Multimodal Foundation Models 1

Self-supervised Learning

Sangdoo Yun and Jin-Hwa Kim

NAVER AI LAB

Today's lecture

- Contents
  - Pre-training
  - Supervised Learning
  - Self-supervised Learning

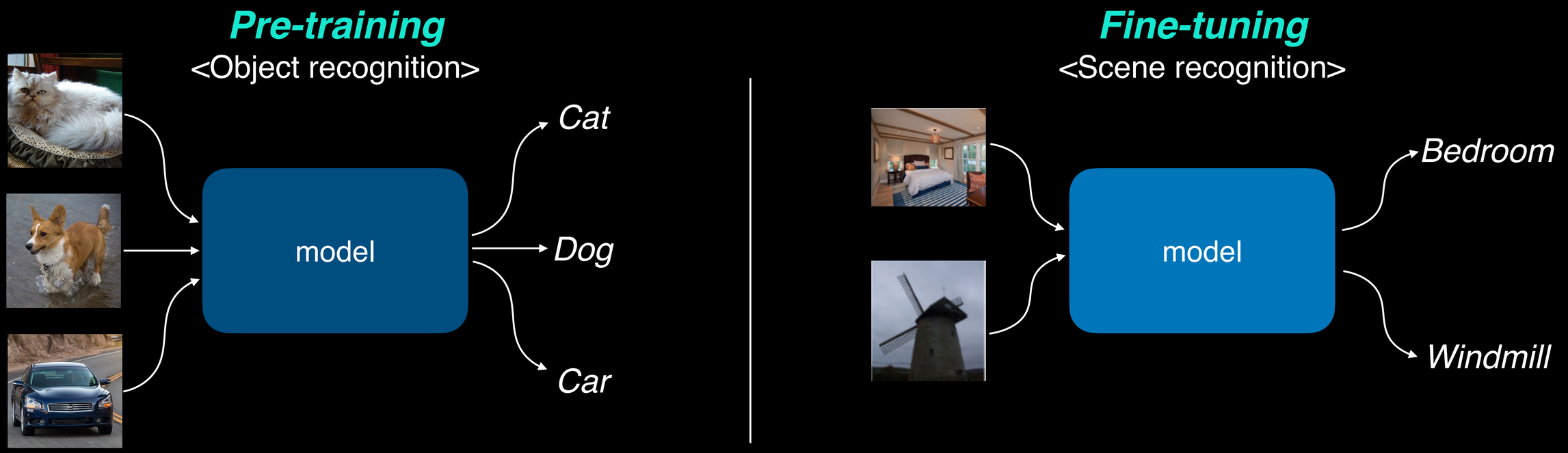
Multimodal Pre-training

Pre-training

- The question is: how are Deep Learning models learned?
- Usually, we use *pre-training* and *fine-tuning* paradigm

Pre-training

- *Pre-training* and *fine-tuning* paradigm on vision models



Pre-training

- *Pre-training* and *fine-tuning* paradigm on multimodal models

Pre-training
<Matching image-text pair>

model

True?

False?

“The man at bat readies to swing at the pitch while the umpire looks on.”

→ Large-scale data, simple task

Fine-tuning
<Visual question answering>

model

banana

“What is the mustache made of?”

→ Small data, complex task

Pre-training

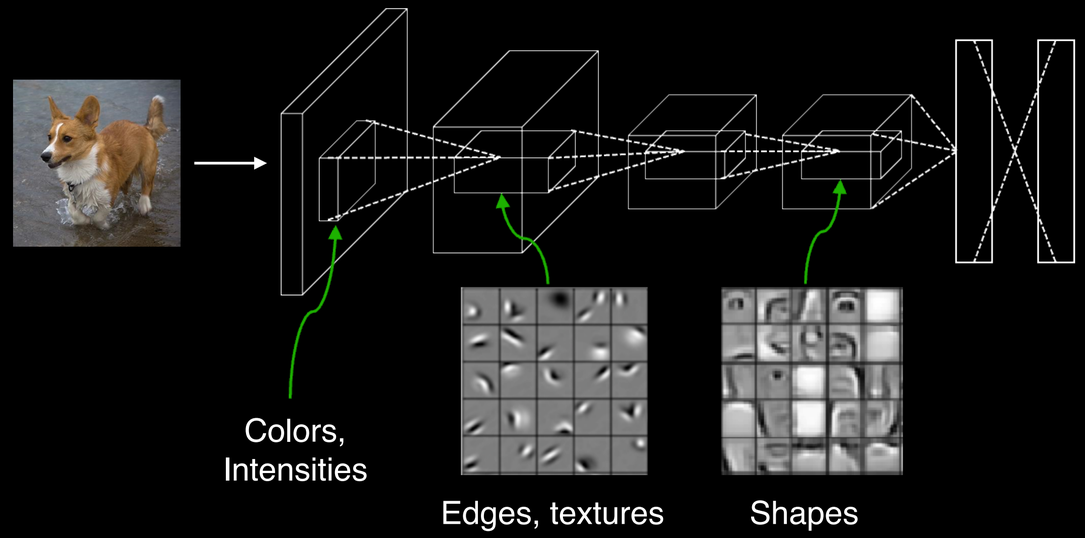
- Today's goal is to understand (general) *pre-training*
- We will learn multimodal pre-training next week 😅

Representation learning

- Continuing from Lecture 2, “Multimodal Representation Learning”
- The goal of pre-training: obtain good representation ability
- What is representation?

Representation learning

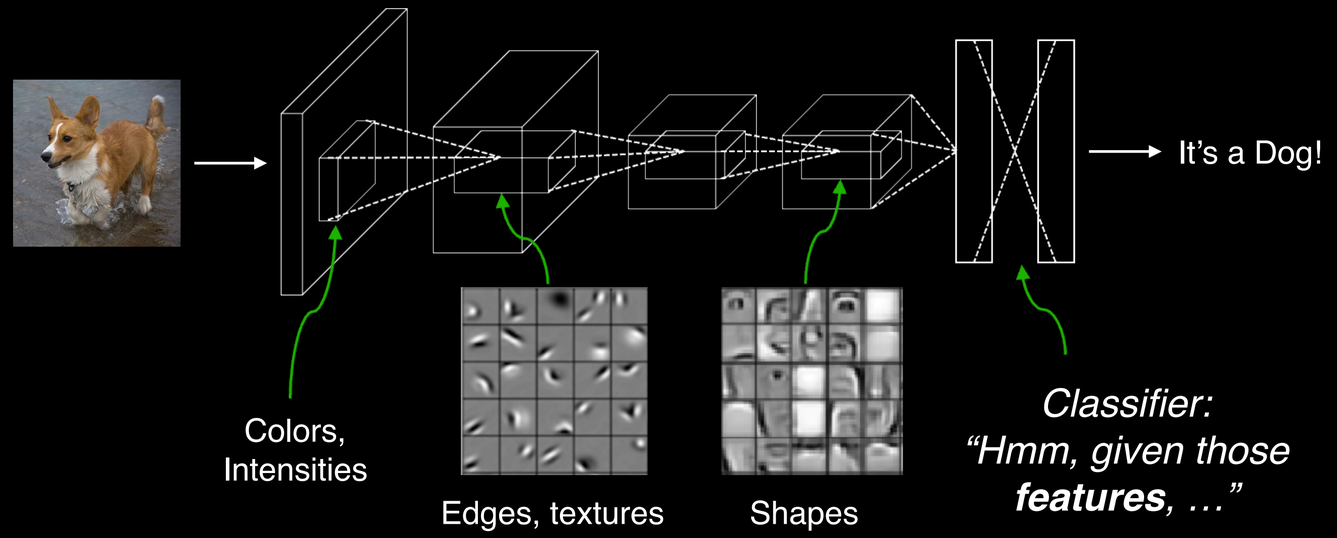
- Continuing from Lecture 3, “Multimodal Representation Learning”
- The goal of pre-training: obtain good representation ability
- What is representation?



Lee et al., "Unsupervised learning of hierarchical representations with convolutional deep belief networks." 2011.

Representation learning

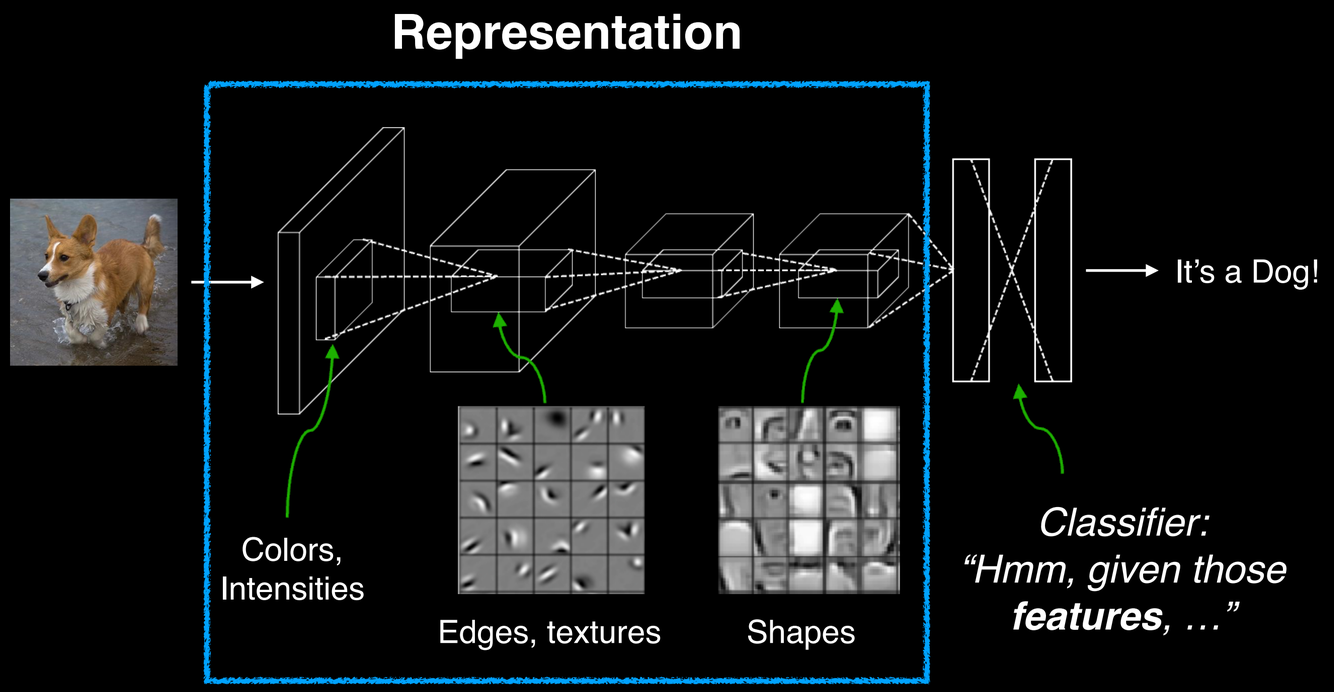
- Continuing from Lecture 3, “Multimodal Representation Learning”
- The goal of pre-training: obtain good representation ability
- What is representation?



Lee et al., "Unsupervised learning of hierarchical representations with convolutional deep belief networks." 2011.

Representation learning

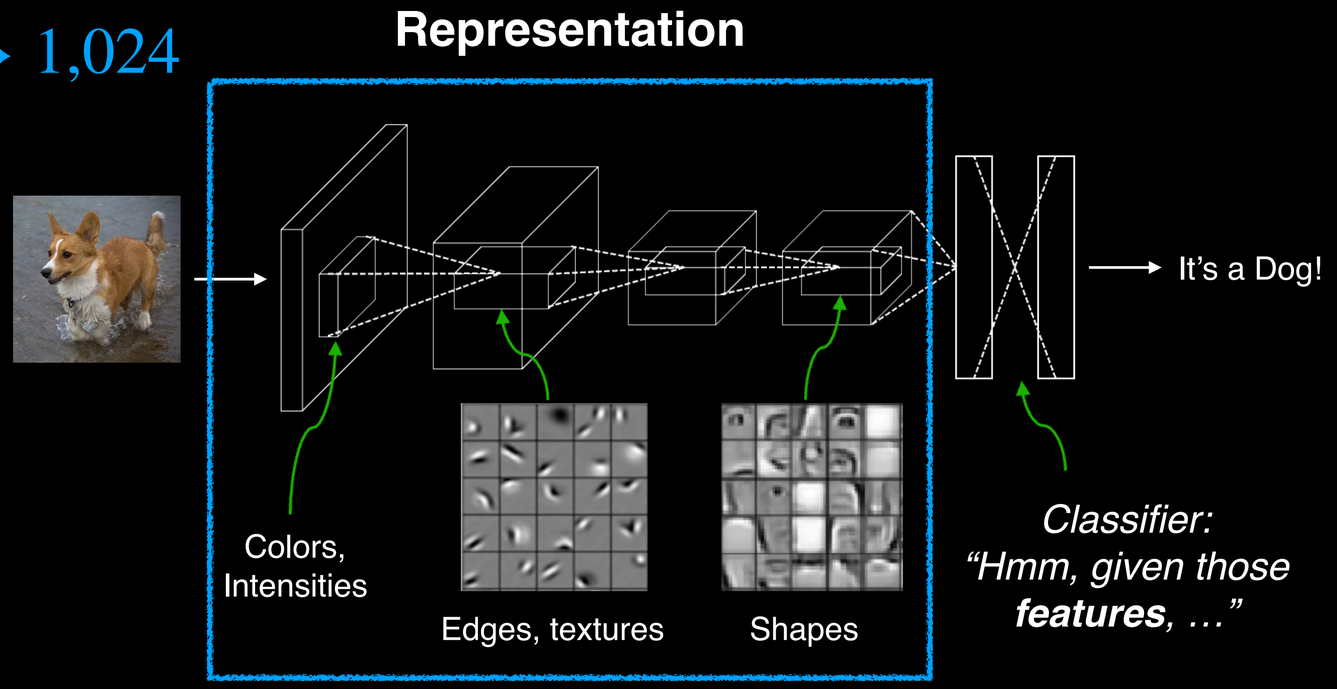
- Continuing from Lecture 3, “Multimodal Representation Learning”
- The goal of pre-training: obtain good representation ability
- What is representation?



Lee et al., "Unsupervised learning of hierarchical representations with convolutional deep belief networks." 2011.

Representation learning

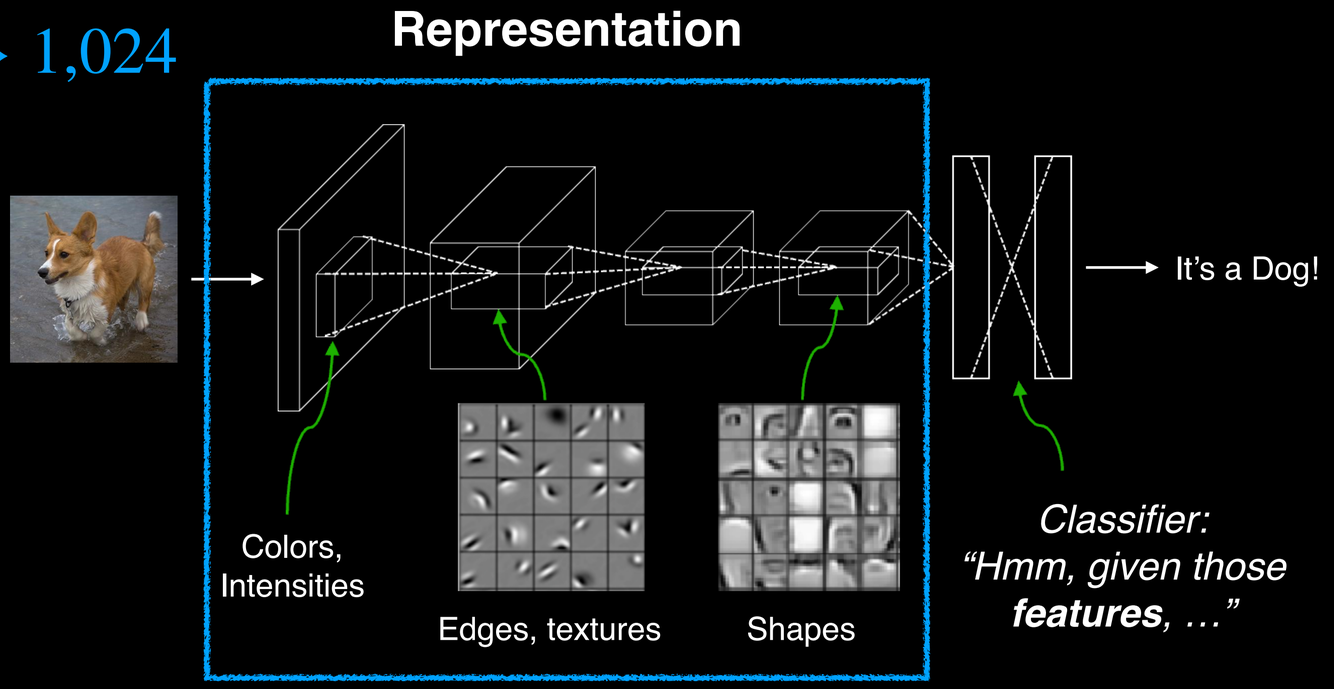
- Continuing from Lecture 3, “Multimodal Representation Learning”
- The goal of pre-training: obtain good representation ability
- What is representation?
  - Image space: $256^{3\times300\times300} \rightarrow 1{,}024$



Lee et al., "Unsupervised learning of hierarchical representations with convolutional deep belief networks." 2011.

Representation learning

- Continuing from Lecture 3, “Multimodal Representation Learning”
- The goal of pre-training: obtain good representation ability
- What is representation?
  - Image space: $256^{3\times300\times300} \rightarrow 1{,}024$
  - Compact vector
  - Represents input contents
  - Can transfer to other tasks



Lee et al., "Unsupervised learning of hierarchical representations with convolutional deep belief networks." 2011.

Supervised Learning

Supervised learning

- A classification model learns representation.

Supervised learning

- A classification model learns representation.
- Data

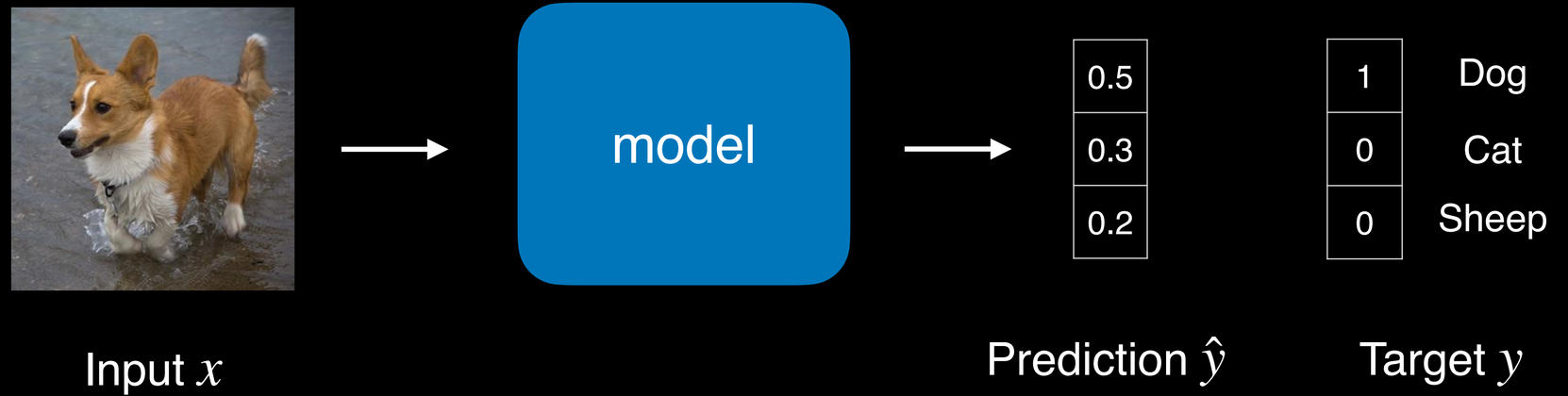
Input x

1	Dog
0	Cat
0	Sheep

Target y

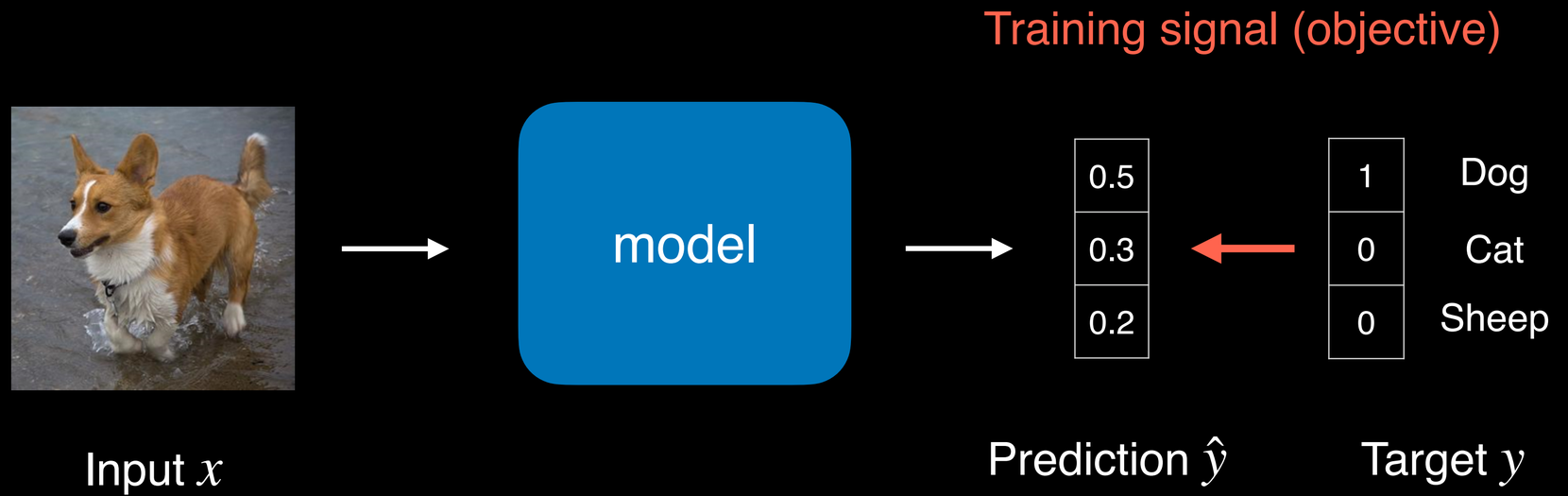
Supervised learning

- A classification model learns representation.
- Data, model



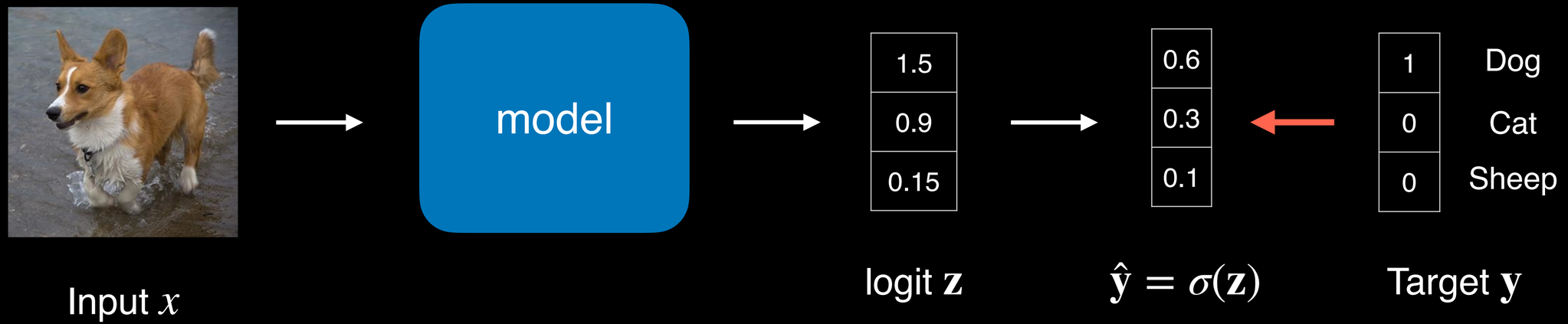
Supervised learning

- A classification model learns representation.
- Data, model, objective



Supervised learning

- Objective



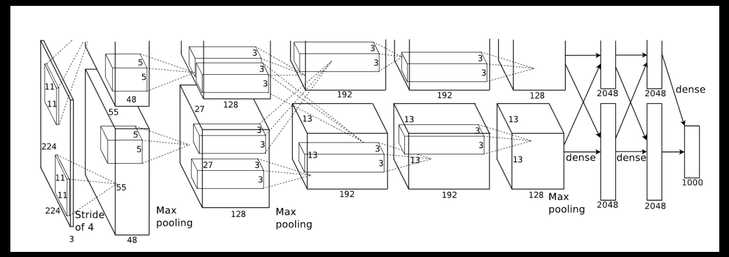
Sigmoid: $\hat{y}_i = \dfrac{1}{1 + e^{-z_i}}$

