SWINBURNE UNIVERSITY OF TECHNOLOGY

COS30018 – Intelligent Systems: Week 10
Self-Supervised Learning (SSL)

1

SWINBURNE UNIVERSITY OF TECHNOLOGY

Outline

- What is Self-Supervised Learning (SSL)
  - Motivation, basic concepts, examples
- Self-supervised learning pretext tasks
  - Self-prediction
  - Contrastive learning
- Transferring a pretrained model to applications
  - Example applications

2

Self-Supervised Learning (SSL)

Self-Supervised Learning (SSL) is a special type of representation learning that enables learning good data representation from unlabelled dataset.

Idea: *constructing supervised learning tasks out of unsupervised datasets.*

Source: https://kvaes.wordpress.com/2013/05/31/data-knowledge-information-wisdom/

3

Self-Supervised Learning (SSL)

Self-Supervised Learning (SSL) is a special type of representation learning that enables learning good data representation from unlabelled dataset.

Idea: *constructing supervised learning tasks out of unsupervised datasets.* **WHY?**

1. Data labelling is expensive and thus high-quality labelled dataset is limited.

4

Self-Supervised Learning (SSL)

Self-Supervised Learning (SSL) is a special type of representation learning that enables learning good data representation from unlabelled dataset.

Idea: *constructing supervised learning tasks out of unsupervised datasets.* **WHY?**

1. Data labelling is expensive and thus high-quality labelled dataset is limited.

But there is a huge amount of (unlabelled) data available
(e.g., trillions of documents on the web, trillions of images and videos, etc.)
→ Can we distil these data to obtain common knowledge (e.g., representation) that can be used in various tasks??

5

Self-Supervised Learning (SSL)

Self-Supervised Learning (SSL) is a special type of representation learning that enables learning good data representation from unlabelled dataset.

Idea: *constructing supervised learning tasks out of unsupervised datasets.* **WHY?**

1. Data labelling is expensive and thus high-quality labelled dataset is limited.

2. Learning good representation makes it easier to **transfer** useful information to a variety of **downstream tasks**.
   - e.g. A downstream task has only a few examples.
   - e.g. Zero-shot transfer to new tasks.

Self-supervised learning tasks are also known as *pretext tasks*.

6

Self-Supervised Learning (SSL)

Self-Supervised Learning (SSL) is a special type of representation learning that enables learning good data representation from unlabelled dataset.

Idea: *constructing supervised learning tasks out of unsupervised datasets.* **WHY?**

2. Learning good representation makes it easier to **transfer** useful information to a variety of **downstream tasks**.

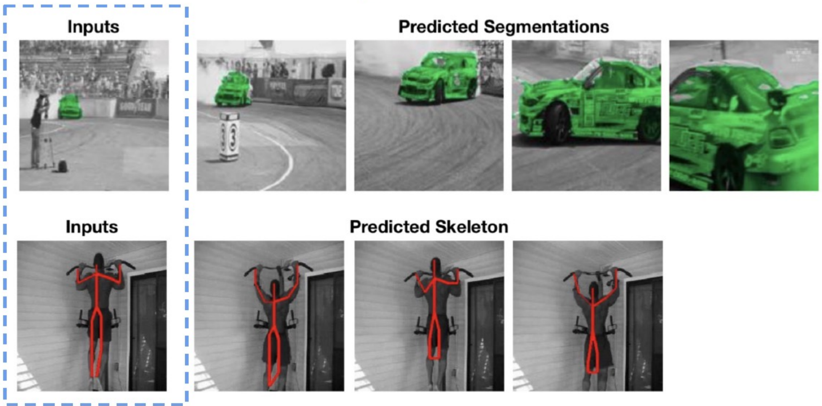
Good representation ~ pretrained (large) model/foundation model

Transfer to downstream tasks ~ adaptation

7

What's Possible with Self-Supervised Learning?

Video colorization (Vondrick et al 2018), as a self-supervised learning method, resulting in a rich representation that can be used for video segmentation and unlabelled visual region tracking, without extra fine-tuning.

