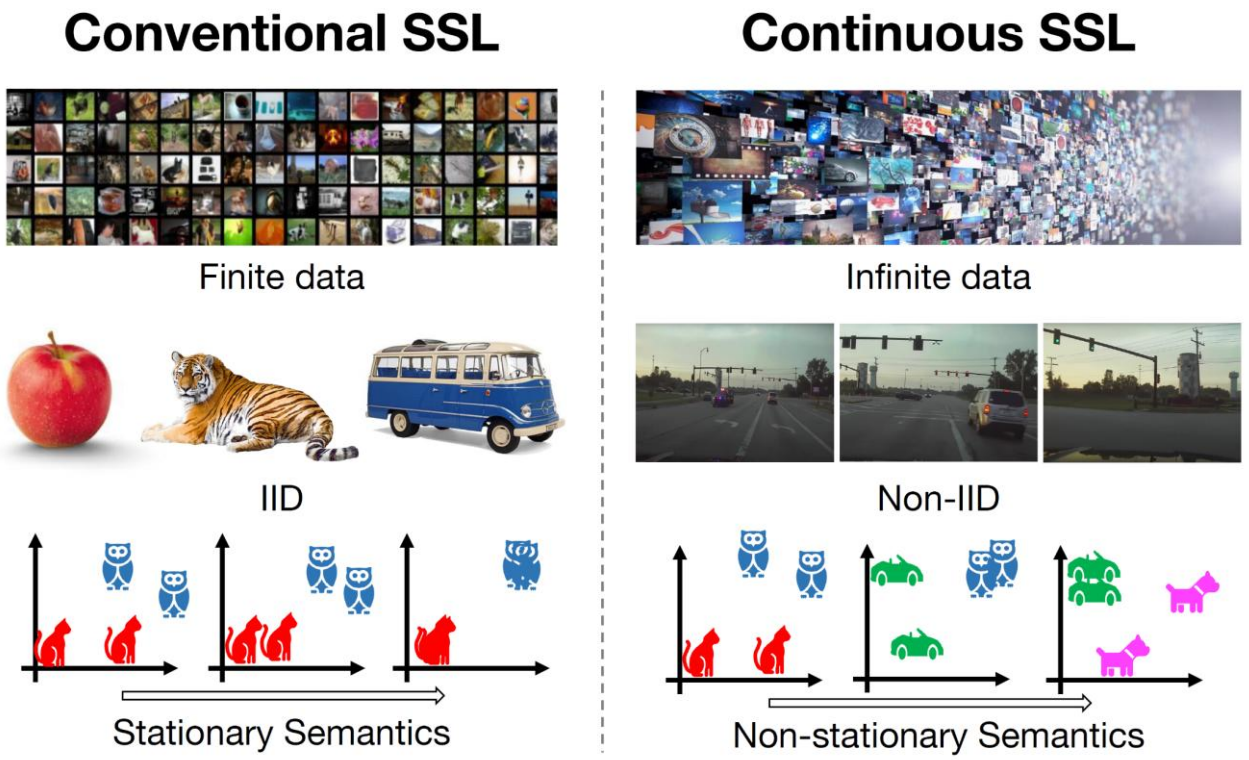
SimCLR

Generate a positive pair by sampling data augmentation functions

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of $f, g, \mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}, t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i, j)$ **as** $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

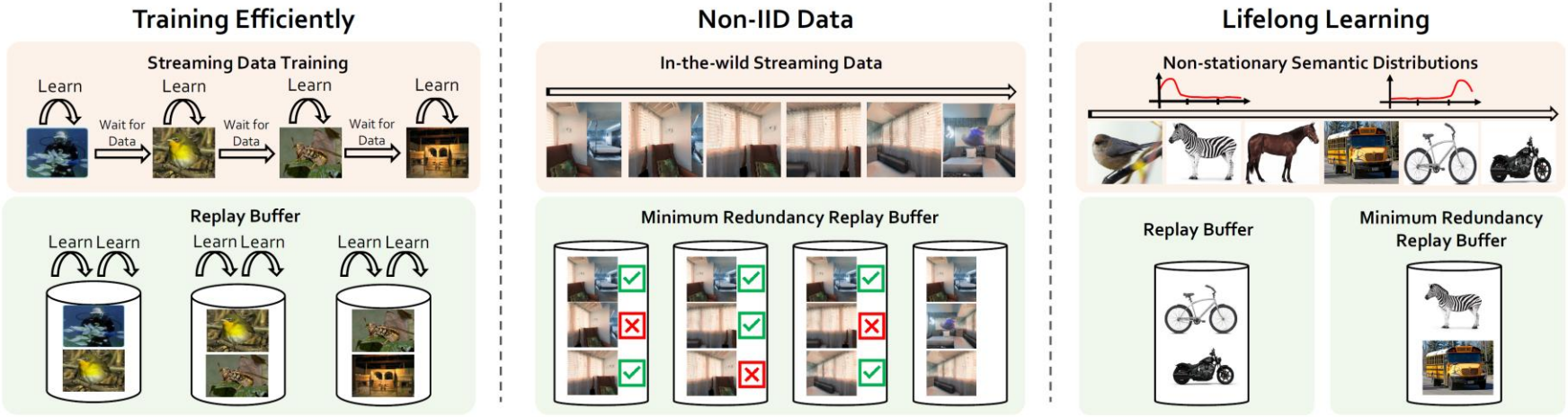
Two Self-supervised Learning Methods



传统自监督学习设定下，数据集是固定的。而自然场景下持续收集到的数据是无限、非独立同分布、具有非平稳语义的。因此，传统设定很难作为自然场景下设定的自监督学习的对比基准。

Purushwalkam, Morgado, and Gupta, The Challenges of Continuous Self-Supervised Learning, https://arxiv.org/abs/2203.12710

Continuous Self-supervised Learning



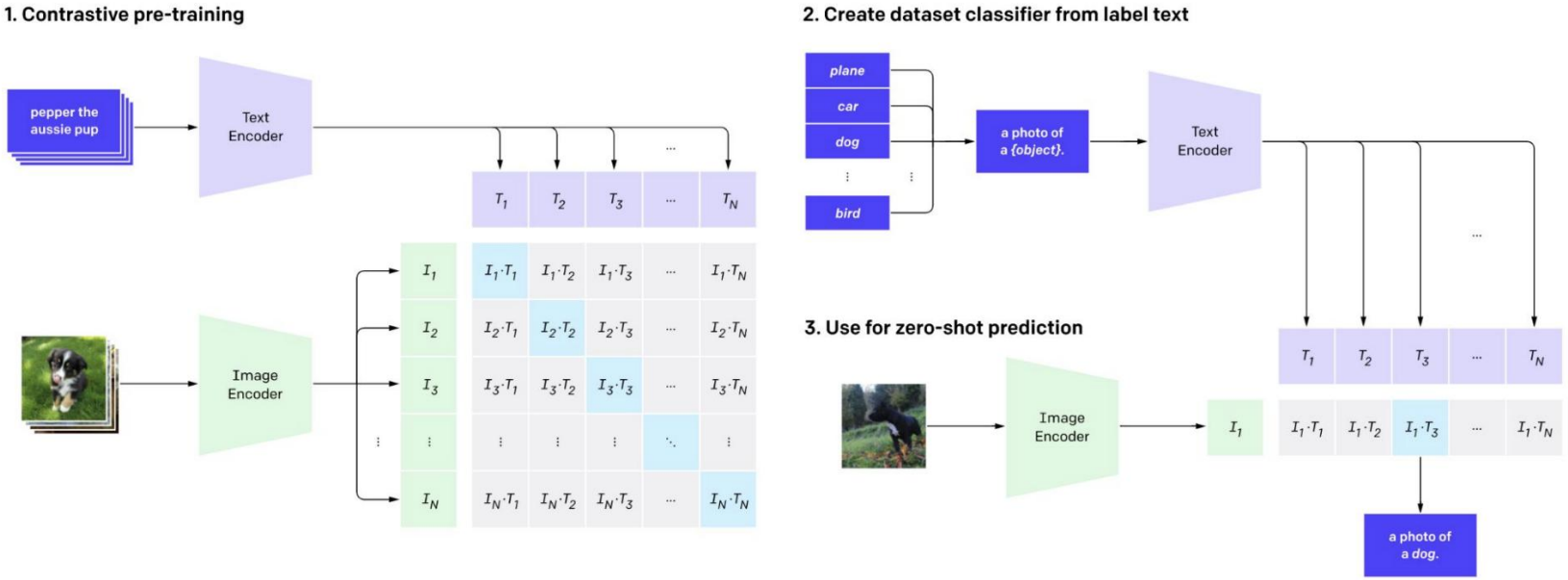
a) 无限数据流中的样本无法重复，使用**回放缓冲区**增强现有的自监督学习方法，显著缓解了该问题。

