

Self-supervised Learning

AI602: Recent Advances in Deep Learning

Lecture 07

Slide made by

Hankook Lee, Sangwoo Mo, Hyunwoo Kang

KAIST EE

1. Introduction

- Overview of Self-supervised Learning (SSL)
- Evaluating Self-supervised Representation

2. SSL via Pretext Tasks

- Pretext Tasks for Vision

3. SSL via Invariance (and Contrast)

- Clustering, Consistency, Contrastive
- Choices for Positive Samples

4. SSL via Generation

- Classic Approaches
- Masked Autoencoder (e.g., BERT, MAE)
- Sequential Prediction (e.g., GPT, world model)

1. Introduction

- Overview of Self-supervised Learning (SSL)
- Evaluating Self-supervised Representation

2. SSL via Pretext Tasks

- Pretext Tasks for Vision

3. SSL via Invariance (and Contrast)

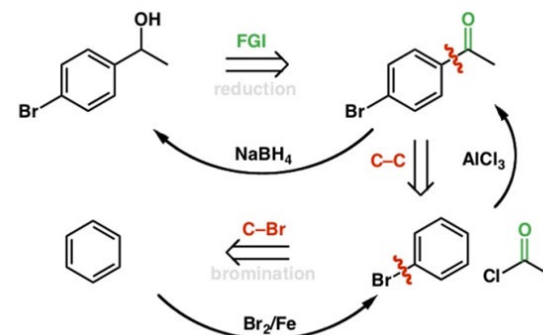
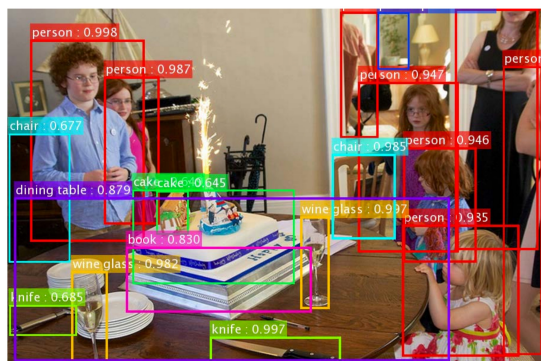
- Clustering, Consistency, Contrastive
- Choices for Positive Samples

4. SSL via Generation

- Classic Approaches
- Masked Autoencoder (e.g., BERT, MAE)
- Sequential Prediction (e.g., GPT, world model)

Motivation

- DNNs achieve **remarkable success** in various applications
 - They usually **require massive amounts of manually labeled data**
 - The **annotation cost is high** because
 - It is **time-consuming**: e.g., annotating bounding boxes of all objects
 - It requires **expert knowledge**: e.g., medical diagnosis and retrosynthesis

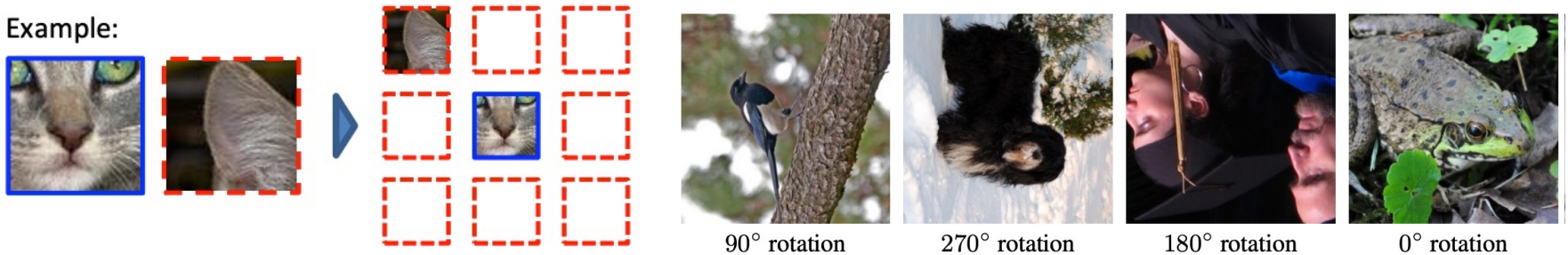


- But, **collecting unlabeled samples is extremely easy** compared to annotation
- **Q.** How to utilize the **unlabeled samples** for learning DNNs?


- **Self-supervision?**

- It is **a label** constructed **from only input signals** without human-annotation
- Using self-supervision, one can apply supervised learning approaches
- Examples: Predicting relative location of patches¹ or rotation degree²

Example:

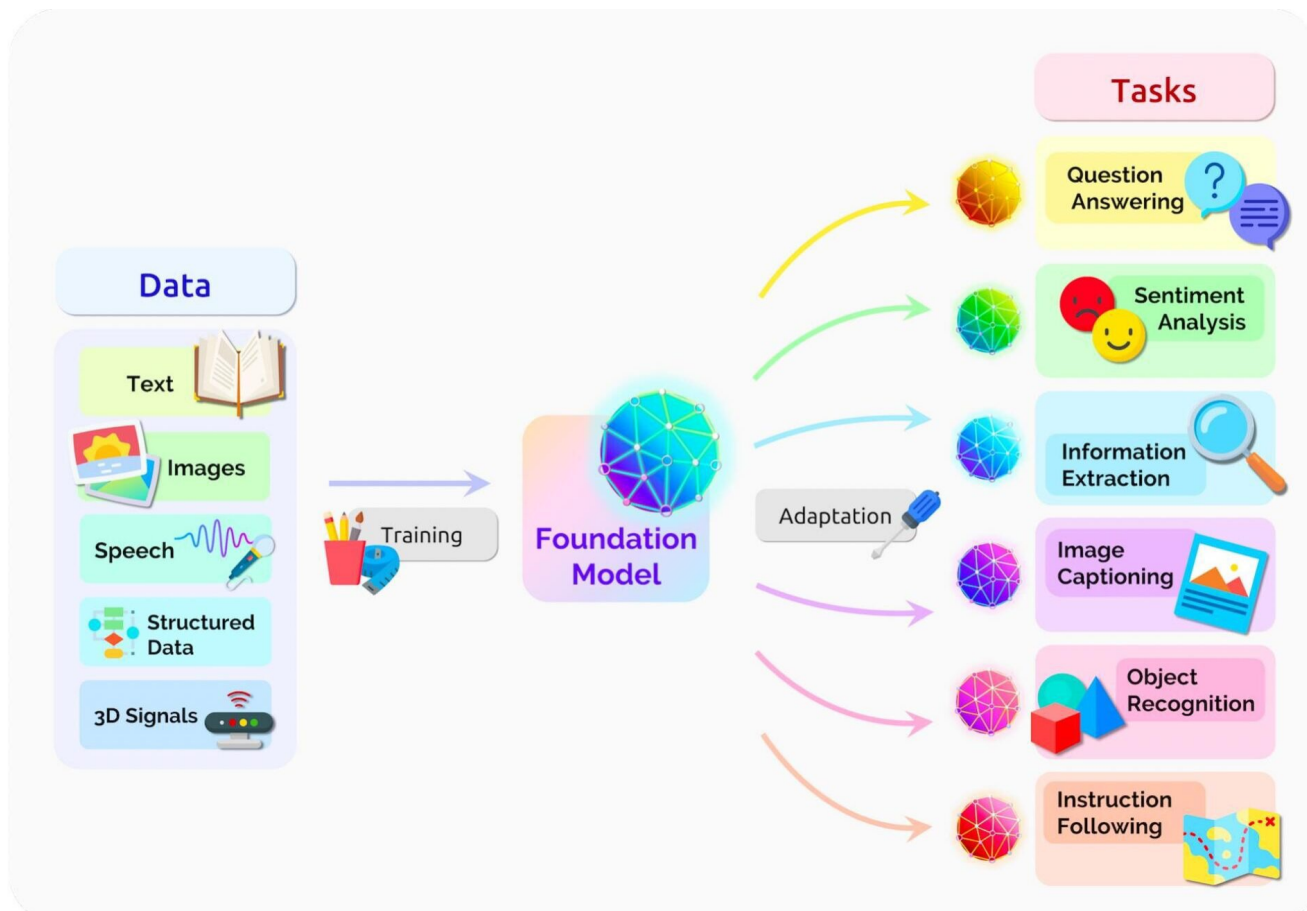


- What can we learn from self-supervised learning?

- To predict (well-designed) self-supervision, one might **require high-level understanding of inputs**
- E.g., we should know  is the right ear of the cat for predicting locations
- Thus, **high-level representations could be learned w/o human-annotation**

• Foundation Models

- Fixing a foundation model (e.g., trained via self-supervised learning) and only adapting a **simple task-specific model** is sufficient for many problems
 - E.g., linear classifier upon the SimCLR/BERT backbone



I now call it “self-supervised learning”, because “unsupervised” is both a loaded and confusing term. ...

Self-supervised learning uses way more supervisory signals than supervised learning, and enormously more than reinforcement learning. That’s why calling it “unsupervised” is totally misleading.

by Yann LeCun (2019. 04. 30)

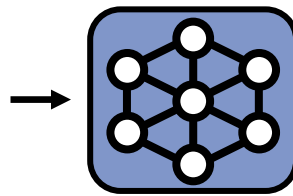
- How to define “unsupervised learning” term? (there is no answer ...)
 - **Q)** We need an objective (or loss) for learning; is the objective not a (self-)supervision?
 - **Q)** Unsupervised learning \supseteq self-supervised learning?
 - **Q)** What are the purely unsupervised learning methods?
 - In classic ML, clustering, grouping and dimensionality reduction ...
- In this lecture,
 - We mainly use the “self-supervised learning” term instead of unsupervised learning
 - We learn recent SSL approaches in vision, NLP, and graph domains

- **How to evaluate the quality of self-supervision?**

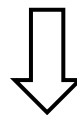
1. Self-supervised learning in a large-scale dataset (e.g., ImageNet)
2. Transfer the pretrained network to various downstream tasks
 - **Linear probing**: freeze the network and training only the linear classifier
⇒ **it directly evaluates the learned representation qualities**
 - **Fine-tuning** whole parameters



ImageNet (1.2M images)



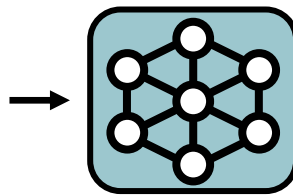
→ Pretraining (self-supervised learning)



Transfer (initialization)



Flowers102 (2k images)



→ Linear evaluation or Fine-tuning

- We mainly follow the history of **“SSL for images”**
 - **2015-2018: Pretext Tasks**
 - Context Prediction ('15), Jigsaw Puzzle ('16), Colorization ('17), Rotation ('18)
 - **2019-2021: Contrastive Learning**
 - NPID ('18), MoCo/SimCLR ('20), BYOL ('20), MoCov3/DINO ('21)
 - Similar idea was also considered in Exemplar-CNN ('14)
 - **2022-: Masked Autoencoder**
 - BEiT ('22), MAE ('22), data2vec ('22)
 - Similar idea was also considered in Context Encoder ('16)
- Current SSL approaches can be categorized into **3 groups**:
 - Let X be data, $Z(X)$ be representation, and Y be pretext label
 - I denotes mutual information (MI) of two random variables
 1. **Pretext task:** Maximize $I(Z(X); Y)$ where Y is pretext label of X
 2. **Invariance:** Maximize $I(Z(X_1); Z(X_2))$ where X_1, X_2 are invariant data
 3. **Generation:** Maximize $I(Z(\tilde{X}); X)$ where \tilde{X} is perturbed version of X

1. Introduction

- Overview of Self-supervised Learning (SSL)
- Evaluating Self-supervised Representation

2. SSL via Pretext Tasks

- Pretext Tasks for Vision

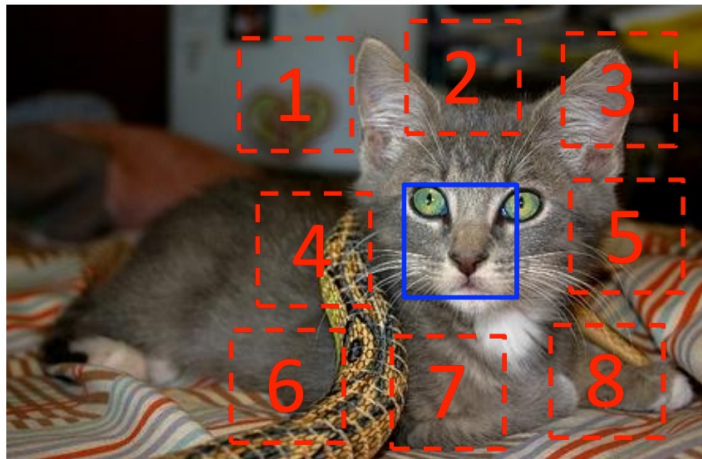
3. SSL via Invariance (and Contrast)

- Clustering, Consistency, Contrastive
- Choices for Positive Samples

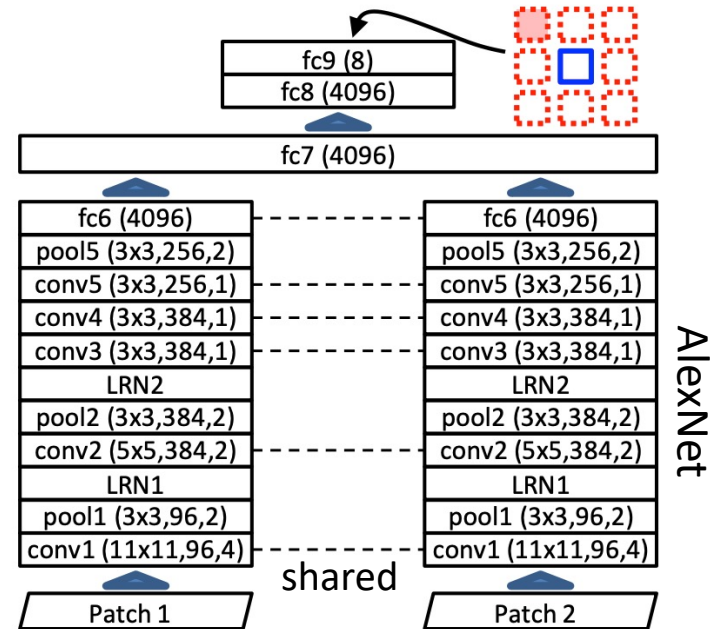
4. SSL via Generation

- Classic Approaches
- Masked Autoencoder (e.g., BERT, MAE)
- Sequential Prediction (e.g., GPT, world model)

- **Context Prediction** [Doersch et al., 2015]
 - From a natural image, extract 3x3 patches
 - **Patch 1**: The center patch
 - **Patch 2**: Select one of other patches randomly
 - **Task**: Given **Patch 1 & 2**, predict the location of the second patch (8-way classification)

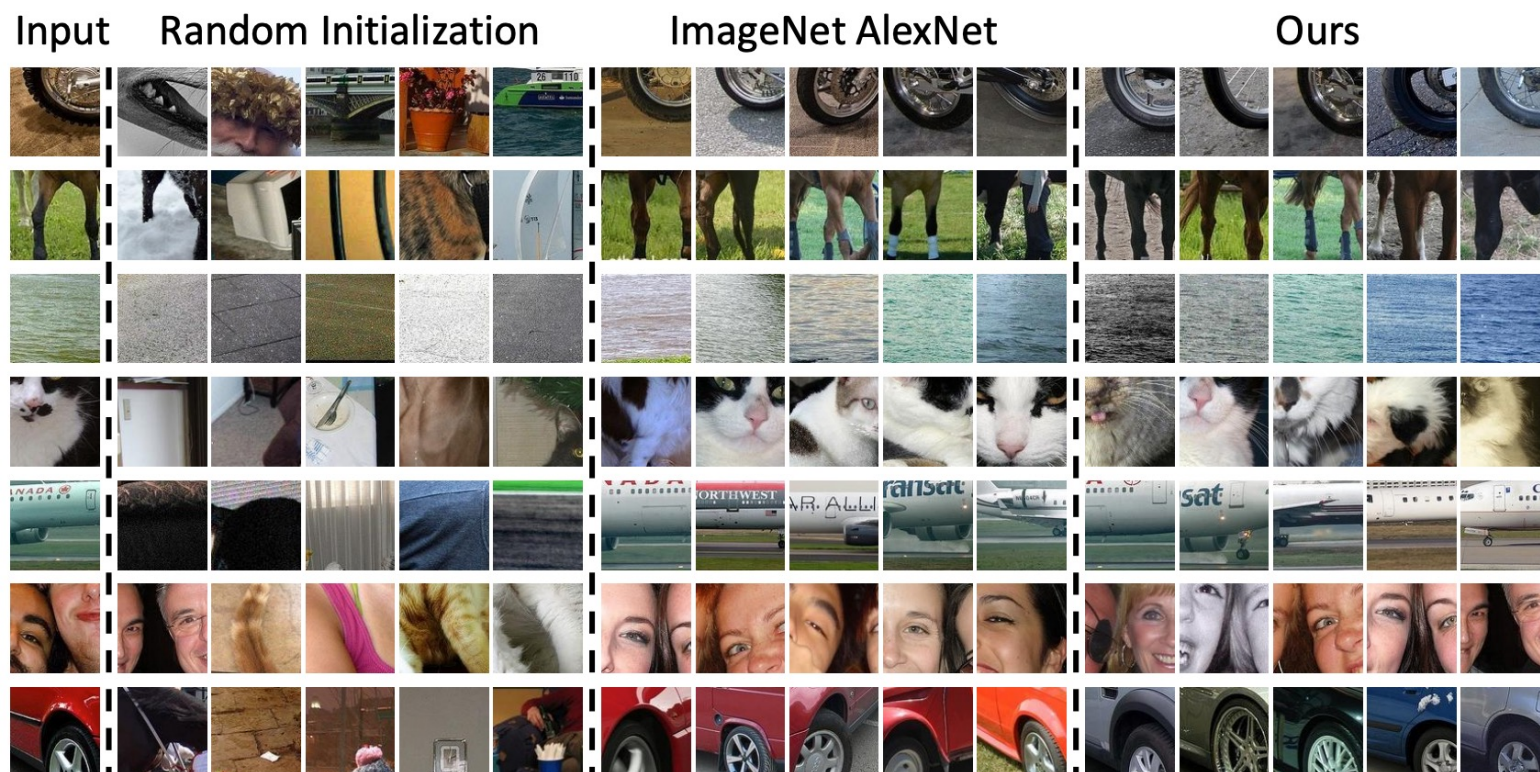


$$X = (\text{Patch 1}, \text{Patch 2}); Y = 3$$

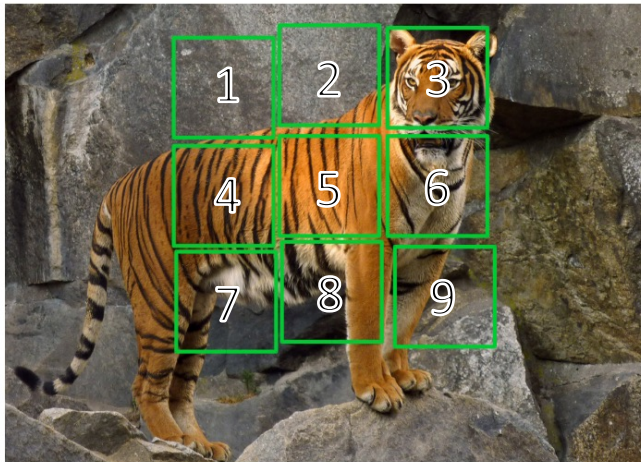


- Each patch's embedding is computed by one **shared** encoder (AlexNet)