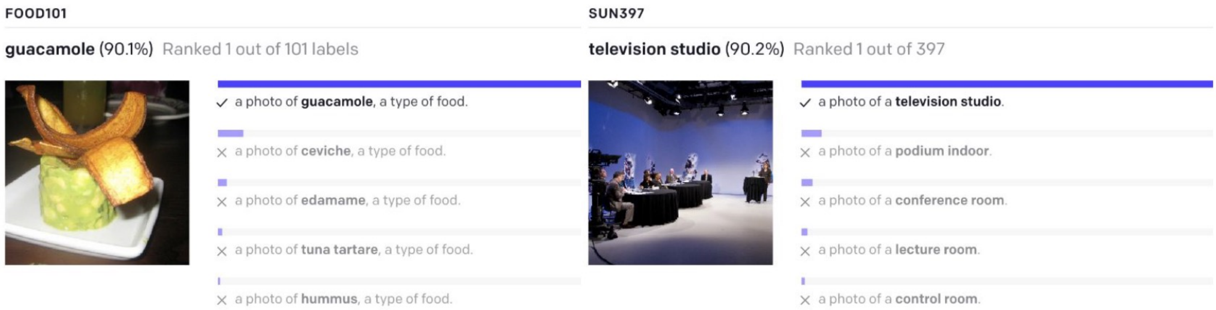


Source: OpenAI's NeurIPS 2021 Tutorial on SSL

8

What's Possible with Self-Supervised Learning?

Despite of not training on supervised labels, the zero-shot CLIP (Radford et al. 2021) classifier achieve great performance on challenging image-to-text classification tasks.

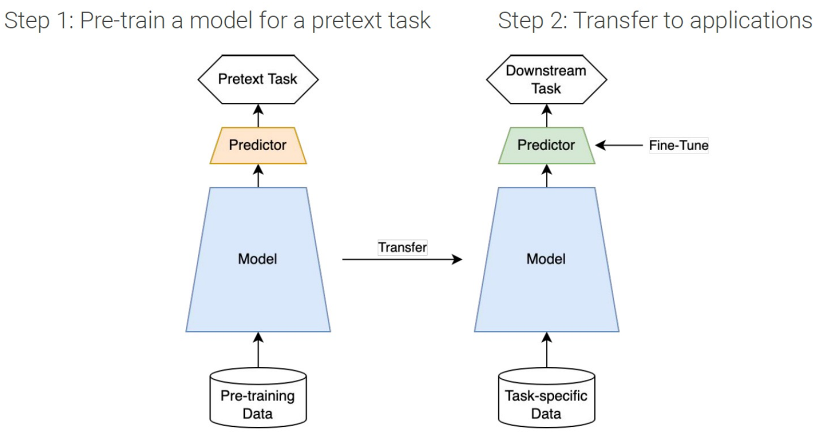


Source: OpenAI's NeurIPS 2021 Tutorial on SSL

9

Self-Supervised Learning (SSL)

Self-supervised learning tasks are also known as *pretext tasks*.



10

Self-Supervised Learning (SSL)

pretext tasks




11

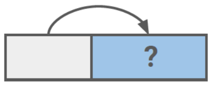
Self-Supervised Learning (SSL)

pretext tasks


12

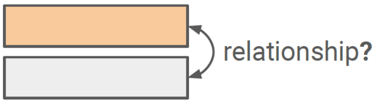
Self-Supervised Learning (SSL) - *pretext tasks*

- **Self-prediction**: Given an individual data sample, the task is to predict one part of the sample given the other part.
  - The part to be predicted pretends to be missing.



"Intra-sample" prediction

- **Contrastive learning**: Given multiple data samples, the task is to predict the relationship among them.
  - The multiple samples can be selected from the dataset based on some known logics (e.g. the order of words / sentences), or fabricated by altering the original version.



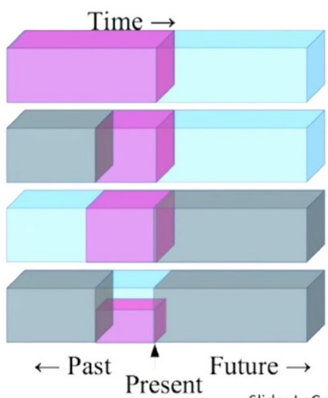
"Inter-sample" prediction

13

Self-Prediction

- Self-prediction construct prediction tasks within every individual data sample: to predict a part of the data from the rest while pretending we don't know that part.