b) 持续从自然场景下收集的数据往往在时间上是相关的，不满足优化算法的独立同分布假设。通过增强回放缓冲区来保留最低限度的冗余样本（MinRed），从而生成相关性较低的数据。

c) 在自然场景下收集到数据的语义分布是非平稳的，模型可能会「遗忘」在过去的分布中看到的概念。MinRed 缓冲区可以通过从各种语义类中收集独特的样本缓解「遗忘」问题。

Other examples

Contrastive learning between image and natural language sentences



CLIP (*Contrastive Language–Image Pre-training*) Radford *et al.*, 2021

Predictive Learning: Language Models

Modern self-supervised learning methods:

- Contrastive Learning
- **Predictive Learning/Generative**

Predictive Learning: Language Models

Next word prediction (GPT)

- Predict the next word (token) given a prefix

Radford, et al., "Improving Language Understanding by Generative Pre-Training", 2018

Predictive Learning: Language Models

Masked language modeling (BERT)

- Predict the masked words (tokens) in a text

Devlin, et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018

Predictive Learning: Computer Vision

Masked image modeling (Context Encoders)

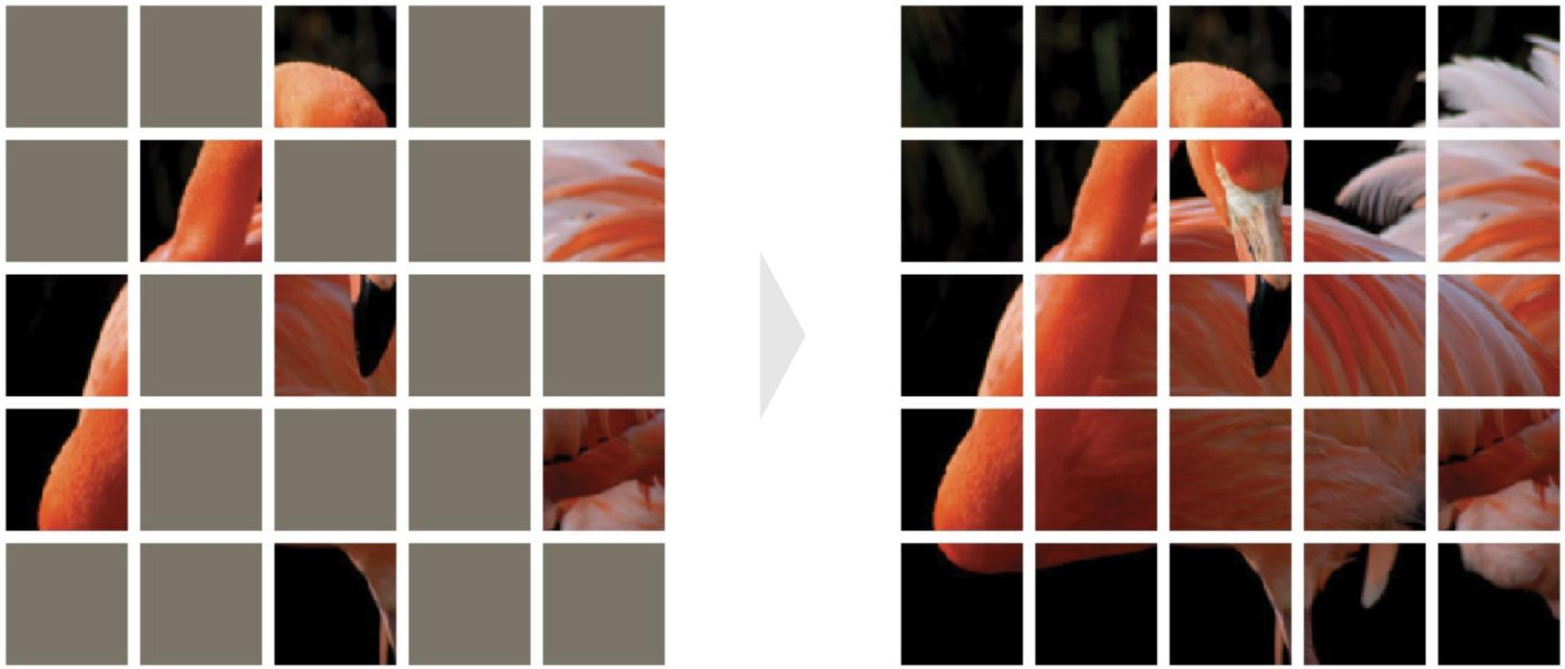
- Predict the masked regions using ConvNets

Pathak, et al., “Context Encoders: Feature Learning by Inpainting”, CVPR 2016

Predictive Learning: Computer Vision

Masked image modeling (Masked Autoencoder)

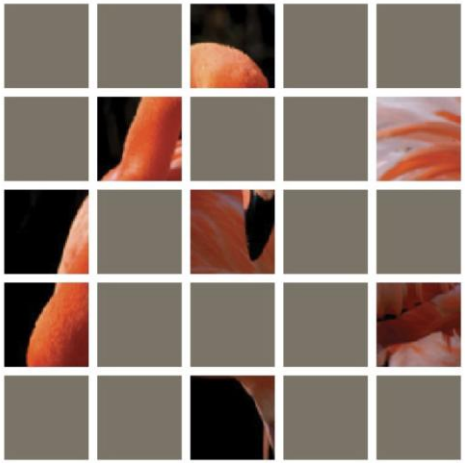
- Predict the masked patches using Transformers



He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

Predictive Learning: Computer Vision

Masked image modeling (Masked Autoencoder)

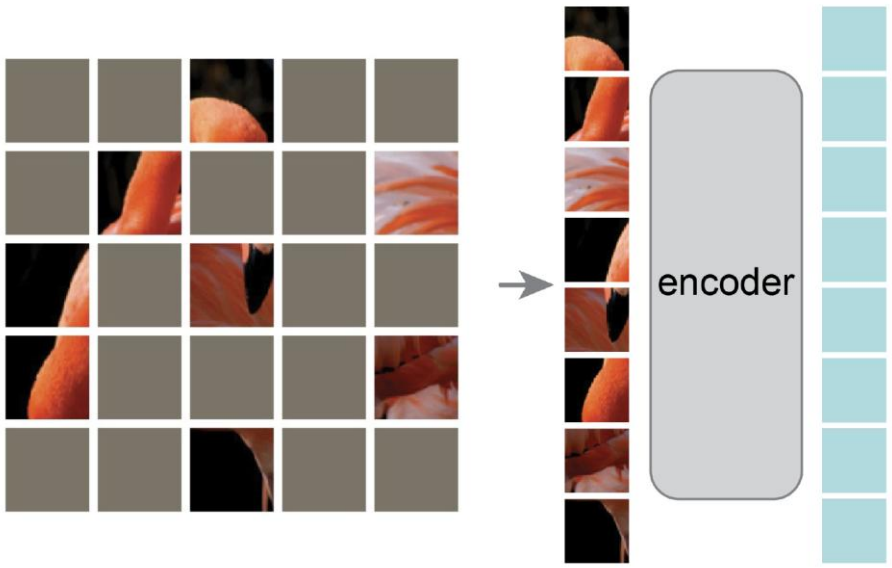


random masking

He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

Predictive Learning: Computer Vision

Masked image modeling (Masked Autoencoder)