Softmax: $\hat{y}_i = \dfrac{e^{z_i}}{\sum_j e^{z_j}}$

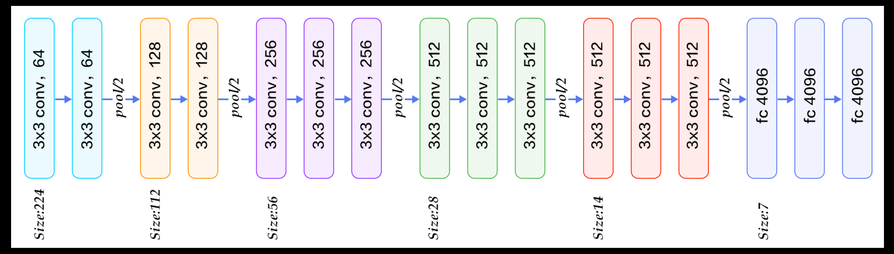
Loss: $-\sum_i y_i \cdot \log \hat{y}_i$

Deep model architecture

- AlexNet (2012): 8 Layers

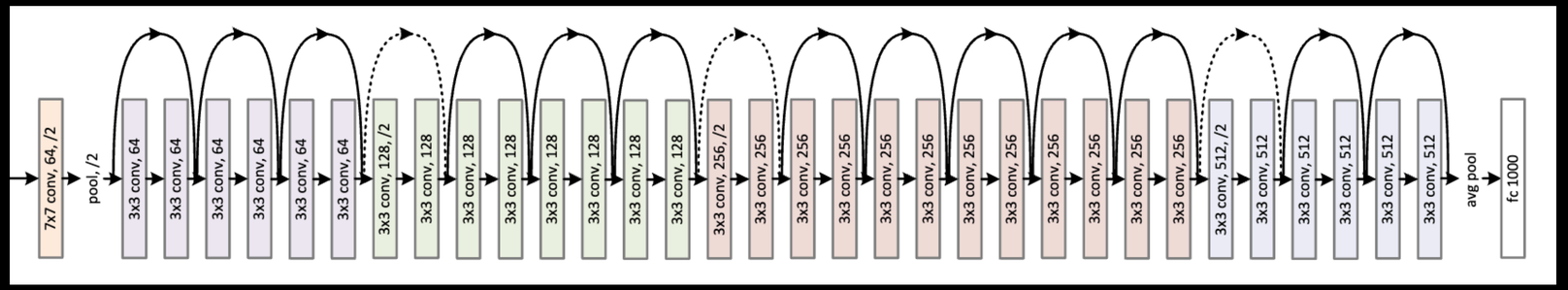


- VGGNet (2014): 16/19 Layers



https://www.kaggle.com/code/blurredmachine/vggnet-16-architecture-a-complete-guide

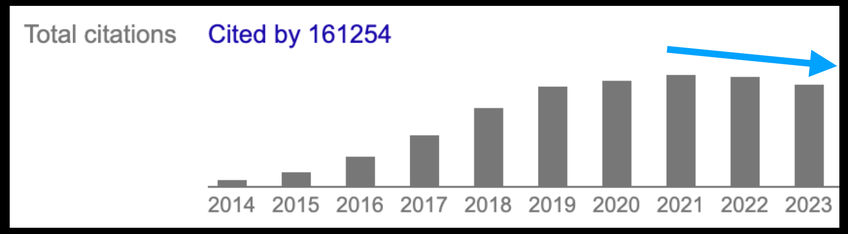
- ResNet (2016): >100 Layers with *Residual Connection*



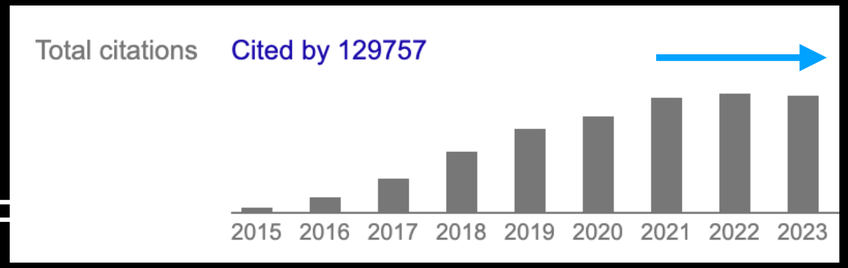
"Imagenet classification with deep convolutional neural networks", NIPS 2012.
"Very Deep Convolutional Networks for Large-Scale Image Recognition", ICLR 2015.
"Deep Residual Learning for Image Recognition", CVPR 2016.

Deep model architecture

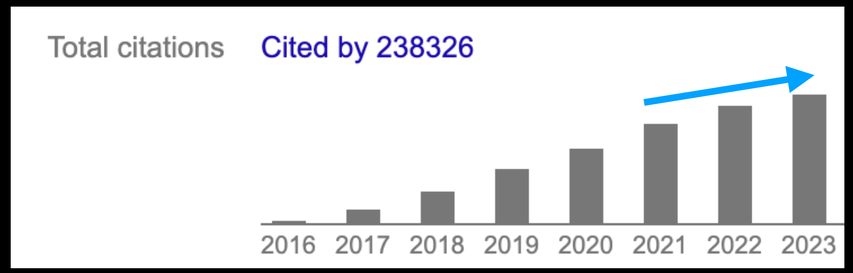
- AlexNet (2012)



- VGGNet (2014)

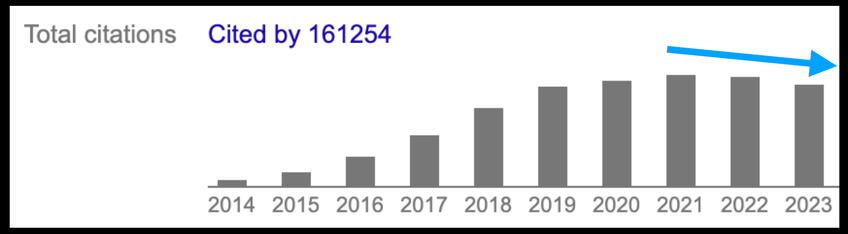


- R

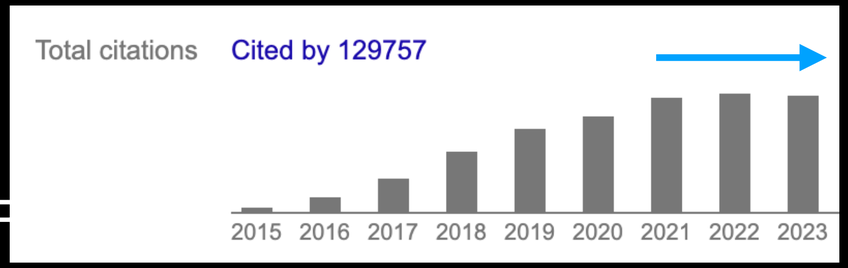


Deep model architecture

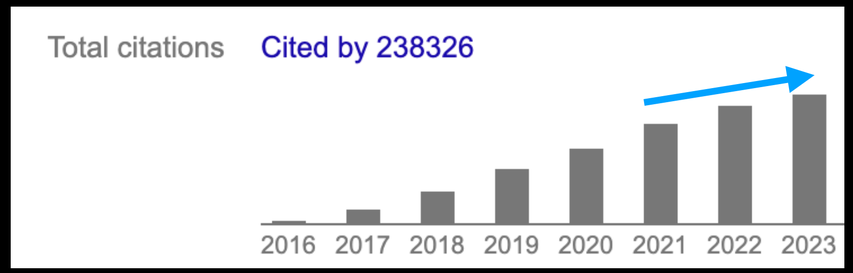
- AlexNet (2012)



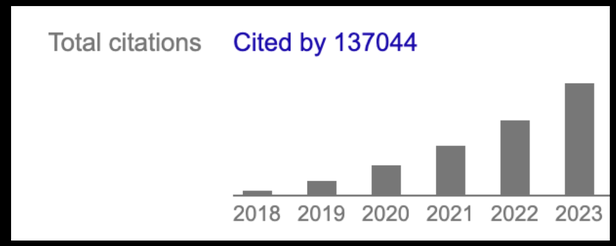
- VGGNet (2014)



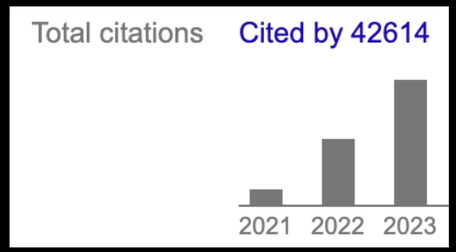
- R



- ? (2018)

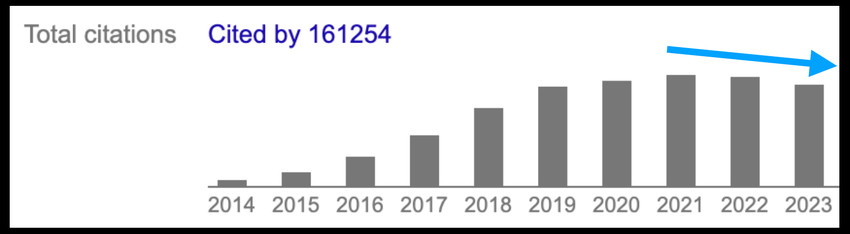


- ? (2020)

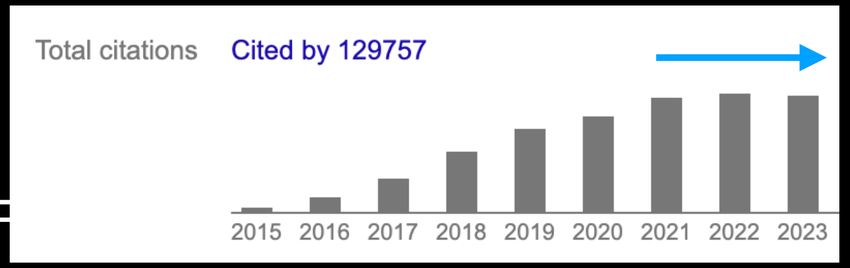


Deep model architecture

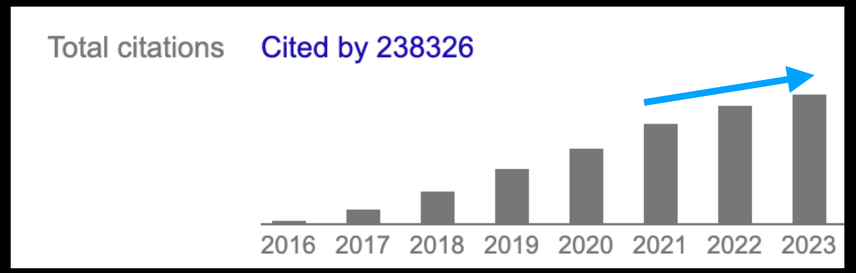
- AlexNet (2012)



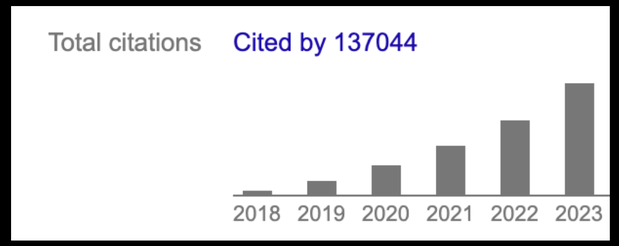
- VGGNet (2014)



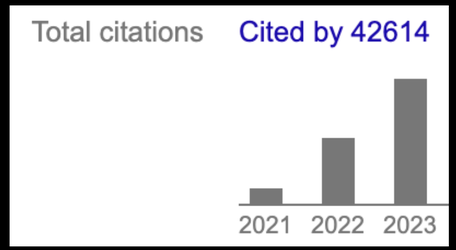
- R



- *Transformers* (2018)

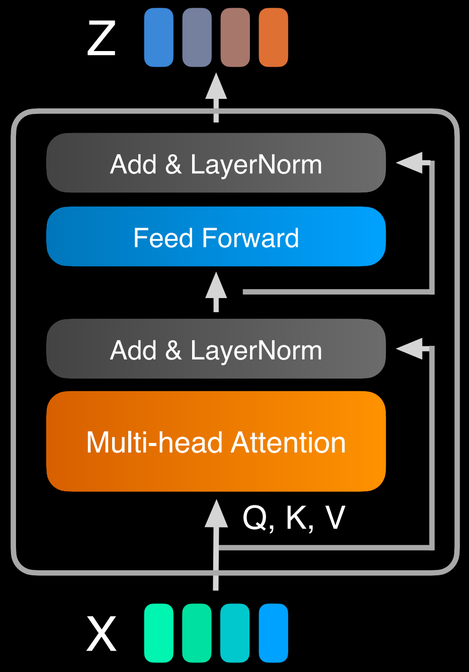


- *Vision Transformers (ViTs)* (2020)

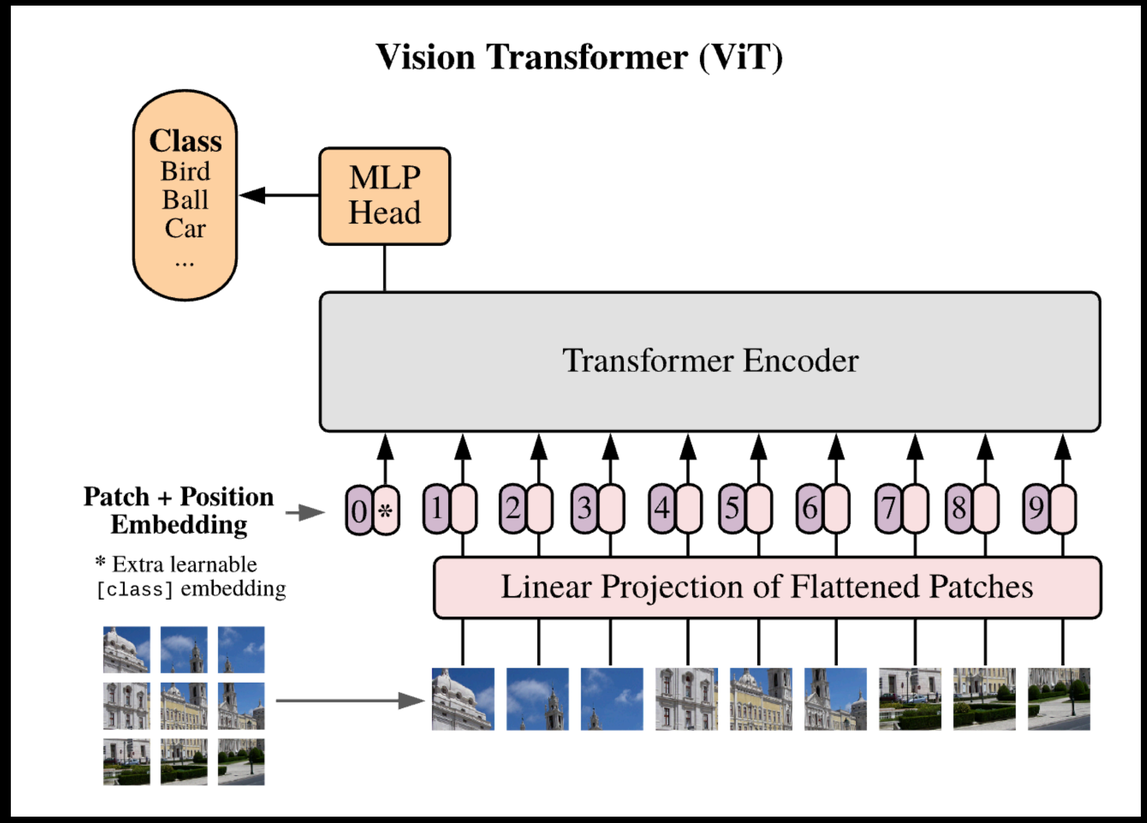


Vision Transformers

- Remind lecture 2's Transformers

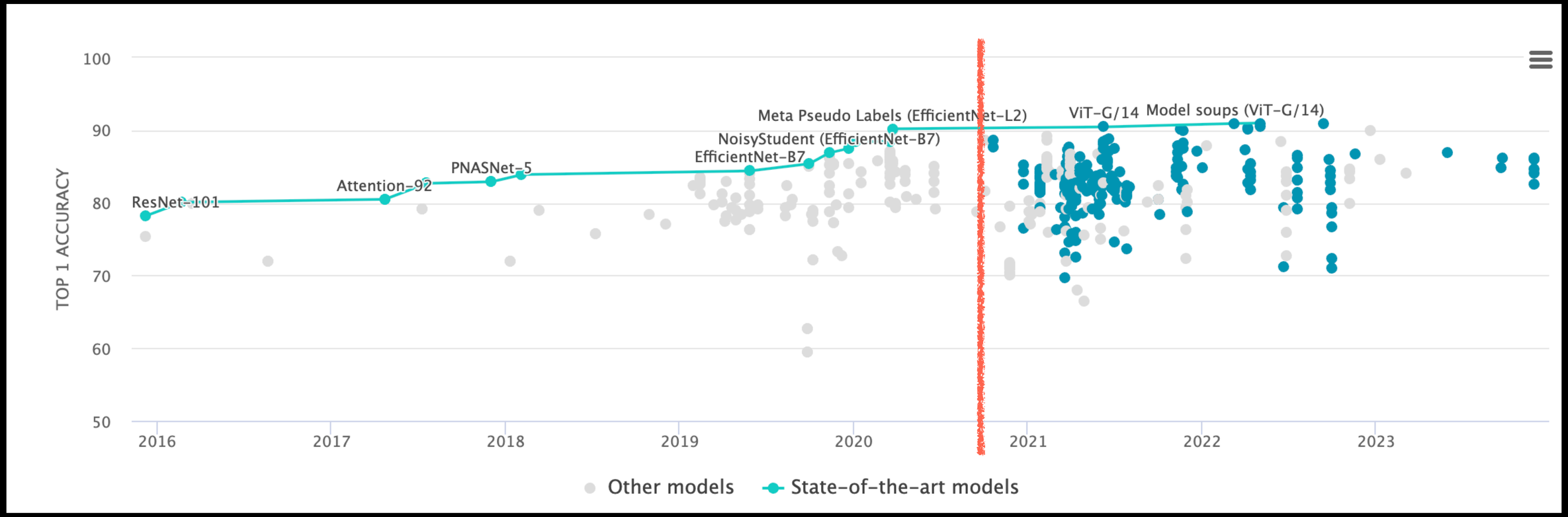


Self-attention block



Vision Transformers

- Vision Transformers on ImageNet benchmark



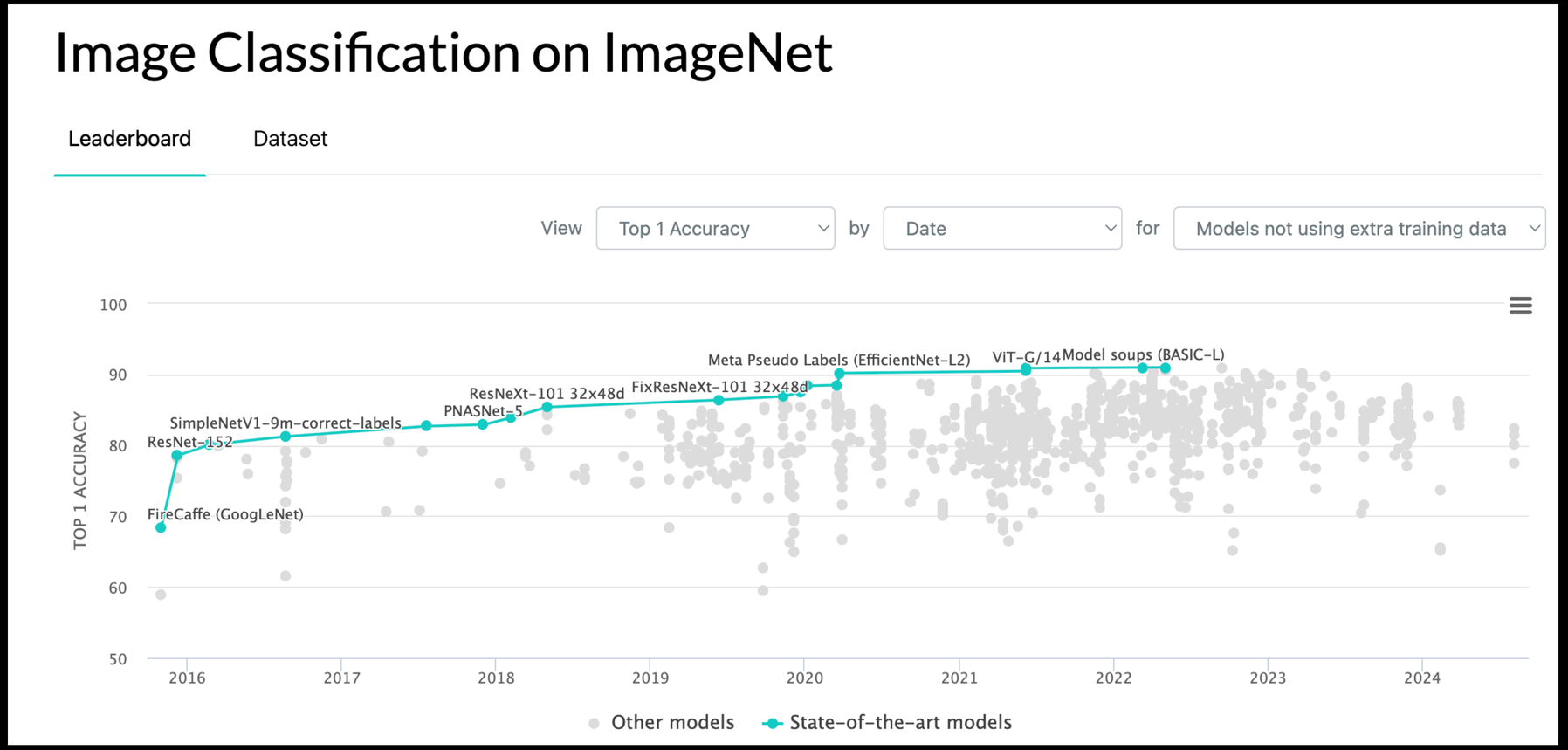
Supervised learning — Data

- *ImageNet* — Most famous vision dataset and benchmark

Deng et al., "Imagenet: A large-scale hierarchical image database". CVPR 2009.