- **Predict any part of the input from any other part.**
- **Predict the future from the past.**
- **Predict the future from the recent past.**
- **Predict the past from the present.**
- **Predict the top from the bottom.**
- **Predict the occluded from the visible**
- **Pretend there is a part of the input you don't know and predict that.**



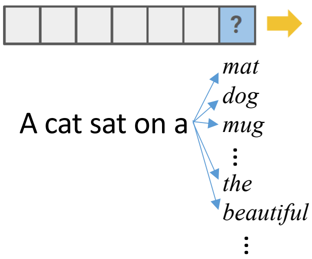
Slide: LeCun

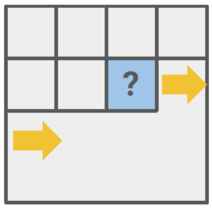
(Famous illustration from Yann LeCun)

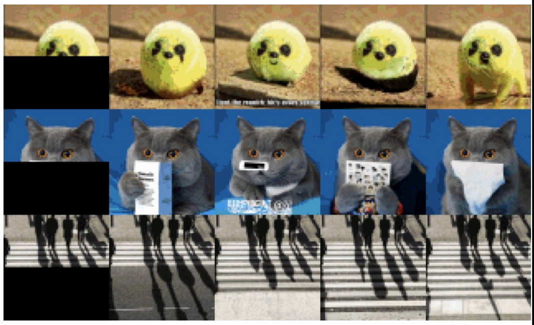
14

Self-Prediction: Autoregressive Generation

- The autoregressive model predicts future behavior based on past behavior. Any data that comes with an innate sequential order can be modeled with regression.





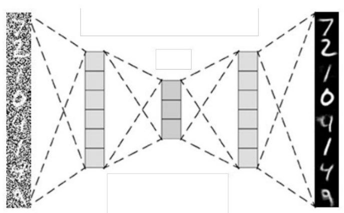


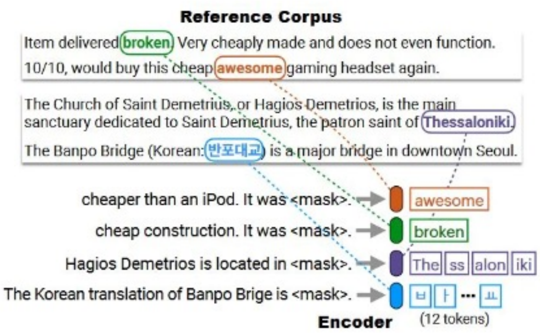
Examples:
- Audio (WaveNet, WaveRNN)
- Autoregressive language modeling (GPT, XLNet)
- Images in raster scan (PixelCNN, PixelRNN, iGPT)

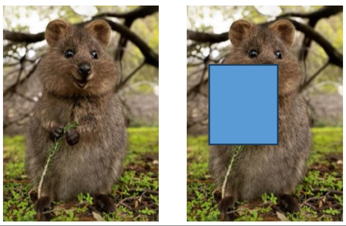
15

Self-Prediction: Masked Generation

- We mask a random portion of information and pretend it is missing, irrespective of the natural sequence. The model learns to predict the missing portion given other unmasked information.





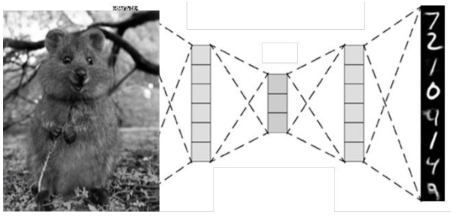


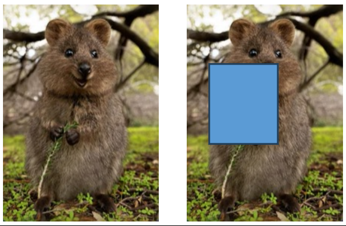
Examples:
- Masked language modeling (BERT)
- Images with masked patch (denoising autoencoder, context autoencoder, colorization)

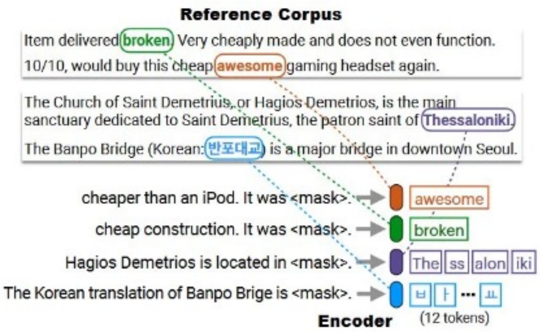
16

Self-Prediction: Masked Generation

- We mask a random portion of information and pretend it is missing, irrespective of the natural sequence. The model learns to predict the missing portion given other unmasked information.