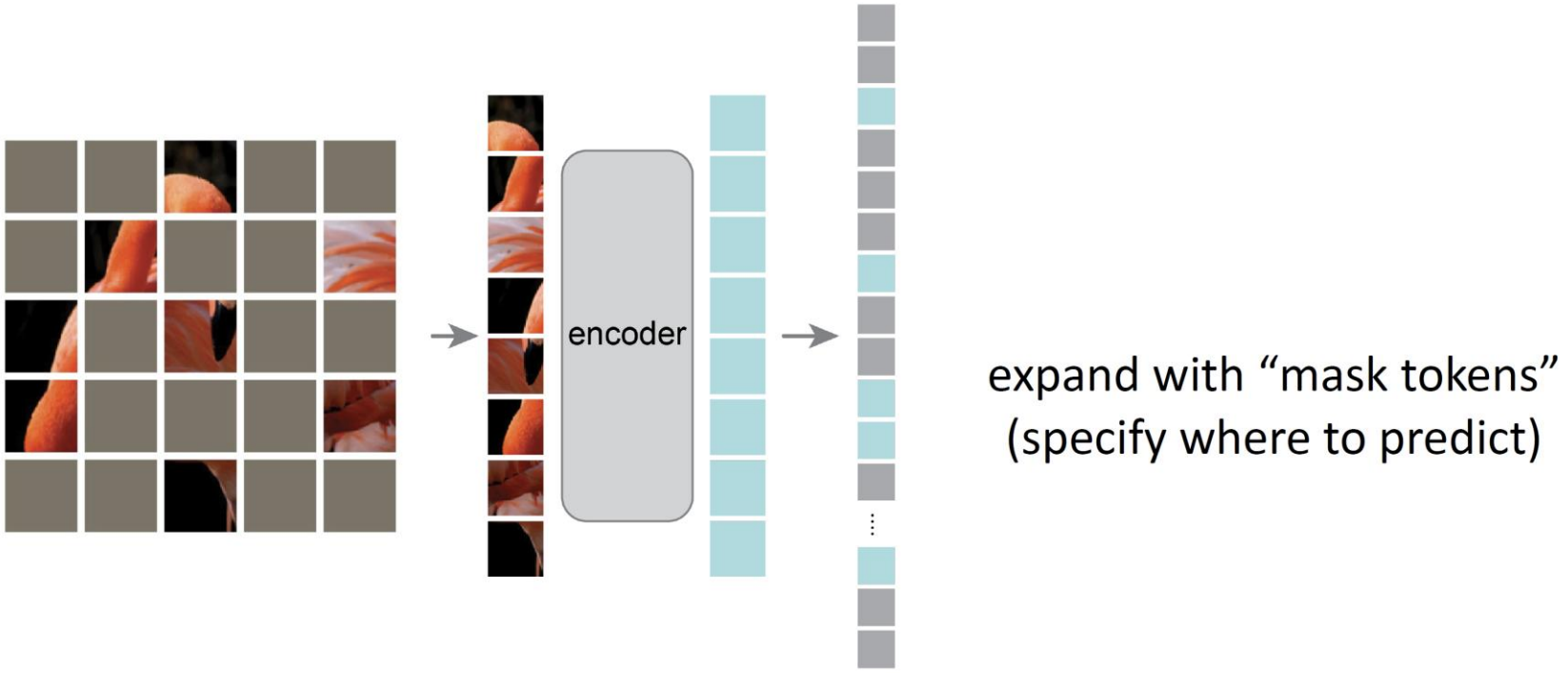


encode visible patches
w/ Transformer

He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

Predictive Learning: Computer Vision

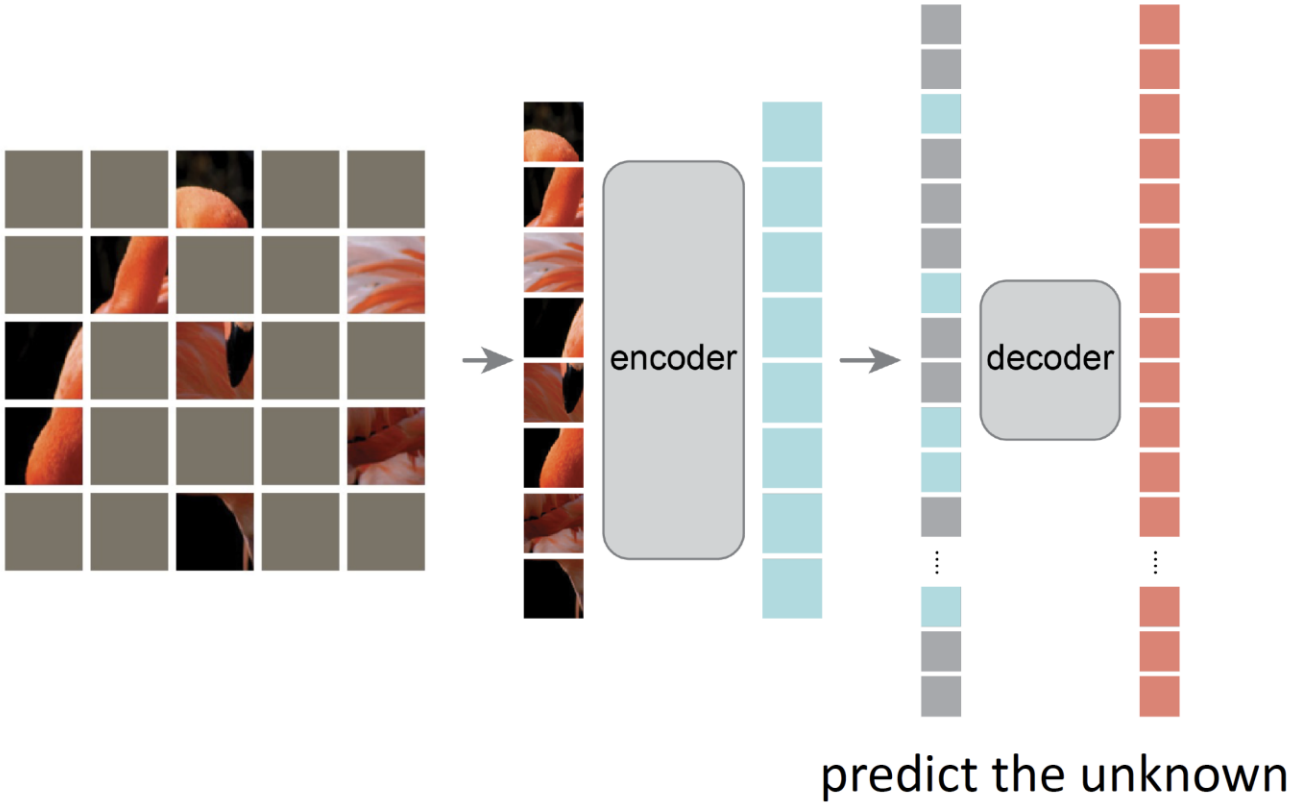
Masked image modeling (Masked Autoencoder)