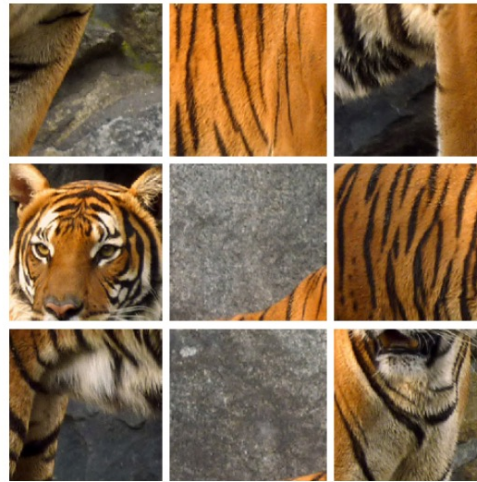
- **Context Prediction** [Doersch et al., 2015]
 - **Task:** Given **Patch 1 & 2**, predict the location of the second patch (8-way classification)
 - This pretext task assigns similar representations to semantically similar patches
 - Qualitative analysis of nearest neighbors of learned representations



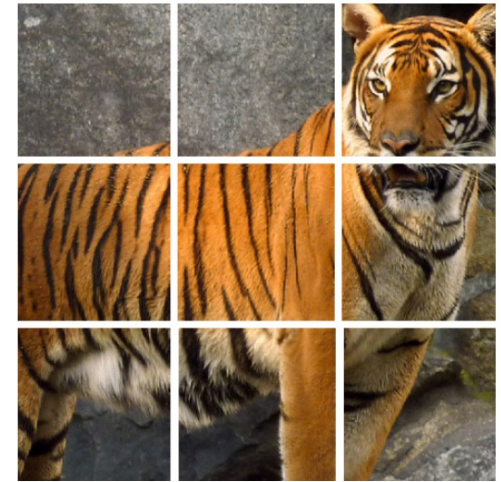
- **Jigsaw Puzzle** [Noroozi & Favaro, 2016]
 - Extension of [Doersch et al., 2015]
 - (a) Extract 3x3 patches from a natural image; (b) **permute** the patches randomly
 - **Task:** From (b) the shuffled patches, find **which permutation is applied**



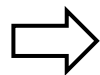
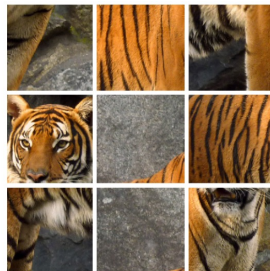
(a)



(b)



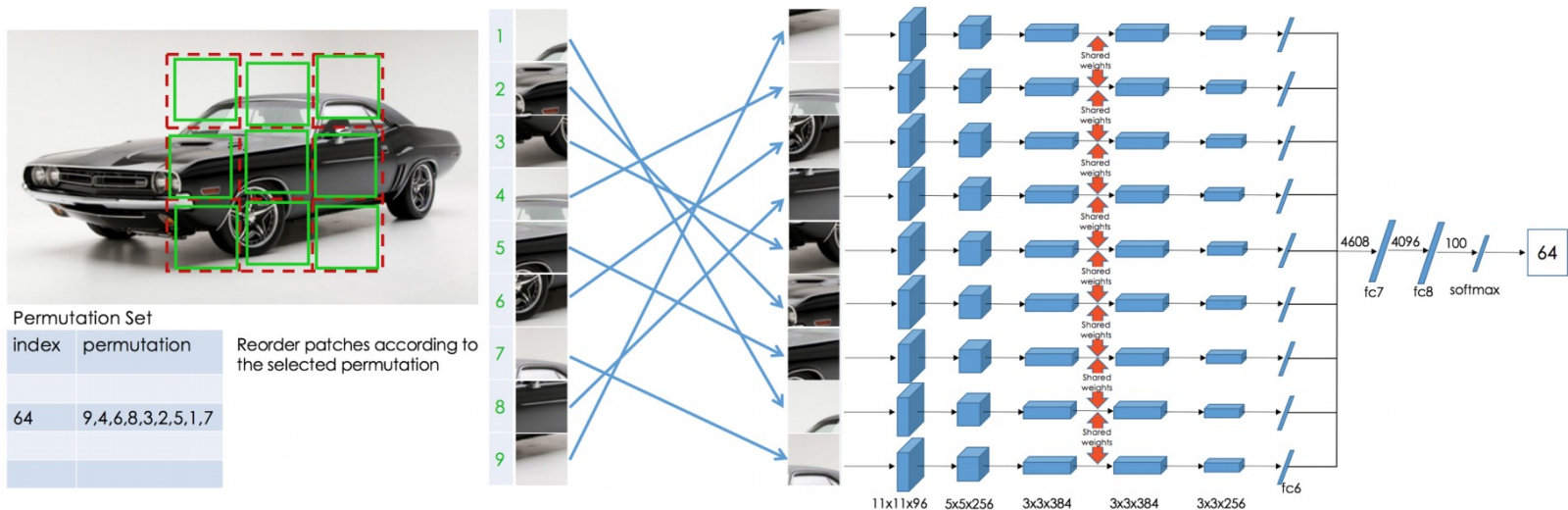
(c)



How to solve this Jigsaw puzzle?

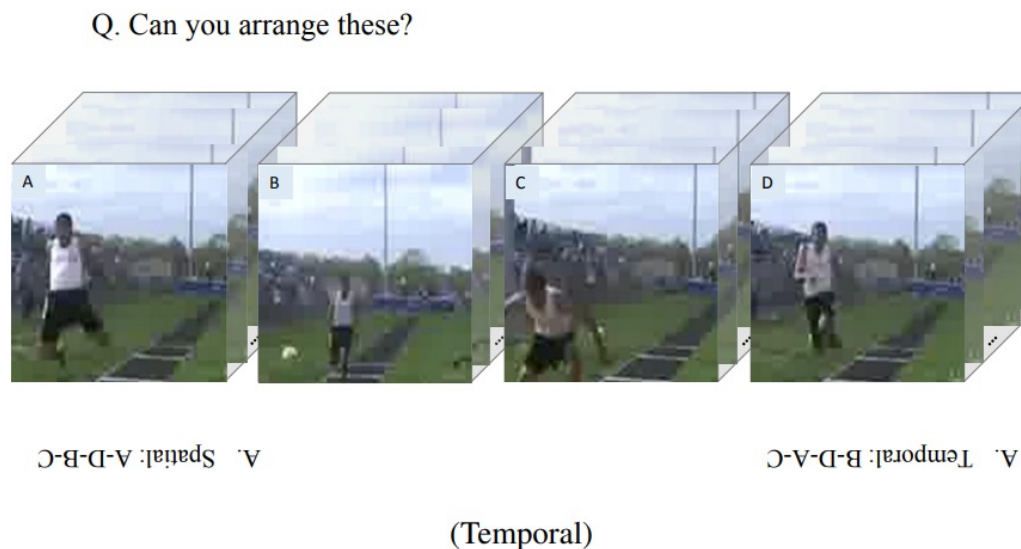


- **Jigsaw Puzzle** [Noroozi & Favaro, 2016]
 - Extension of [Doersch et al., 2015]
 - (a) Extract 3x3 patches from a natural image; (b) **permute** the patches randomly
 - **Task:** From (b) the shuffled patches, find **which permutation is applied**



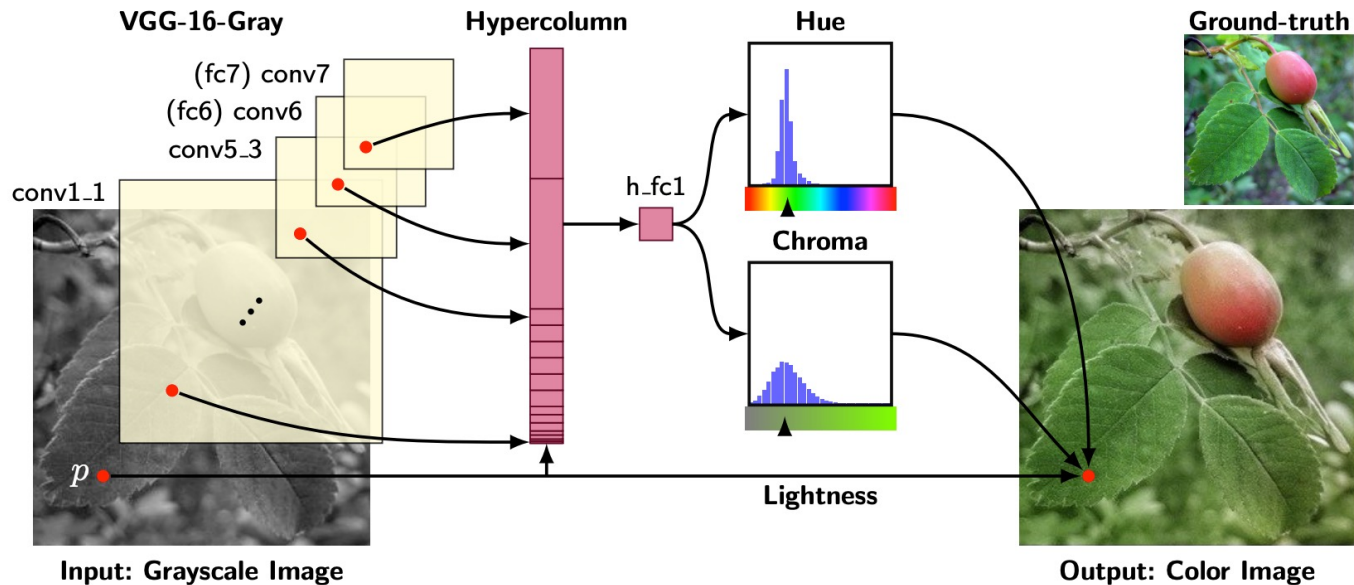
- Each patch's embedding is computed by one **shared** encoder
- There are too many permutations ($9! = 362k$) \Rightarrow choose a subset of them
 - Empirically, **neither simple nor ambiguous tasks** achieve better performance

- **Space-Time Cubic Puzzles** [Kim et al., 2019] for video representation
 - **Self-supervised pretext task** for **3D CNNs** with **temporal dimension**
 - Train a network to **predict** their **original spatio-temporal arrangement**
 - Pretraining with the learned representation outperforms supervised pretraining methods in video action recognition

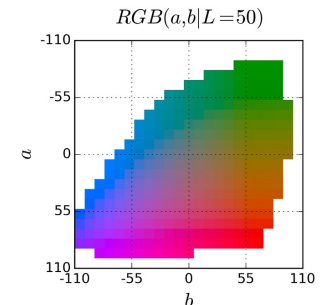


- **Colorization** [Zhang et al., 2016]

- **Task:** Predict color information from a grayscale image
- Colorization requires **dense prediction**
- \Rightarrow **Hypercolumn:** concatenate feature maps of different layers

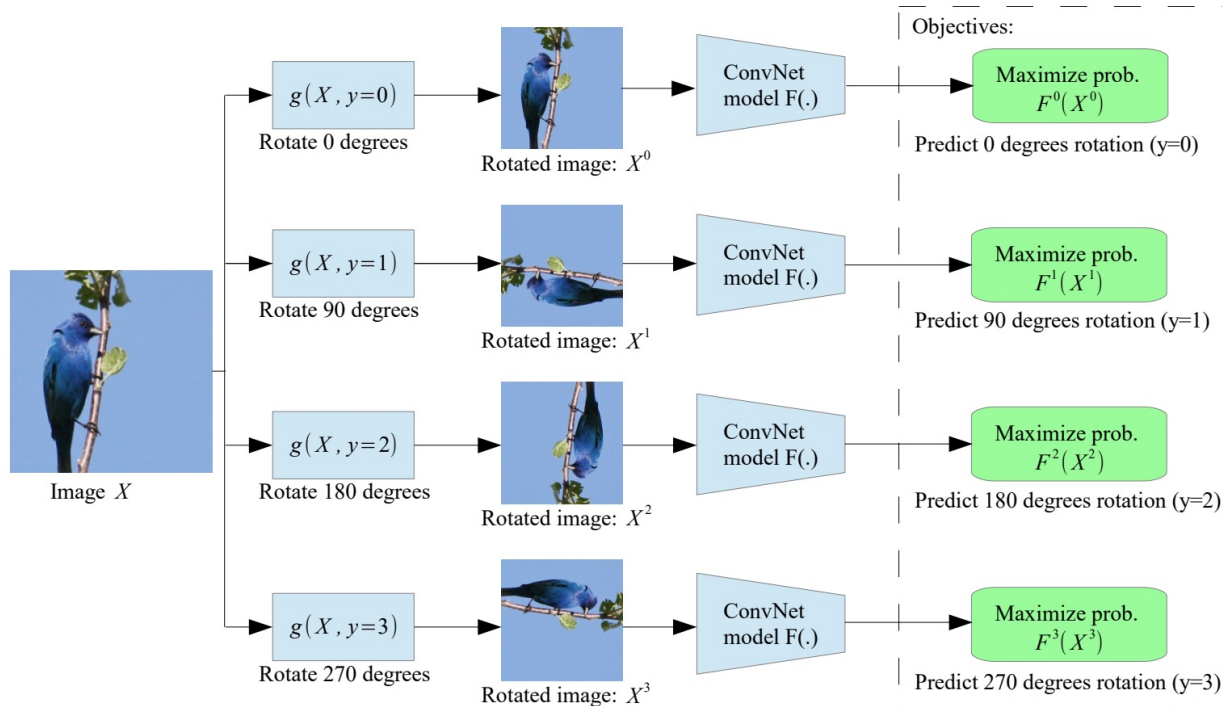


- Formulate colorization as **classification** instead of regression
 - Quantize Lab color space for classification
 - This can handle **multi-modal** color distributions well



* source : [Zhang et al., 2016]

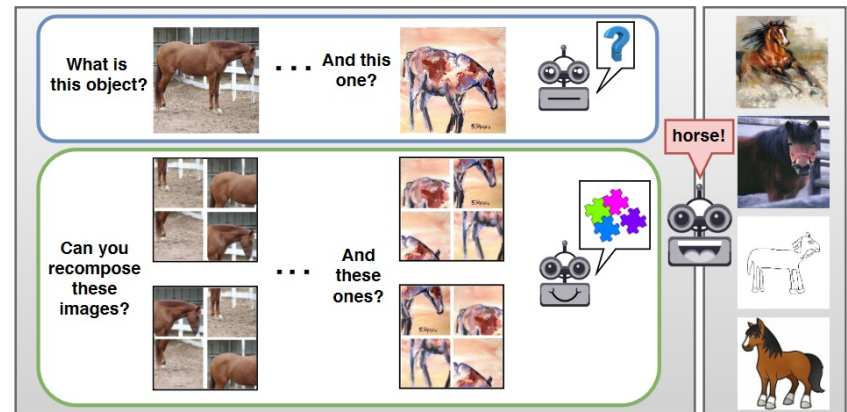
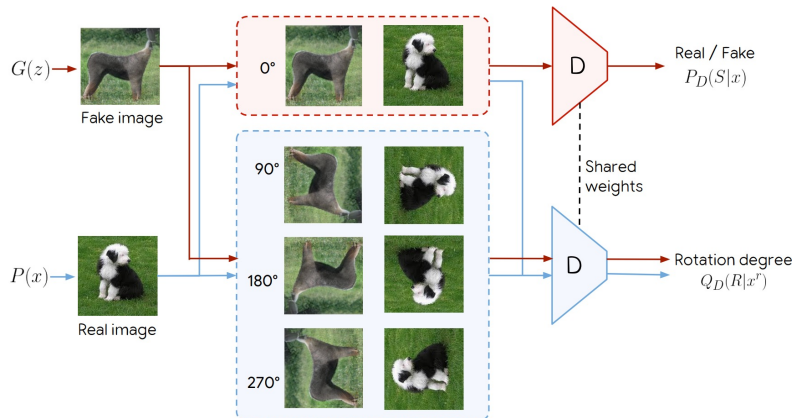
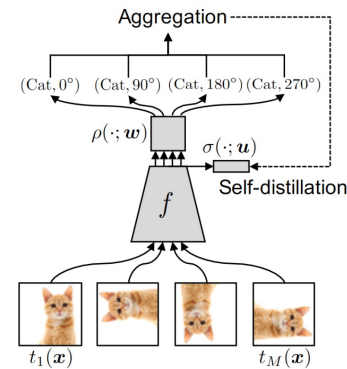
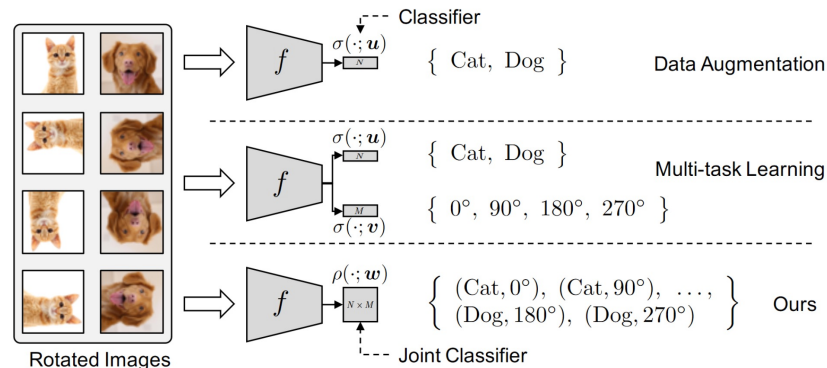
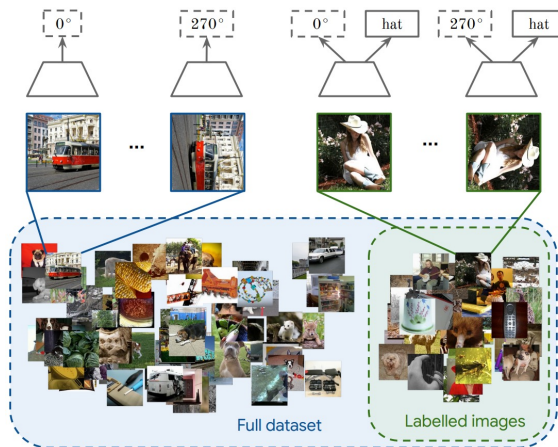
- **Rotation** [Gidaris et al., 2018]
 - **Task:** Predict the rotation degree from a rotated image (4-way classification)



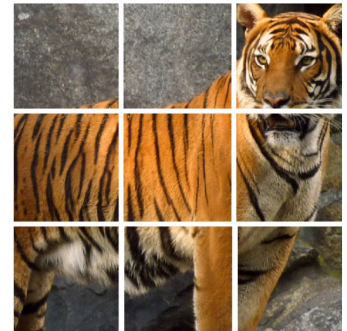
- What is the optimal number of classes (rotations)?
 - Empirically, using 4 rotations (0° , 90° , 180° , 270°) is best

- **Rotation** [Gidaris et al., 2018]

- **Task:** Predict the rotation degree from a rotated image
- Due to **its simplicity**, this approach is widely used for other applications
 - Semi-supervised Learning [Zhai et al., 2019], Supervised Learning [Lee et al., 2020], GAN [Chen et al., 2019] Domain Generalization [Carlucci et al., 2019]



- **Limitations** on handcrafted pretext tasks
 1. **Domain-specific knowledge is required to design self-supervision**
 - In different domains (e.g., audio), existing methods might be not working
 2. **The use of self-supervision is limited**
 - Patch-based tasks for small-sized datasets, e.g., CIFAR
 - Colorization-based tasks for single-channel inputs, e.g., gray images
 3. **Pre-processing is important to avoid trivial solutions**
 - For example, one can solve Jigsaw puzzle by using color information in **boundaries**
- **Next:** more general approaches
 - Invariance-based approaches
 - Generation-based approaches



1. Introduction

- Overview of Self-supervised Learning (SSL)
- Evaluating Self-supervised Representation

2. SSL via Pretext Tasks

- Pretext Tasks for Vision

3. SSL via Invariance (and Contrast)

- Clustering, Consistency, Contrastive
- Choices for Positive Samples


4. SSL via Generation

- Classic Approaches
- Masked Autoencoder (e.g., BERT, MAE)
- Sequential Prediction (e.g., GPT, world model)

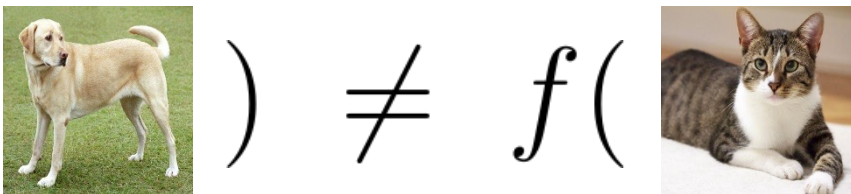
Core idea of invariance-based learning:

- **Invariance:** Representations of related samples should be similar
- **Contrast** (optional): Representations of unrelated samples should be dissimilar

Positive pair $f\left(\text{img}_1\right) \approx f\left(\text{img}_2\right)$



Negative pair $f\left(\text{img}_1\right) \neq f\left(\text{img}_2\right)$




- **Q)** How to construct positive/negative pairs in the unsupervised setting?