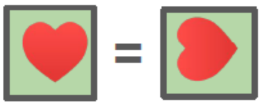


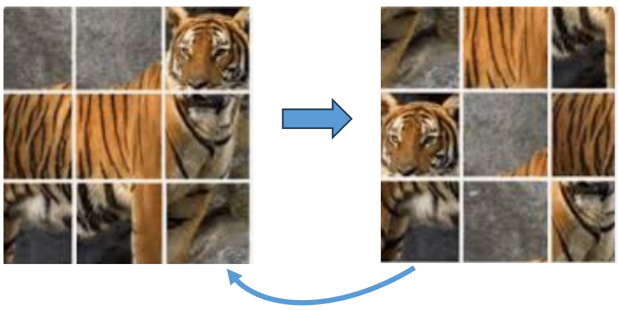
Examples:
- Masked language modeling (BERT)
- Images with masked patch (denoising autoencoder, context autoencoder, colorization)

17

Self-Prediction: Innate Relationship Prediction

- Some transformation (e.g. segmentation, rotation) of one data sample should maintain the original information or follow the desired innate logic.



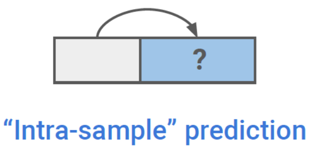


Examples:
- Order of image patches (e.g., relative position, jigsaw puzzle)
- Image rotation
- Counting features across patches

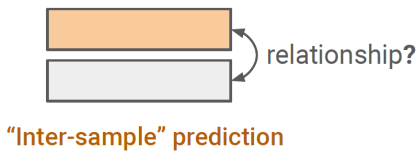
18

Self-Supervised Learning (SSL) - *pretext tasks*

- **Self-prediction**: Given an individual data sample, the task is to predict one part of the sample given the other part.
  - The part to be predicted pretends to be missing.



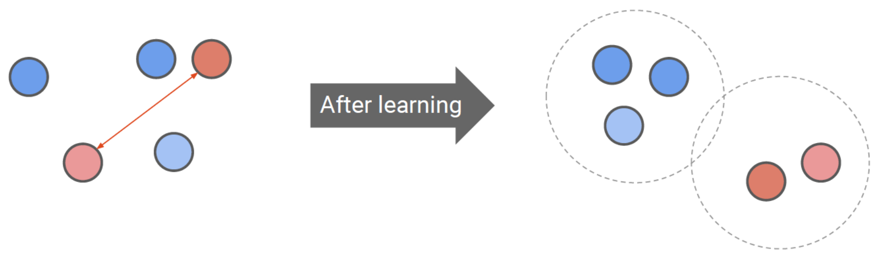
- **Contrastive learning**: Given multiple data samples, the task is to predict the relationship among them.
  - The multiple samples can be selected from the dataset based on some known logics (e.g. the order of words / sentences) or fabricated by altering the original version.



19

Contrastive Learning

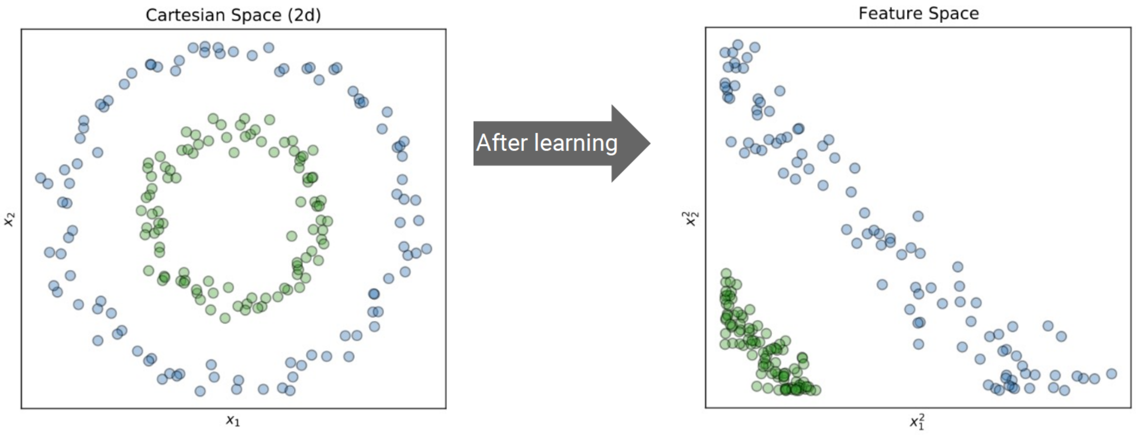
- The goal of contrastive representation learning is to learn such an embedding space in which *similar* sample pairs stay *close* to each other while *dissimilar* ones are *far apart*.