He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

Predictive Learning: Computer Vision

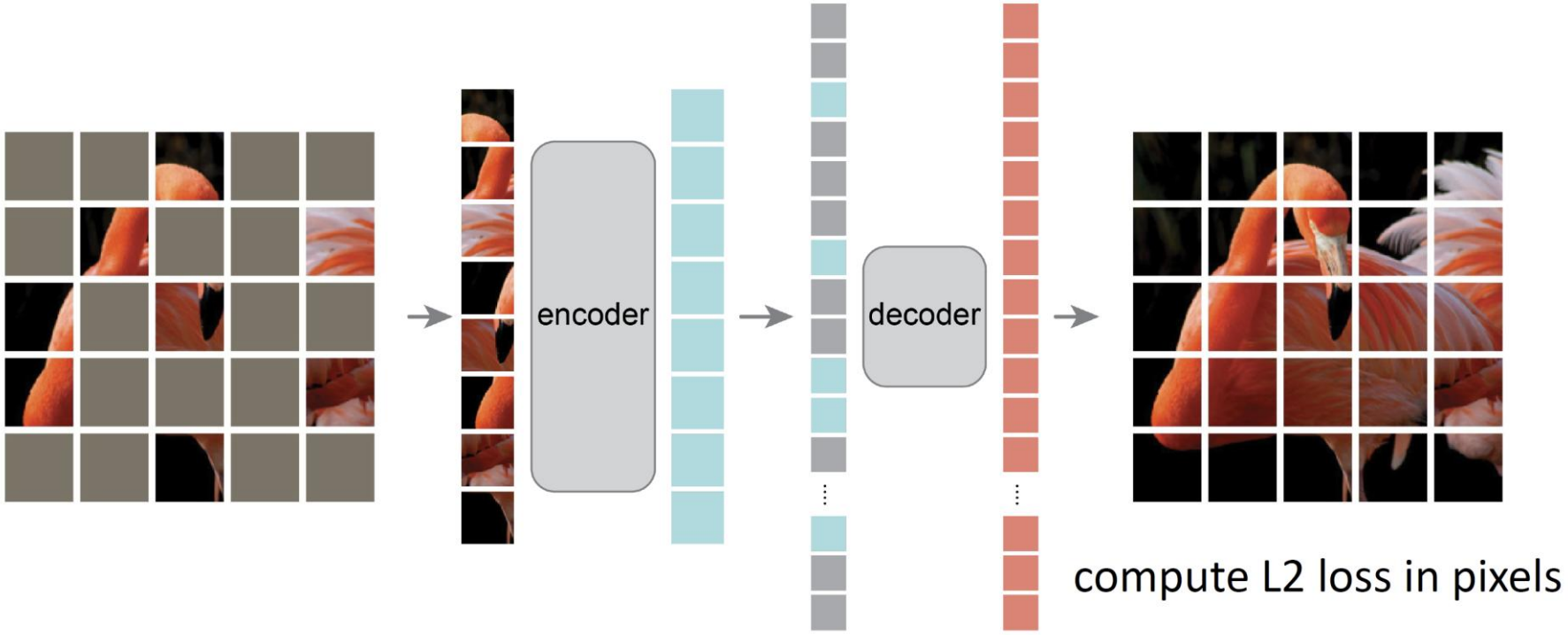
Masked image modeling (Masked Autoencoder)



He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

Predictive Learning: Computer Vision

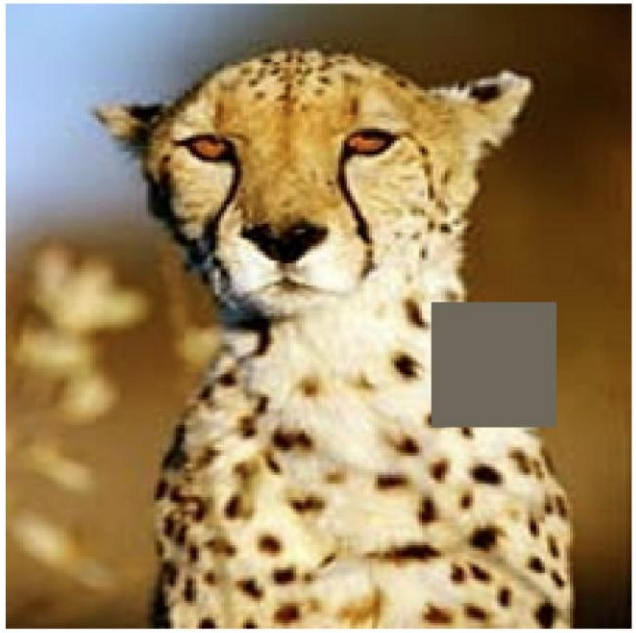
Masked image modeling (Masked Autoencoder)



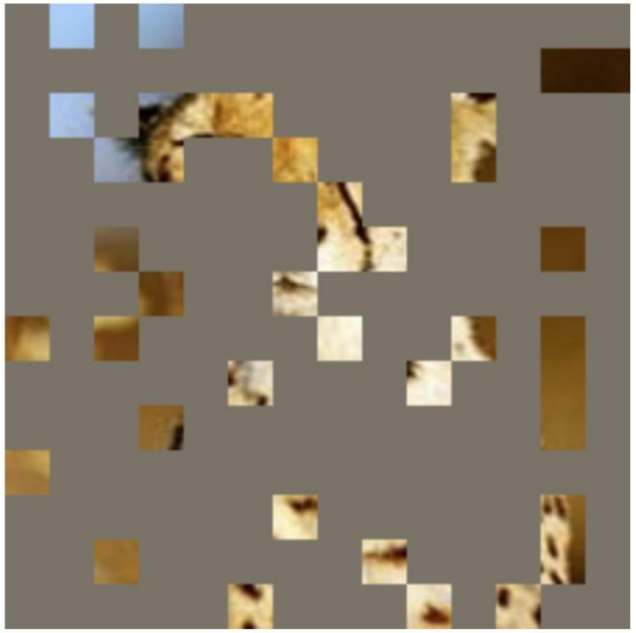
He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

Predictive Learning: Computer Vision

How to learn good representations by predicting?



- predicting a small portion may not require high-level understanding

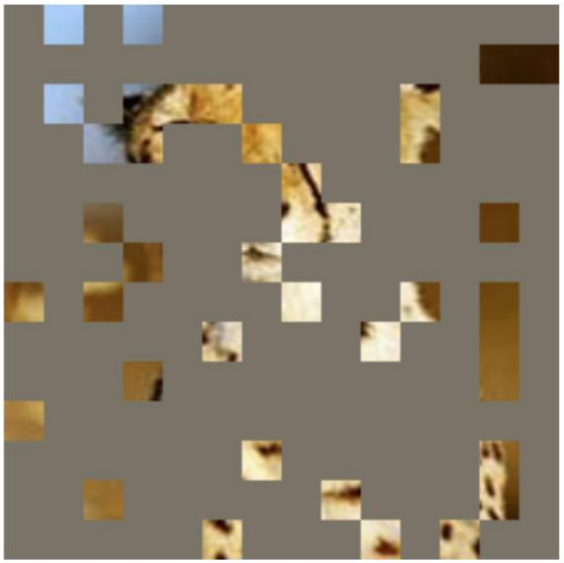


- predicting a large portion of unknown patches encourages to learn semantic features