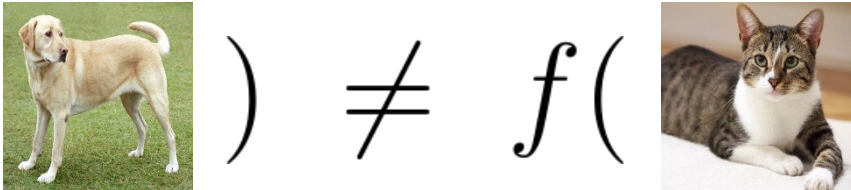
Core idea of invariance-based learning:

- **Invariance:** Representations of related samples should be similar
- **Contrast** (optional): Representations of unrelated samples should be dissimilar

Positive pair $f\left(\text{img}_1\right) \approx f\left(\text{img}_2\right)$



Negative pair $f\left(\text{img}_1\right) \neq f\left(\text{img}_2\right)$

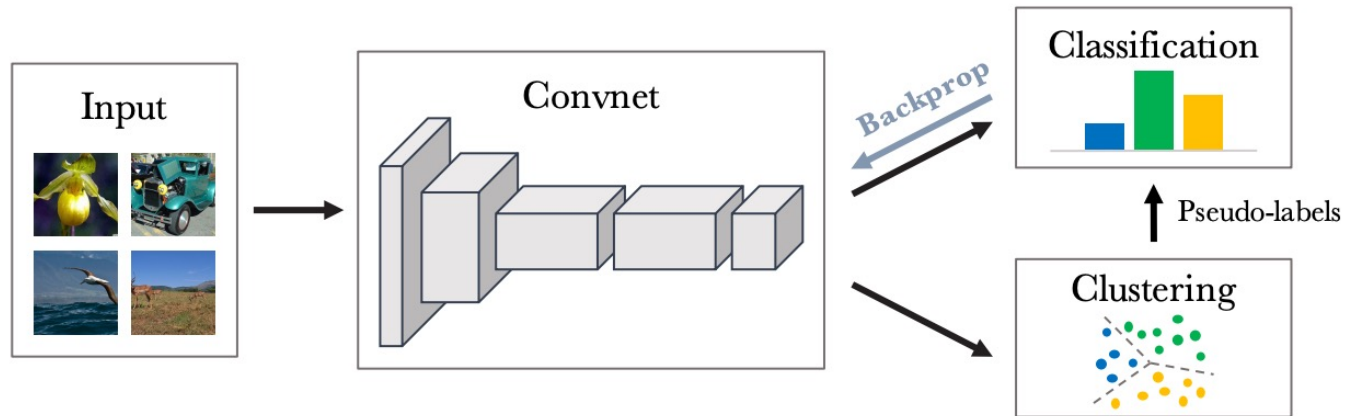


- **Q)** How to construct positive/negative pairs in the unsupervised setting?
- **A)** Positive samples are constructed from
 - Similar samples (e.g., in the same cluster)
 - Same instance of different data augmentation
 - Additional structures (e.g., multi-view images, video)(negative samples = not positive samples)

- **Instantiations of invariance-based approach**

- Many classes of self-supervised learning can be viewed as invariance-based
- **Clustering & pseudo-labeling**
 - **Cluster** data into K groups, and assume they are **pseudo-labels**
 - Distill pseudo-labels to the self-supervised classifier (strengthen the similarity)
 - E.g., DeepCluster, SwAV, DINO
- **Consistency regularization**
 - **Attract** similar samples
 - E.g., MixMatch, UDA, BYOL
- **Contrastive learning**
 - **Attract** similar samples and **dispel** dissimilar samples
 - E.g., MoCo, SimCLR, CLIP

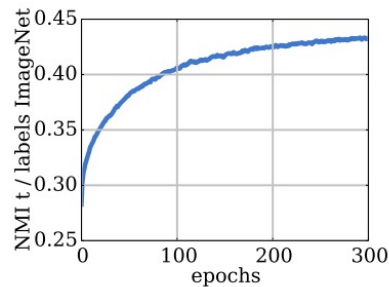
- **DeepCluster** [Caron et al., 2018]
 - **Idea:** Clustering on embedding space provides pseudo-labels



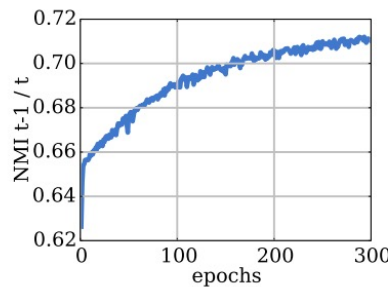
- **Simple method:** Alternate between
 1. Clustering the features to produce pseudo-labels
 2. Updating parameters by predicting these pseudo-labels
- How to avoid **trivial solutions**?
 - Empty cluster \Leftarrow feature quantization (it reassigns empty clusters)
 - Imbalanced sizes of clusters \Leftarrow over-sampling

- **DeepCluster** [Caron et al., 2018]

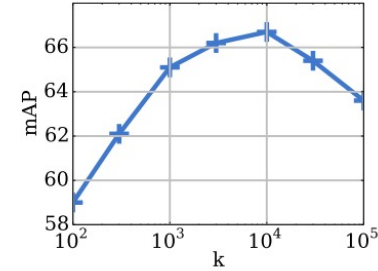
- Is the **clustering quality** improved during training?
 - a. Clustering overlap between DeepCluster and ImageNet
 - b. Clustering overlap between the current and previous epochs
 - c. Influence of the number of clusters



(a) Clustering quality



(b) Cluster reassignment

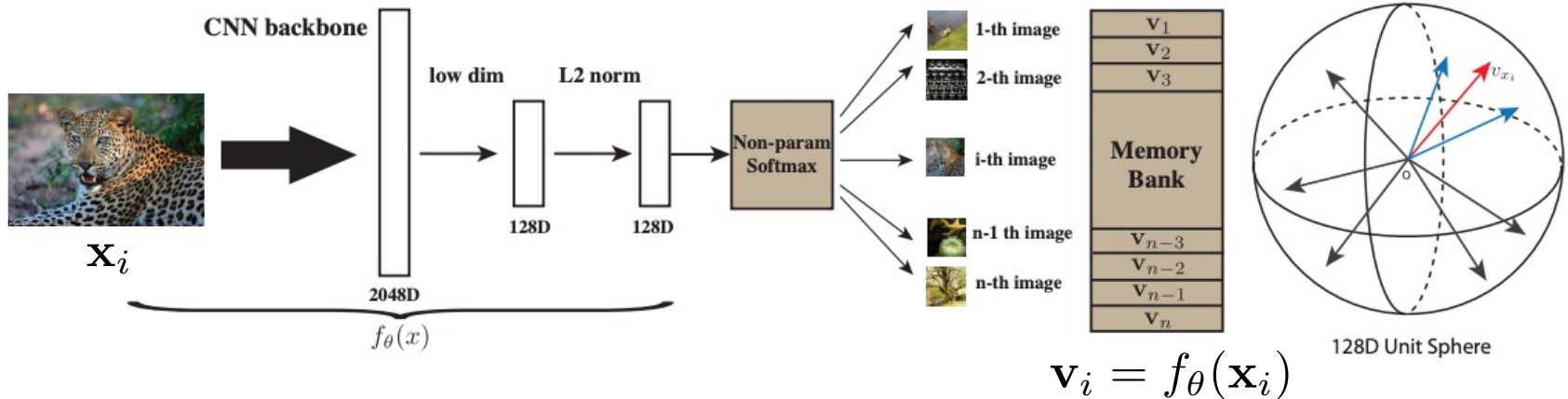


(c) Influence of k

- **Which images activate the target filters** in the last convolutional layer?



- **Instance Discrimination** [Wu et al., 2018]
 - **Idea:** Each image belongs to an unique class



- **Non-parameteric classifier**

$$P(i|\mathbf{v}) = \frac{\exp(\mathbf{v}_i^\top \mathbf{v} / \tau)}{\sum_{j=1}^n \exp(\mathbf{v}_j^\top \mathbf{v} / \tau)}$$

- Each class has only one instance $\Rightarrow \mathbf{v}_i$ can be used directly as a class prototype

- **Instance Discrimination** [Wu et al., 2018]

- **Idea:** Each image belongs to an unique class

- **Non-parameteric classifier**

$$P(i|\mathbf{v}) = \frac{\exp(\mathbf{v}_i^\top \mathbf{v} / \tau)}{\sum_{j=1}^n \exp(\mathbf{v}_j^\top \mathbf{v} / \tau)}$$

- Computing $P(i|\mathbf{v})$ is **inefficient** because it requires all $\mathbf{v}_j = f_\theta(\mathbf{x}_j)$ and $\mathbf{v}_j^\top \mathbf{v}$
- **Solution 1:** Memory bank
 - Store all \mathbf{v}_j in memory and update them for each mini-batch
 - To stabilize training, representations in memory bank are **momentum-updated**

$$\mathbf{v}_i^{(t)} \leftarrow m \mathbf{v}_i^{(t-1)} + (1 - m) \mathbf{v}_i^{\text{new}}$$

Representations in memory bank

Computed by current encoder

- **Instance Discrimination** [Wu et al., 2018]

- **Idea:** Each image belongs to an unique class

- **Non-parameteric classifier**

$$P(i|\mathbf{v}) = \frac{\exp(\mathbf{v}_i^\top \mathbf{v} / \tau)}{\sum_{j=1}^n \exp(\mathbf{v}_j^\top \mathbf{v} / \tau)}$$

- Computing $P(i|\mathbf{v})$ is **inefficient** because it requires all $\mathbf{v}_j = f_\theta(\mathbf{x}_j)$ and $\mathbf{v}_j^\top \mathbf{v}$
- **Solution 1:** Memory bank
 - Store all \mathbf{v}_j in memory and update them for each mini-batch
 - To stabilize training, representations in memory bank are **momentum-updated**
- **Solution 2:** Noise-Contrastive Estimation [Gutmann & Hyvarinen, 2010]
 - It casts multi-class classification into a set of binary classification problems

Positive sample: $P(D = 1|i, \mathbf{v}) = P(i|\mathbf{v}) = \frac{\exp(\mathbf{v}_i^\top \mathbf{v})}{\exp(\mathbf{v}_i^\top \mathbf{v}) + \sum_{k=1}^m \exp(\mathbf{v}_{j_k}^\top \mathbf{v})}$

m negative samples

Objective: $\mathcal{L}_{\text{NCE}} = -\mathbb{E}_{P_d}[\log P(D = 1|i, \mathbf{v})] - m\mathbb{E}_{P_n}[\log P(D = 0|i, \mathbf{v}')]$

data distribution
noise distribution (uniform)

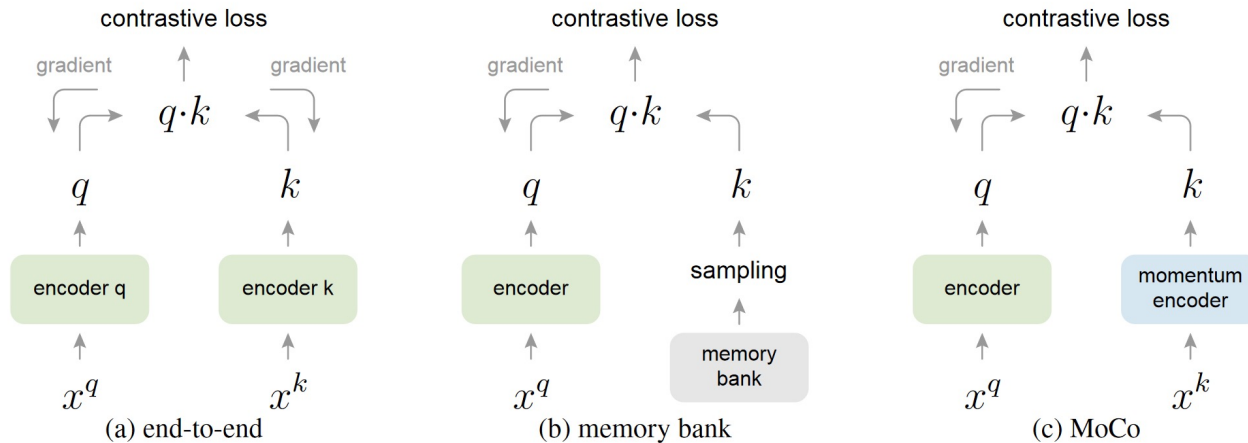
- **Instance Discrimination** [Wu et al., 2018]
 - Ablation Study
 - **Non-parametric softmax is better** than the parametric version
 - **NCE with many negative samples** approaches to the no-approximation version

Training / Testing	Linear SVM	Nearest Neighbor
Param Softmax	60.3	63.0
Non-Param Softmax	75.4	80.8
NCE $m = 1$	44.3	42.5
NCE $m = 10$	60.2	63.4
NCE $m = 512$	64.3	78.4
NCE $m = 4096$	70.2	80.4

- Large embedding size increases the performance, but it is saturated at 256

embedding size	32	64	128	256
top-1 accuracy	34.0	38.8	41.0	40.1

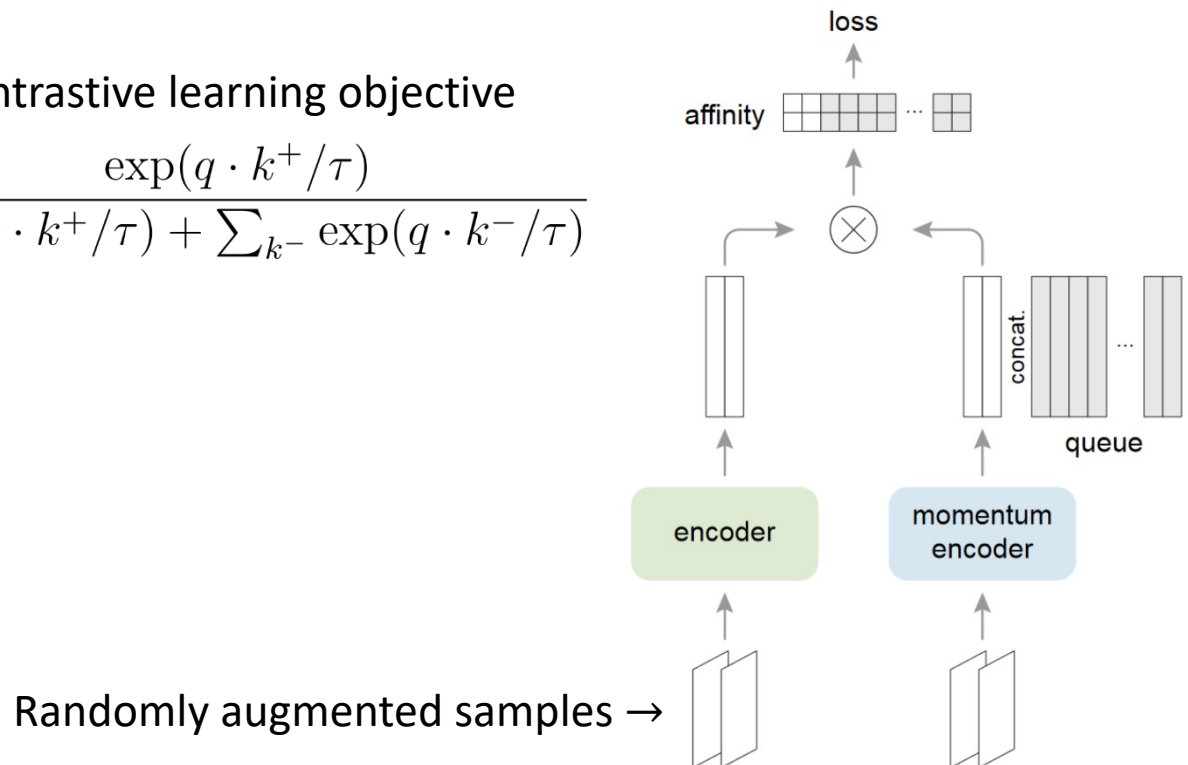
- **Momentum Contrast (MoCo)** [He et al., 2019]
 - **Key issue:** the number of negatives is very crucial in contrastive learning
 - How to resolve this issue in prior works? **Memory Bank**
 - Note: representations in the memory bank are momentum-updated
 - **MoCo's idea:** use a **momentum-updated encoder** and maintain a **queue**



- **Momentum encoder** increases the **key representations' consistency**
- **Queue** allows us to use **recent and many negative** samples

- **Momentum Contrast (MoCo)** [He et al., 2019]
 - **Key issue:** the number of negatives is very crucial in contrastive learning
 - How to resolve this issue in prior works? **Memory Bank**
 - Note: representations in the memory bank are momentum-updated
 - **MoCo's idea:** use a **momentum-updated encoder** and maintain a **queue**
- MoCo also optimizes contrastive learning objective

$$\mathcal{L}_{q,k^+,\{k^-\}} = -\log \frac{\exp(q \cdot k^+ / \tau)}{\exp(q \cdot k^+ / \tau) + \sum_{k^-} \exp(q \cdot k^- / \tau)}$$



- **Momentum Contrast (MoCo)** [He et al., 2019]
 - **Key issue:** the number of negatives is very crucial in contrastive learning
 - How to resolve this issue in prior works? **Memory Bank**
 - Note: representations in the memory bank are momentum-updated

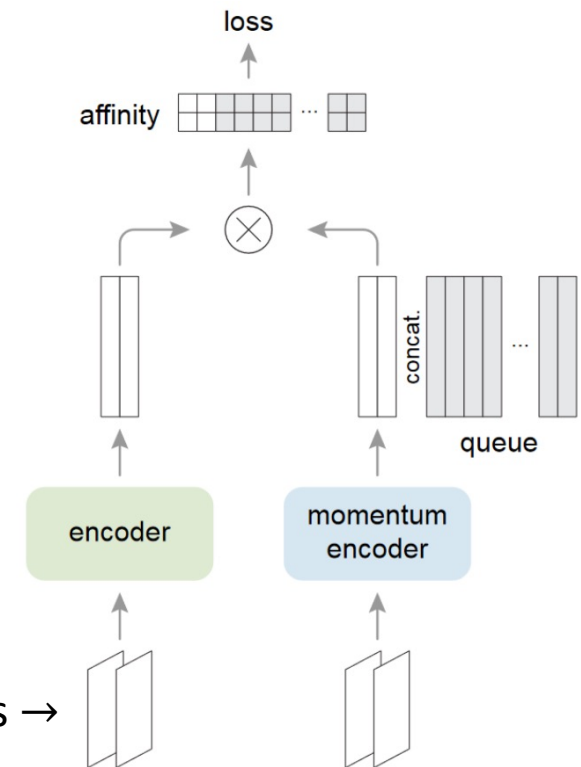
- **MoCo's idea:** use a **momentum-updated encoder** and maintain a **queue**