20

Contrastive Learning

- The goal of contrastive representation learning is to learn such an embedding space in which *similar* sample pairs stay *close* to each other while *dissimilar* ones are *far apart*.



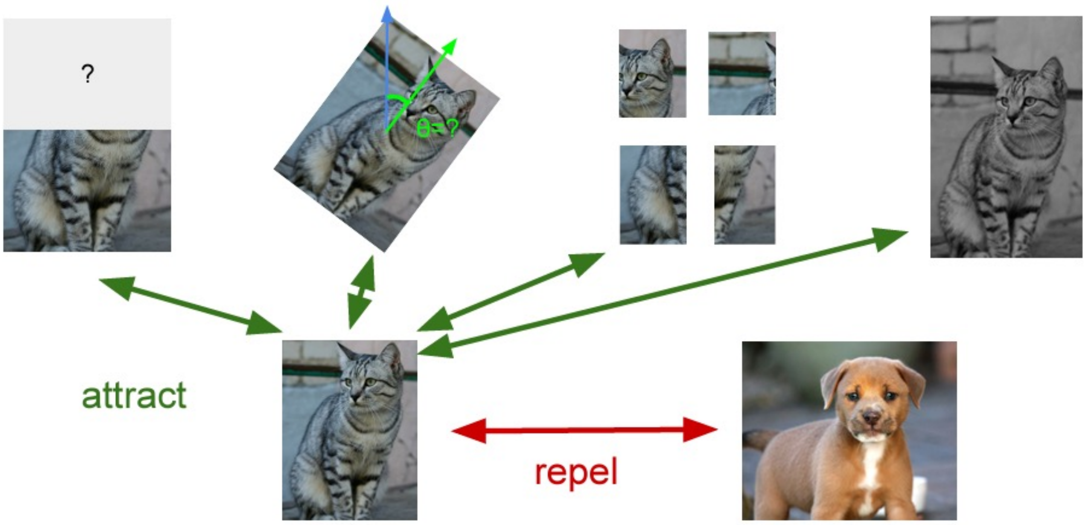
21

Contrastive Learning: Inter-Sample Classification

- Given both similar ("positive") and dissimilar ("negative") candidates, to identify which ones are similar to the anchor data point is a *classification* task.
- There are creative ways to construct a set of data point candidates:
  1. The original input and its distorted version
  2. Data that captures the same target from different views