Supervised learning — Data

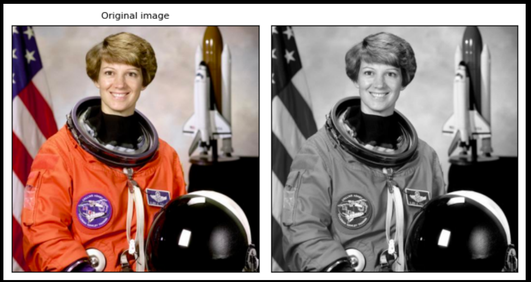
- *ImageNet* — Most famous vision dataset and benchmark



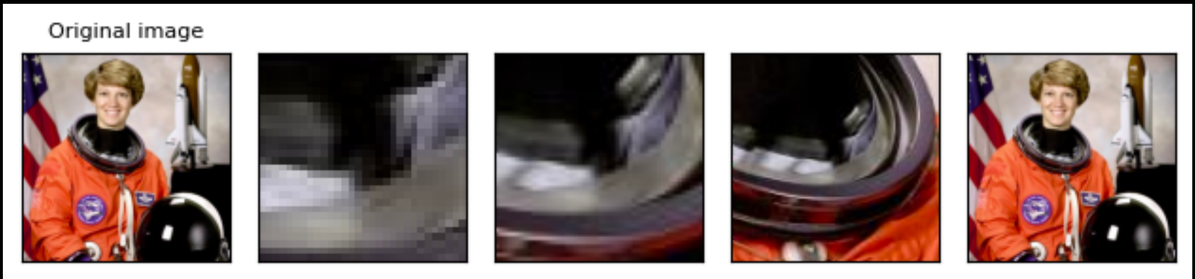
Data augmentation — Input level

```
import torchvision.transforms as T
out_img = T.grayscale()(org_img)
out_img = T.Centercrop(size=30)(org_img)
out_img = T.RandomAffine(degrees=(30,70),
translate=(0.1,0.3), scale=(0.5, 0.75))
```

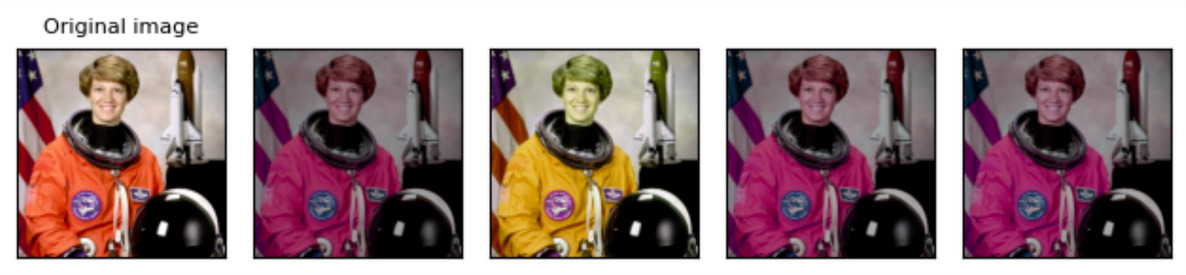
Grayscale



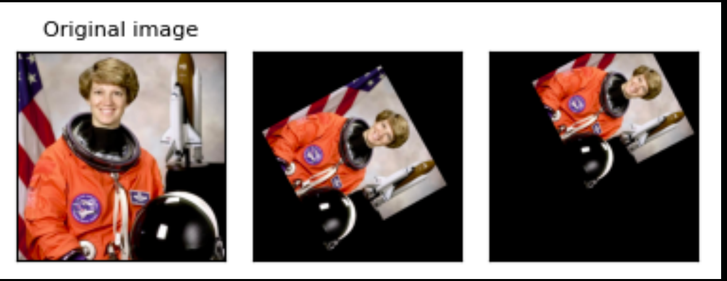
CenterCrop



ColorJitter



RandomAffine



...

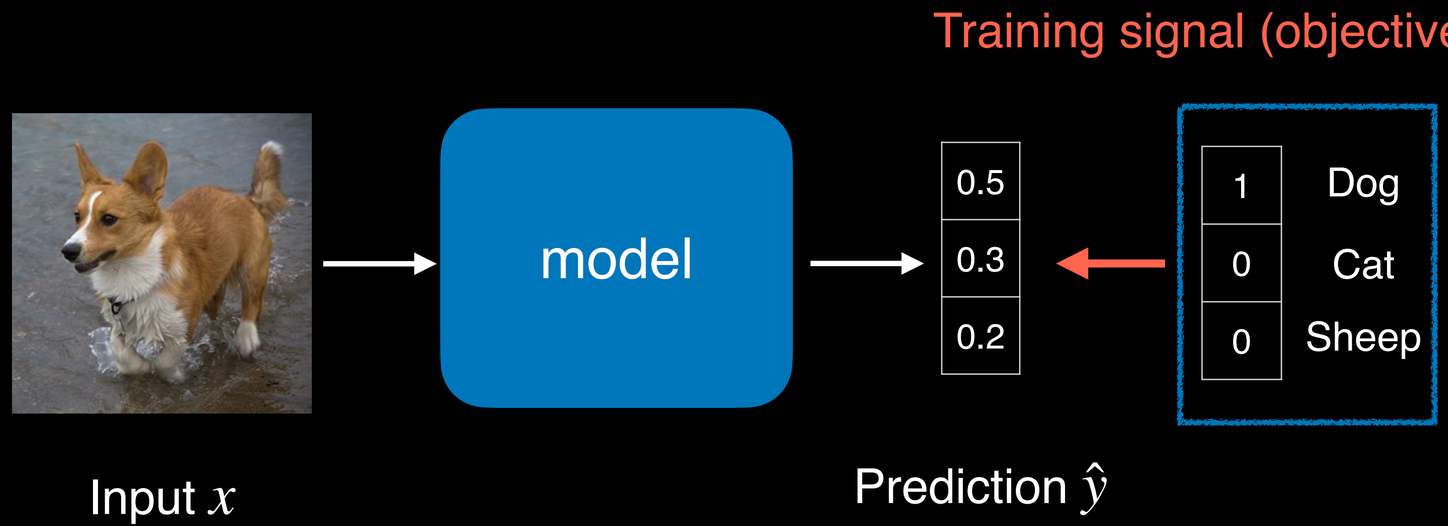
Documentation of PyTorch

Data augmentation — Input level

Cutout, RandomErasing

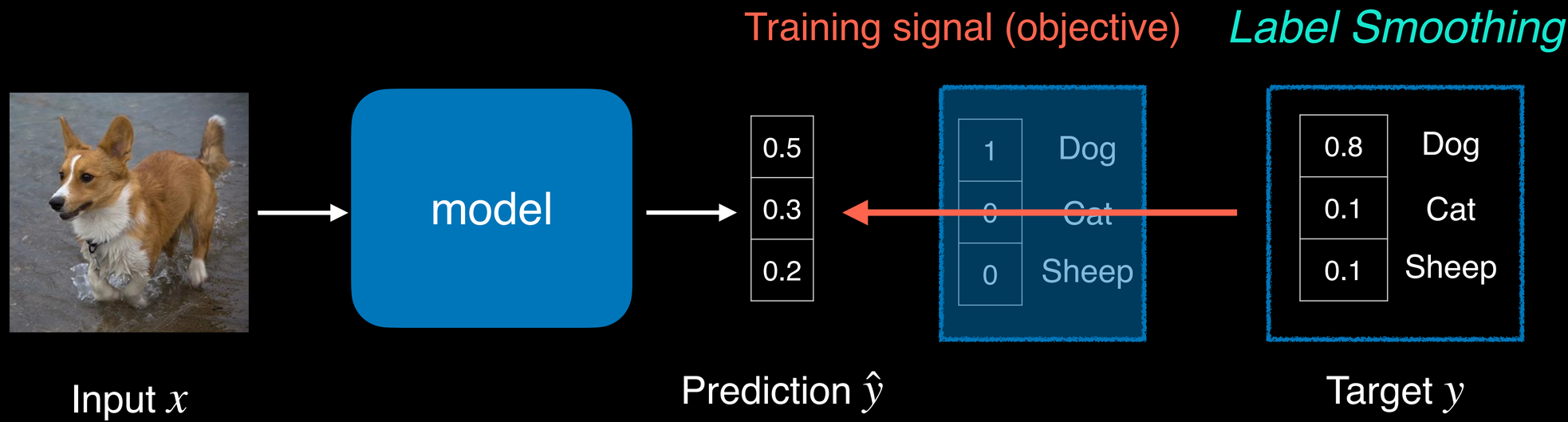
Devries et al., “Improved regularization of convolutional neural networks with cutout”, arXiv 2017.
Zhong et al., “Random erasing data augmentation”, arXiv 2017.

Data augmentation — Labels



Data augmentation — Labels

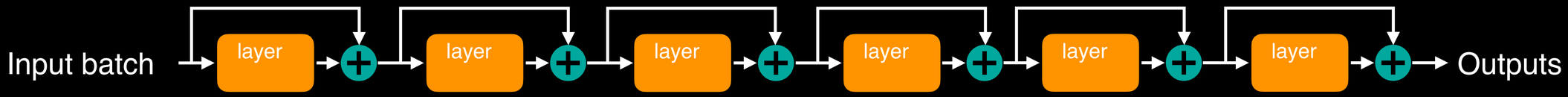
- Label Smoothing: Resolve the *over-confident* problem
- Can be seen as a regularizer



"Rethinking the inception architecture for computer vision." CVPR 2016.

Data augmentation — Features

- Deep networks with *residual* connections

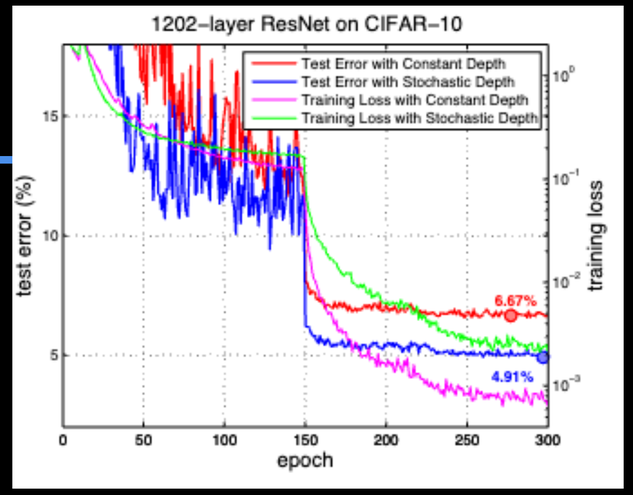


- Deep residual networks can be seen as “Exponential *Ensembles* of Shallow Networks” (Veit et al., 2016)

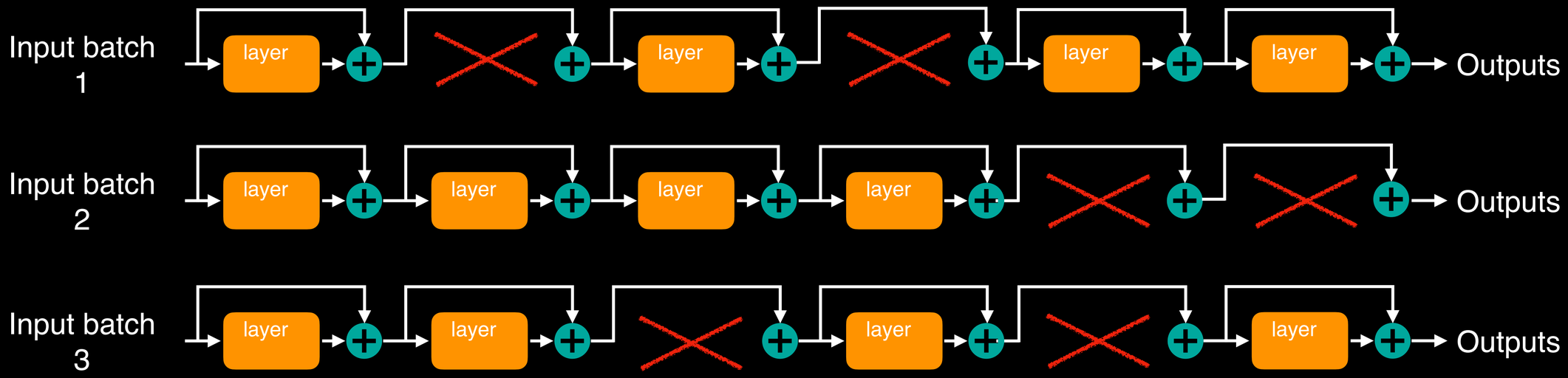
Data augmentation — Features

- *Stochastic Depth*: randomly drop l-th layer with probability p_l
  - Linear decay rule: early layer with low p_l, layer layer with high p_l

Data augmentation — Features



- *Stochastic Depth*: randomly drop l-th layer with probability p_l
  - Linear decay rule: early layer with low p_l, layer layer with high p_l



- It can be seen as a *feature augmentation* (randomly dropping intermediate features)

Data augmentation — Mixed Samples

- Combination of pairs of examples and their labels

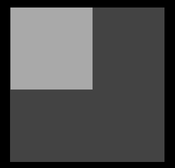
Mixup input image

$$\text{Input} : \tilde{x} = \lambda x_A + (1 - \lambda) x_B$$

$$\text{Label} : \tilde{y} = \lambda y_A + (1 - \lambda) y_B$$

CutMix input image

Binary mask



$$\text{Input} : \tilde{x} = \mathbf{M} \odot x_A + (1 - \mathbf{M}) \odot x_B$$

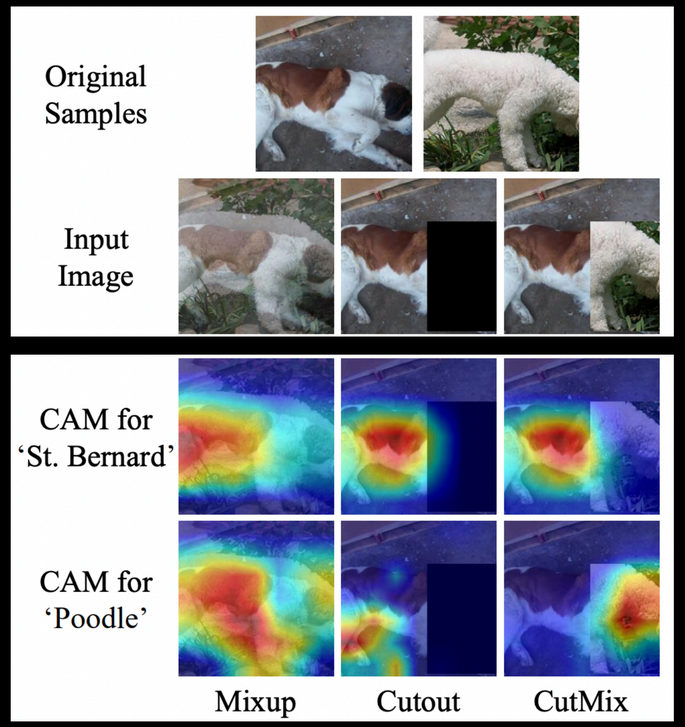
$$\text{Label} : \tilde{y} = \lambda y_A + (1 - \lambda) y_B$$

Zhang et al., “Mixup: Beyond empirical risk minimization” ICLR (2018)

Yun et al., “CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features” ICCV (2019)

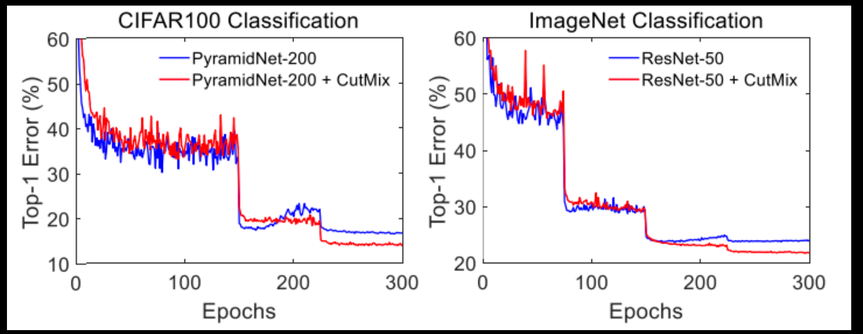
Data augmentation — Mixed Samples

- *CutMix* performance and analysis



Model	# Params	Top-1 Err (%)	Top-5 Err (%)
ResNet-152*	60.3 M	21.69	5.94
ResNet-101 + SE Layer* [15]	49.4 M	20.94	5.50
ResNet-101 + GE Layer* [14]	58.4 M	20.74	5.29
ResNet-50 + SE Layer* [15]	28.1 M	22.12	5.99
ResNet-50 + GE Layer* [14]	33.7 M	21.88	5.80
ResNet-50 (Baseline)	25.6 M	23.68	7.05
ResNet-50 + Cutout [3]	25.6 M	22.93	6.66
ResNet-50 + StochDepth [17]	25.6 M	22.46	6.27
ResNet-50 + Mixup [48]	25.6 M	22.58	6.40
ResNet-50 + Manifold Mixup [42]	25.6 M	22.50	6.21
ResNet-50 + DropBlock* [8]	25.6 M	21.87	5.98
ResNet-50 + Feature CutMix	25.6 M	21.80	6.06
ResNet-50 + CutMix	25.6 M	**21.40**	**5.92**

Table 3: ImageNet classification results based on ResNet-50 model. '*' denotes results reported in the original papers.



Yun et al., "CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features" ICCV (2019)

Do we need more data? Yes

- *ImageNet-21K* (aka ImageNet-full dataset)
  - Noisy labeling

ImageNet-1K

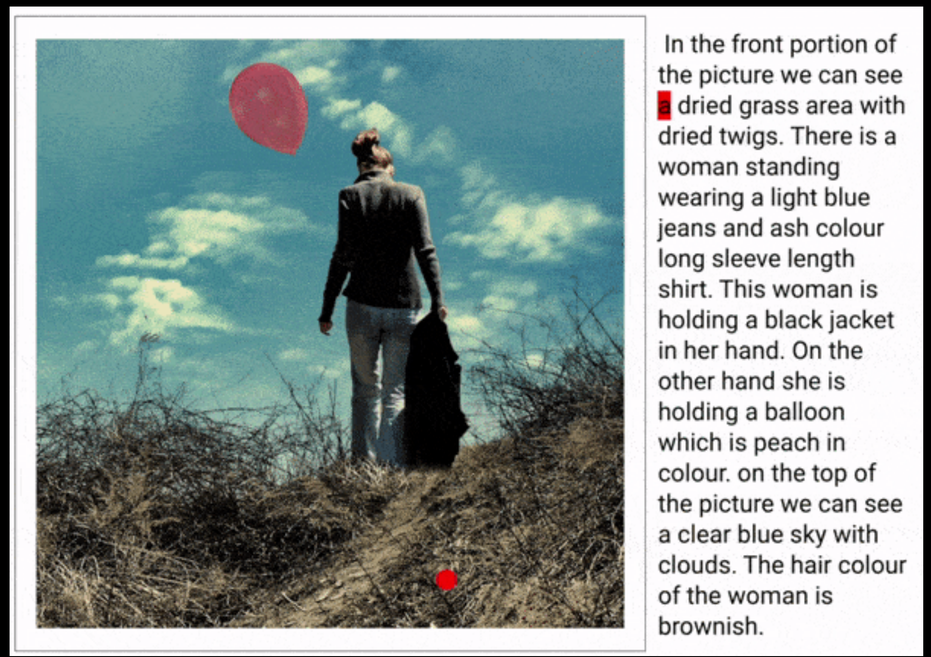
1.28M images
1,000 classes

ImageNet-21K

14.2M images
21,841 classes

OpenImages

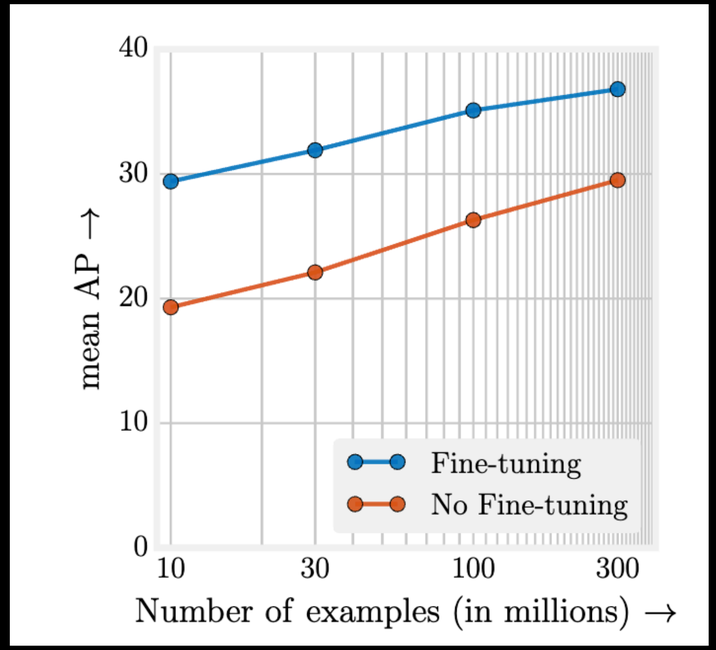
- 9M images with human-verified annotations
- Image-level labels (positive/negative), object bounding boxes, object segmentation masks, visual relationships, and *localized narratives*



OpenImages: A public dataset for large-scale multi-label and multi-class image classification, 2017

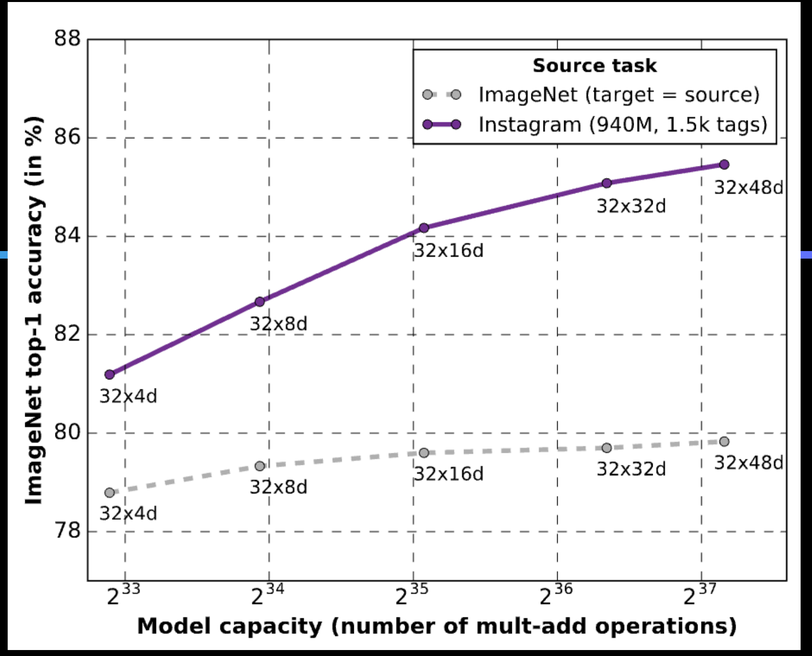
JFT-300M

- Google's internal dataset
- 300M images, 375M labels, 18k categories.
- Automatic labeling → Labels will be noisy and inconsistent.
- Objective: *Multi-class Sigmoid loss*
  - Loss: $-\sum_i y_i \cdot \log \hat{y}_i$, where $\hat{y}_i = \text{Sigmoid}(z_i)$
- Learning by weak (noisy) annotation → *Weakly supervised learning*
- Will extend to *JFT-3B dataset,* for <u>multimodal tasks</u>



Revisiting Unreasonable Effectiveness of Data in Deep Learning Era, ICCV 2017.