- MoCo also optimizes contrastive learning objective

$$\mathcal{L}_{q,k^+,\{k^-\}} = -\log \frac{\exp(q \cdot k^+ / \tau)}{\exp(q \cdot k^+ / \tau) + \sum_{k^-} \exp(q \cdot k^- / \tau)}$$

- After **encoder** is updated,
 - **Momentum encoder** is updated by

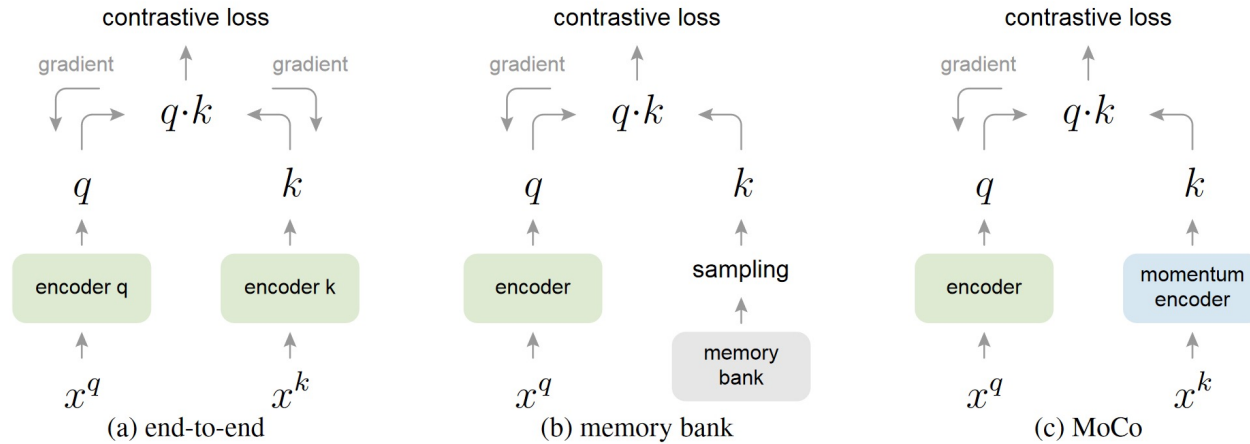
$$\theta_{\text{momentum}} \leftarrow m\theta_{\text{momentum}} + (1 - m)\theta$$
 - Add the current positive keys k^+ into the queue



Randomly augmented samples →

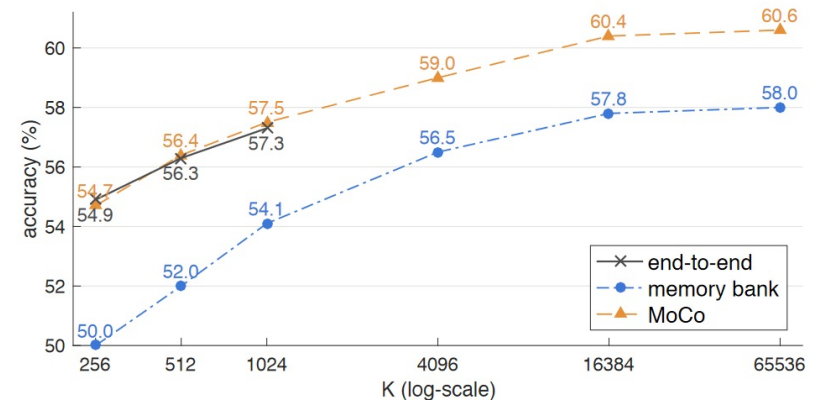
• Momentum Contrast (MoCo) [He et al., 2019]

- **MoCo's idea:** use a **momentum-updated encoder** and maintain a **queue**



- **Momentum encoder** increases the **key representations' consistency**
- **Queue** allows us to use **recent and many negative** samples

momentum m	0	0.9	0.99	0.999	0.9999
accuracy (%)	<i>fail</i>	55.2	57.8	59.0	58.9



- **SimCLR** [Chen et al., 2020]
 - A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank
 - This paper founds that contrastive learning benefits from ...
 1. **Strong augmentation** (i.e., composition of multiple data augmentation operations)
 2. **A nonlinear MLP** between the representation and the contrastive loss
 3. **Large batch** sizes and **longer training**

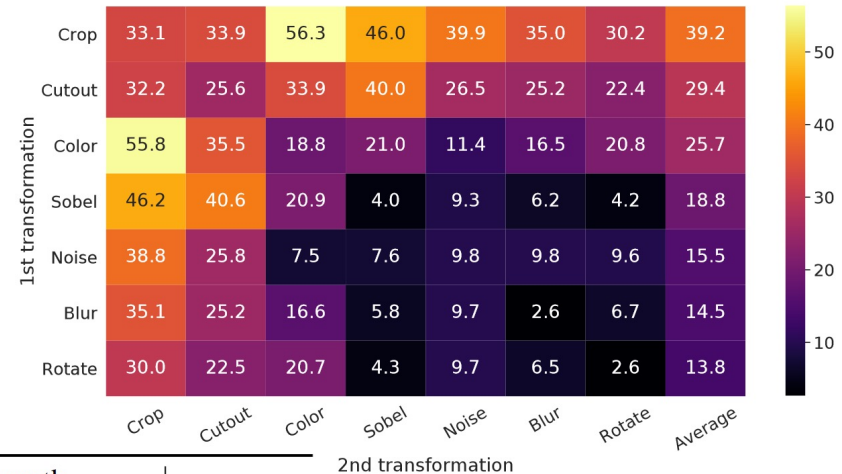
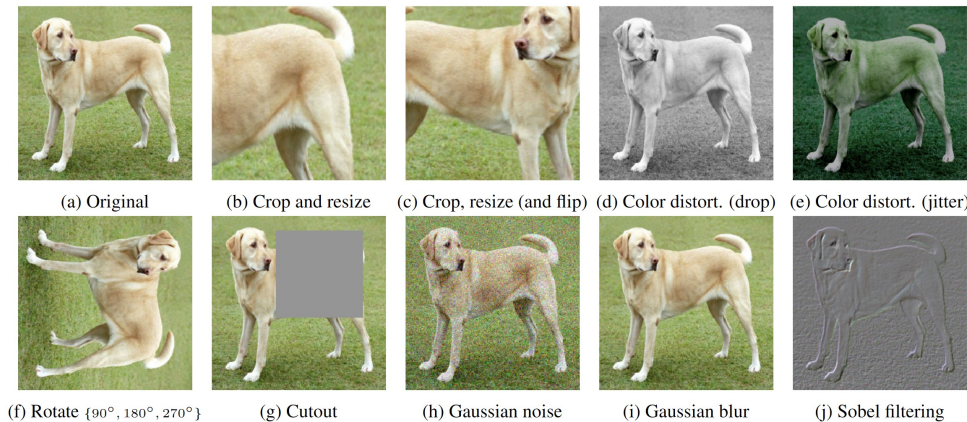
- **SimCLR** [Chen et al., 2020]

- A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank

- This paper finds that contrastive learning benefits from ...

1. **Strong augmentation** (i.e., composition of multiple data augmentation operations)

- Strong color distortion degrades supervised learning, but improves SimCLR
- A stronger augmentation (AutoAugment) degrades SimCLR



Methods	Color distortion strength					AutoAug
	1/8	1/4	1/2	1	1 (+Blur)	
SimCLR	59.6	61.0	62.6	63.2	64.5	61.1
Supervised	77.0	76.7	76.5	75.7	75.4	77.1

• SimCLR [Chen et al., 2020]

- A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank

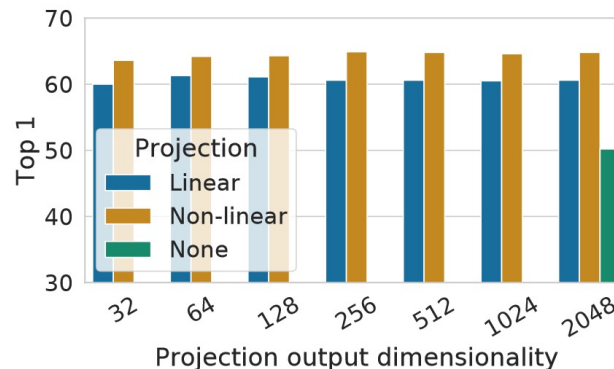
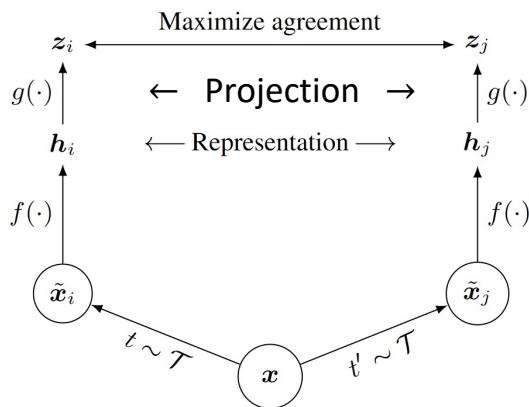
- This paper founds that contrastive learning benefits from ...

2. A nonlinear MLP between the representation and the contrastive loss

- Contrastive learning objective learns \mathbf{z} to be **invariant to augmentations**

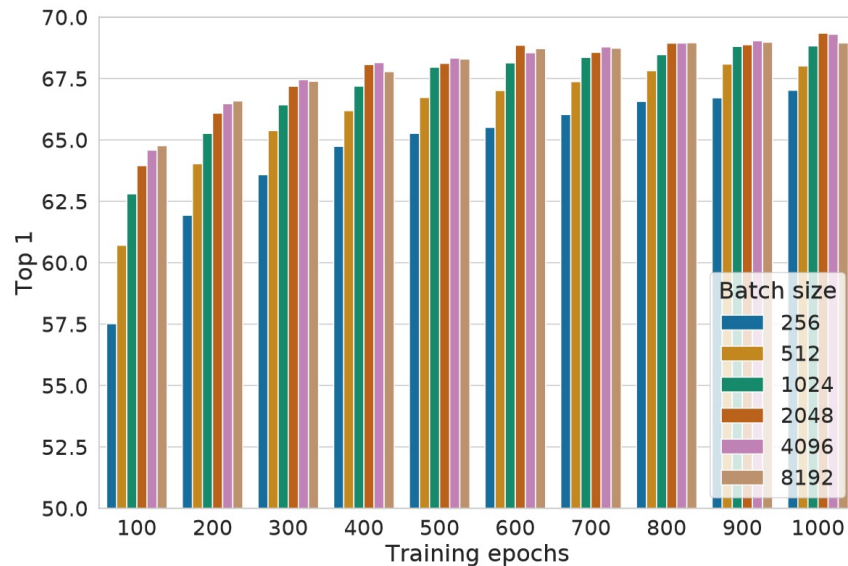
$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)},$$

- $g(\cdot)$ can remove information that may be useful such as color
- Using nonlinear $g(\cdot)$ allows \mathbf{h} to contain more information



What to predict?	Random guess	Representation \mathbf{h}	Representation $g(\mathbf{h})$
Color vs grayscale	80	99.3	97.4
Rotation	25	67.6	25.6
Orig. vs corrupted	50	99.5	59.6
Orig. vs Sobel filtered	50	96.6	56.3

- **SimCLR** [Chen et al., 2020]
 - A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank
 - This paper finds that contrastive learning benefits from ...
3. Large batch sizes and longer training



- **SimCLR** [Chen et al., 2020]
 - A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank
 - SimCLR achieves outstanding performance in various downstream tasks

Fine-grained image classification tasks

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
<i>Linear evaluation:</i>												
SimCLR (ours)	76.9	95.3	80.2	48.4	65.9	60.0	61.2	84.2	78.9	89.2	93.9	95.0
Supervised	75.2	95.7	81.2	56.4	64.9	68.8	63.8	83.8	78.7	92.3	94.1	94.2
<i>Fine-tuned:</i>												
SimCLR (ours)	89.4	98.6	89.0	78.2	68.1	92.1	87.0	86.6	77.8	92.1	94.1	97.6
Supervised	88.7	98.3	88.7	77.8	67.0	91.4	88.0	86.5	78.8	93.2	94.2	98.0
Random init	88.3	96.0	81.9	77.0	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

Semi-supervised learning in ImageNet

Method	Architecture	Label fraction	
		1%	10%
		Top 5	
Supervised baseline	ResNet-50	48.4	80.4
<i>Methods using other label-propagation:</i>			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
<i>Methods using representation learning only:</i>			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(*)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	85.8	92.6

Linear evaluation in ImageNet

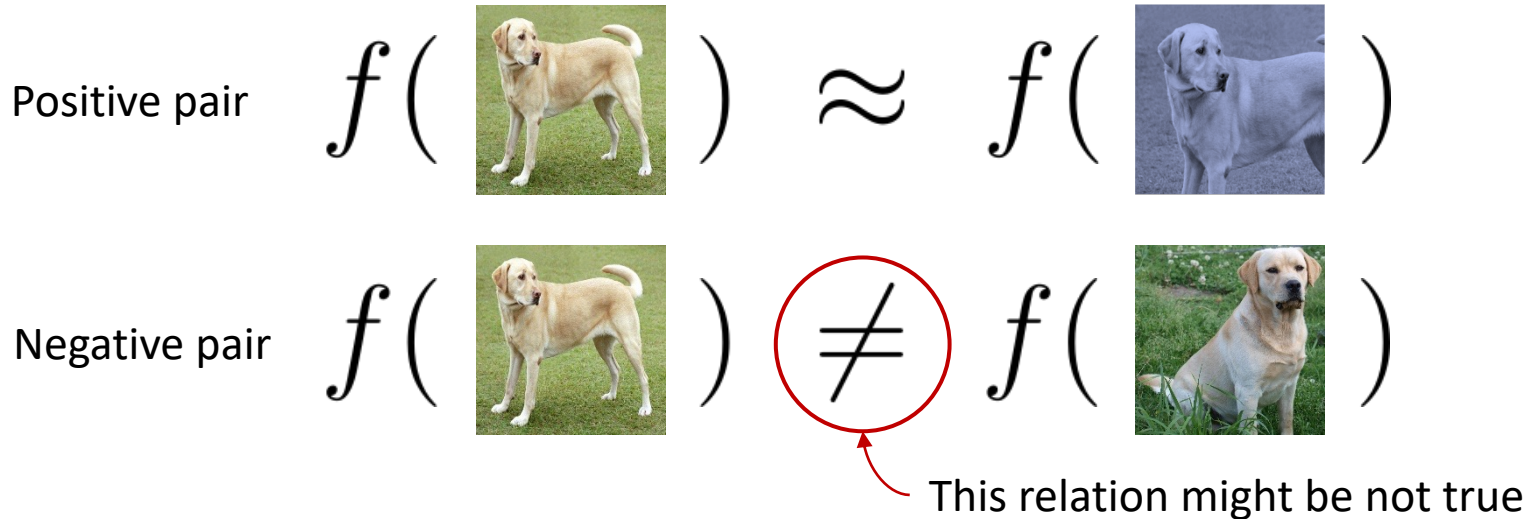
Method	Architecture	Param (M)	Top 1	Top 5
<i>Methods using ResNet-50:</i>				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	69.3	89.0
<i>Methods using other architectures:</i>				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (*)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	76.5	93.2

- **Limitations** in contrastive learning (with negatives)
 - It is sensitive to the number of negative \Rightarrow a large batch size or a queue is required
 - Are all the different instances negative?

Positive pair $f(\text{img1}) \approx f(\text{img2})$

Negative pair $f(\text{img3}) \neq f(\text{img4})$

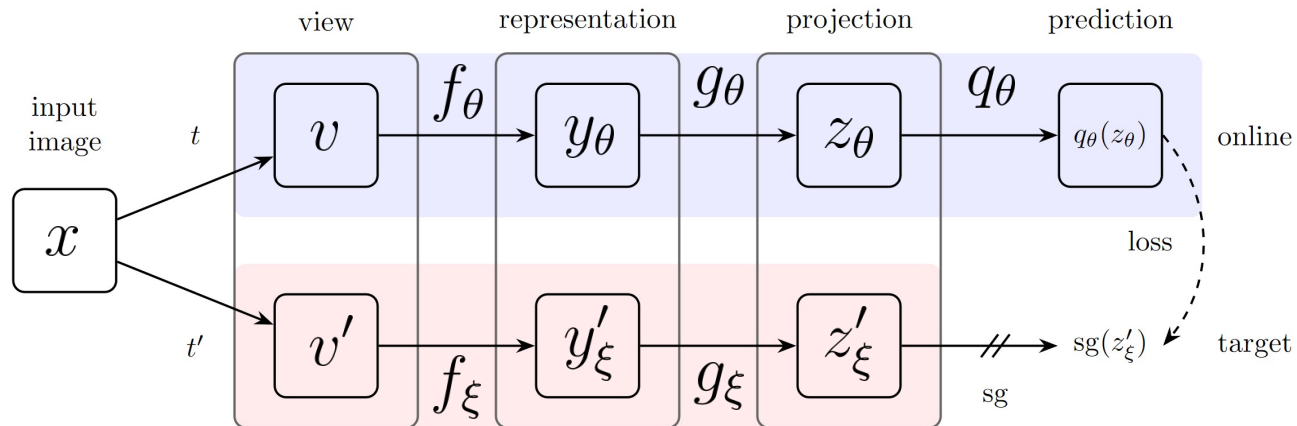
This relation might be not true



- **Q)** can we learn representations without negative samples?
- Simply minimizing $\|f(\text{img1}) - f(\text{img2})\|$ leads to mode collapse, i.e., $\forall x, f(x) = c$
- **Next:** Positive-only approaches

- **Bootstrap Your Own Latent (BYOL)** [Grill et al., 2020]

- **Idea:** directly bootstrap the representations



Objective

$$\mathcal{L}_{\text{BYOL}} = \left\| \frac{q_\theta(z_\theta)}{\|q_\theta(z_\theta)\|} - \frac{z'_\xi}{\|z'_\xi\|} \right\|^2$$

Update

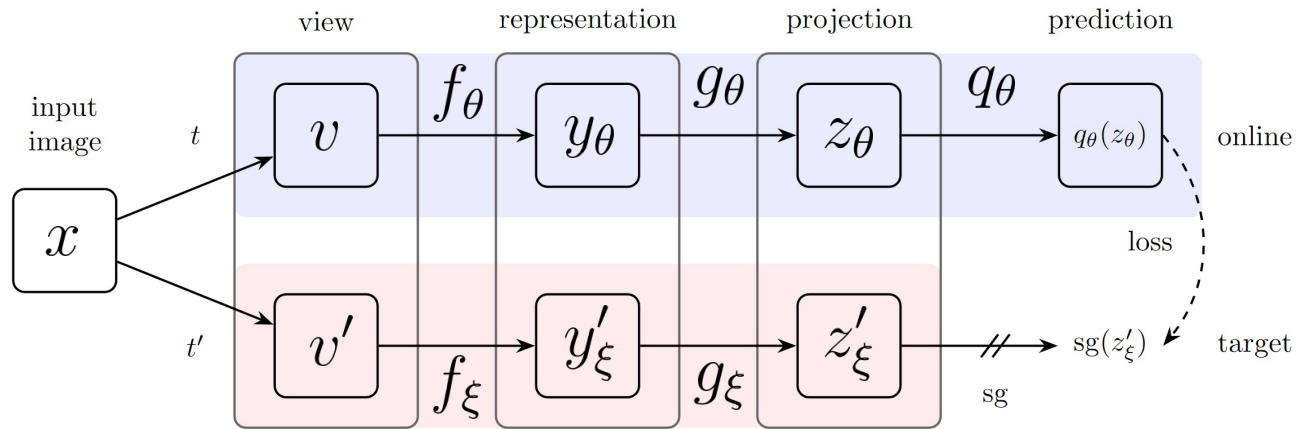
$$\theta \leftarrow \text{optimizer}(\theta, \nabla_\theta \mathcal{L}_{\text{BYOL}})$$

$$\xi \leftarrow \tau \xi + (1 - \tau) \theta$$

- **Key components:** target (momentum) network, predictor, stop-gradient (sg)

• Bootstrap Your Own Latent (BYOL) [Grill et al., 2020]

- **Idea:** directly bootstrap the representations



Objective

$$\mathcal{L}_{\text{BYOL}} = \left\| \frac{q_\theta(z_\theta)}{\|q_\theta(z_\theta)\|} - \frac{z'_\xi}{\|z'_\xi\|} \right\|^2$$

Update

$$\theta \leftarrow \text{optimizer}(\theta, \nabla_\theta \mathcal{L}_{\text{BYOL}})$$

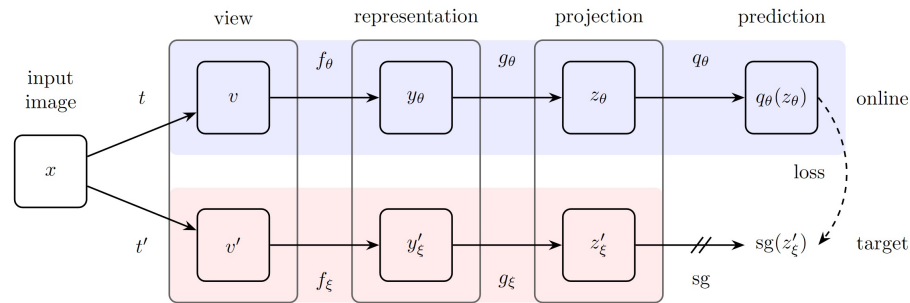
$$\xi \leftarrow \tau \xi + (1 - \tau) \theta$$

• **Q)** How does BYOL avoid the undesired collapsed solutions?