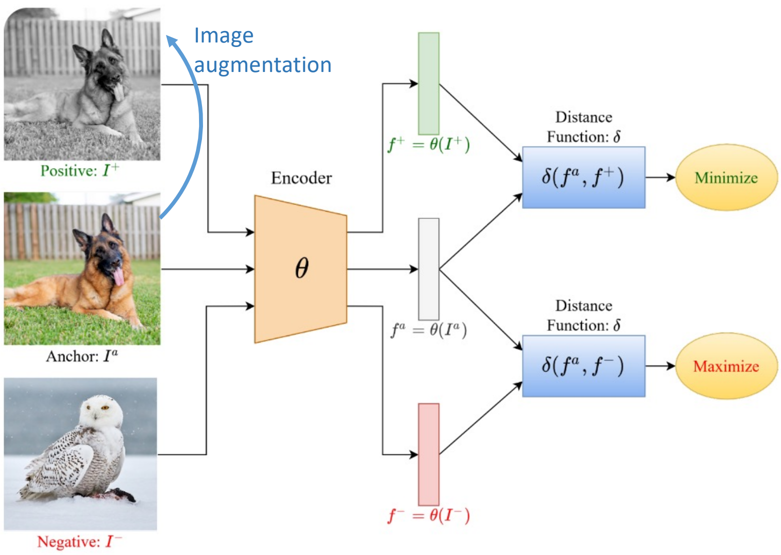
22

Contrastive Learning: Intuition



23

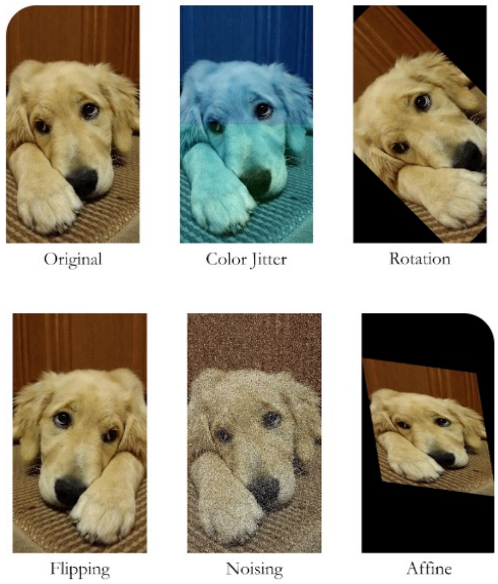
Contrastive Learning: Inter-Sample Classification



24

Image augmentation methods:

Contrastive Learning: Inter-Sample Classification



25

Contrastive Learning: Inter-Sample Classification

Common loss functions:

- Contrastive loss (Chopra et al. 2005)
- Triplet loss (Schroff et al. 2015; FaceNet)
- Lifted structured loss (Song et al. 2015)
- Multi-class n-pair loss (Sohn 2016)
- Noise contrastive estimation (“NCE”; Gutmann & Hyvarinen 2010)
- InfoNCE (van den Oord, et al. 2018)
- Soft-nearest neighbors loss (Salakhutdinov & Hinton 2007, Frosst et al. 2019)

26

Contrastive Learning: Inter-Sample Classification - Formulation idea

- What we want:

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

- x: anchor;
- 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^-).

27

Contrastive Learning: Inter-Sample Classification

- **N-pair loss** (Sohn 2016) generalizes triplet loss to include comparison with multiple negative samples.
- Given one positive and N-1 negative samples,

$$\{\mathbf{x}, \mathbf{x}^+, \mathbf{x}_1^-, \ldots, \mathbf{x}_{N-1}^-\}$$

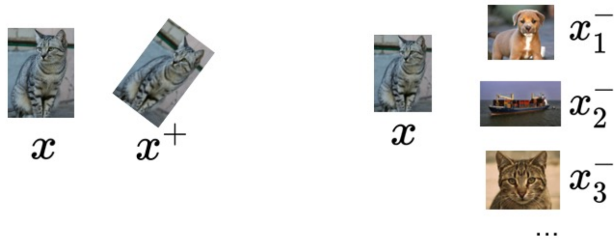
$$\begin{aligned}\mathcal{L}_{\text{N-pair}}(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{N-1}) &= \log\Big(1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-) - f(\mathbf{x})^\top f(\mathbf{x}^+))\Big) \\ &= -\log \frac{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))}{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+)) + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-))}\end{aligned}$$

28

Contrastive Learning: Inter-Sample Classification

- **N-pair loss** (Sohn 2016) generalizes triplet loss to include comparison with multiple negative samples.

$$\begin{aligned}\mathcal{L}_{\text{N-pair}}(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{N-1}) &= \log\left(1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-) - f(\mathbf{x})^\top f(\mathbf{x}^+))\right) \\ &= -\log \frac{\overline{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))}}{\underline{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))} + \underline{\sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-))}}\end{aligned}$$



29

Contrastive Learning: Inter-Sample Classification

- **N-pair loss** (Sohn 2016) generalizes triplet loss to include comparison with multiple negative samples.

$$\begin{aligned}\mathcal{L}_{\text{N-pair}}(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{N-1}) &= \log\left(1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-) - f(\mathbf{x})^\top f(\mathbf{x}^+))\right) \\ &= -\log \frac{\overline{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))}}{\underline{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))} + \underline{\sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-))}}\end{aligned}$$

- $\mathcal{L}$ is decreasing in the green term and increasing in the red term.
- Thus, to minimise the loss $\mathcal{L}$, we have to maximise the green term (similarity to the positive sample) and minimise the red term (similarity to the negative samples)

30

Contrastive Learning: Inter-Sample Classification

- **N-pair loss** (Sohn 2016) generalizes triplet loss to include comparison with multiple negative samples.