Instagram-3B dataset

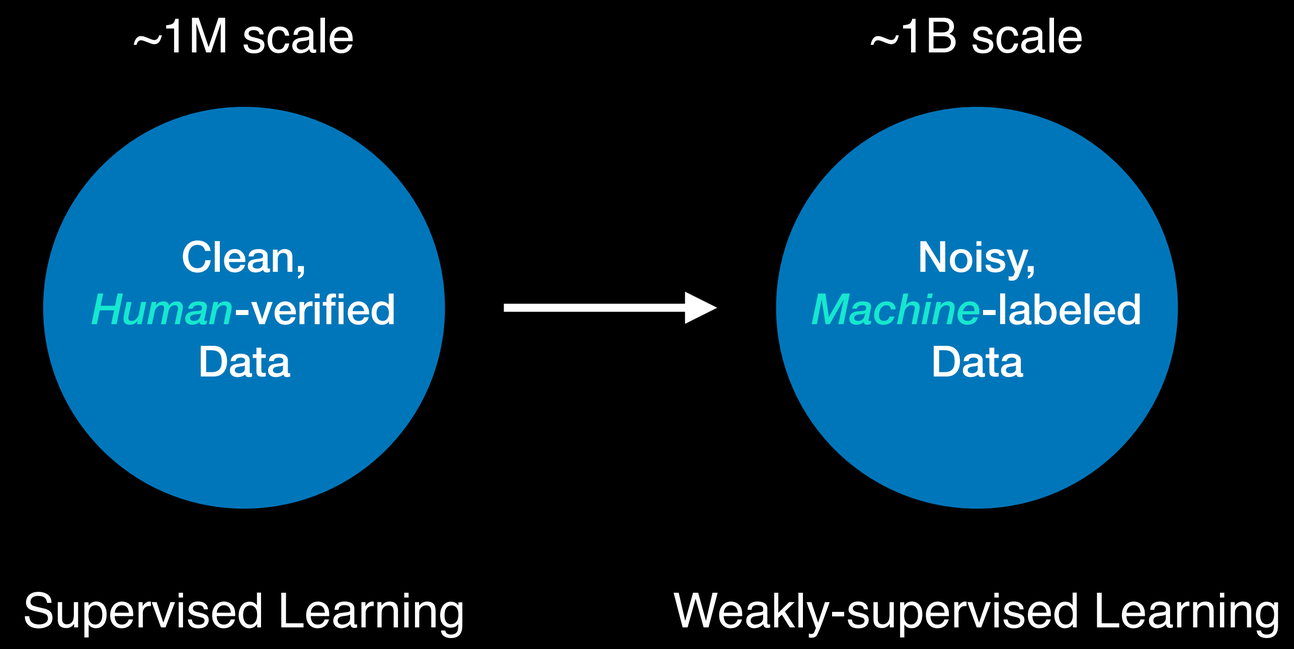


- Meta's internal dataset
- 3.5B images, 17k categories
- Automatic labels → *#hashtag*-based categorization
- Objective: *Softmax CE Loss with multi-hot labels* (which is a sum to 1)
  - A label *has **k** non-zero entries of **1/k** value (i.e., an image can have multiple hashtags)*
  - Loss: $-\sum_i y_i \cdot \log \hat{y}_i$, where $\hat{y}_i = \text{Softmax}(z_i)$
- Learning by weak (noisy) annotation → *Weakly supervised learning*

Exploring the Limits of Weakly Supervised Pretraining, ECCV 2018.

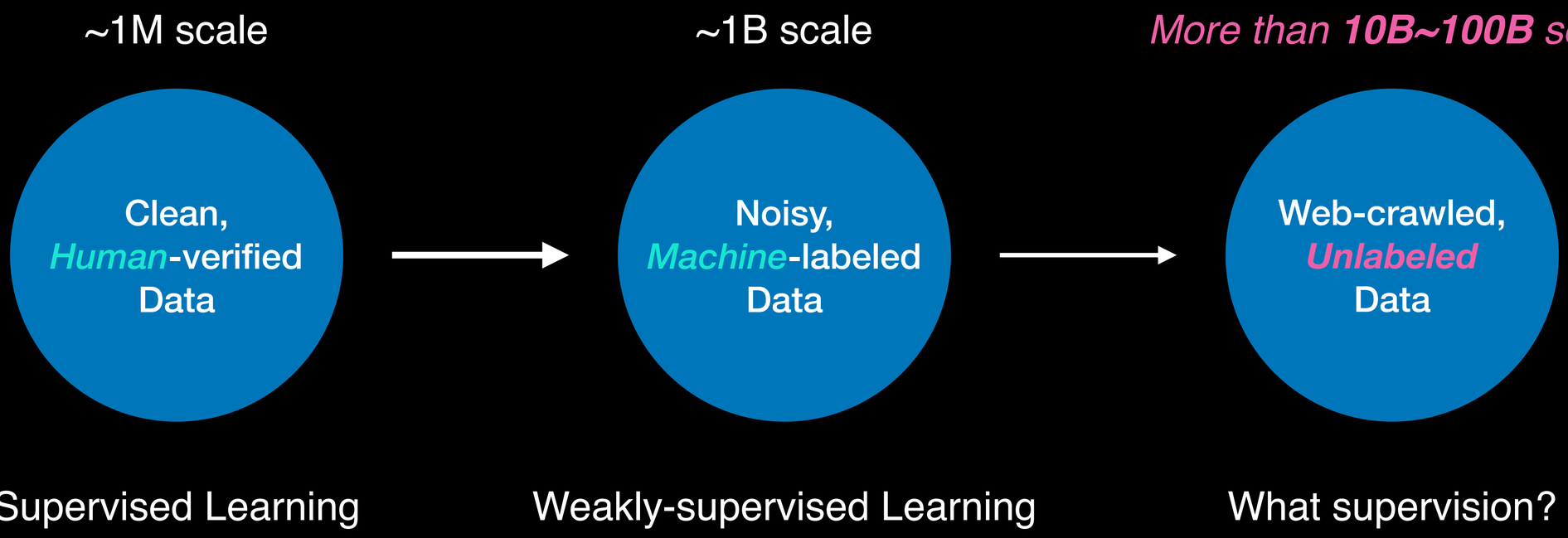
So, do we need even more data?

- In summary,



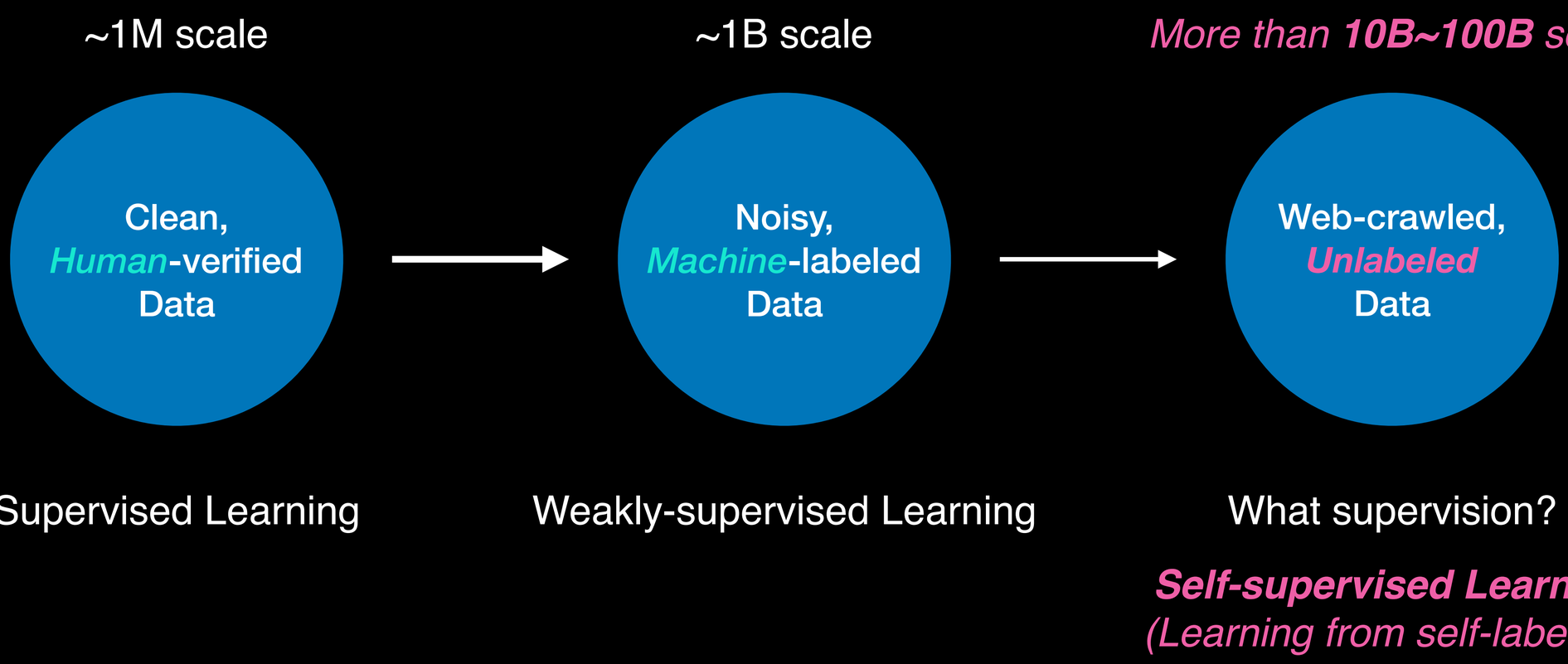
So, do we need even more data?

- In summary,



So, do we need even more data?

- In summary,



Self-supervised Learning

Self-supervised learning

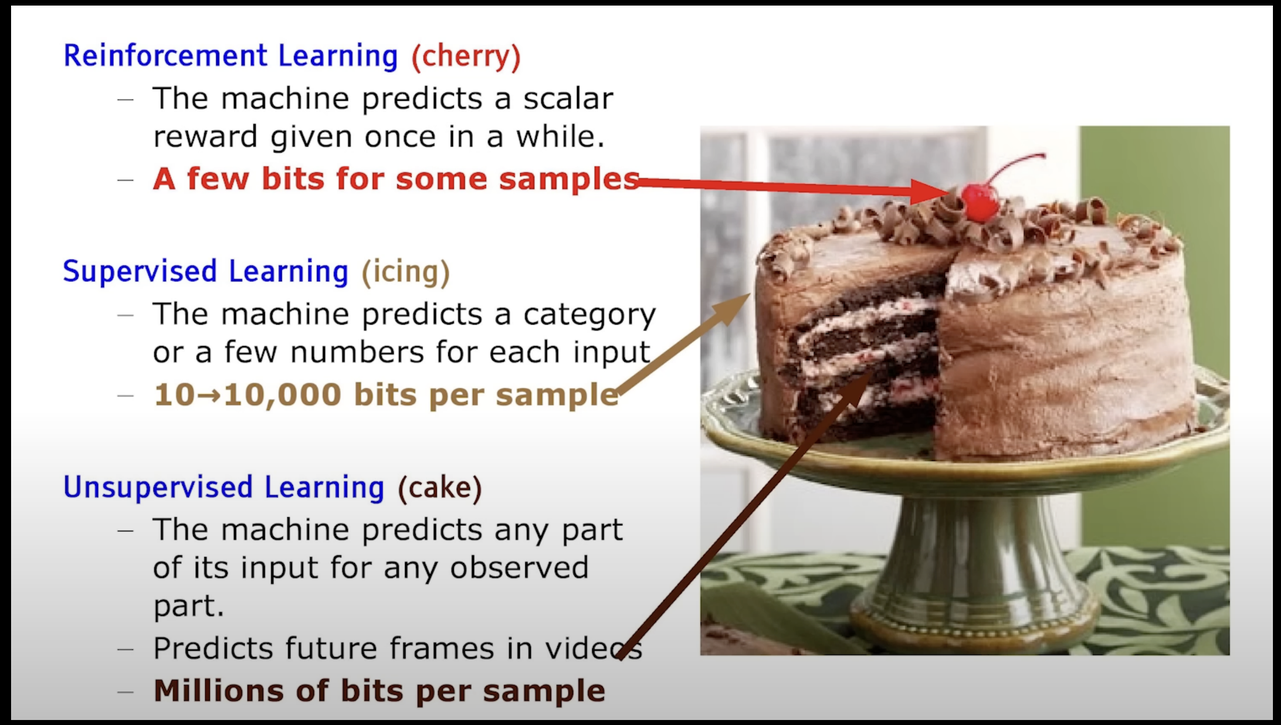
- What is *self-supervised learning*?
  - A learning paradigm where the model is trained using *labels generated from the data itself*
  - This creates a *self-generated problem* that the model must solve, encouraging a model to understand and represent the input data effectively.

Self-supervised learning

- Benefits of self-supervised learning:
  - *Beyond labels* — Enlarge dataset size (as we have seen in previous slides)
  - *Beyond task* — Mitigate *wrong correlations* between samples and labels.

Self-supervised learning

- Benefits of self-supervised learning:
  - *Beyond labels* — Enlarge dataset size (as we have seen in previous slides)
  - *Beyond task* — Mitigate *wrong correlations* between samples and labels.



https://www.youtube.com/watch?v=_1Cyyt-4-n8 Yann LeCun's talk

Self-supervised learning

- What will the model say?

(a) Texture image

81.4% **Indian elephant**
10.3% indri
8.2% black swan

(b) Content image

71.1% **tabby cat**
17.3% grey fox
3.3% Siamese cat

?

Self-supervised learning

- What will the model say?

(a) Texture image

81.4%	**Indian elephant**
10.3%	indri
8.2%	black swan

(b) Content image

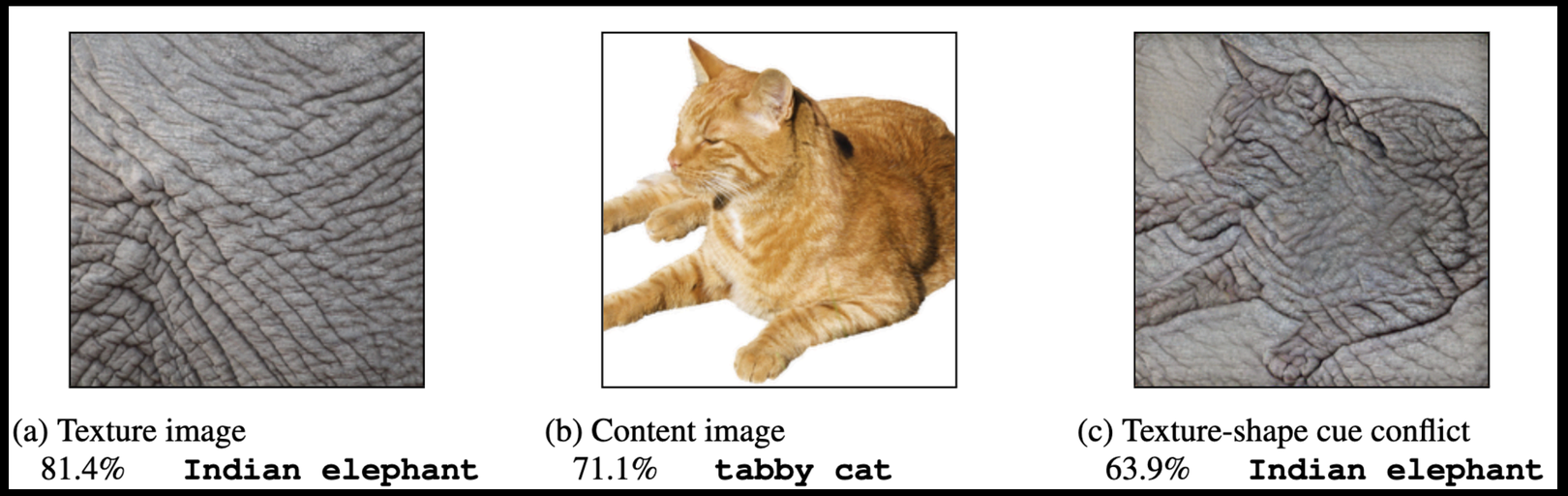
71.1%	**tabby cat**
17.3%	grey fox
3.3%	Siamese cat

(c) Texture-shape cue conflict

63.9%	**Indian elephant**
26.4%	indri
9.6%	black swan

Self-supervised learning

- Models are lazy (they usually cheat).
- Models trained by image recognition tasks usually learn *spurious correlation* (shortcut)



- Self-supervised learning can *reduce* this correlation *since there are no labels*!

ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, ICLR 2019.

Self-supervised learning

- Benefits of self-supervised learning:
  - *Beyond labels* — Enlarge dataset size (as we have seen in previous slides)
  - *Beyond task* — Mitigate wrong correlations between samples and labels.
- Self-supervision for *better representation learning*!

Self-supervised learning — Method

- *Generate artificial labels* and train models to *predict the generated labels*.
- Self-prediction
- Inter-sample prediction

Break

Self-supervised Learning

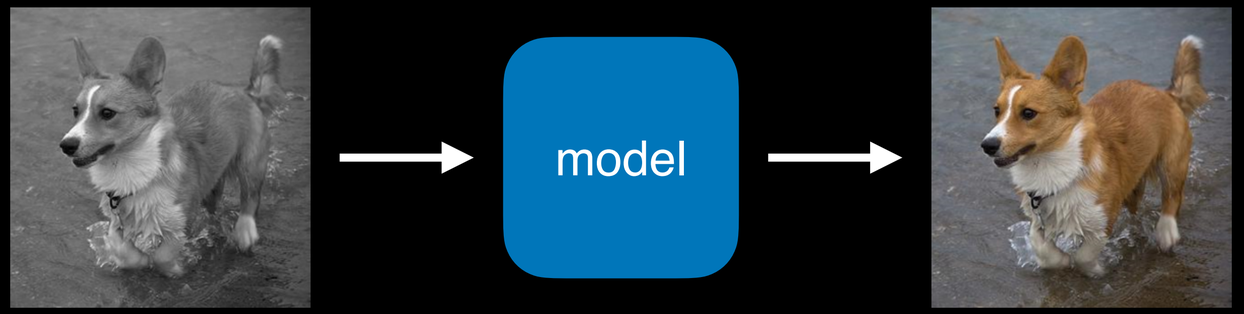
Self-prediction

Self-prediction

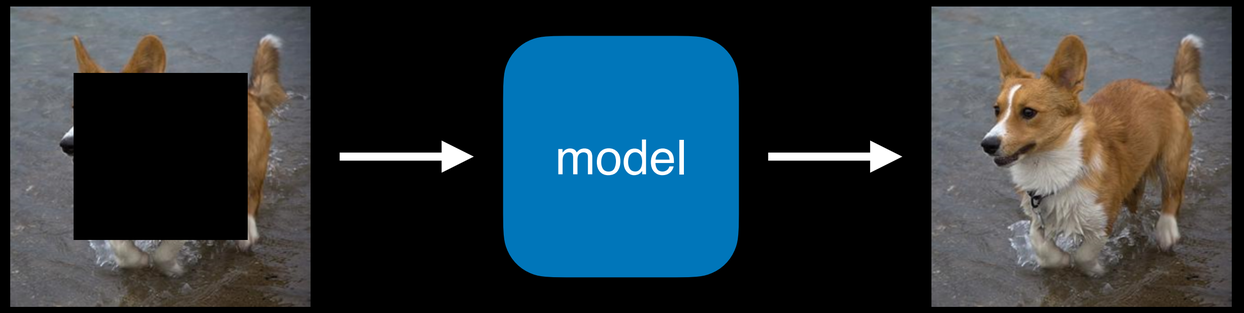
- Apply a *transformation* or *distortion* on data.
- Train a model to either restore the original data or identify the transformation.