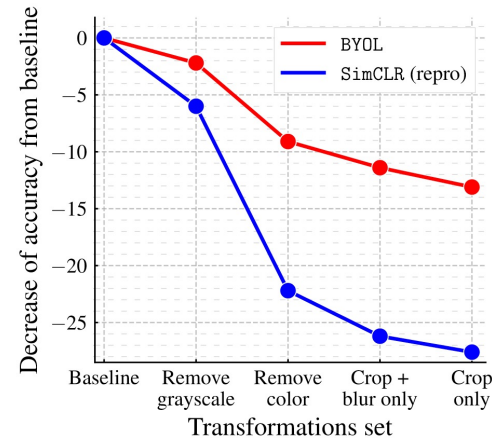
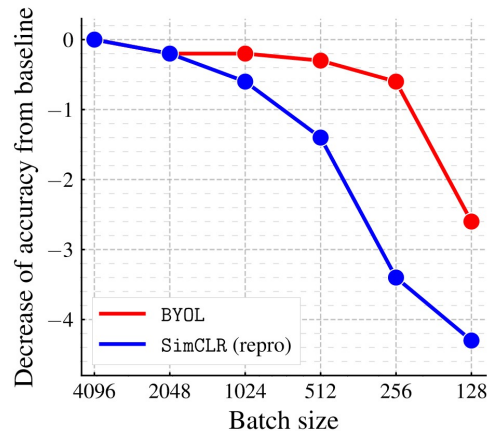
- ξ is not updated in the direction of $\nabla_\xi \mathcal{L}_{\text{BYOL}}$
- When the predictor is optimal, i.e., $q^*(z_\theta) = \mathbb{E}[z'_\xi | z_\theta]$, $\mathcal{L}_{\text{BYOL}} = \mathbb{E}[\sum_i \text{Var}(z'_{\xi,i} | z_\theta)]$ z'_ξ 's i-th feature \searrow
- For any constant c , $\text{Var}(z'_{\xi,i} | z_\theta) \leq \text{Var}(z'_{\xi,i} | c) \Rightarrow$ constant equilibria is unstable

- **Bootstrap Your Own Latent (BYOL)** [Grill et al., 2020]

- **Idea:** directly bootstrap the representations

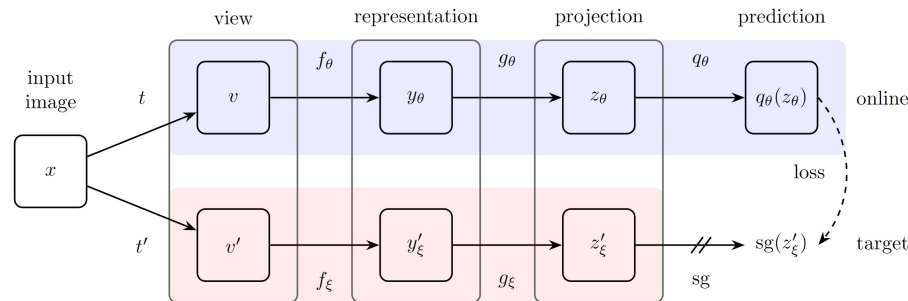


- BYOL is **more robust** to the choice of **batch sizes** and **augmentations**

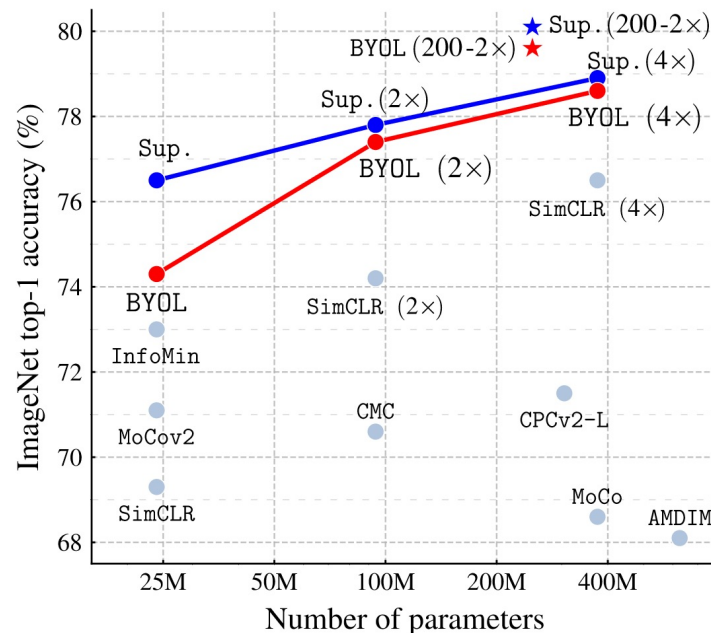


- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]

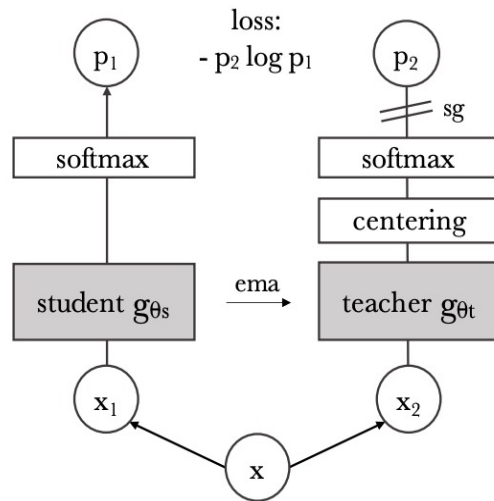
- **Idea:** directly bootstrap the representations



- BYOL is **more robust** to the choice of **batch sizes** and **augmentations**
- BYOL achieves 74.3% linear evaluation accuracy; supervised learning does 76.5%



- **DINO** [Caron et al., 2021]
 - **Idea**: representation learning via self knowledge-distillation



Objective

$$\mathcal{L}_{DINO} = H(P_t(x), P_s(x))$$

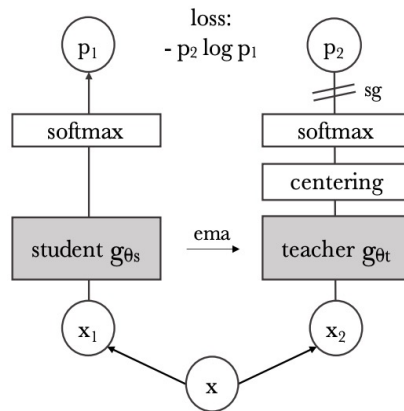
Update

$$\begin{aligned}\theta_s &\leftarrow \text{optimizer}(\theta_s, \nabla_{\theta_s} \mathcal{L}_{DINO}) \\ \theta_t &\leftarrow \lambda \theta_t + (1 - \lambda) \theta_s\end{aligned}$$

- **Key components:**
 - (self) knowledge-distillation
 - Distill the teacher (EMA version of a student) knowledge to the student
 - multi-crop: a strategy to generate positive views
 - centering and sharpening: a strategy to avoid collapse

- **DINO** [Caron et al., 2021]

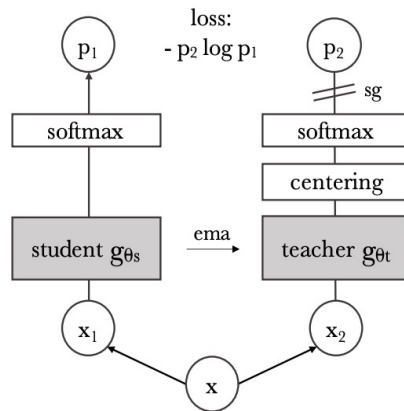
- **Idea**: representation learning via self knowledge-distillation



- DINO constructs a set of views V via **multi-crop** strategy:
 - (1) global views: x_1^g, x_2^g
 - (2) local views with smaller resolution
- All crops are passed through the student; only the global views are passed through the teacher: “**local-to-global**” correspondences
 - Therefore, the loss is written as:

$$\min_{\theta_s} \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x'))$$

- **DINO** [Caron et al., 2021]
 - **Idea**: representation learning via self knowledge-distillation



- DINO **avoids the collapse** via **centering** and **sharpening**
 - Centering: adding a bias term c to the teacher
$$g_t(x) \leftarrow g_t(x) + c$$
 - The center c is updated with an exponential moving average

$$c \leftarrow mc + (1 - m) \frac{1}{B} \sum_{i=1}^B g_{\theta_t}(x_i)$$

- Sharpening: using a low value for the temperature τ_t in the teacher softmax normalization

• DINO [Caron et al., 2021]

- DINO outperforms previous contrastive methods in classification tasks
- Self-supervised ViT features contain explicit information about the semantic segmentation of an image

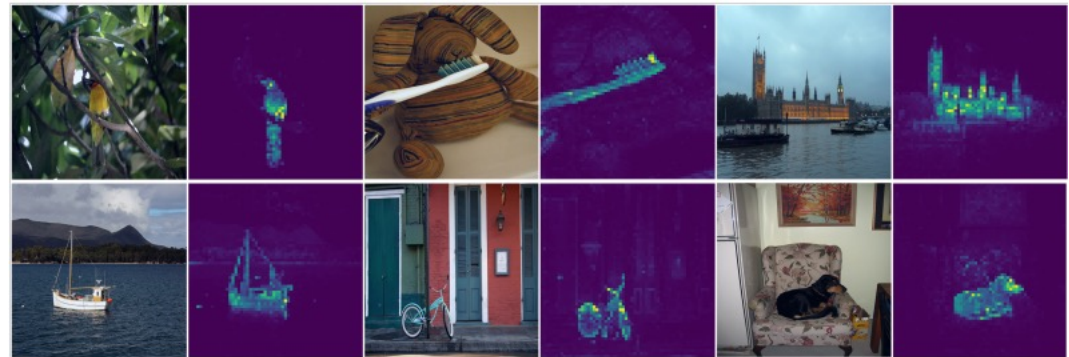
Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [15]	RN50	23	1237	71.1	61.9
InfoMin [67]	RN50	23	1237	73.0	65.3
BarlowT [81]	RN50	23	1237	73.2	66.0
OBoW [27]	RN50	23	1237	73.8	61.9
BYOL [30]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2	67.1
SwAV [10]	RN50	23	1237	75.3	65.7
DINO	RN50	23	1237	75.3	67.5

Supervised	ViT-S	21	1007	79.8	79.8
BYOL* [30]	ViT-S	21	1007	71.4	66.6
MoCov2* [15]	ViT-S	21	1007	72.7	64.4
SwAV* [10]	ViT-S	21	1007	73.5	66.3
DINO	ViT-S	21	1007	77.0	74.5

Comparison across architectures

SCLR [12]	RN50w4	375	117	76.8	69.3
SwAV [10]	RN50w2	93	384	77.3	67.3
BYOL [30]	RN50w2	93	384	77.4	—
DINO	ViT-B/16	85	312	78.2	76.1
SwAV [10]	RN50w5	586	76	78.5	67.1
BYOL [30]	RN50w4	375	117	78.6	—
BYOL [30]	RN200w2	250	123	79.6	73.9
DINO	ViT-S/8	21	180	79.7	78.3
SCLRv2 [13]	RN152w3+SK	794	46	79.8	73.1
DINO	ViT-B/8	85	63	80.1	77.4

Top-1 accuracy for linear and k-NN evaluations
on the validation set of ImageNet



Self-attention map on [CLS] of self-supervised ViT

Method	Data	Arch.	$(\mathcal{J} \& \mathcal{F})_m$	\mathcal{J}_m	\mathcal{F}_m
<i>Supervised</i>					
ImageNet	INet	ViT-S/8	66.0	63.9	68.1
STM [48]	I/D/Y	RN50	81.8	79.2	84.3
<i>Self-supervised</i>					
CT [71]	VLOG	RN50	48.7	46.4	50.0
MAST [40]	YT-VOS	RN18	65.5	63.3	67.6
STC [37]	Kinetics	RN18	67.6	64.8	70.2
DINO	INet	ViT-S/16	61.8	60.2	63.4
DINO	INet	ViT-B/16	62.3	60.7	63.9
DINO	INet	ViT-S/8	69.9	66.6	73.1
DINO	INet	ViT-B/8	71.4	67.9	74.9

Video instance segmentation on top of
self-supervised feature

- **Choices for Positive Samples**

- We discussed **how** to make positive samples invariant
- By the way, **what** are the positive samples?
- **Similar data (e.g., by clustering)**
 - Discussed before (e.g., DeepCluster)
- **Same data with different augmentation**
 - Discussed image domain before (e.g., SimCLR)
 - How about **other domains** (e.g., language, graph, or domain-agnostic)?
- **Same data with different modality**
 - Different **channel** (e.g., multi-view) or **domain** (e.g., vision & language)
- **Utilize sequential structure**
 - (a) Predict **future state** from past states (positive = true future)
 - (b) Use states from **same sequence** as positives (positive = same sequence)

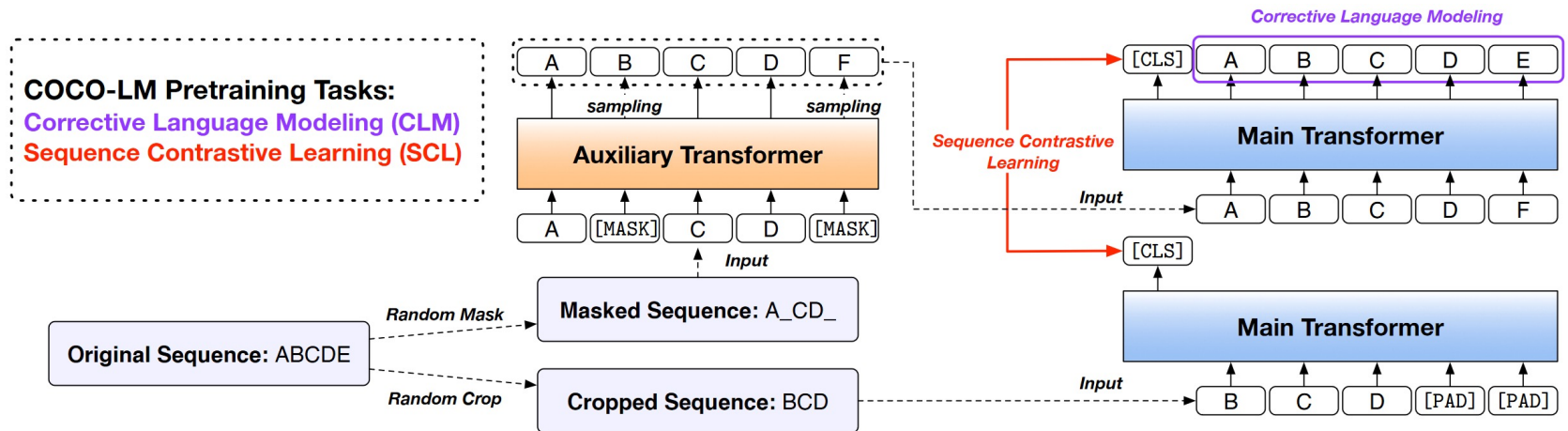
- **Choices for Positive Samples**

- We discussed **how** to make positive samples invariant
- By the way, **what** are the positive samples?
- **Similar data (e.g., by clustering)**
 - Discussed before (e.g., DeepCluster)
- **Same data with different augmentation**
 - Discussed image domain before (e.g., SimCLR)
 - How about **other domains** (e.g., language, graph, or domain-agnostic)?
- **Same data with different modality**
 - Different channel (e.g., multi-view) or domain (e.g., vision & language)
- **Utilize sequential structure**
 - (a) Predict future state from past states (positive = true future)
 - (b) Use states from same sequence as positives (positive = same sequence)

- **COCO-LM** [Meng et al., 2021]

- **Idea:**

- **Corrective Language Modeling:** Recover original tokens from corrupted ones
- **Sequence Contrastive Learning** between corrupted and augmented sentences



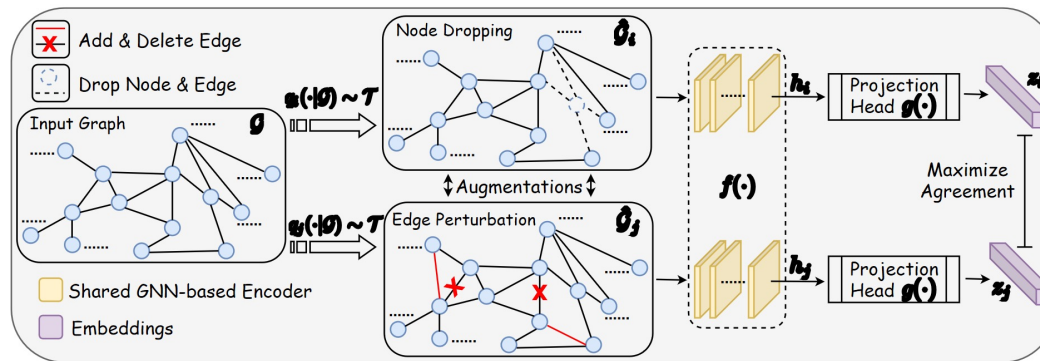
- Both **CLM** and **SCL** improves **Baseline**

- Improvements are observed on different tasks, e.g., CLM: CoLA, SCL: RTE (CoLA: grammatical validity of one sentence, RTE: relation of two sentences)