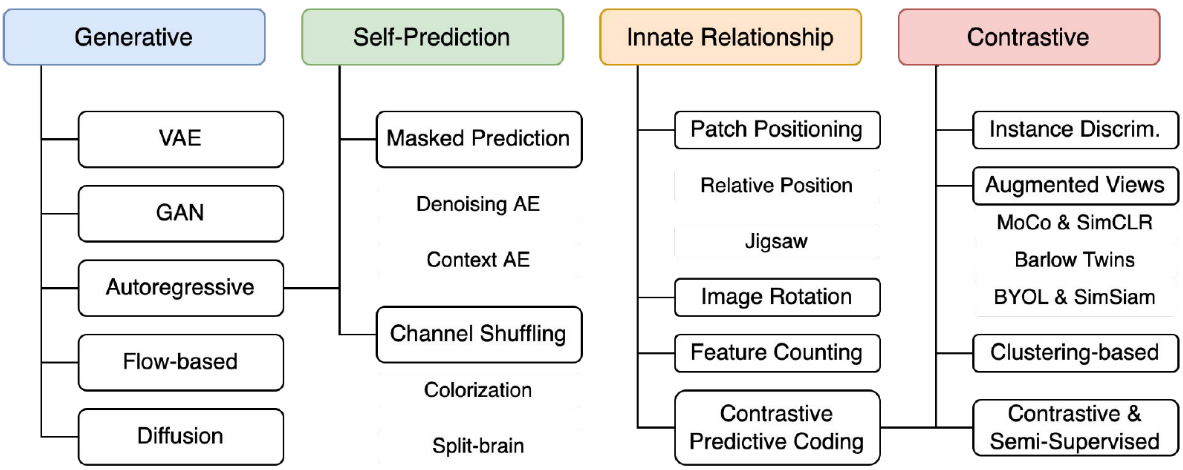
$$\mathcal{L}_{\text{N-pair}}(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{N-1}) = \log\left(1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-) - f(\mathbf{x})^\top f(\mathbf{x}^+))\right)$$

$$= -\log \frac{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))}{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+)) + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-))}$$

This is:
Cross entropy loss for a *N*-way softmax classifier!
i.e., learn to find the positive sample from the *N* samples

31

Pretext Tasks: Taxonomy



Source: OpenAI’s NeurIPS 2021 Tutorial on SSL

32

Adapting a FM to the downstream tasks

- In the adaptation phase, we train a new model that depends on pre-trained Foundation Model (FM) parameters θ that parameterize the FM Φ_θ
- We are given a downstream dataset $(x^{(1)},y^{(1)}),\ldots,(x^{(n)},y^{(n)})$ sampled from a downstream task distribution P_{task}
- We minimize some parameters γ from a family of parameters Γ on a task loss ℓ_{task} (e.g., *cross entropy* loss).
- The family of parameters Γ may represent a subset of the existing parameters or introduce new parameters.
- The output of the optimization problem are the adapted parameters γ_{adapt}, which parameterizes the adapted model Φ_{adapt}:

$$\gamma_{adapt} = \arg\min_{\gamma\in\Gamma} \frac{1}{n}\sum_{i=1}^{n} \ell_{task}\,(\gamma, \theta, x^{(i)}, y^{(i)})$$