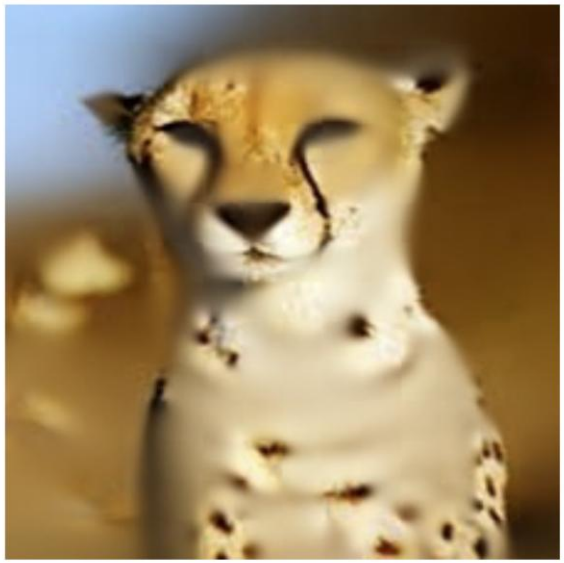
He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

Predictive Learning: Computer Vision

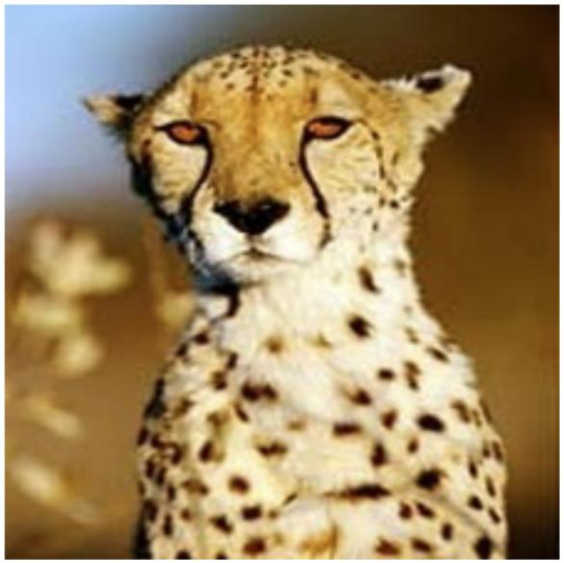
How to learn good representations by predicting?



input



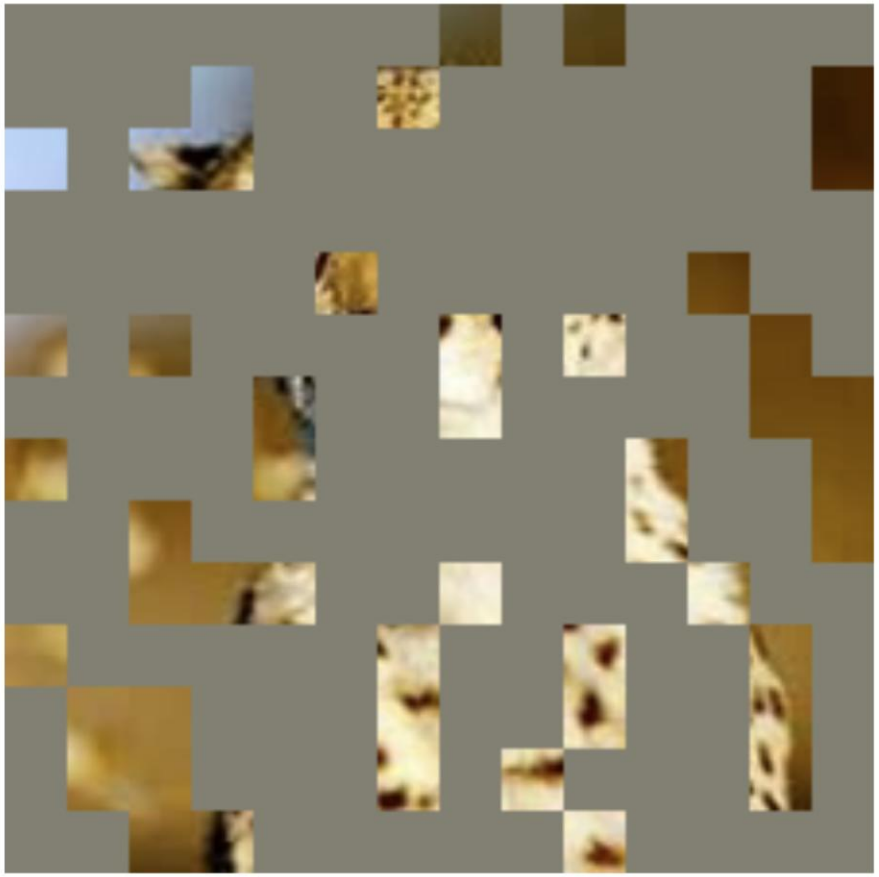
MAE prediction



original

He, et al., “Masked Autoencoders Are Scalable Vision Learners”, CVPR 2022

Predictive Learning: Computer Vision



- **The learning process**: the network gradually makes sense of the semantic patterns

He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

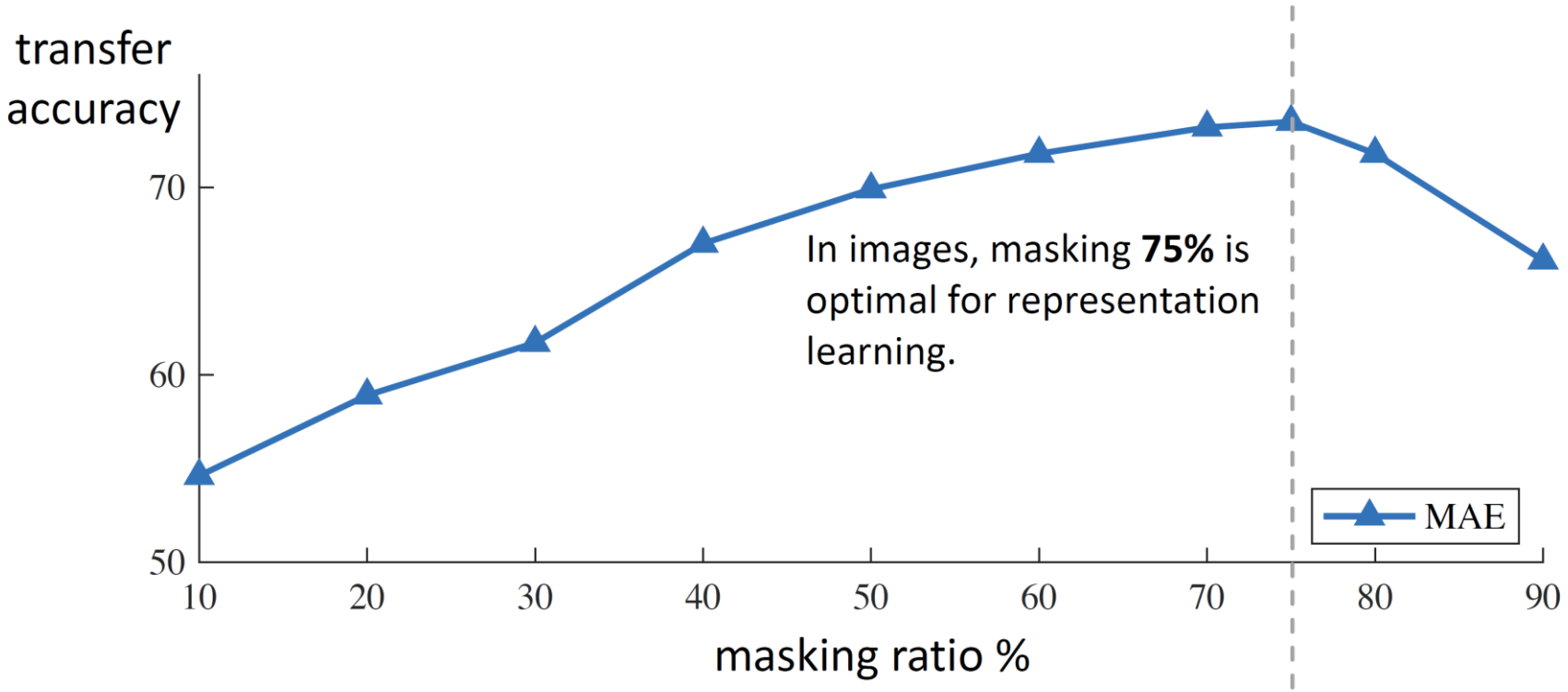
Predictive Learning: Computer Vision

- **The learning process**: the network gradually makes sense of the semantic patterns