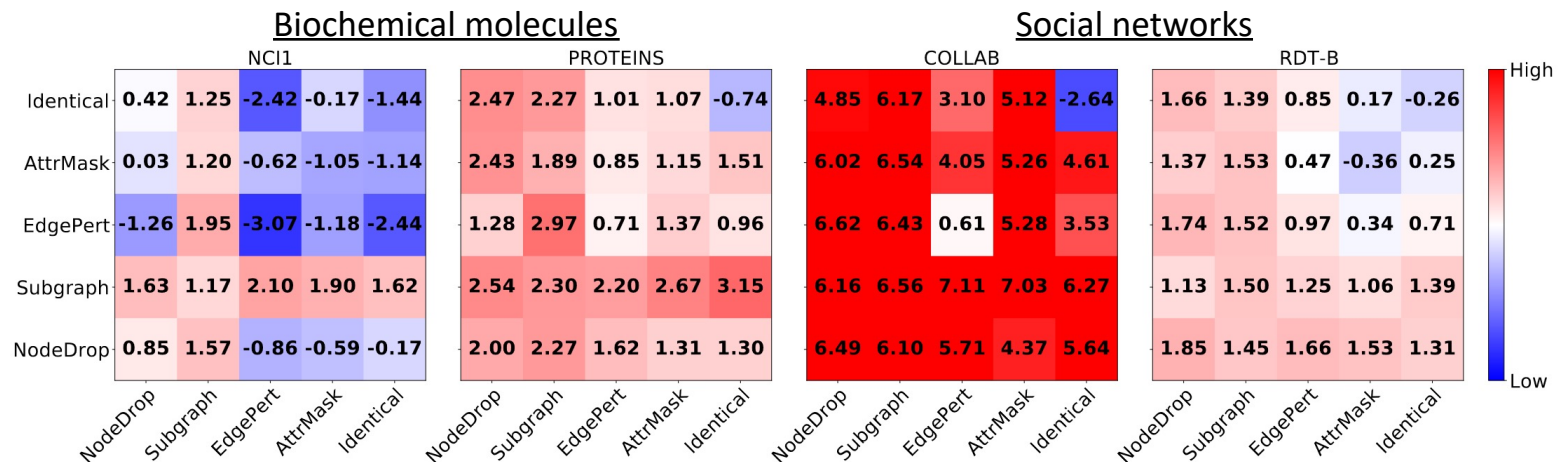
Group	Method	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	RTE	MRPC	STS-B	AVG
Baseline	RoBERTa (Ours)	85.61/85.51	91.34	91.80	93.86	58.64	69.03	87.50	86.53	83.03
	ELECTRA (Ours)	86.92/86.72	91.86	92.56	93.64	66.50	75.28	88.46	88.04	85.39
Original	COCO-LM Base	88.67/88.35	92.02	93.00	94.08	65.41	85.42	91.51	88.61	87.05
Pretraining Task	CLM Only	88.64/88.40	92.03	93.14	93.86	66.95	80.90	89.90	88.45	86.72
	SCL Only	88.62/88.14	92.14	93.45	93.86	64.70	82.57	90.38	89.35	86.86

• GraphCL [You et al., 2020]

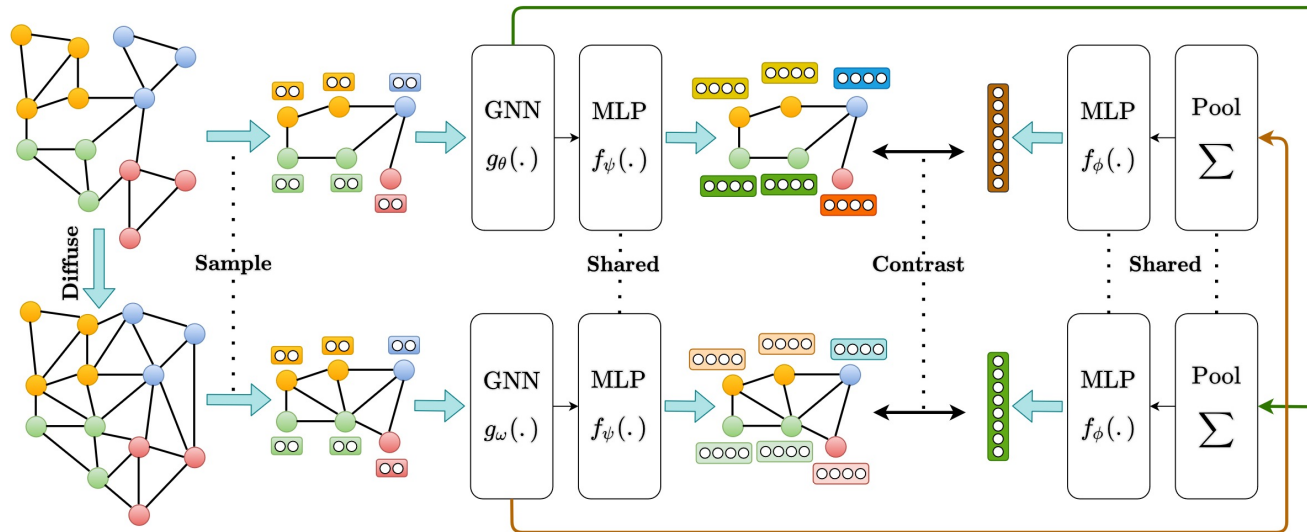
- This paper studies contrastive learning with diverse **graph augmentations**
 - Node dropping, edge perturbation, attribute masking, subgraph sampling
 - GraphCL's architecture and objective are almost the same as SimCLR



- The choice of graph augmentations is critical depending on downstream tasks



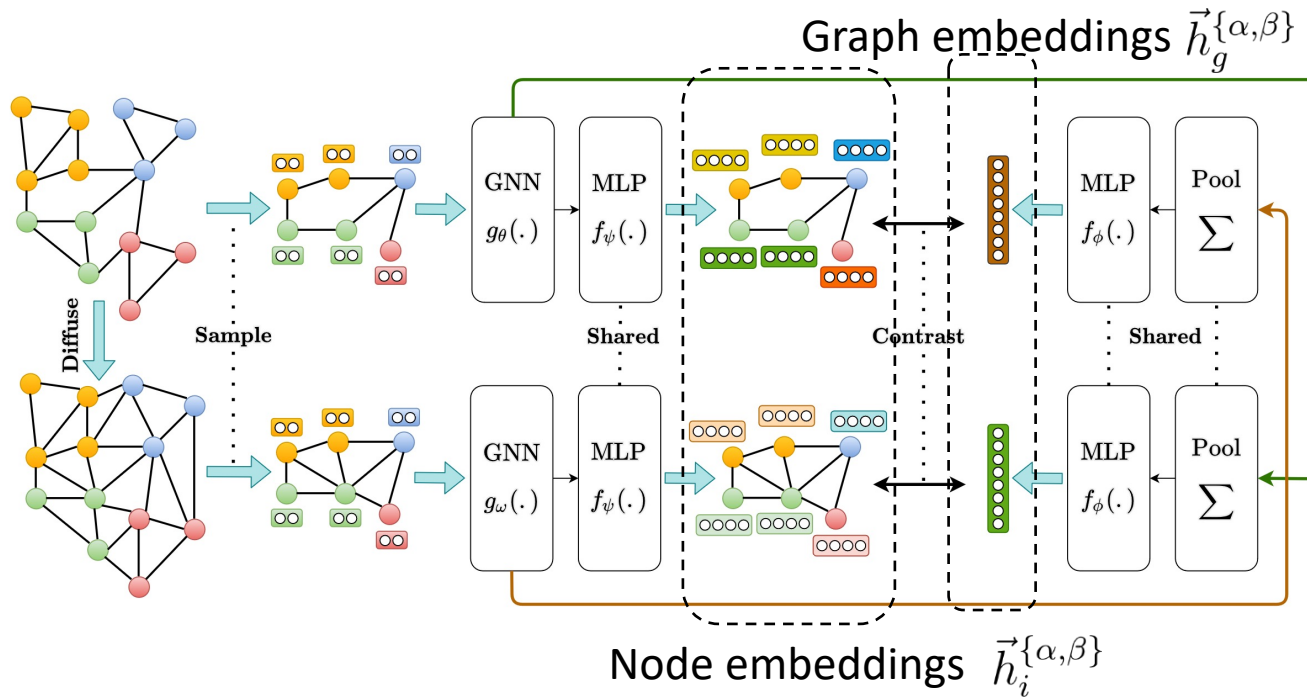
- **Multi-View Contrastive Learning on Graphs** [Hassani & Khasahmadi, 2020]
 - **Idea:** Use a **graph diffusion** as the second view



Objective

$$\max_{\theta, \omega, \phi, \psi} \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \left[\frac{1}{|g|} \sum_{i=1}^{|g|} \left[\text{MI} \left(\vec{h}_i^\alpha, \vec{h}_g^\beta \right) + \text{MI} \left(\vec{h}_i^\beta, \vec{h}_g^\alpha \right) \right] \right]$$

- **Multi-View Contrastive Learning on Graphs** [Hassani & Khasahmadi, 2020]
 - **Idea:** Use a **graph diffusion** as the second view



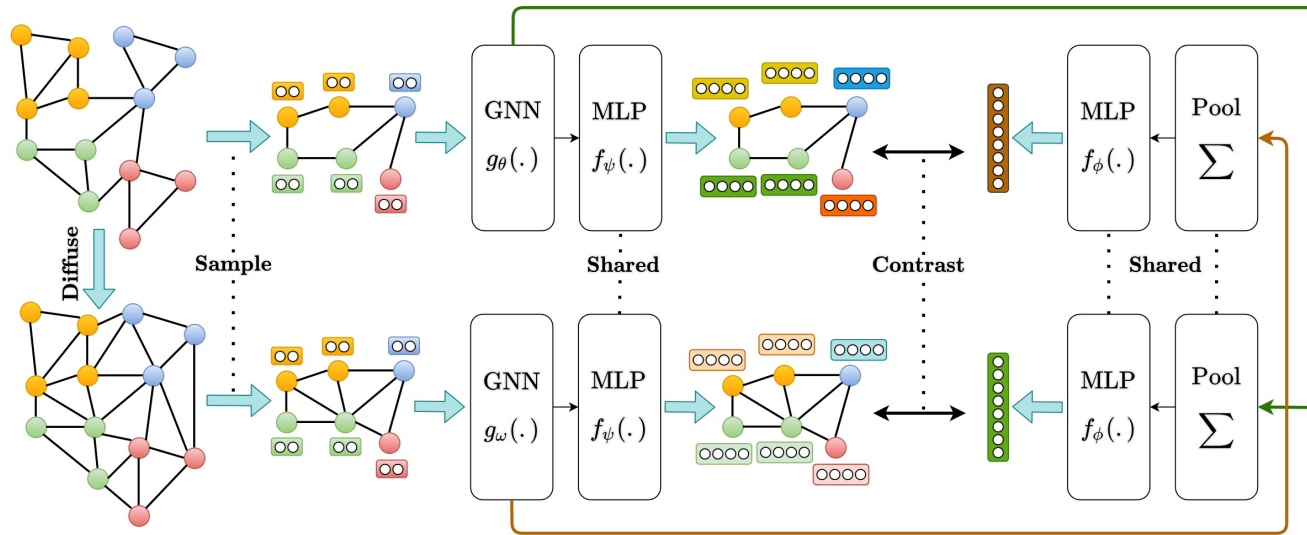
Objective

$$\max_{\theta, \omega, \phi, \psi} \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \left[\frac{1}{|g|} \sum_{i=1}^{|g|} \left[\text{MI} \left(\vec{h}_i^\alpha, \vec{h}_g^\beta \right) + \text{MI} \left(\vec{h}_i^\beta, \vec{h}_g^\alpha \right) \right] \right]$$

Maximize MI by contrastive learning

- **Multi-View Contrastive Learning on Graphs** [Hassani & Khasahmadi, 2020]

- **Idea:** Use a graph diffusion as the second view



- For graph diffusion, this paper uses Personalized PageRank and Heat Kernel
- Unlike visual representation learning, increasing # of views does not help

#VIEWS	CORA	CITeseer	PUBMED
2	86.8 \pm 0.5	73.3 \pm 0.5	80.1 \pm 0.7
3	85.3 \pm 0.5	71.2 \pm 0.7	79.9 \pm 0.6

- Different diffusion matrices may have similar information about the graph

- **i-Mix** [Lee et al., 2021]
 - **Idea:** Introduce **virtual labels** in a batch and apply MixUp or CutMix
 - It is a **domain-agnostic regularization** strategy for contrastive learning
 - **General form of i-Mix**
 - Let $\mathcal{B} = \{(x_i, \tilde{x}_i)\}_{i=1}^N$ be a batch of positive data pairs for contrastive learning
 - For each anchor x_i , \tilde{x}_i is a positive sample, $\tilde{x}_{i \neq j}$ are negative samples
 - Then, i-Mix defines the **one-hot virtual label** $v_i \in \{0,1\}^N$ of x_i and \tilde{x}_i
 - $v_{i,i} = 1$ and $v_{i,j \neq i} = 0$
 - With virtual labels, we can re-write a general contrastive loss: $\ell(x_i, v_i)$
 - Then, i-Mix loss is defined as:
$$\ell^{i\text{-Mix}}((x_i, v_i), (x_j, v_j); \mathcal{B}, \lambda) = \ell(\text{Mix}(x_i, x_j; \lambda), \lambda v_i + (1 - \lambda)v_j; \mathcal{B})$$
 - i-Mix uses MixUp and CutMix functions as a Mix operator
 - i-Mix can be applied for different contrastive objectives, such as SimCLR, MoCo and BYOL

- **i-Mix** [Lee et al., 2021]
 - **Idea:** Introduce **virtual labels** in a batch and apply MixUp or CutMix
 - It is a **domain-agnostic regularization** strategy for contrastive learning
- i-Mix consistently improves the classification accuracy on different domains

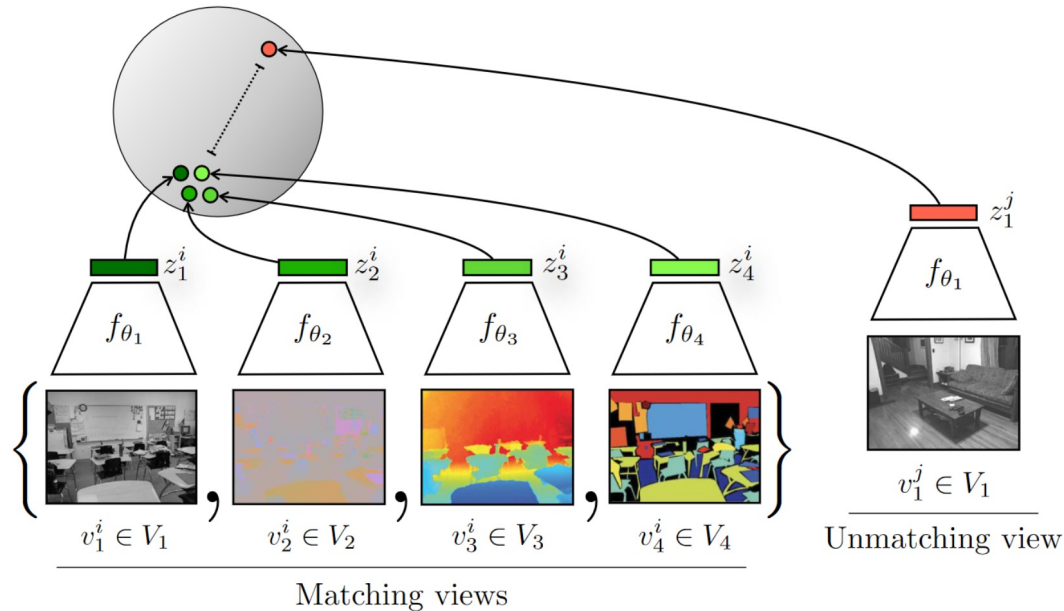
Domain	Dataset	N-pair	+ <i>i</i> -Mix	MoCo v2	+ <i>i</i> -Mix	BYOL	+ <i>i</i> -Mix
Image	CIFAR-10	93.3 \pm 0.1	95.6 \pm 0.2	93.5 \pm 0.2	96.1 \pm 0.1	94.2 \pm 0.2	96.3 \pm 0.2
	CIFAR-100	70.8 \pm 0.4	75.8 \pm 0.3	71.6 \pm 0.1	78.1 \pm 0.3	72.7 \pm 0.4	78.6 \pm 0.2
Speech	Commands	94.9 \pm 0.1	98.3 \pm 0.1	96.3 \pm 0.1	98.4 \pm 0.0	94.8 \pm 0.2	98.3 \pm 0.0
Tabular	CovType	68.5 \pm 0.3	72.1 \pm 0.2	70.5 \pm 0.2	73.1 \pm 0.1	72.1 \pm 0.2	74.1 \pm 0.2

Table 1: Comparison of contrastive representation learning methods and *i*-Mix in different domains.