Pretext modeling

- Colorization (2016)



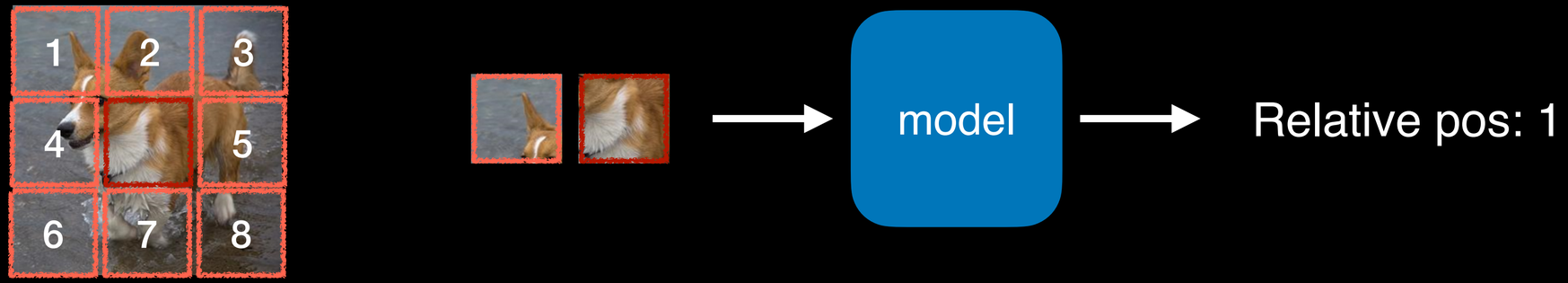
- Context Auto-Encoder (2016); inpainting



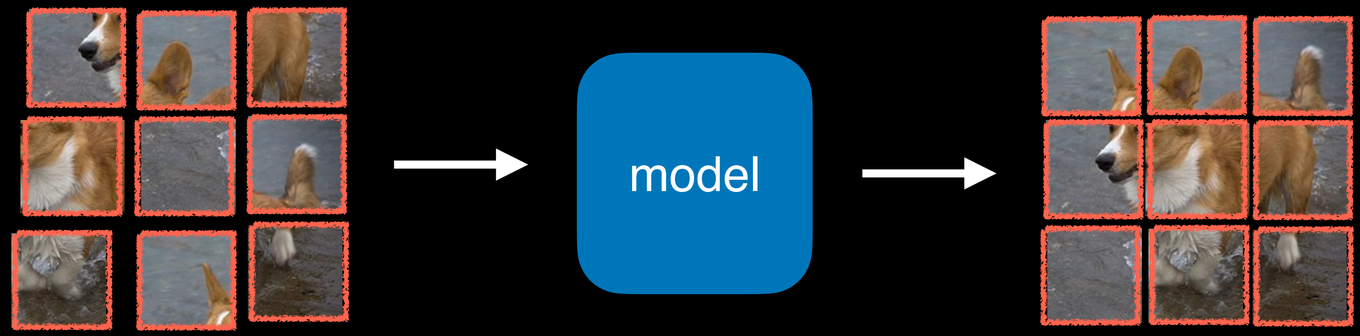
Zhang et al., “Colorful image colorization”, ECCV 2016.
Pathak et al., “Context Encoders: Feature Learning by Inpainting”, CVPR 2016.

Pretext modeling

- Relative location prediction (2015)



- Solve Jigsaw puzzle (2016)

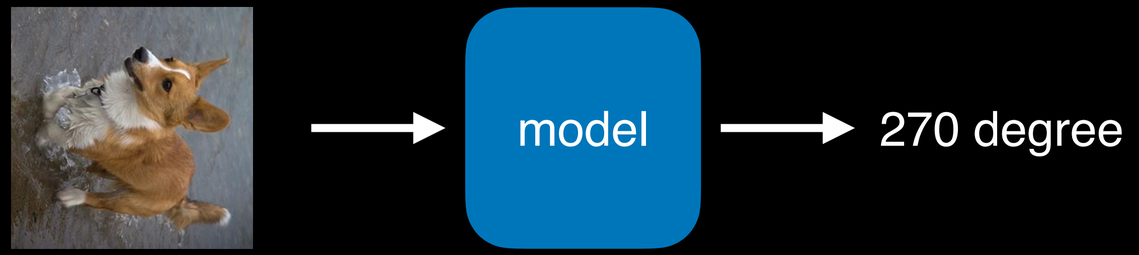


Doersch et al., “Unsupervised Visual Representation Learning by Context Prediction”, ICCV 2015.
Noroozi and Paolo, “Unsupervised learning of visual representations by solving jigsaw puzzles”, ECCV 2016.

Pretext modeling

- Counting features across grid patches (2017)

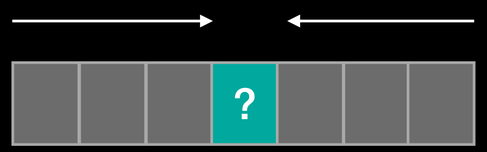
- Rotation (2018)
  - Predict which rotation is applied (classification among 0, 90, 180, and 270)



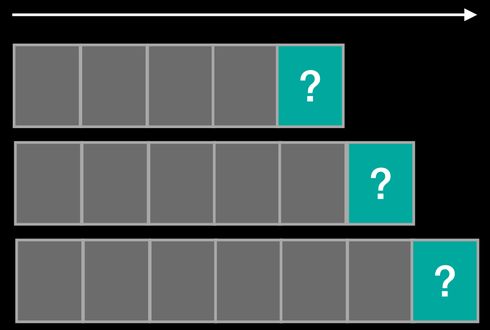
Noroozi et al., “Representation Learning by Learning to Count”, ICCV 2017.
Gidaris et al., “Unsupervised Representation Learning by Predicting Image Rotations”, ICLR 2018.

Pretext modeling (Text)

- Masked language modeling (BERT style)



- Auto-regressive generation (GPT style)

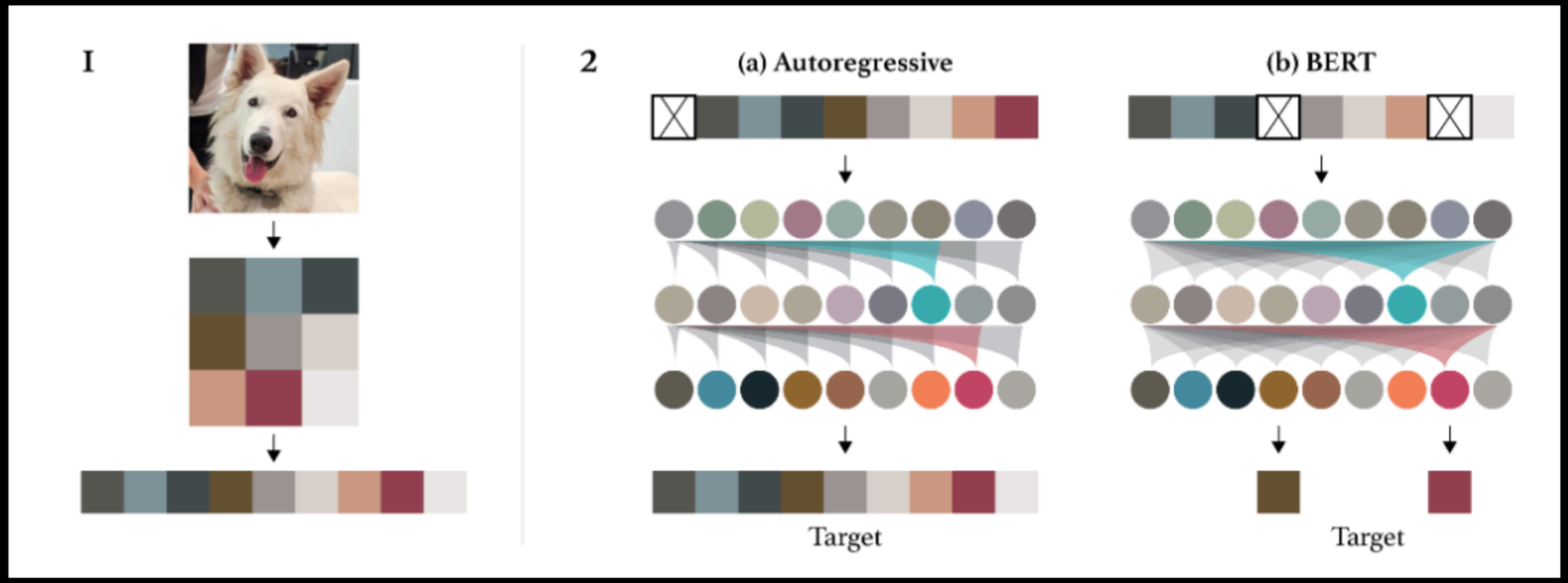



```
input=text[:-1]
target=text[1:]
```

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL 2019.
Improving Language Understanding by Generative Pre-Training, 2018.

ImageGPT

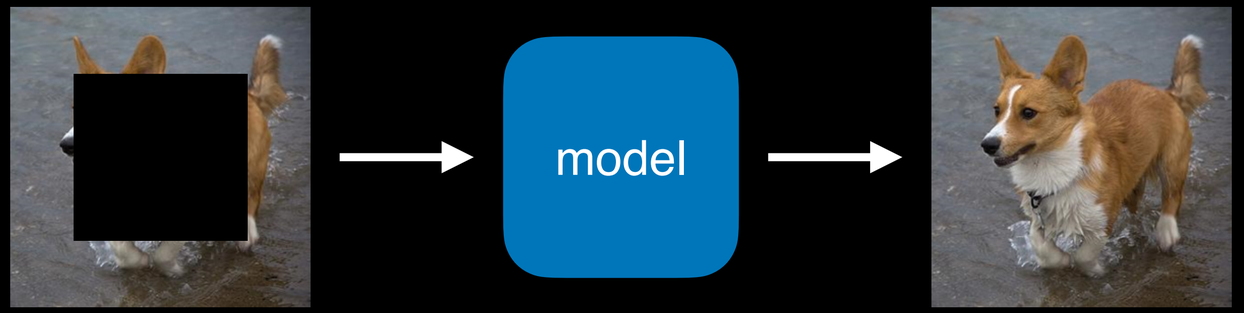
- ImageGPT (*iGPT*) (Chen et al., 2020)



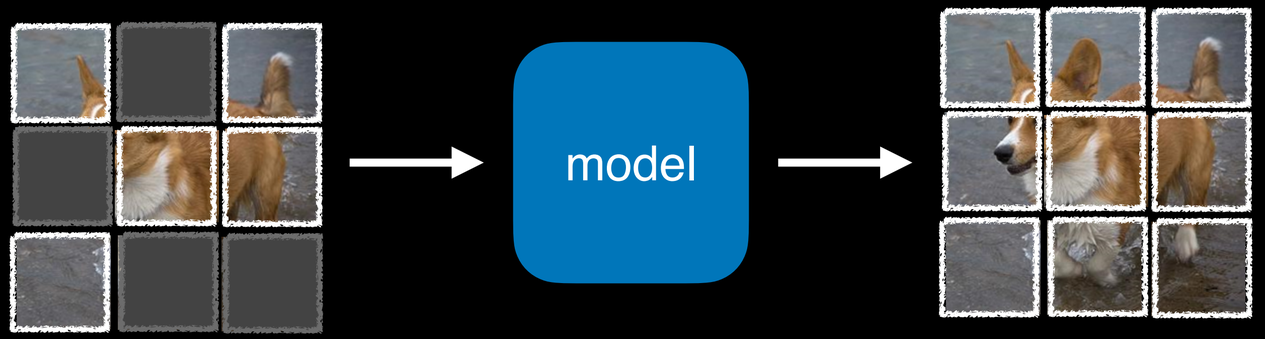
Generative Pretraining from Pixels, ICML 2020.

Masked image modeling

- Context Auto-Encoder (Pathak et al., 2016)

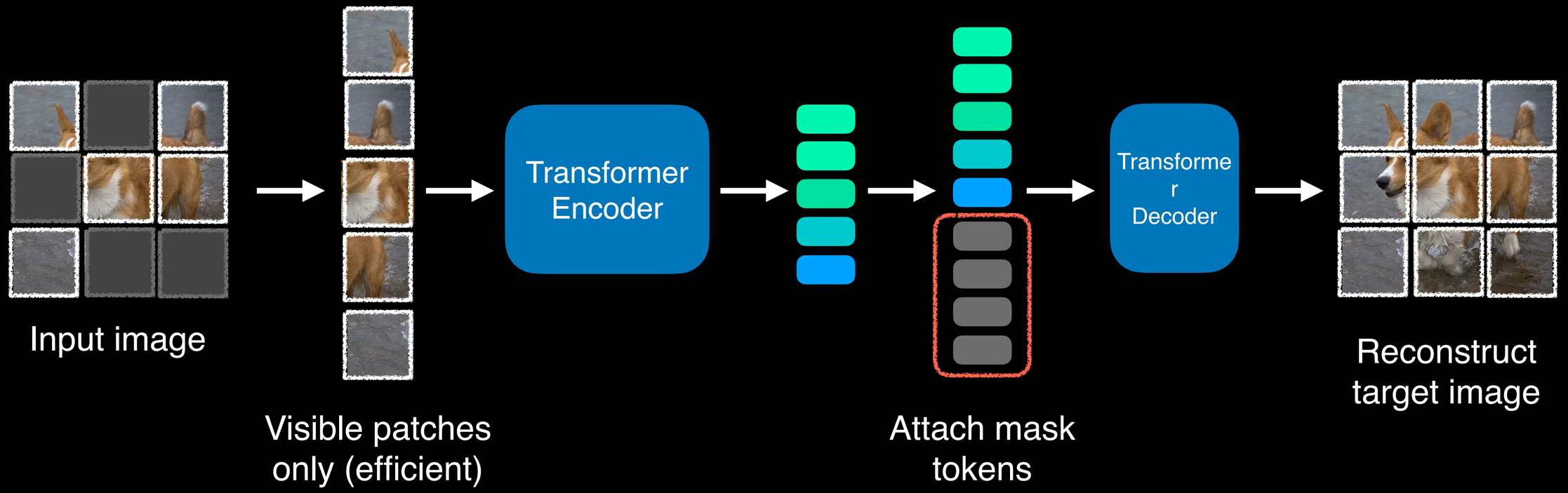


- Masked Auto-Encoder (He et al., 2022); state-of-the-art performance



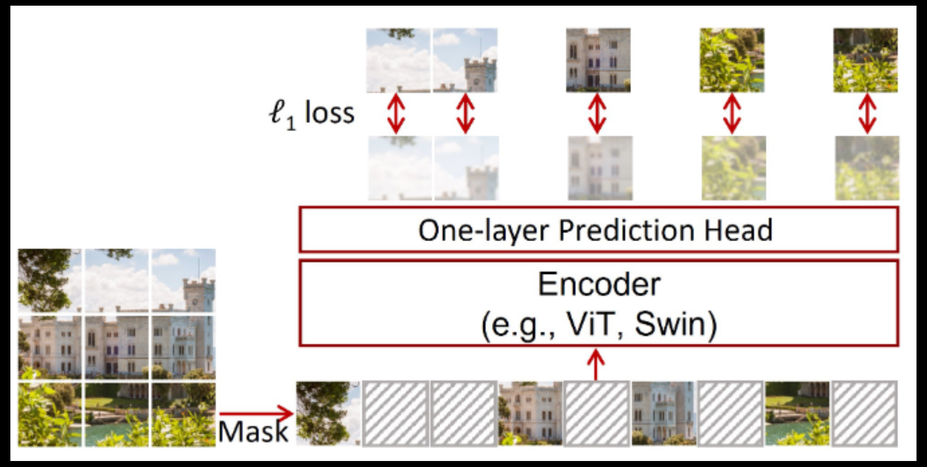
Masked image modeling

- Masked Auto-Encoder (MAE) (He et al., 2022)

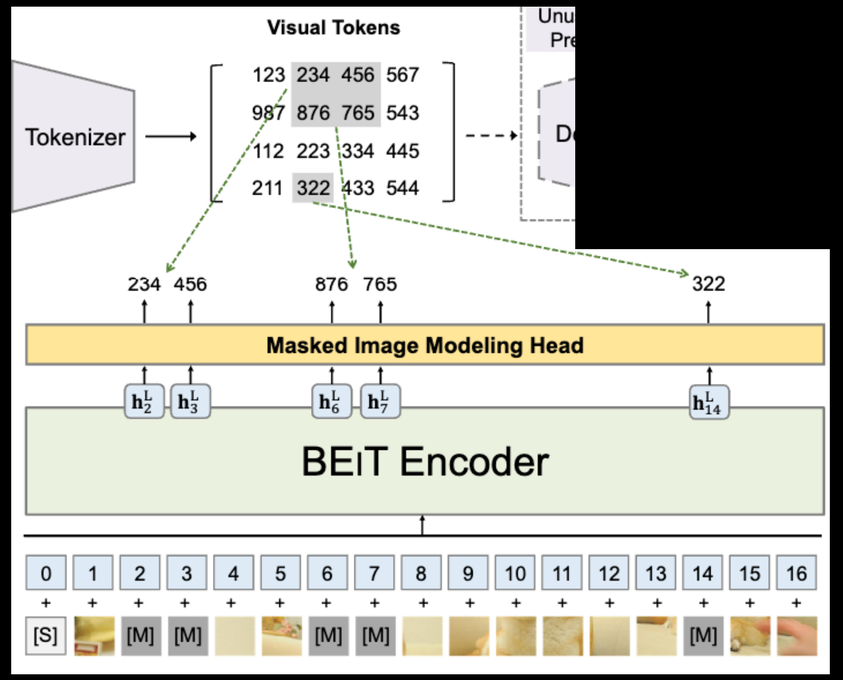


Masked image modeling

- SimMIM
  - *Simple* masked image modeling
  - Similar to MAE, *but no patch-drop*



- BEiT
  - Similar to MAE, but *no patch reconstruction*
  - *Image token* classification



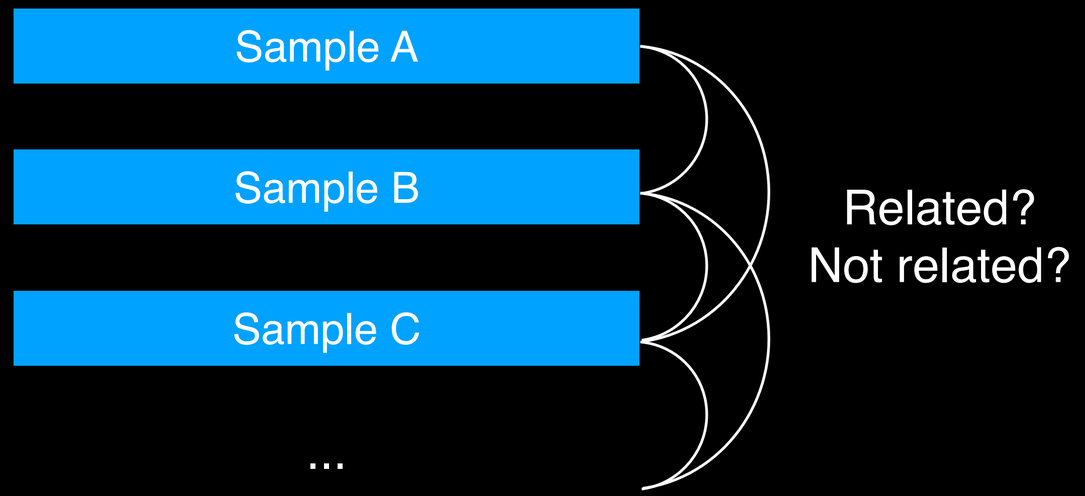
"SimMIM: a Simple Framework for Masked Image Modeling", CVPR 2022.
"BEIT: BERT Pre-Training of Image Transformers", ICLR 2022.

Self-supervised Learning

Inter-sample Prediction

Inter-sample prediction

- Beyond sample-wise prediction, learn the *relationship* between samples
- *Inter-sample prediction*



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*.
- Contrastive learning can be formulated as a *classification task* to classify *positive (similar) samples* from *negative (dissimilar) samples*.
- But we don’t have positive and negative labels.

https://lilianweng.github.io/posts/2021-05-31-contrastive/

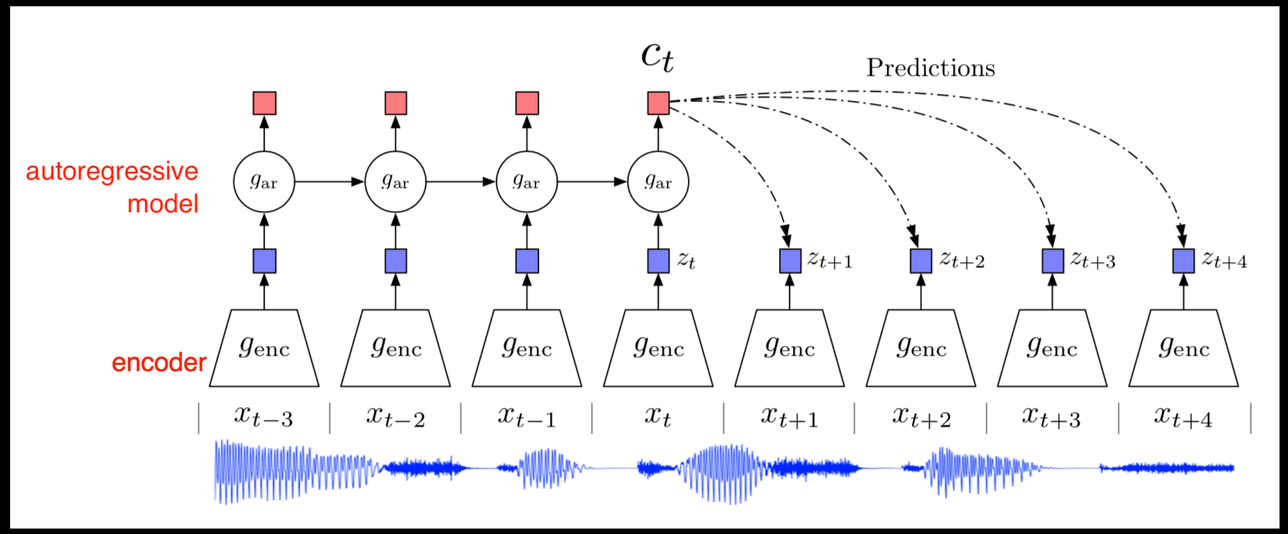
Contrastive learning

- InfoNCE (Oord et al., 2018) uses cross-entropy loss to identify the positive sample from unrelated noise samples (*e.g.,* random samples). (Remind p.32 in lecture 3)

$$\mathcal{L}_N = -\mathbb{E}\left[\log \frac{f(x_{pos}, h)}{\sum_j^N f(x_j, h)}\right]$$

$$[x_1, \ldots, x_{N-1}, x_{pos}]$$

(N-1) negatives + One positive

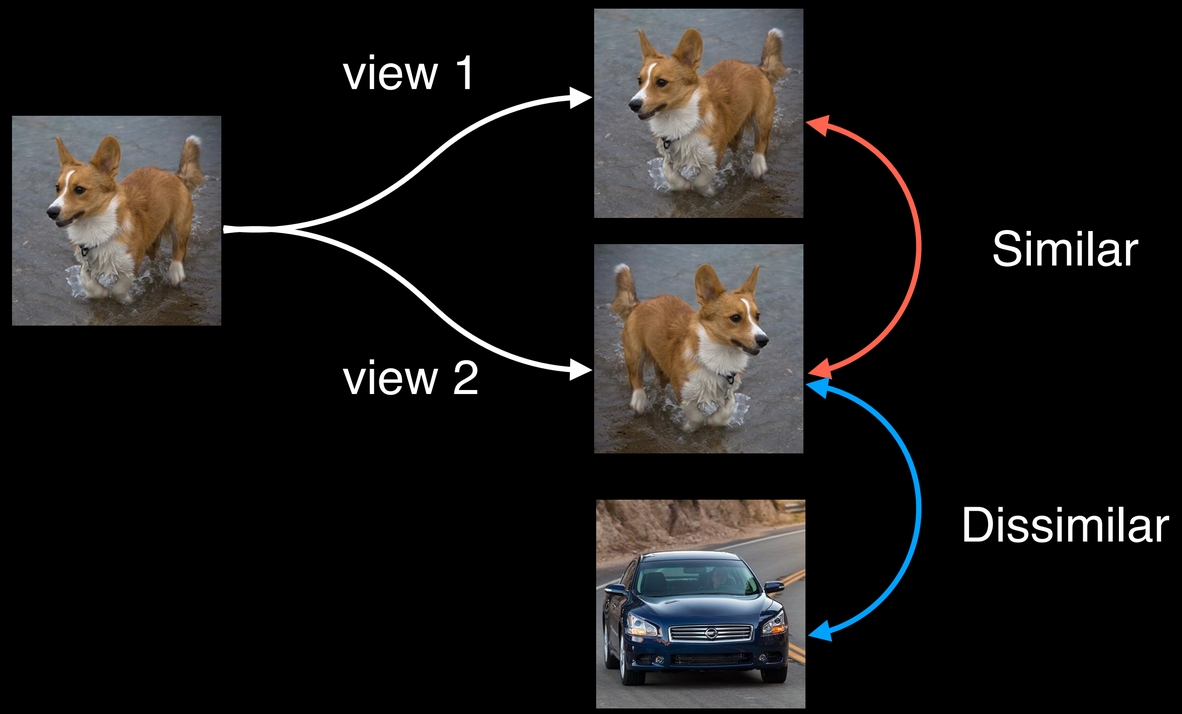


Classify the “future” (*positive*) representation from unrelated *negative* samples