He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

Predictive Learning: Computer Vision

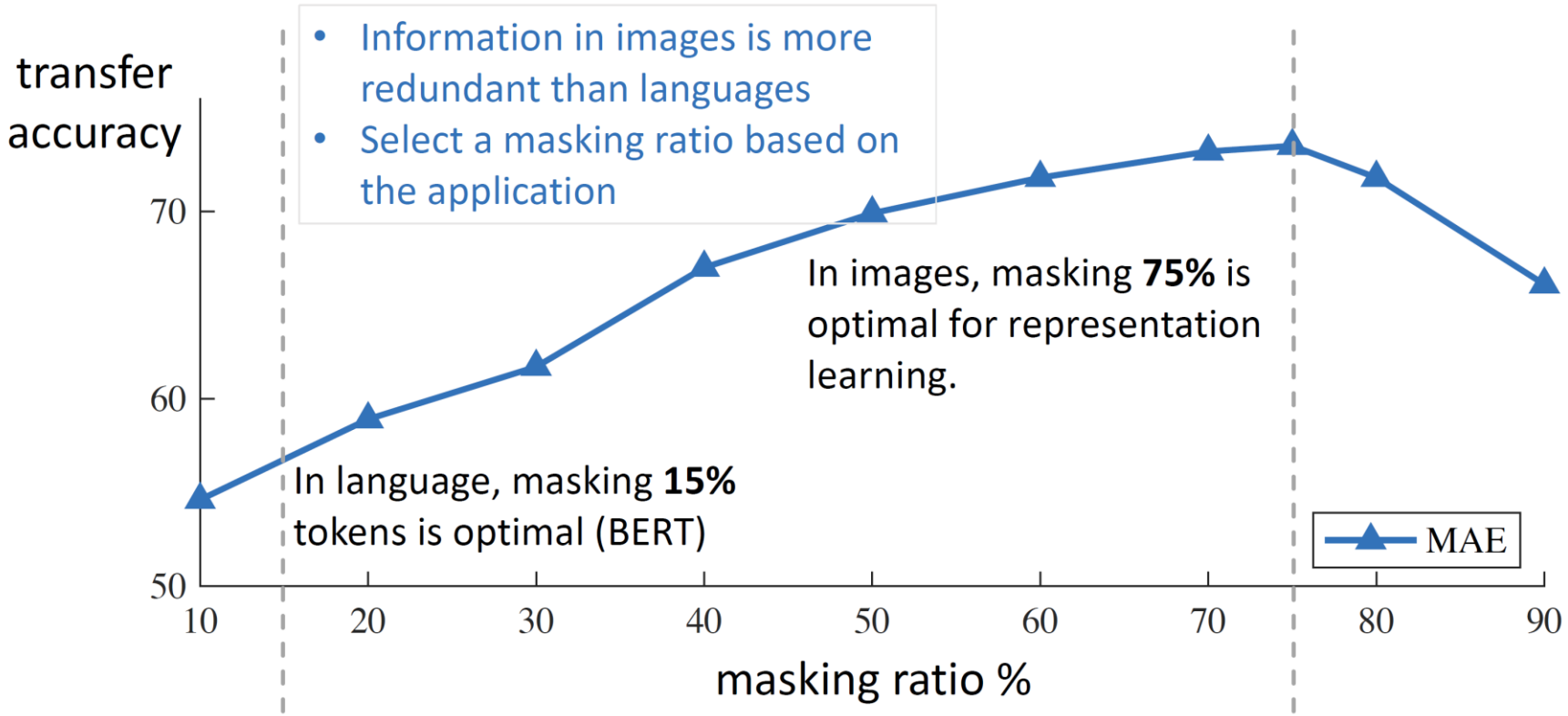
How to learn good representations by predicting?



He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

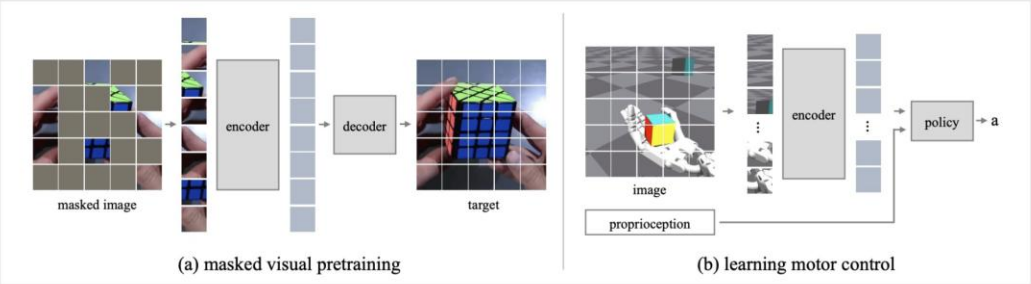
Predictive Learning: Computer Vision

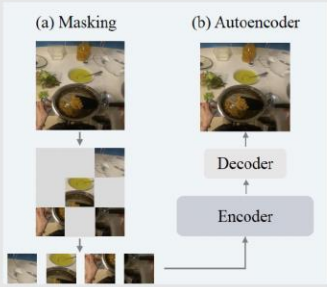
How to learn good representations by predicting?

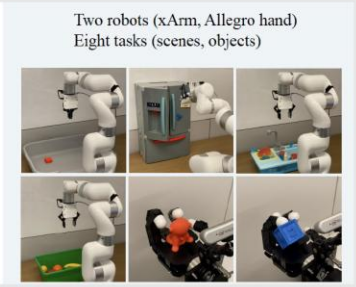


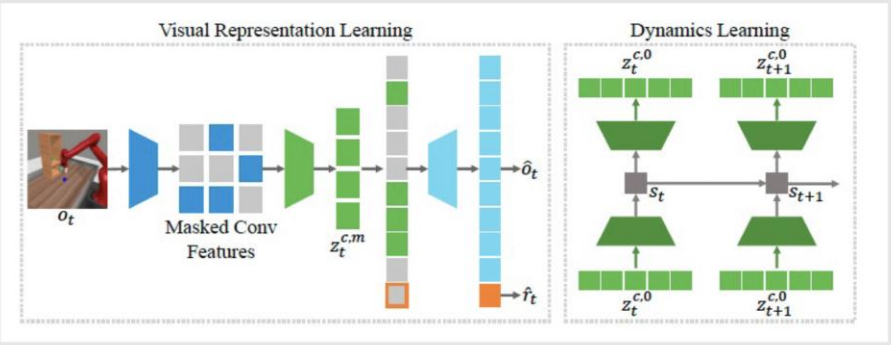
He, et al., "Masked Autoencoders Are Scalable Vision Learners", CVPR 2022