- **Choices for Positive Samples**

- We discussed **how** to make positive samples invariant
- By the way, **what** are the positive samples?
- **Similar data (e.g., by clustering)**
 - Discussed before (e.g., DeepCluster)
- **Same data with different augmentation**
 - Discussed image domain before (e.g., SimCLR)
 - How about other domains (e.g., language, graph, or domain-agnostic)?
- **Same data with different modality**
 - Different **channel** (e.g., multi-view) or **domain** (e.g., vision & language)
- **Utilize sequential structure**
 - (a) Predict future state from past states (positive = true future)
 - (b) Use states from same sequence as positives (positive = same sequence)

- **Contrastive Multiview Coding (CMC)** [Tian et al., 2019]
 - **Idea:** Use **multiple views** of the same instance as positive samples

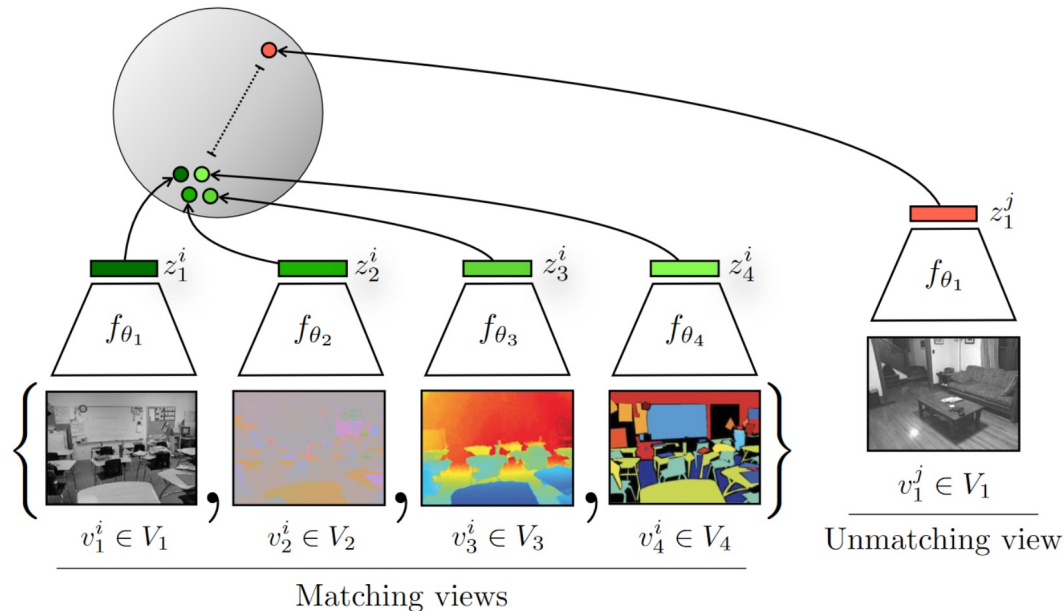


$$\mathcal{L}_{\text{contrast}}^{V_1, V_2} = -\mathbb{E}_{\{v_1^1, v_2^1, \dots, v_2^{k+1}\}} \left[\log \frac{h_{\theta}(\{v_1^1, v_2^1\})}{\sum_{j=1}^{k+1} h_{\theta}(\{v_1^1, v_2^j\})} \right]$$

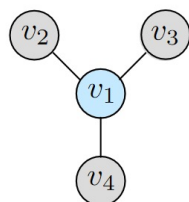
where $h_{\theta}(\{v_1, v_2\}) = \exp \left(\frac{f_{\theta_1}(v_1)^{\top} f_{\theta_2}(v_2)}{\|f_{\theta_1}(v_1)\| \|f_{\theta_2}(v_2)\|} \frac{1}{\tau} \right)$

Neural network

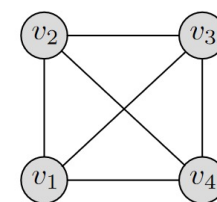
- **Contrastive Multiview Coding (CMC)** [Tian et al., 2019]
 - **Idea:** Use **multiple views** of the same instance as positive samples



- By minimizing $\mathcal{L}(V_1, V_2) = \mathcal{L}_{\text{contrast}}^{V_1, V_2} + \mathcal{L}_{\text{contrast}}^{V_2, V_1}$, $f_{\theta_1}(\cdot), f_{\theta_2}(\cdot)$ learns to extract common information in two different views
- For $M > 2$ views, use $\mathcal{L} = \sum_{j=1}^M \mathcal{L}(V_1, V_j)$ or $\mathcal{L} = \sum_{1 \leq i < j \leq M} \mathcal{L}(V_i, V_j)$

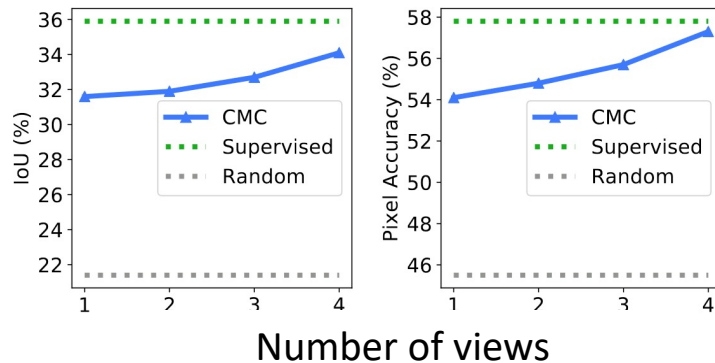


Core-view



Full-graph

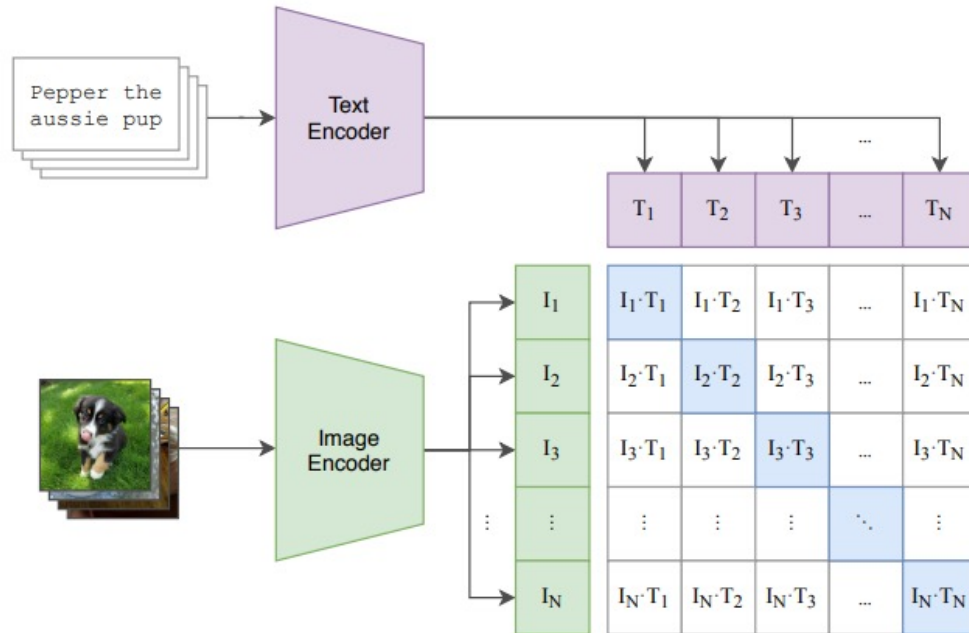
- **Contrastive Multiview Coding (CMC)** [Tian et al., 2019]
 - **Idea:** Use **multiple views** of the same instance as positive samples
 - Using more views is effective
 - NYU-Depth-V2 dataset have 4 views: (1) luminance (L), (2) chrominance (ab), (3) depth, (4) surface normal
 - **Task:** semantic segmentation



Core-view vs Full-graph

	Pixel Accuracy (%)	mIoU (%)
Random	45.5	21.4
CMC (core-view)	57.1	34.1
CMC (full-graph)	57.0	34.4
Supervised	57.8	35.9

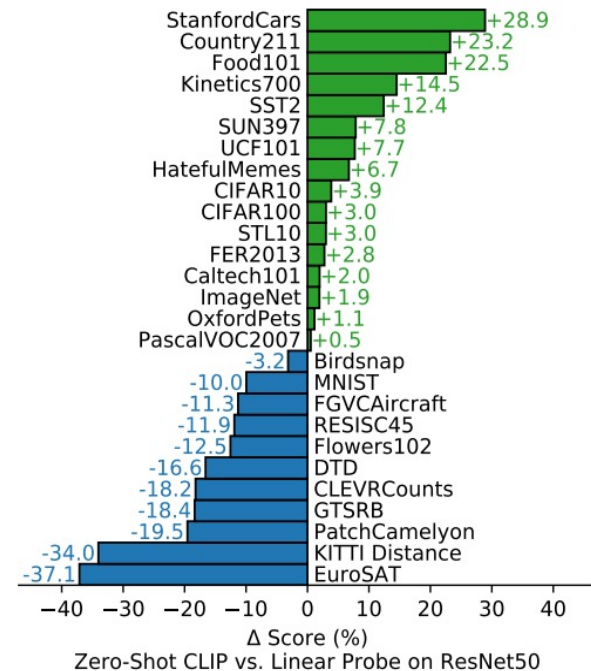
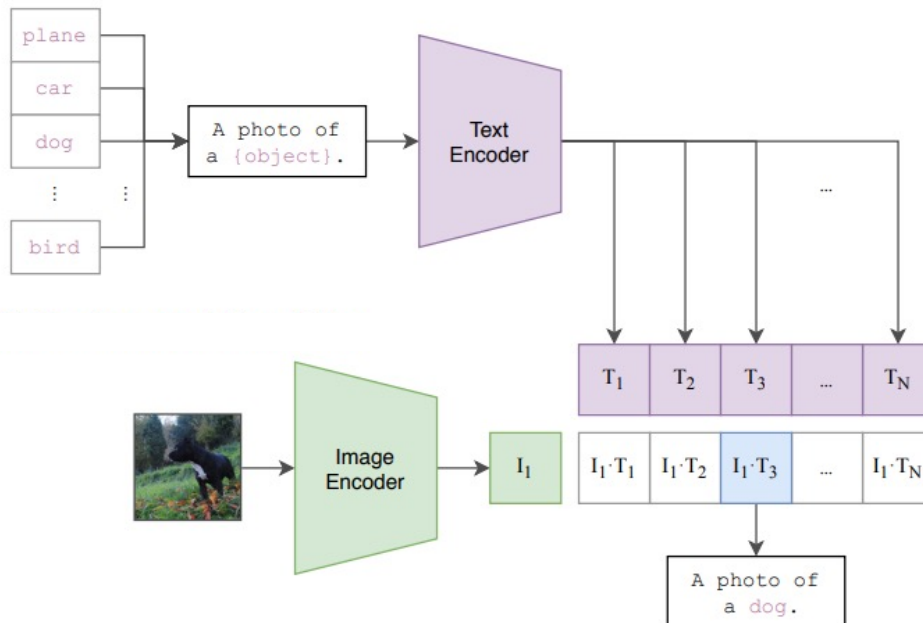
- **CLIP** [Radford et al., 2021]
 - **Idea:** Use **text description** of the given image as positive samples
 - Negative pair: an image and text that describes different image
 - CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of (image, text)



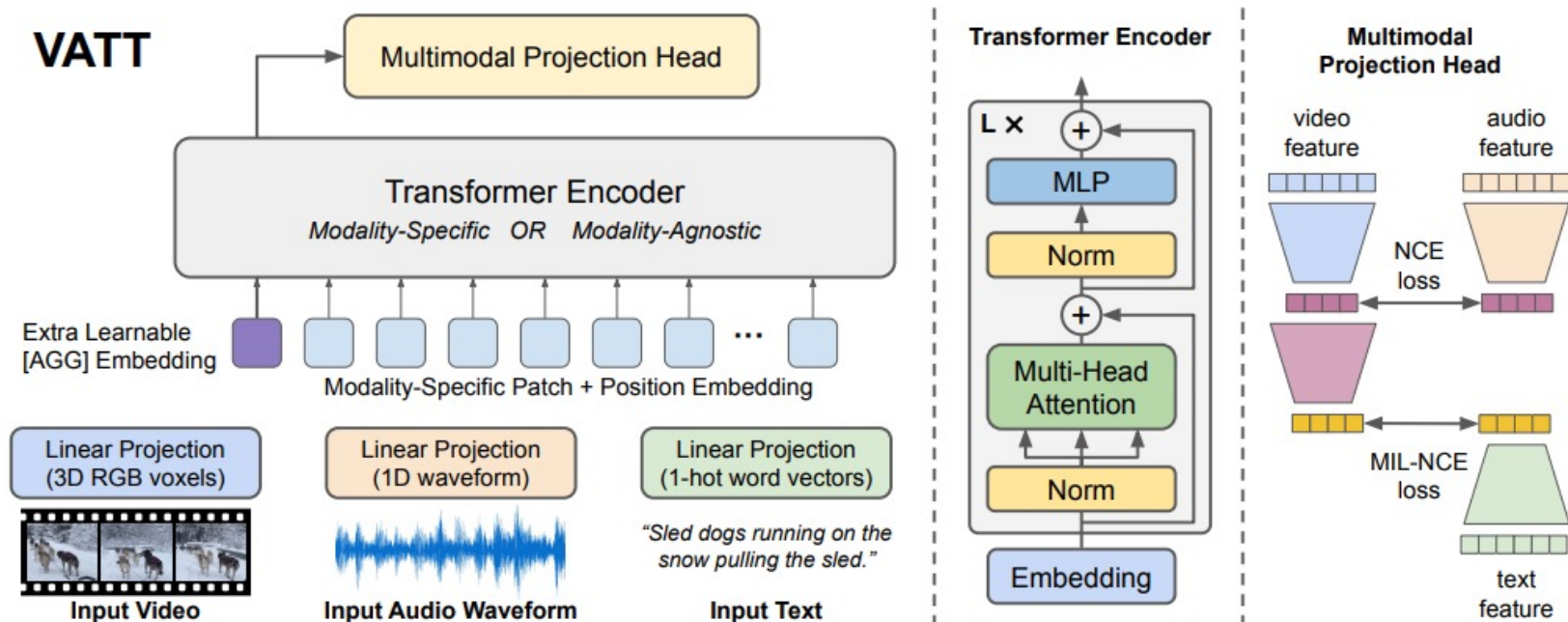
- 400 million (image, text) pairs are collected from the internet

SSL via Invariance – Different Modality (HW)

- **CLIP** [Radford et al., 2021]
 - **Idea:** Use **text description** of the given image as positive samples
 - After the pre-train, the model can be **zero-shot** transferred to downstream task:
 - 1) Embed the descriptions of target classes with the text encoder
 - 2) Select the maximally similar text embedding to the given image
 - A zero-shot CLIP classifier shows a competitive performance with a fully supervised linear classifier fitted on ResNet-50 features



- **VATT** [Akbari et al., 2021]
 - VATT matches **video**, **audio**, and description **text** via contrastive learning
 - Similar to CLIP, but uses Transformer encoder to apply on various data modalities



- **VATT** [Akbari et al., 2021]
 - VATT matches **video**, **audio**, and description **text** via contrastive learning
 - Similar to CLIP, but uses Transformer encoder to apply on various data modalities
- VATT is effective on various downstream tasks, e.g., video classification, audio classification, image classification, and text-to-video retrieval

METHOD	<u>Kinetics-400</u>		<u>Kinetics-600</u>		<u>Moments in Time</u>		TFLOPs
	TOP-1	TOP-5	TOP-1	TOP-5	TOP-1	TOP-5	
I3D [13]	71.1	89.3	71.9	90.1	29.5	56.1	-
R(2+1)D [26]	72.0	90.0	-	-	-	-	17.5
bLVNet [27]	73.5	91.2	-	-	31.4	59.3	0.84
S3D-G [96]	74.7	93.4	-	-	-	-	-
Oct-I3D+NL [20]	75.7	-	76.0	-	-	-	0.84
D3D [83]	75.9	-	77.9	-	-	-	-
I3D+NL [93]	77.7	93.3	-	-	-	-	10.8
ip-CSN-152 [87]	77.8	92.8	-	-	-	-	3.3
AttentionNAS [92]	-	-	79.8	94.4	32.5	60.3	1.0
AssembleNet-101 [77]	-	-	-	-	34.3	62.7	-
MoViNet-A5 [47]	78.2	-	82.7	-	39.1	-	0.29
LGD-3D-101 [69]	79.4	94.4	81.5	95.6	-	-	-
SlowFast-R101-NL [30]	79.8	93.9	81.8	95.1	-	-	7.0
X3D-XL [29]	79.1	93.9	81.9	95.5	-	-	1.5
X3D-XXL [29]	80.4	94.6	-	-	-	-	5.8
TimeSFormer-L [9]	80.7	94.7	82.2	95.6	-	-	7.14
VATT-Base	79.6	94.9	80.5	95.5	38.7	67.5	9.09
VATT-Medium	81.1	95.6	82.4	96.1	39.5	68.2	15.02
VATT-Large	82.1	95.5	83.6	96.6	41.1	67.7	29.80
VATT-MA-Medium	79.9	94.9	80.8	95.5	37.8	65.9	15.02

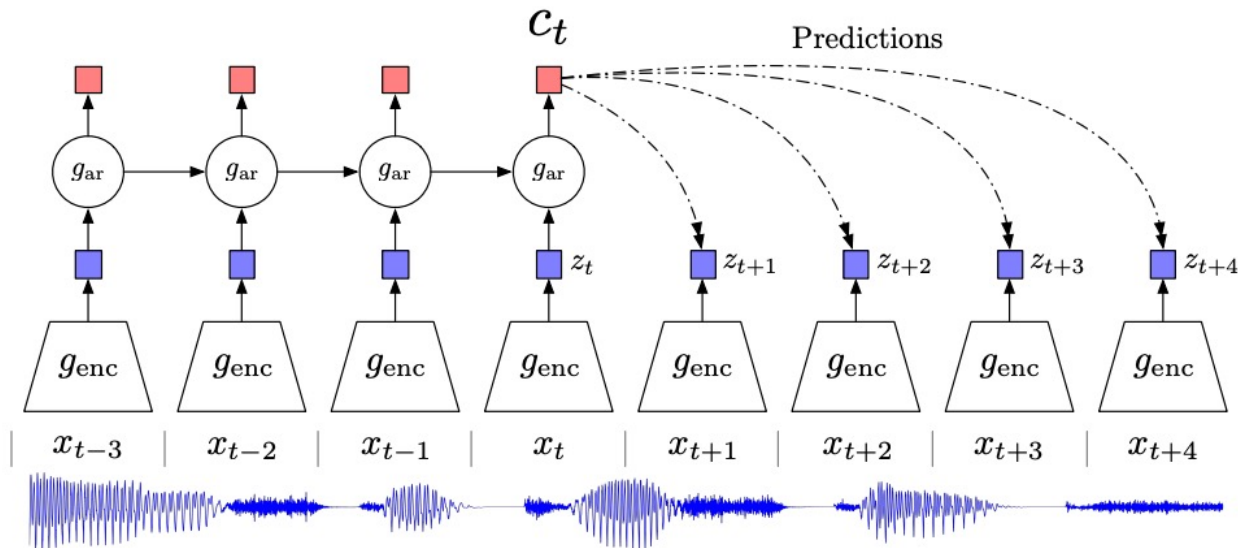
- **Choices for Positive Samples**

- We discussed **how** to make positive samples invariant
- By the way, **what** are the positive samples?
- **Similar data (e.g., by clustering)**
 - Discussed before (e.g., DeepCluster)
- **Same data with different augmentation**
 - Discussed image domain before (e.g., SimCLR)
 - How about other domains (e.g., language, graph, or domain-agnostic)?
- **Same data with different modality**
 - Different channel (e.g., multi-view) or domain (e.g., vision & language)
- **Utilize sequential structure**
 - (a) Predict **future state** from past states (positive = true future)
 - (b) Use states from **same sequence** as positives (positive = same sequence)
 - (a) Is also related to SSL via generation (sequential prediction)

- **Contrastive Predictive Coding (CPC)** [Oord et al., 2018]

- **Idea:** Predicting future information with discarding low-level information

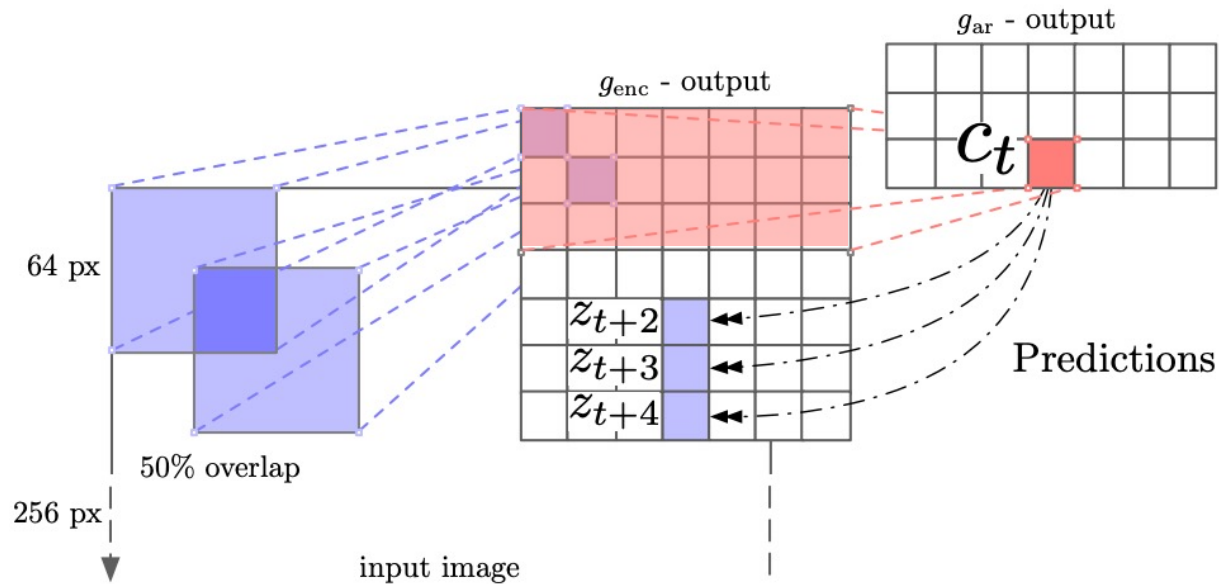
- x_t : data at time t
- $z_t = g_{\text{enc}}(x_t)$: high-level latent representation of x_t
- $c_t = g_{\text{ar}}(x_1, x_2, \dots, x_t)$: context latent representation summarizing all $z_{\leq t}$