“Representation learning with contrastive predictive coding” arXiv (2018)

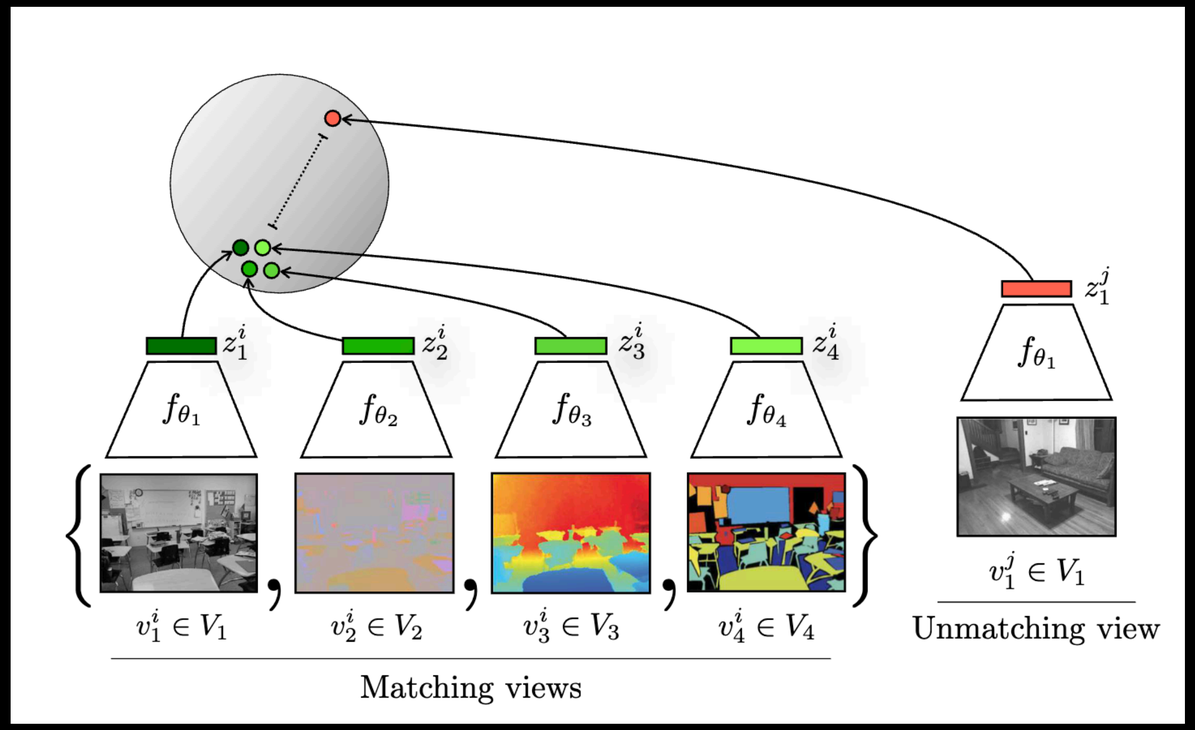
Contrastive learning with different views

- InfoNCE loss to two or more *different views* of input data

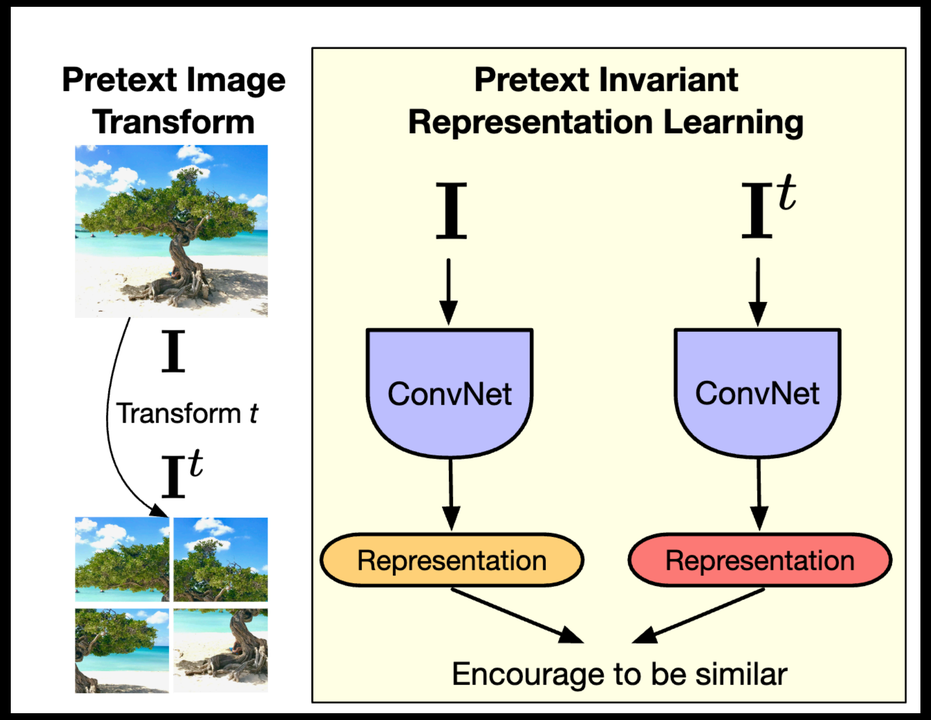


Contrastive learning with different views

- Gray, RGB, Depth, ... (Tian et al., 2019)



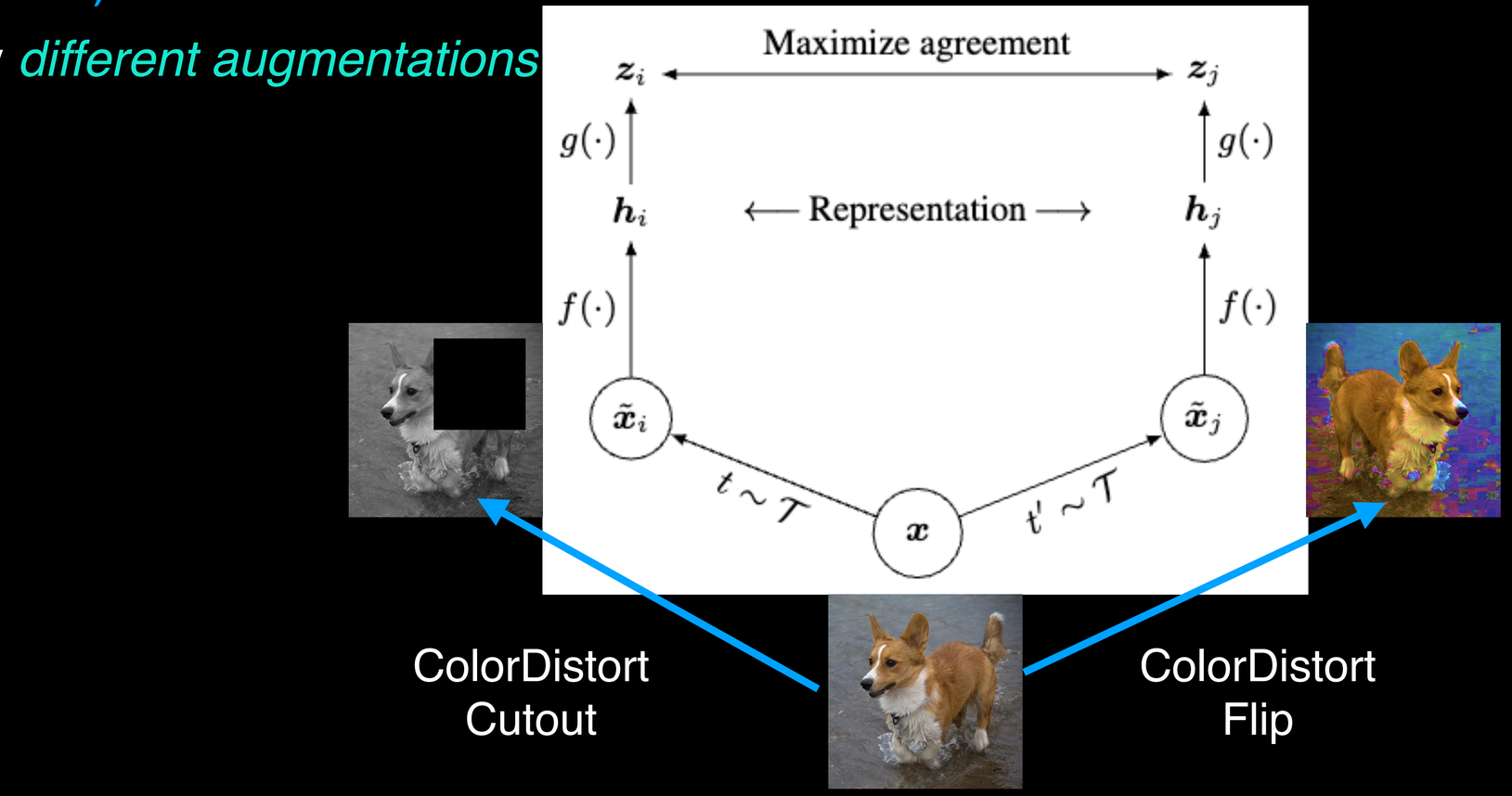
- Jigsaw transform (Misra et al., 2019)



Contrastive Multiview Coding, 2019.
Self-Supervised Learning of Pretext-Invariant Representations, 2019.

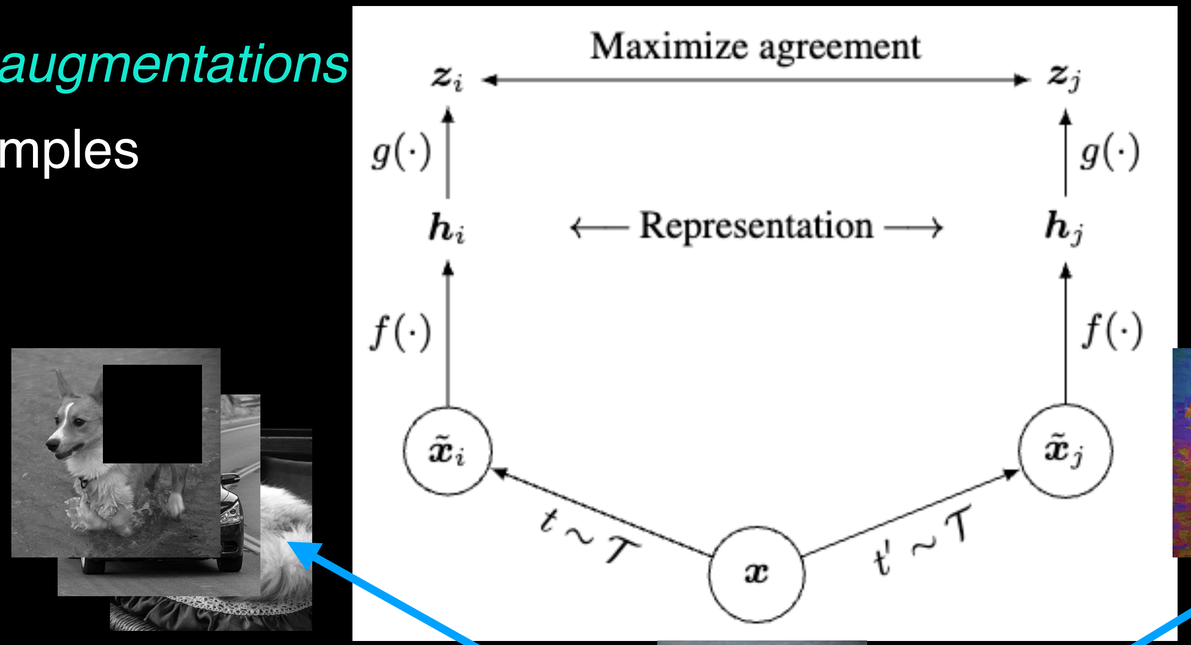
Contrastive learning

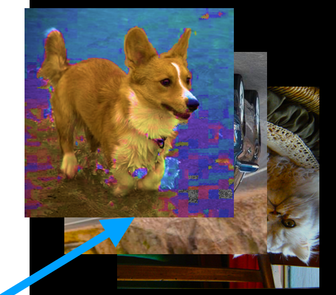
- SimCLR (Chen et al., 2020)
  - Generate two views by *different augmentations*

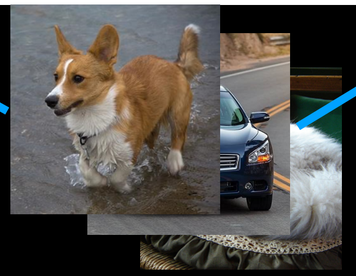


Contrastive learning

- SimCLR (Chen et al., 2020)
  - Generate two views by *different augmentations*
  - N samples → 2N augmented samples







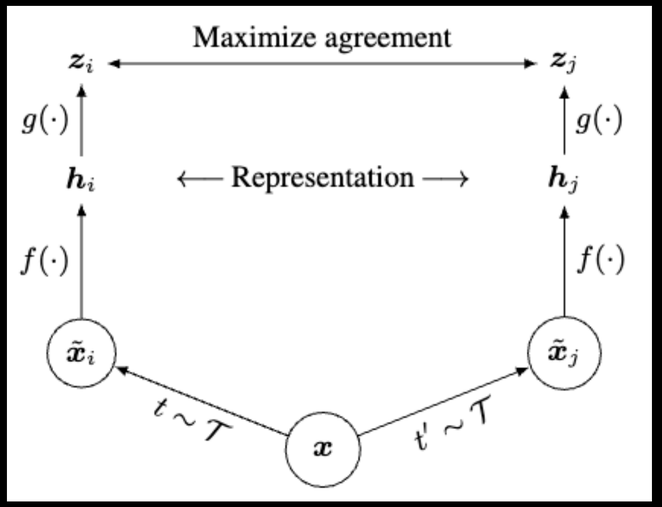
N samples

Contrastive learning

- SimCLR (Chen et al., 2020)
  - Generate two views by *different augmentations*
  - Given one positive pair, other 2(N-1) samples are negative
  - *InfoNCE* loss:

$$\mathscr{L}^{(i,j)}_{\text{SimCLR}} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbf{1}_{k \neq i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$

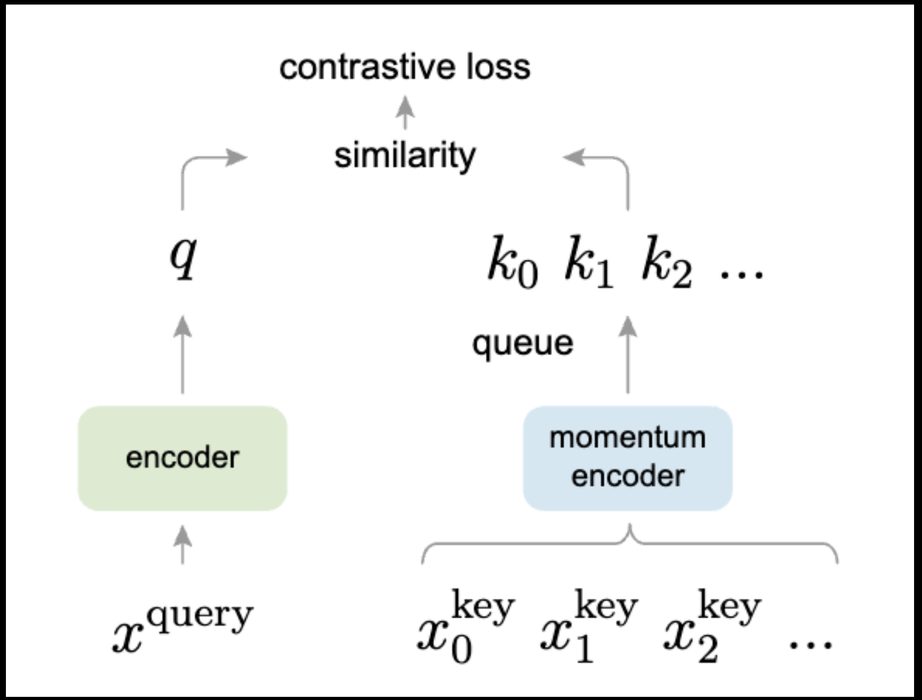
  - SimCLR needs a *large batch-size* (>4K) for performance.



Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020

Contrastive learning

- MoCo (He et al, 2020)
  - Momentum encoder: $\theta_k \leftarrow m\theta_k + (1-m)\theta_q$
  - Negative samples x_i^{key}: samples of previous batches
  - Query input $\mathbf{q}$
  - Positive sample $\mathbf{k}^+$: augmented sample of input $\mathbf{q}$
  - *InfoNCE loss*: $\mathscr{L}_{\text{MoCo}} = -\log \frac{\exp(\mathbf{q}\cdot\mathbf{k}^+/\tau)}{\sum_{i=1}^{N}\exp(\mathbf{q}\cdot\mathbf{k}_i/\tau)}$



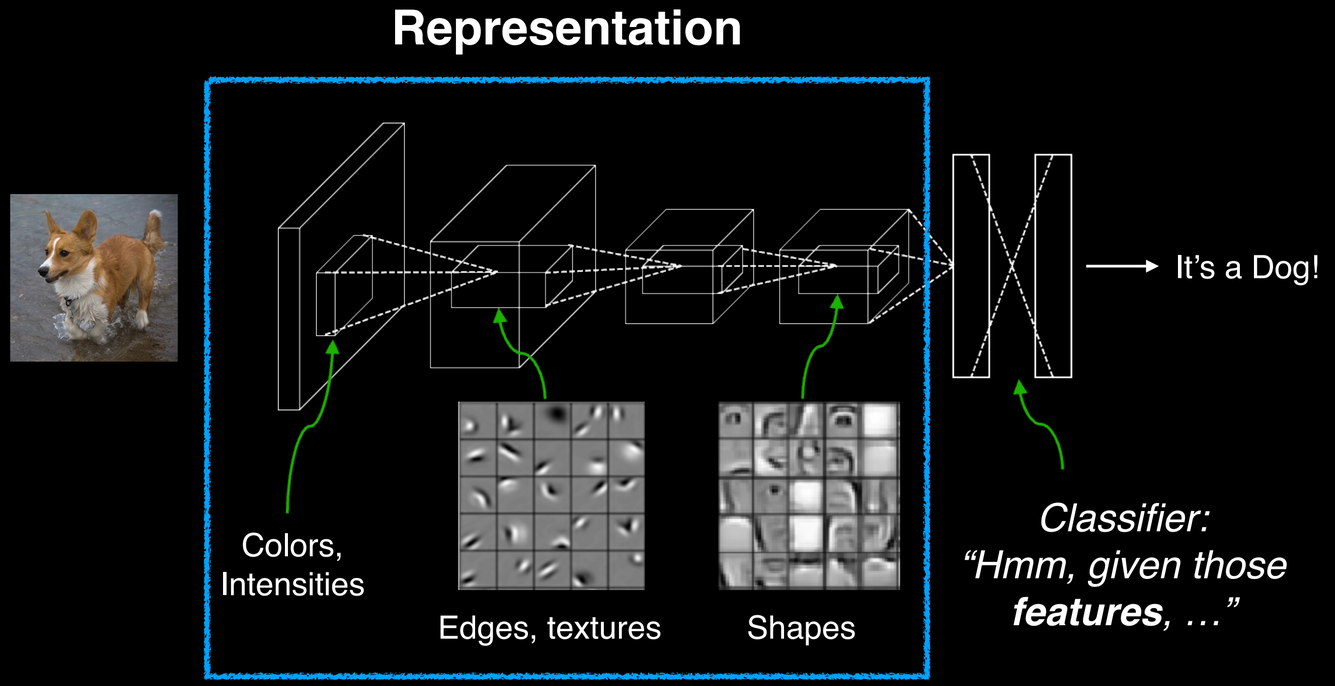
- Using queue, *decouple # of negative samples from batch-size*
- Going to *MoCo-v2*, apply *strong data augmentation* and *MLP projection head* as in SimCLR

He et al., "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020.