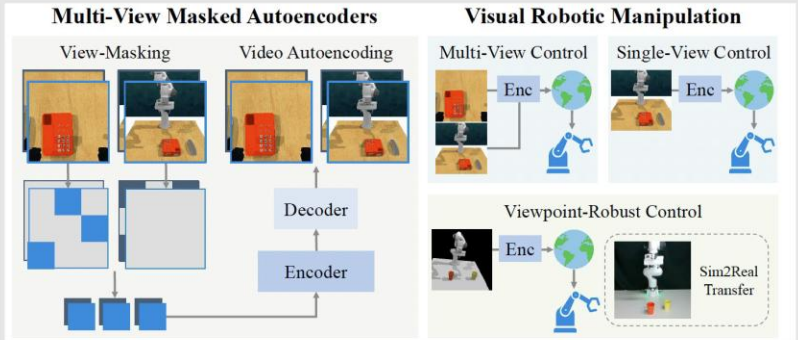
Masked Autoencoder: Extensions







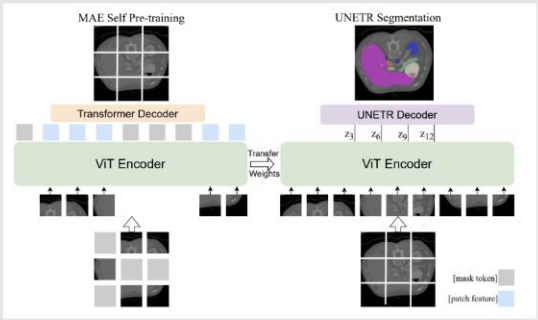


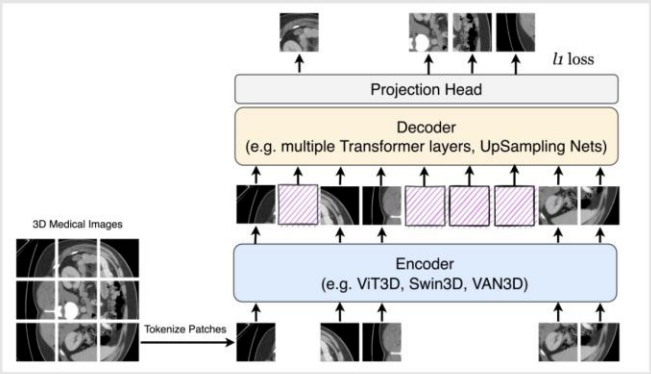


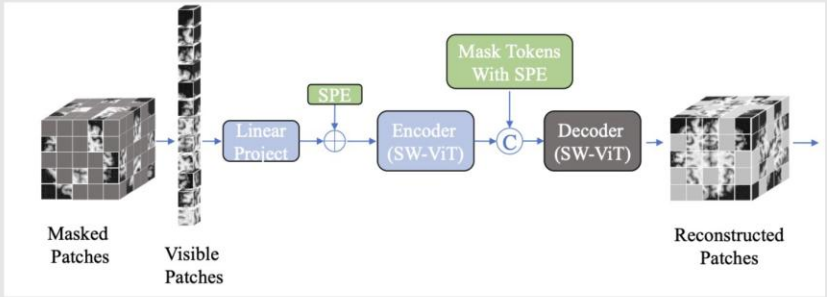
- **Robotics**

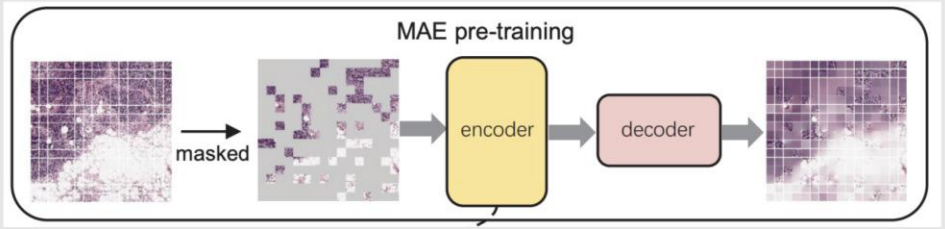
Xiao+, ’22; Radosavovic+, ’22; Seo+, ’22; Seo+, ’23

Masked Autoencoder: Extensions





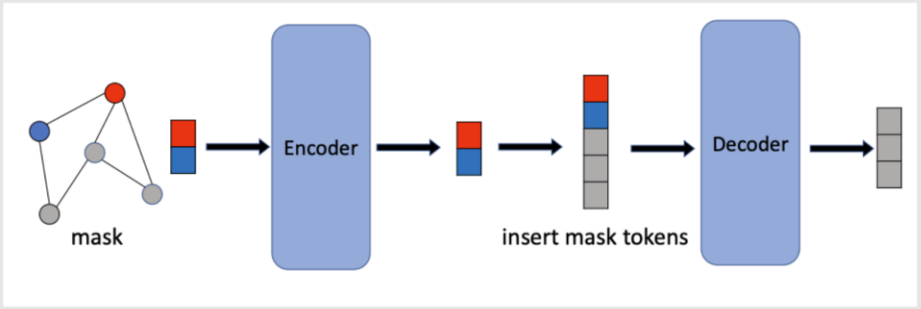


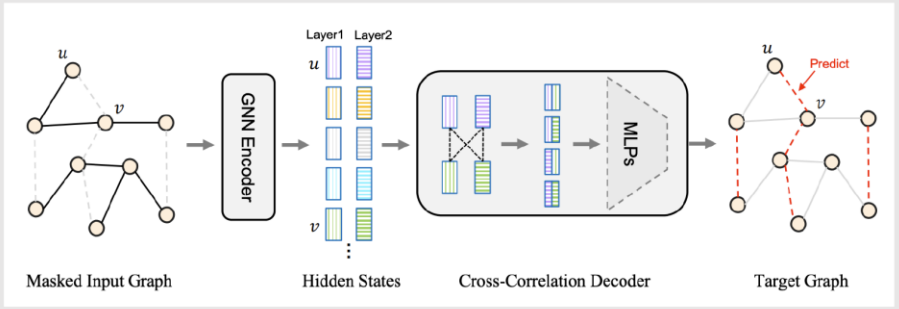


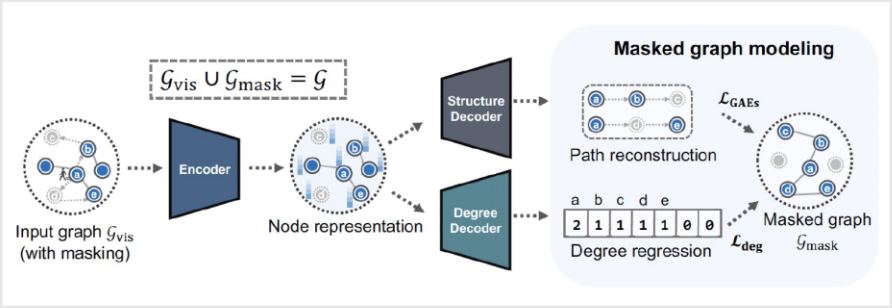
- **Medical Imaging**

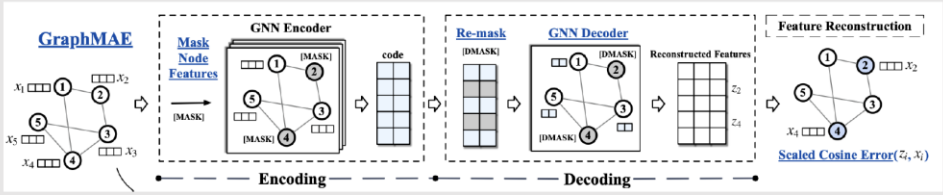
Zhou+,'22; Chen+,'22; Huang+,'22; An+,'22