33

Applications

- Example:
  - Text-to-Image Diffusion Models

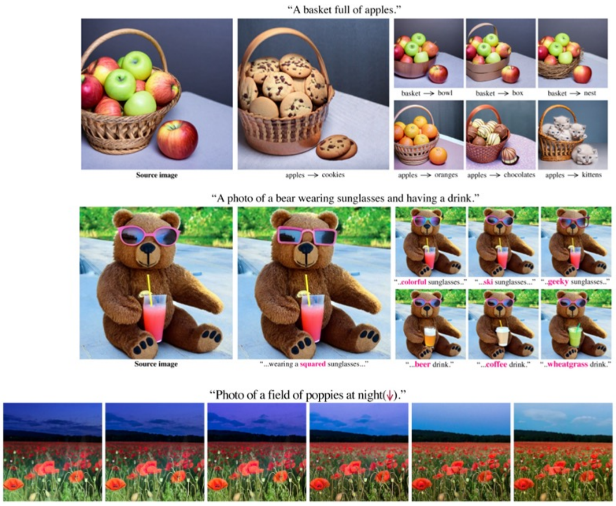
34

Prompt-to-Prompt edits high-quality images with only text modification

Word Swap

Prompt Refinement

Attention Re-weighting



Text-to-Image Diffusion Models

35

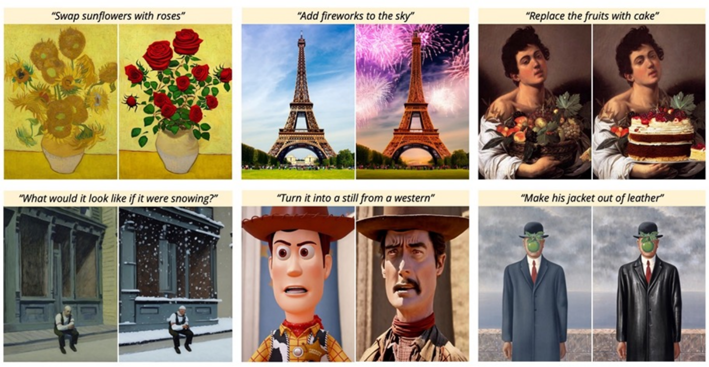
InstructPix2Pix [Brooks et al., 2023]

InstructPix2Pix: Learning to Follow Image Editing Instructions [Brooks et al., 2023]

Motivation: Image editing with **detailed prompt** or **extra information** are cumbersome

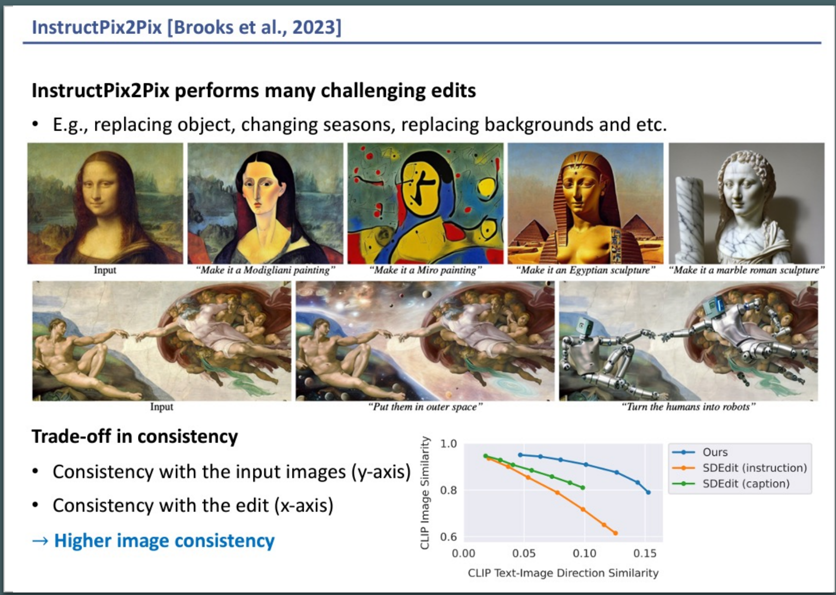
💡 How about editing images with **human instructions** (e.g., make it big)?

Contribution: Fine-tune a generative model to follow **human instructions**



Text-to-Image Diffusion Models

36



Text-to-Image Diffusion Models

37

Summary

- Self-supervised learning (SSL)
  - revolutionizes AI & ML by taking advantage of the large amounts of (unlabelled) data available
- SSL typically consists of two phases:
  - Pretraining
    - To obtain a pretrained model (aka. foundation model)
  - Adaptation
    - To customize the model to a downstream task
- SSL tasks (during pretraining phase) are also known as ***pretext tasks***.
  - Important pretext tasks: self-prediction and contrastive learning

38