Evaluation of Self-supervised Models

Evaluation of learned representation

- Remind what was representation



Evaluation of learned representation

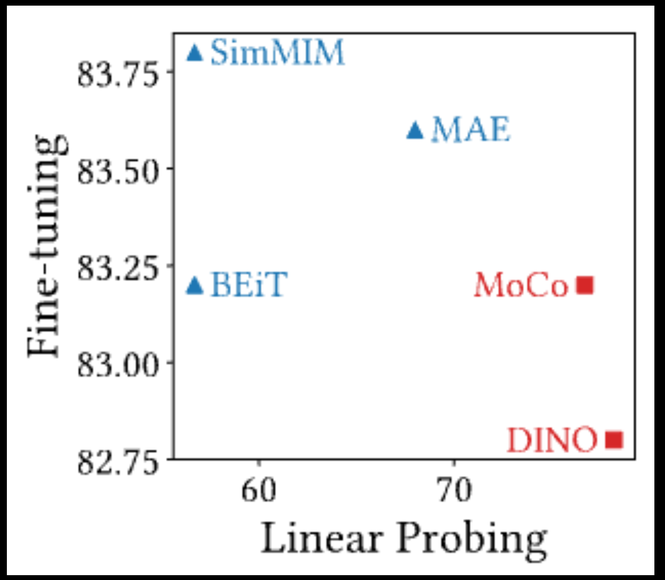
- We have an evaluation train/val dataset (e.g., ImageNet)
- Method: Linear SVM, k-NN classifier, *Linear probing*
  - *Freeze* the representation parts (e.g., encoder, layers except the final layer, …)
  - *Extract* features of train data
  - Attach a *classification layer* (e.g., linear layer) and train it
  - *Evaluate* the classifier on validation data
- Measure “How good the learned representation is.”
- *Contrastive learning* shows good performance here

Evaluation of learned representation

- Not freeze the encoder, *fine-tuning* them? 🤔
- We also have an evaluation train/val dataset (e.g., *ImageNet*)
- End-to-end fine-tuning
  - Attach a *classification layer* on top of the encoder
  - *Train* the *entire model* on train data
  - *Evaluate* the model on validation data
- Measure “transfer learning performance of the learned representation.”
- *Masked image modeling* shows good performance here

Contrastive learning vs. Masked image modeling

- Contrastive learning (CL) — distinguish positive pair from negatives
  - Captures global patterns
  - Later layers play a crucial role
  - Good at linear probing (well-separated final feature space)
- Masked image modeling (MIM) — reconstruct masked regions
  - Captures local patterns
  - Early layers play a crucial role
  - Good at fine-tuning (well transferrable to vision tasks)
- Will the *harmonization of CL and MIM* benefit? Yes! (Park et al., 2023)



Self-supervision and Model Adaptation

Self-supervision and model adaptation

- Things change after training

At training time

At test time

Self-supervision and model adaptation

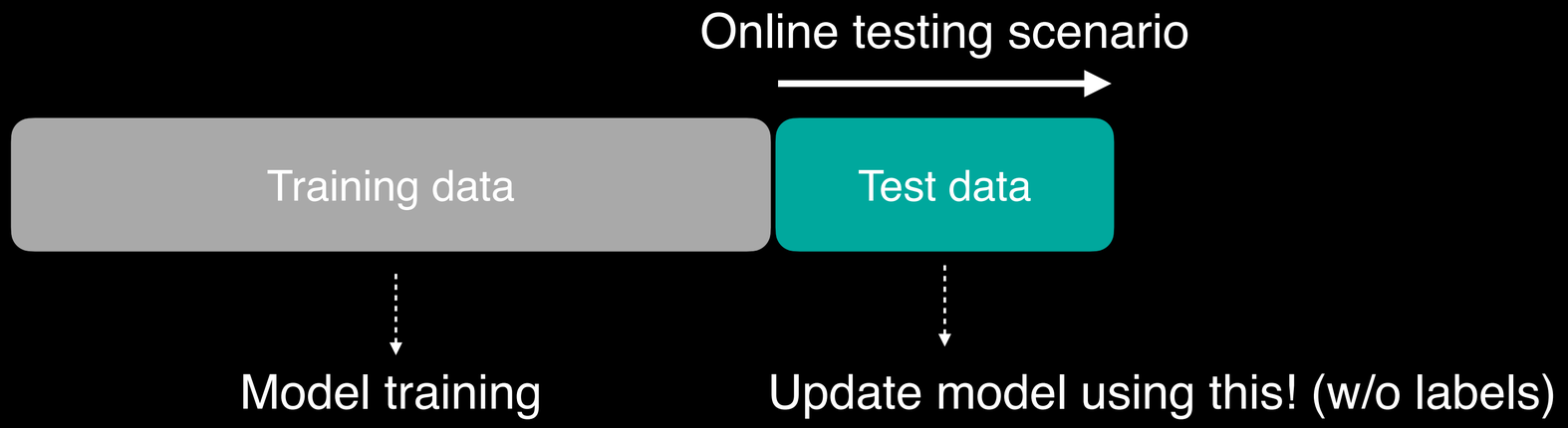
- Things change after training
- Need to adapt to test scenarios

At training time

At test time

Self-supervision and model adaptation

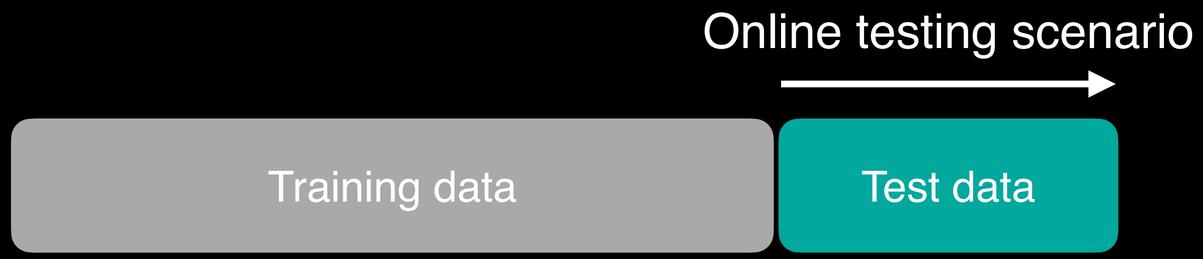
- Model adaptation *at the test time*
- Q) How can we adapt (or update) our model at the test time?



Reference: lecture 17 @ https://cs280-berkeley.github.io/

Self-supervision and model adaptation

- Model adaptation *at the test time*
- Q) How can we adapt (or update) our model at the test time?

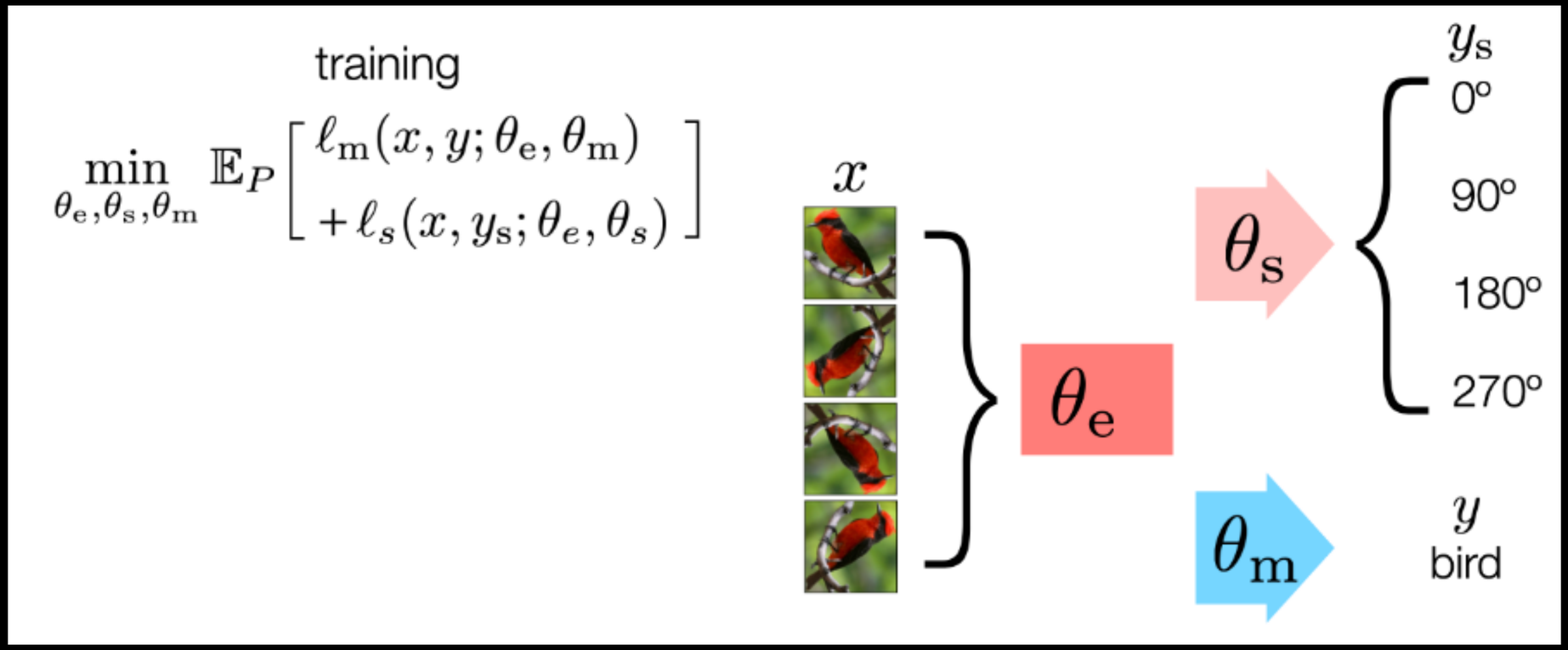


- A) Self-supervised learning

Reference: lecture 17 @ https://cs280-berkeley.github.io/

Self-supervision and model adaptation

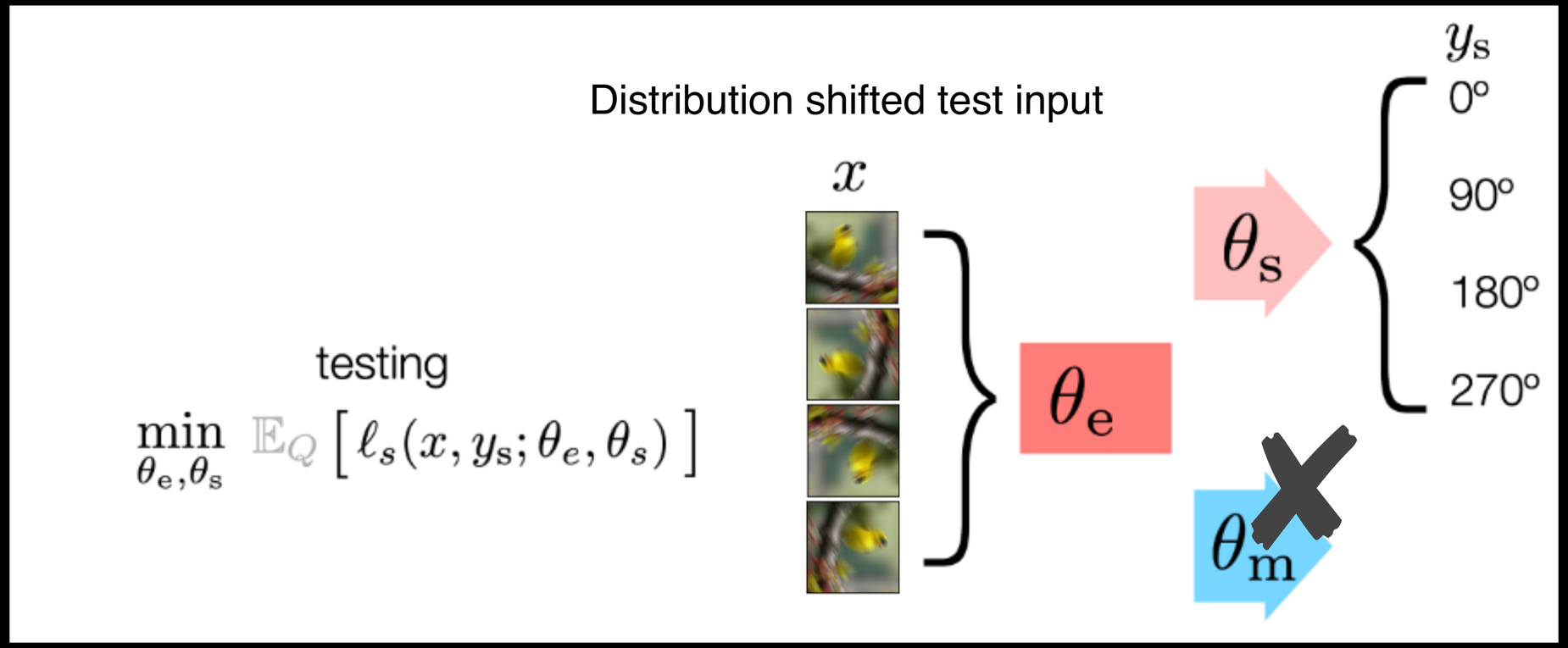
- *Test-Time Training* with Self-Supervision for Generalization under Distribution Shifts (Sun et al., ICML 2020)



Reference: lecture 17 @ https://cs280-berkeley.github.io/

Self-supervision and model adaptation

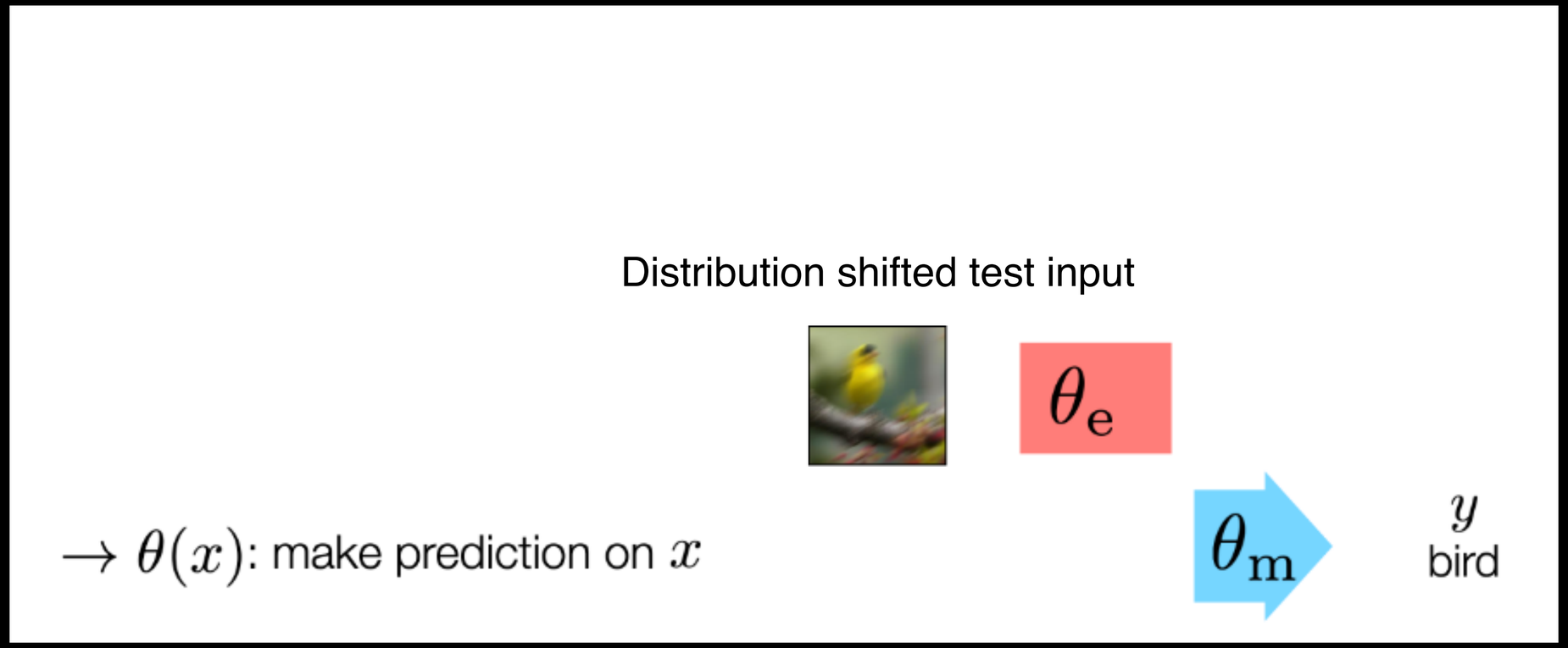
- *Test-Time Training* with Self-Supervision for Generalization under Distribution Shifts (Sun et al., ICML 2020)



Reference: lecture 17 @ https://cs280-berkeley.github.io/

Self-supervision and model adaptation

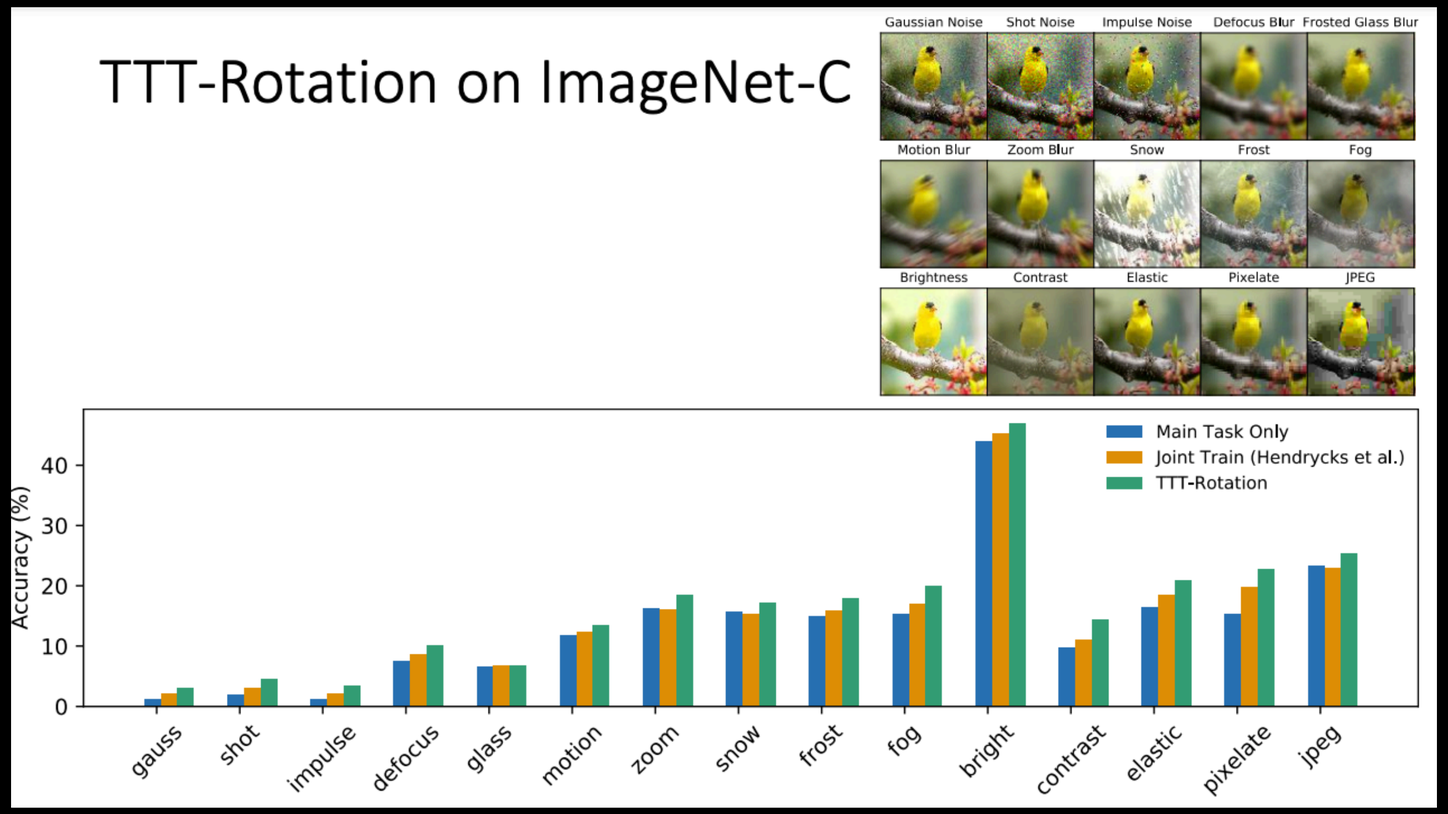
- *Test-Time Training* with Self-Supervision for Generalization under Distribution Shifts (Sun et al., ICML 2020)



Reference: lecture 17 @ https://cs280-berkeley.github.io/

Self-supervision and model adaptation

- *Test-Time Training* with Self-Supervision for Generalization under Distribution Shifts (Sun et al., ICML 2020)



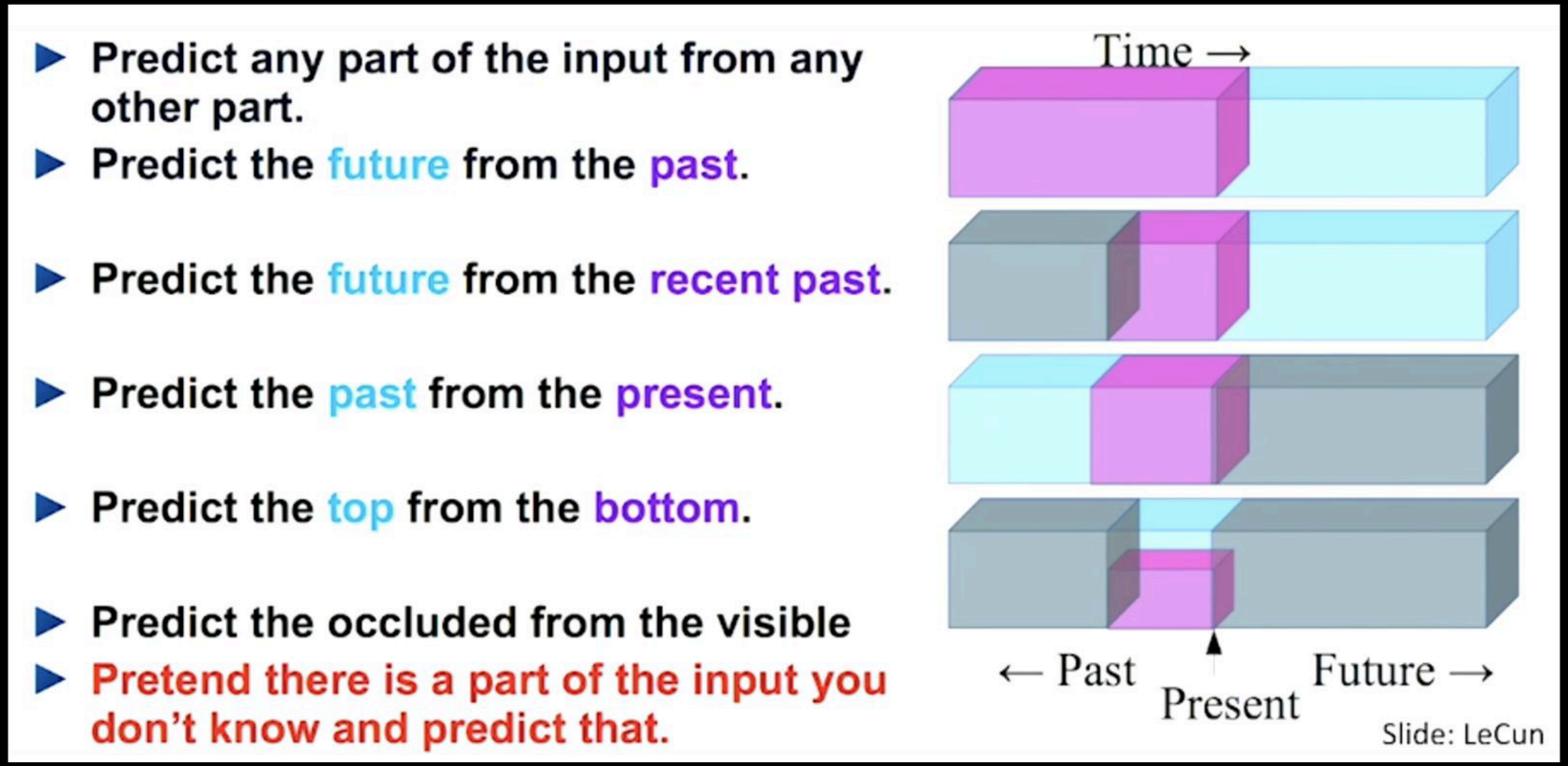
Reference: lecture 17 @ https://cs280-berkeley.github.io/