Masked Autoencoder: Extensions





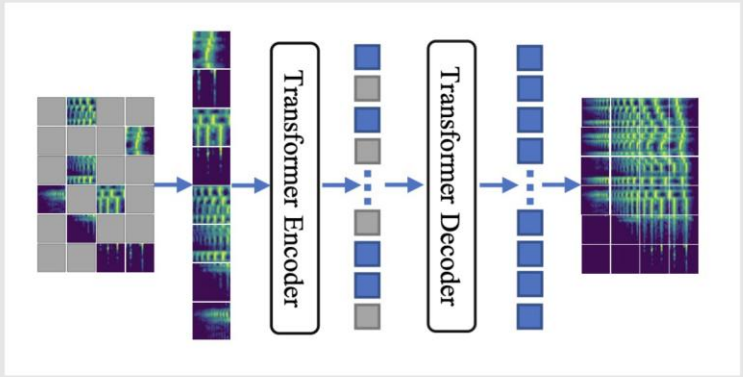


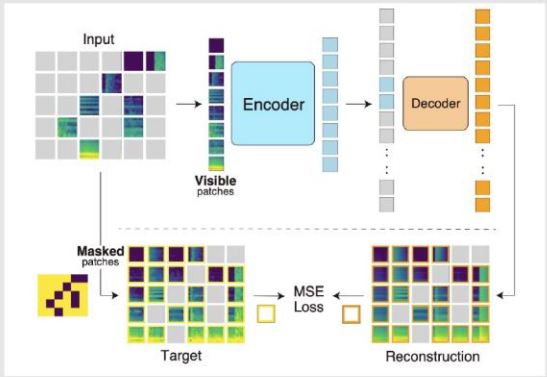


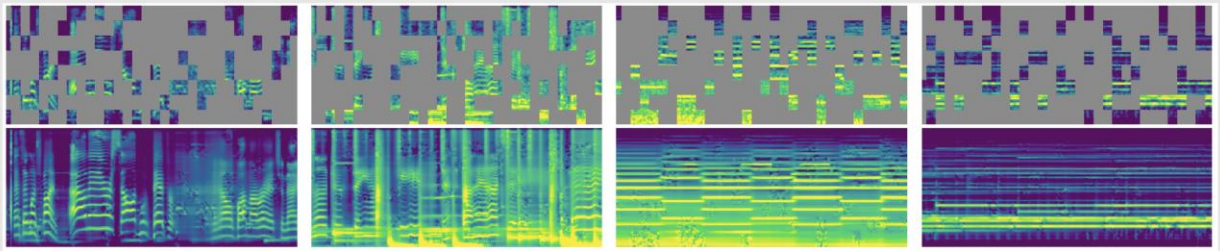
- **Graphs**

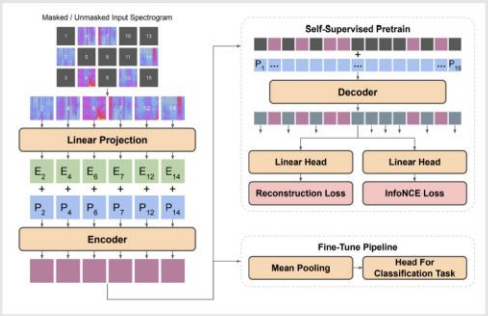
Tan+, '22; Zhang+, '22; Hou+, '22; Li+, '22

Masked Autoencoder: Extensions





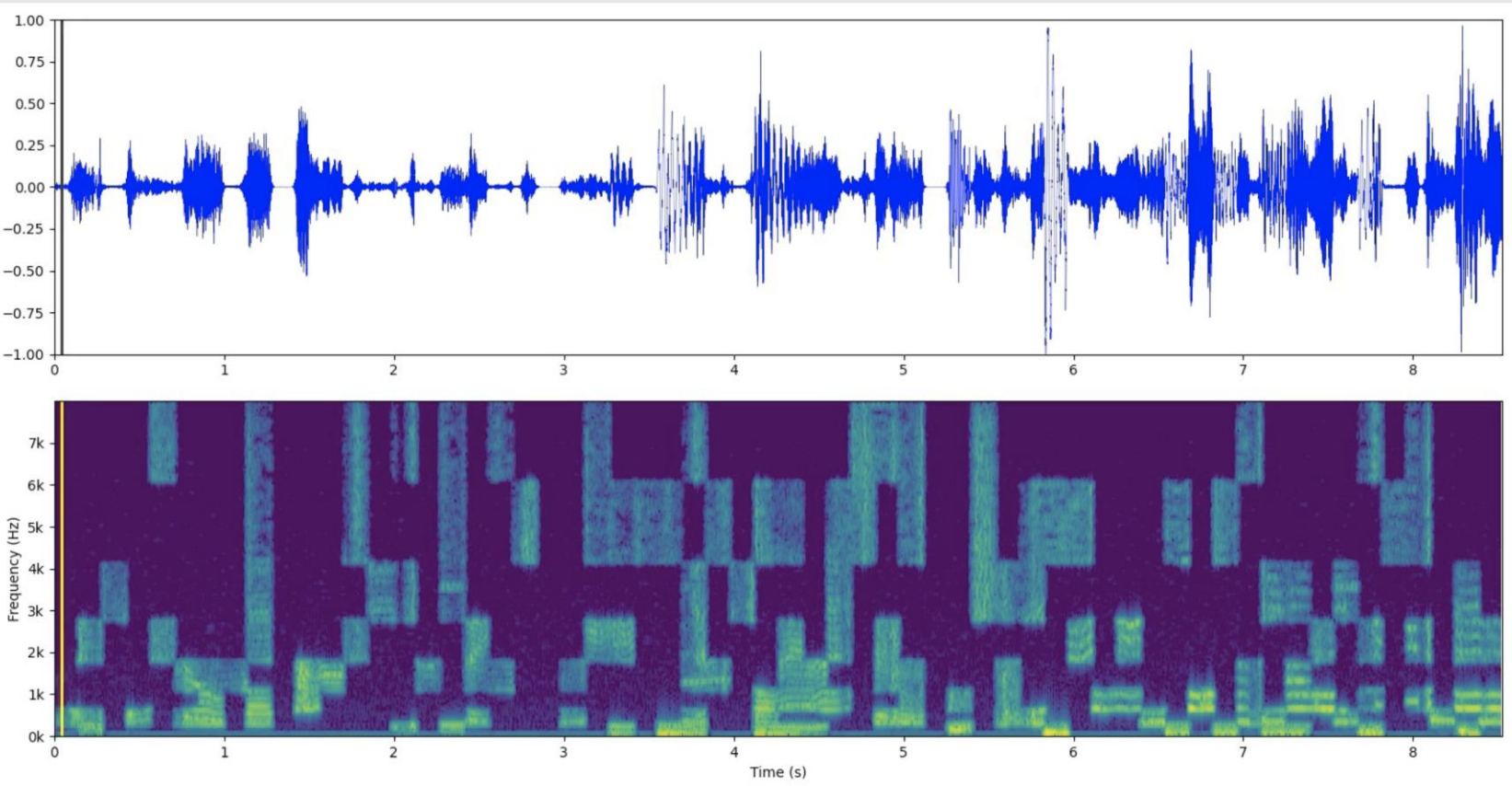




- **Audio**

Baade+, ’22; Chong+, ’22; Niizumi, ’22; Huang+, ’22

Masked Autoencoder: Extensions



Audio MAE

Huang, et al., *Masked Autoencoders that Listen*. NeurIPS '22

Suggested Reading

- **Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey**
  - **https://arxiv.org/abs/1902.06162**
- **Self-supervised Learning: Generative or Contrastive**
  - **https://arxiv.org/abs/2006.08218**
- **Contrastive Self-Supervised Learning**
  - **https://ankeshanand.com/blog/2020/01/26/contrative-self-supervised-learning.html**

下周课见！