Self-supervised Learning

Video/Audio Pretext Modeling

Self-supervised learning on video

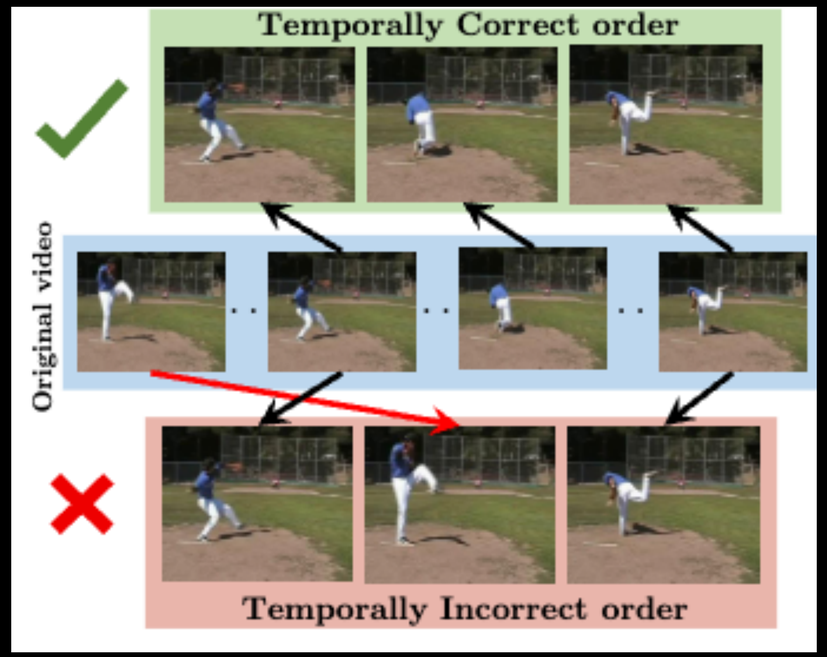
- Video data adds a temporal (time) axis → More versatile self-supervisions

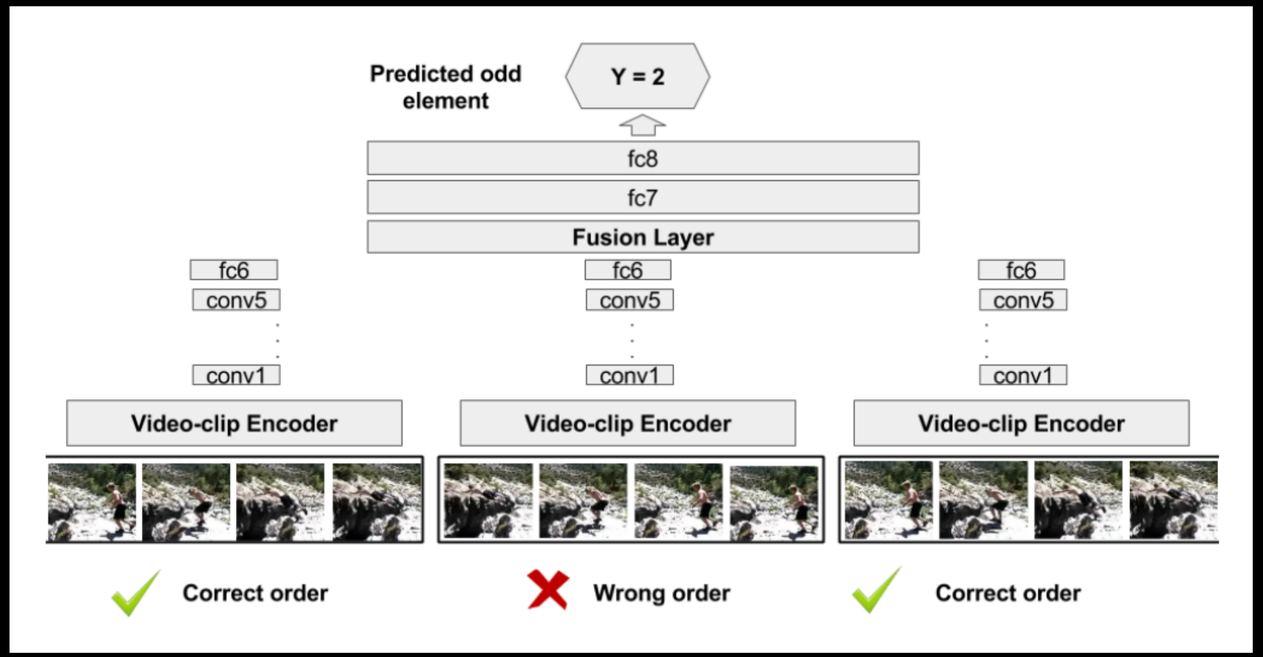


Slide from Yann LeCun

Video pretext tasks — Temporal order

- Temporal order verification (Misra et al. 2016, Fernando et al. 2017)
- *Shuffle* the order and predict

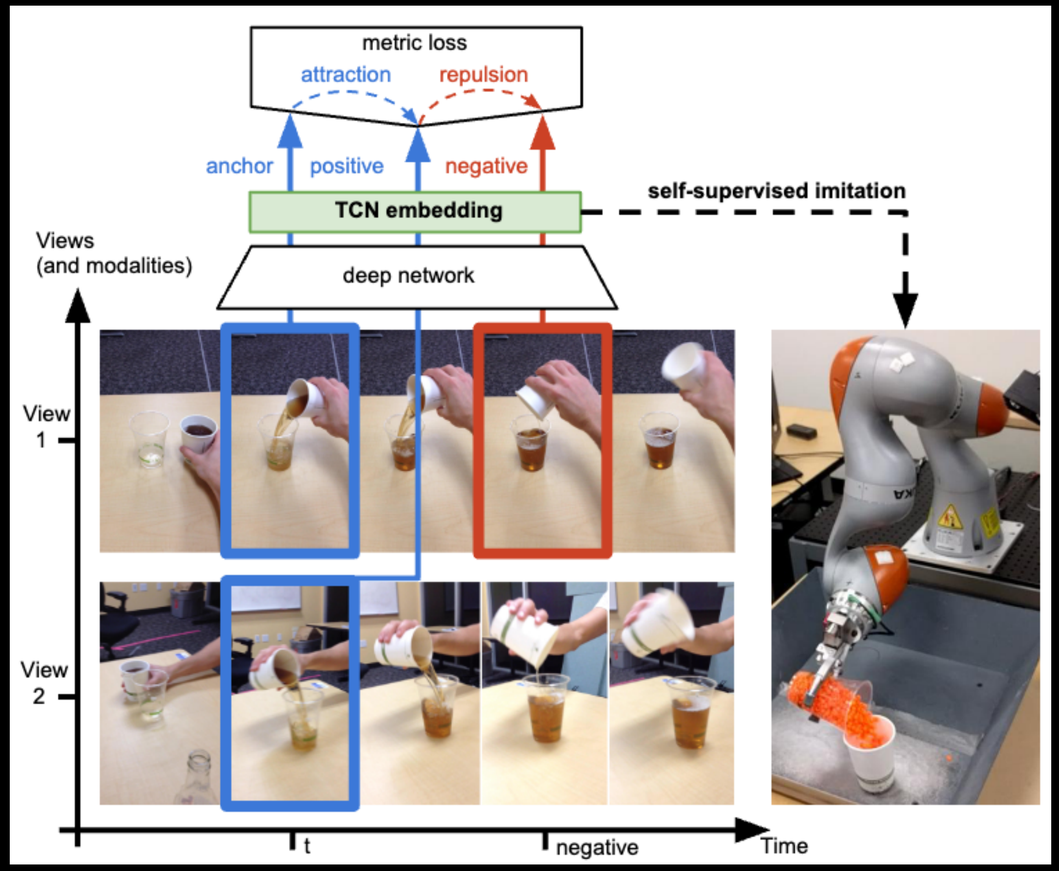




Misra et al., Shuffle and Learn: Unsupervised Learning using Temporal Order Verification, ECCV 2016
Fernando et al., Self-Supervised Video Representation Learning With Odd-One-Out Networks, CVPR 2017

Video pretext tasks — Contrastive learning

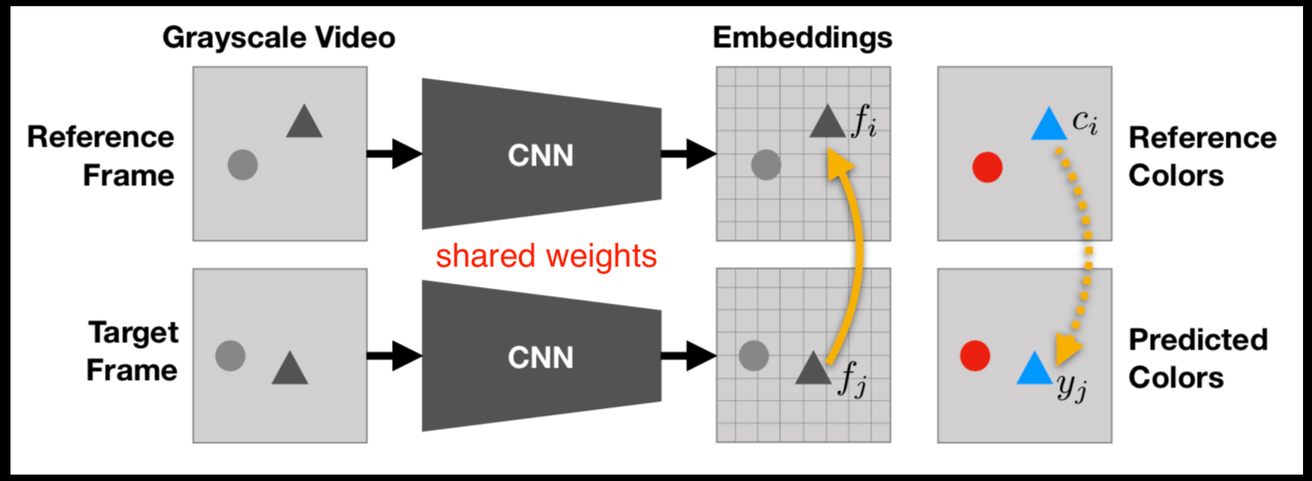
- Time-contrastive learning
- Multiple viewpoints with video frames
- Positive: *same time, different view*
- Negative: *different time, same view*
- (Use triplet loss)



Sermanet et al., “Time-Contrastive Networks: Self-Supervised Learning from Video”, ICLR 2018

Video pretext tasks — Colorization

- Video colorization (Vondrick et al., 2018)
  - Unlike the image-based colorization method (Zhang et al., 2016), uses two frames in video—reference frame (colorful) and target frame (gray)
  - Task: *colorize* the target frame given the reference frame
  - The model learns to *correlate pixels in different frames*.



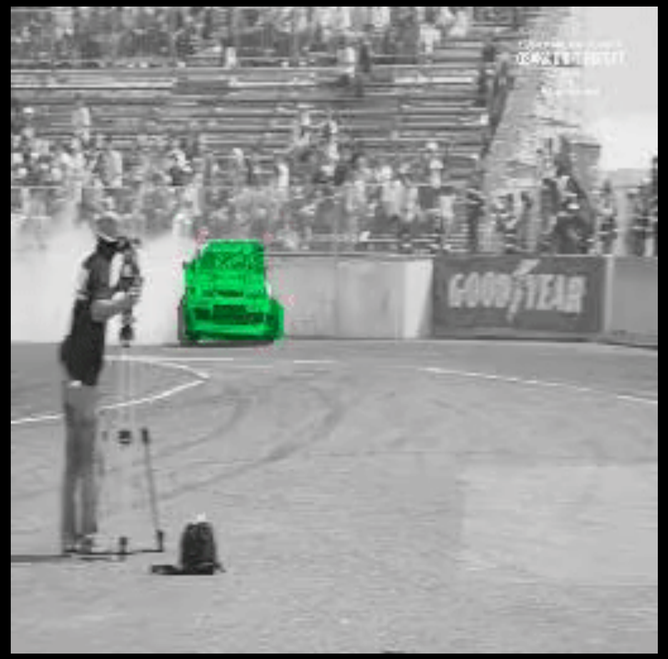
$$\hat{c}_j = \sum_i A_{ij} c_i \text{ where } A_{ij} = \frac{\exp(f_i f_j)}{\sum_{i'} \exp(f_{i'} f_j)}$$

Weighted sum Similarity of f_i and f_j

Vondrick et al., “Tracking Emerges by Colorizing Videos”, ECCV 2018

Video pretext tasks — Colorization

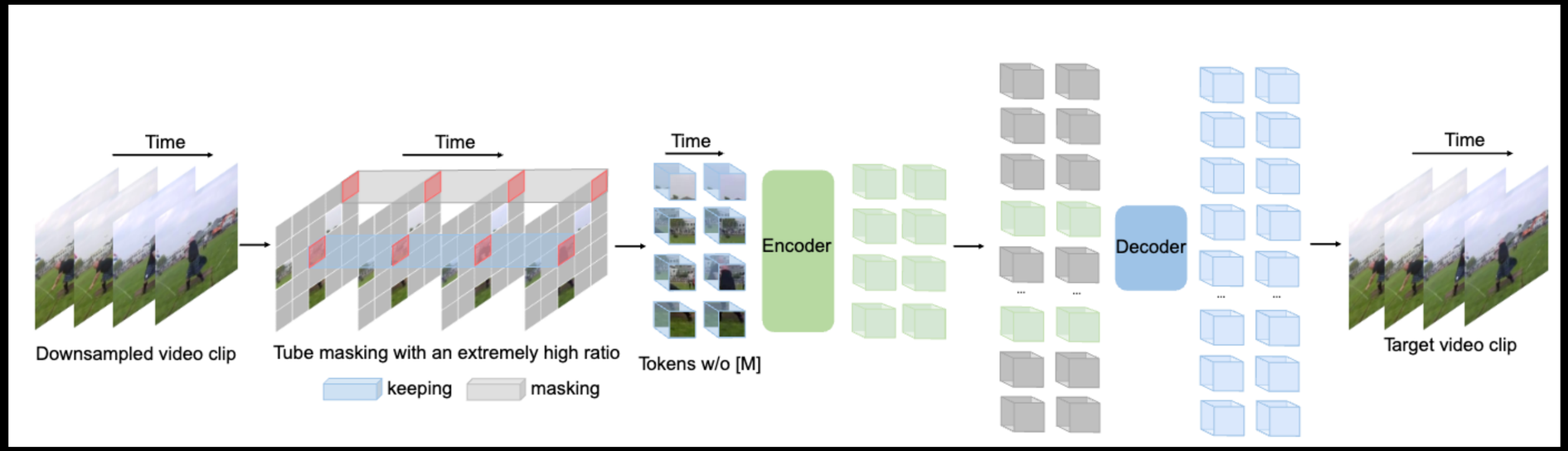
- Video colorization (Vondrick et al., 2018)
  - Get rich representation ability
  - Video segmentation and visual region tracking, *without extra fine-tuning*.



Vondrick et al., “Tracking Emerges by Colorizing Videos”, ECCV 2018

Video pretext tasks — Masked modeling

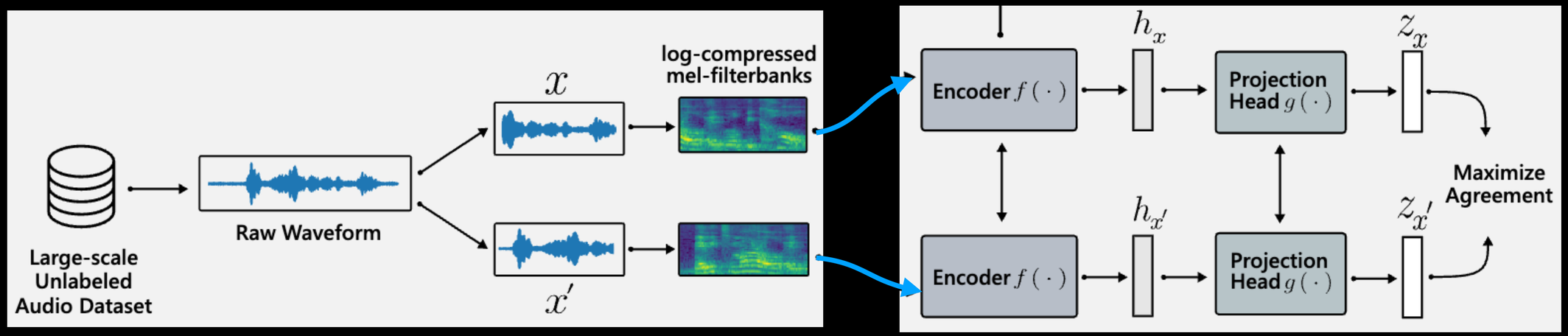
- Video Masked Auto-encoder (Tong et al., 2022, Feichtenhofer et al., 2022)
- *Mask* spatiotemporal tokens and *predict* the dropped tokens.



Tong et al., “VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training”, NeurIPS 2022
Feichtenhofer et al., “Masked Autoencoders As Spatiotemporal Learners”, NeurIPS 2022

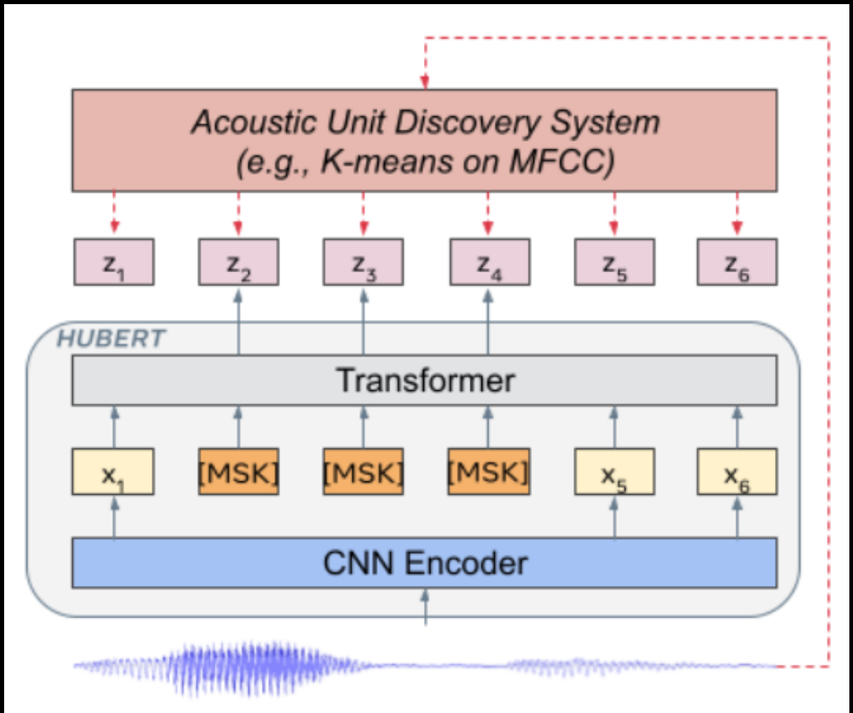
Audio pretext task — Contrastive learning

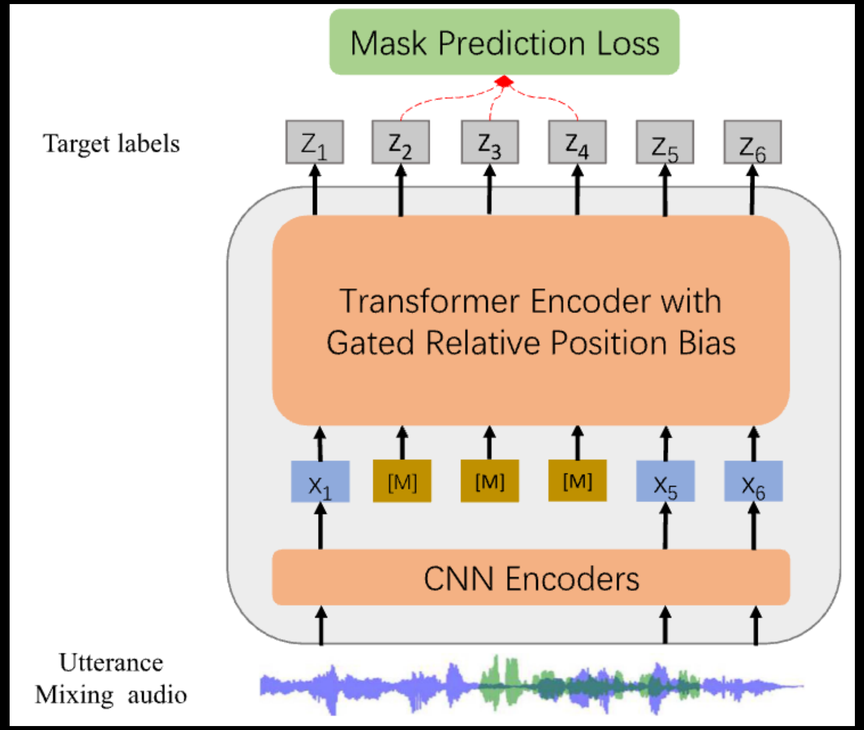
- Contrastive learning on audio data (Saeed et al., 2021)
  - *High similarity* between audio clips extracted from *the same recording*
  - *Low similarity* to clips from *different recordings*



Audio pretext task — Masked modeling

- HuBERT (Hsu et al., 2021) and WavLM (Chen et al., 2022)





Hsu et al., “HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units”, 2021.
WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing, 2022

Conclusion

- What is Representation Learning?
- Supervised Learning
- Self-supervised Learning

- Next class: *Multimodal Foundation Model 2 – Multimodal Pre-Training*

Thank You!

Reference materials

- https://cs280-berkeley.github.io/
- https://nips.cc/media/neurips-2021/Slides/21895.pdf