- **Contrastive Predictive Coding (CPC)** [Oord et al., 2018]

- **Idea:** Predicting future information with discarding low-level information

- x_t : data at time t
- $z_t = g_{\text{enc}}(x_t)$: high-level latent representation of x_t
- $c_t = g_{\text{ar}}(x_1, x_2, \dots, x_t)$: context latent representation summarizing all $z_{\leq t}$



- **Contrastive Predictive Coding (CPC)** [Oord et al., 2018]

- **Idea:** Predicting future information with discarding low-level information
- How to maximize mutual information between x_{t+k} and c_t ?
 - Randomly choose one positive sample x_{t+k} and N-1 negative samples $\{x\}$
 - Minimize the following **NCE**-based loss:

$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_x f_k(x, c_t)} \right]$$

where $f_k(x, c) = \exp(z^\top W_k c)$

- $I(x_{t+k}, c_t) \geq \log(N) - \mathcal{L}_N$ and it becomes tighter as N becomes larger

SSL via Invariance – Sequential Structure

- **Contrastive Predictive Coding (CPC)** [Oord et al., 2018]
 - This framework is working on Audio, Vision, NLP and RL

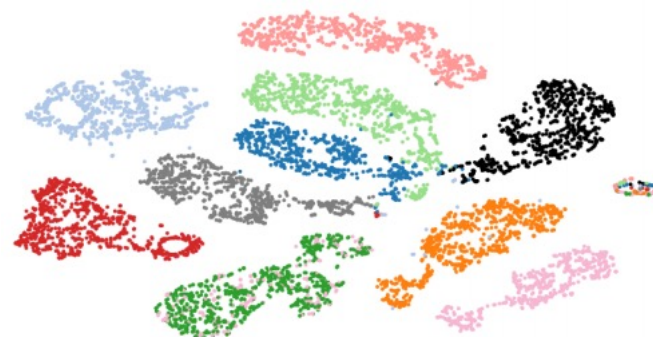
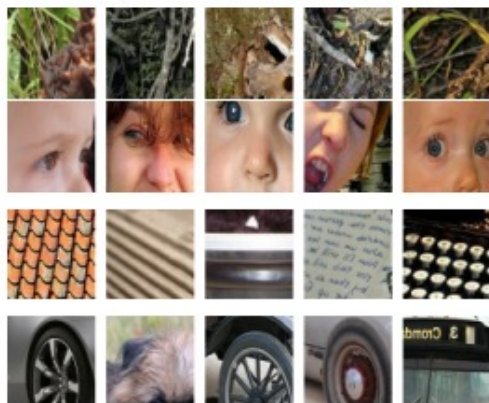
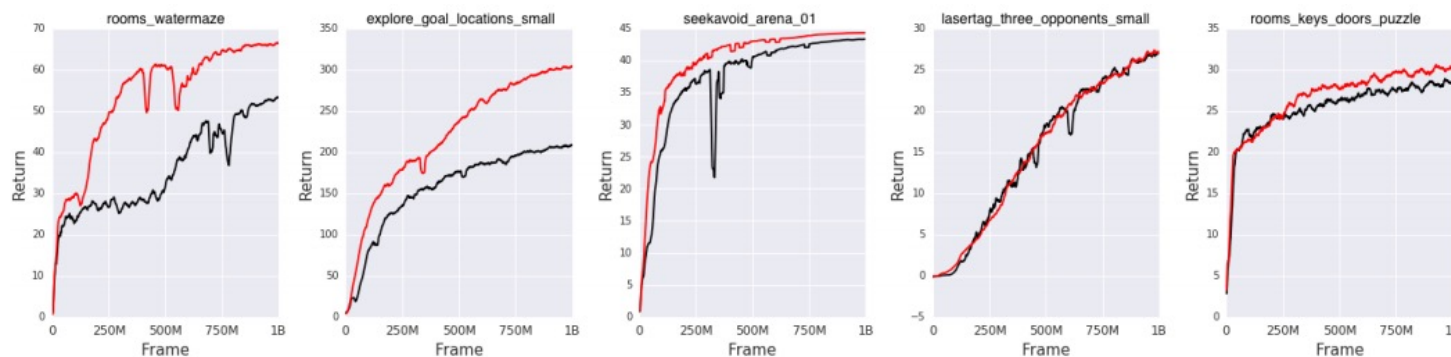


Image patches that activate a certain neuron

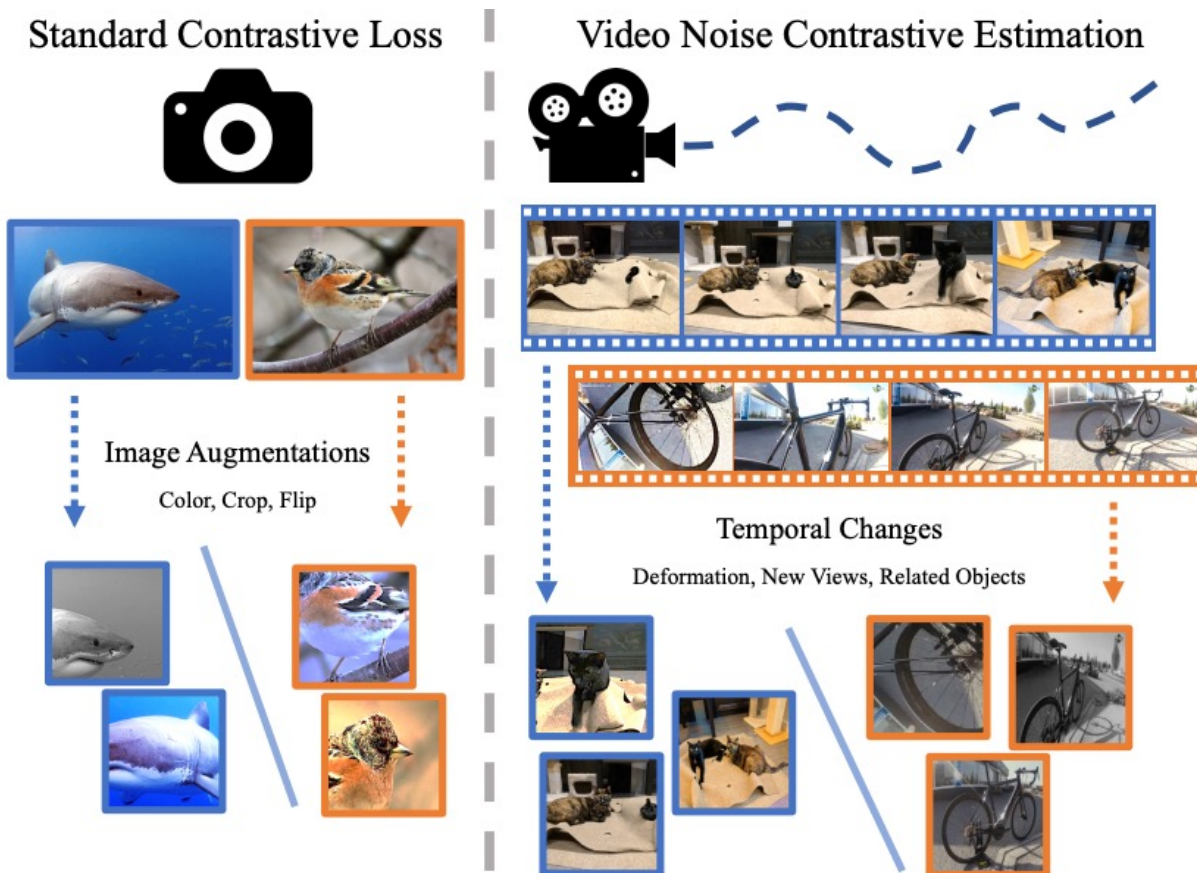
t-SNE of **audio** representations for 10 speakers



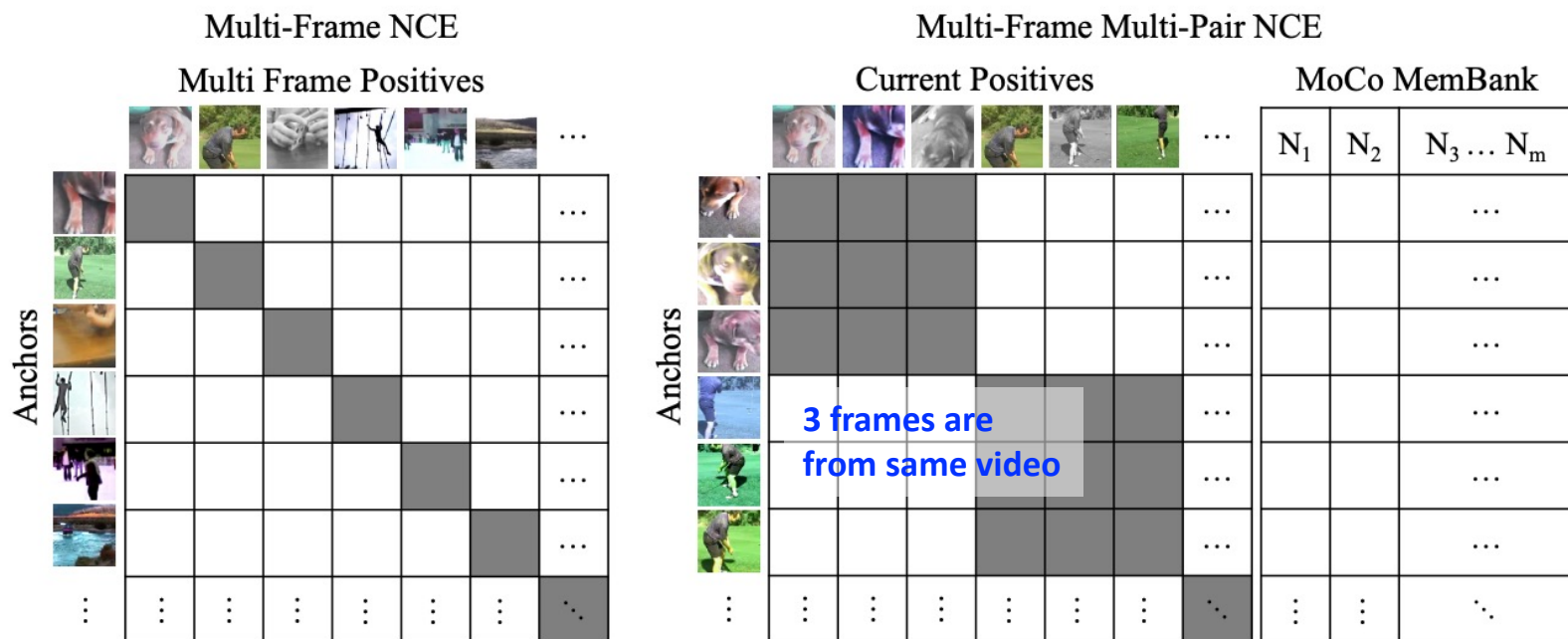
CPC improves agents on **RL environments** (red)

- **VINCE** [Gordon et al., 2019]

- Data augmentations cannot tell the **novel views** and **motions** of the objects
- Instead, use **video data** to provide 3D-aware positive views
- Namely, use **different frames** from the **same video** as positive samples



- **VINCE** [Gordon et al., 2019]
 - Data augmentations cannot tell the **novel views** and **motions** of the objects
 - Instead, use **video data** to provide 3D-aware positive views
 - Namely, use **different frames** from the **same video** as positive samples
- Since video has multiple frames, VINCE attracts **all positives** (not pair-wise)
 - Use 4 positive frames per video for experiments

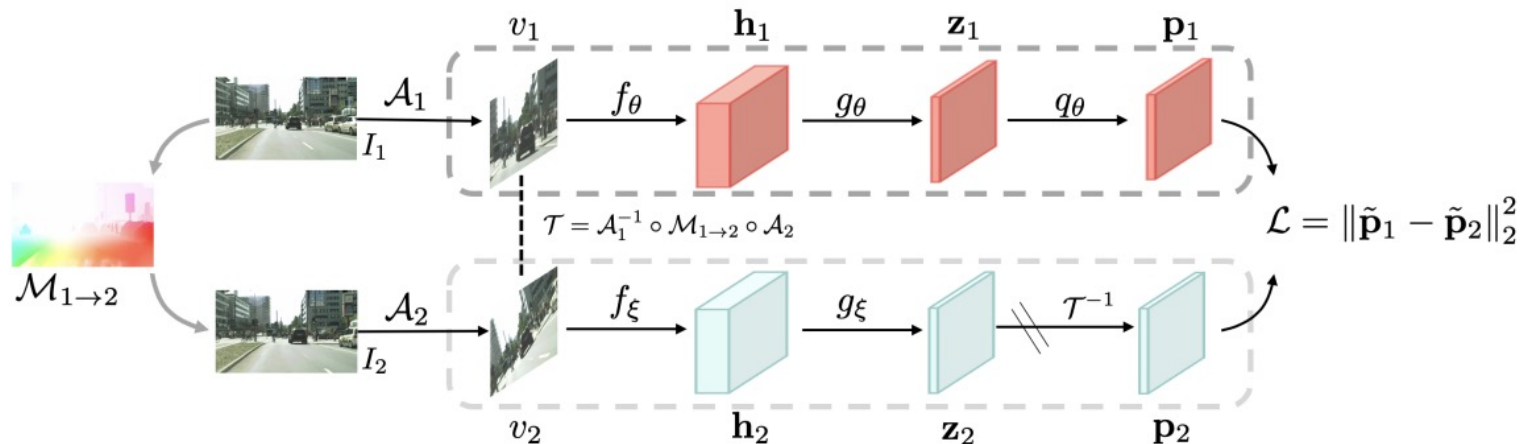


- **VINCE** [Gordon et al., 2019]
 - Data augmentations cannot tell the **novel views** and **motions** of the objects
 - Instead, use **video data** to provide 3D-aware positive views
 - Namely, use **different frames** from the **same video** as positive samples
- Using temporal information provides **better positive** views
 - **Same frame:** Use same frame images but positives are given by the same frame of different image augmentations
 - **Multi-frame (not multi-pair):** Use 2 frames from the same video

Images Per Video	Test Task				
	ImageNet	SUN Scene	Kinetics 400	OTB 2015 Precision	OTB 2015 Success
1: Same Frame	0.358	0.450	0.318	0.555	0.403
2: Multi-Frame	0.381	0.478	0.361	0.622	0.464
8: Multi-Frame Multi-Pair	0.400	0.495	0.362	0.629	0.465

- **FlowE** [Xiong et al., 2021]

- VINCE assumed frames from the same video are invariant
- However, we need to consider their **temporal changes**
- To this end, FlowE relaxes the assumption that the frames are **equivariant**
 - Let $I_2 = \mathcal{T}(I_1)$ where \mathcal{T} is a **transformation** between two frames I_1, I_2
 - Specifically, \mathcal{T} is a composition of data augmentations $\mathcal{A}_1, \mathcal{A}_2$ of each frame I_1, I_2 , respectively, and $\mathcal{M}_{1 \rightarrow 2}$ is an **optical flow**
 - Then the **spatial features** z_1, z_2 should satisfy the equivariance $z_2 = \mathcal{T}(z_1)$

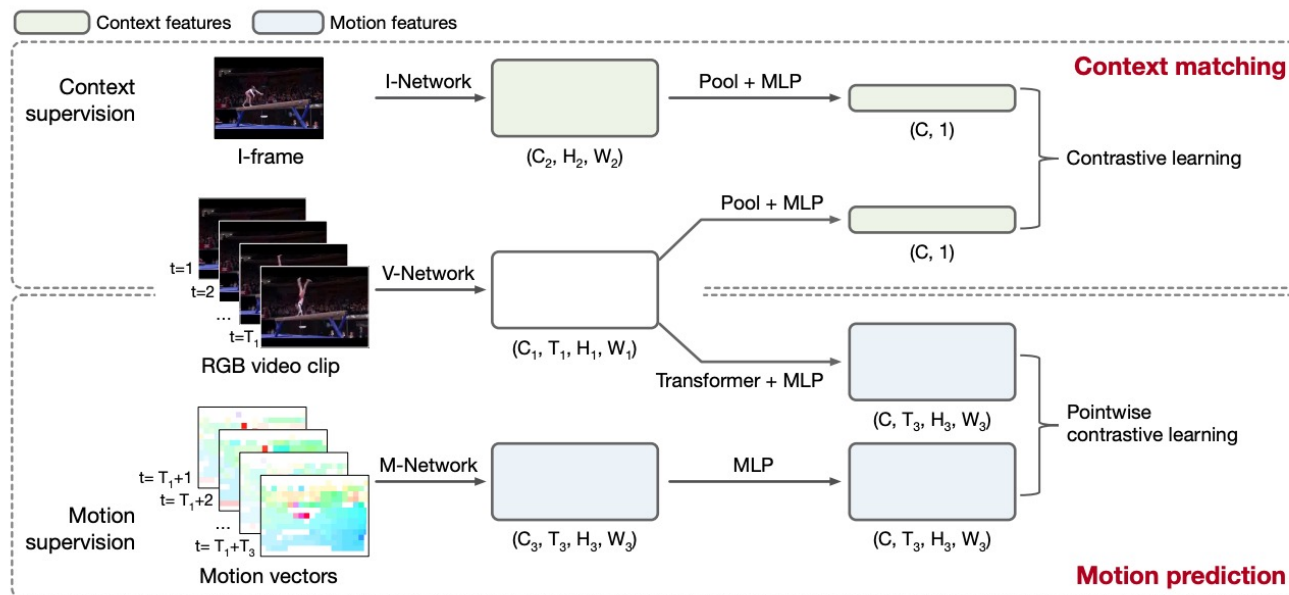


- **FlowE** [Xiong et al., 2021]
 - VINCE assumed frames from the same video are invariant
 - However, we need to consider their **temporal changes**
- To this end, FlowE relaxes the assumption that the frames are **equivariant**
 - Let $I_2 = \mathcal{T}(I_1)$ where \mathcal{T} is a **transformation** between two frames I_1, I_2
 - Specifically, \mathcal{T} is a composition of data augmentations $\mathcal{A}_1, \mathcal{A}_2$ of each frame I_1, I_2 , respectively, and $\mathcal{M}_{1 \rightarrow 2}$ is an **optical flow**
 - Then the **spatial features** z_1, z_2 should satisfy the equivariance $z_2 = \mathcal{T}(z_1)$
- Considering optical flow gives **better positive** than naïve invariance-based (VINCE)

Method	UrbanCity				BDD100K			
	mIoU	mAP	mIoU [†]	mAP [†]	mIoU	mAP	mIoU [†]	mAP [†]
Rand Init	9.4	0.0	27.3	6.4	9.8	0.0	22.0	5.5
CRW [22]	19.0	0.0	31.6	15.2	19.4	1.7	34.7	22.9
VINCE [16]	30.6	0.9	47.4	17.8	23.2	0.1	39.5	23.8
FlowE (Ours)	49.6	5.8	61.7	19.0	37.6	5.8	49.8	24.9
End-to-end supervised	63.3	2.2	67.0	16.5	52.0	8.0	56.6	20.0

- **Context and Motion Decoupling** [Huang et al., 2021]

- For video representation learning, many literature often explicitly **decouples** the **context** and **motion** supervision in the pretext task
- Jointly optimize **two self-supervision**
 - **(Context Matching)** Compare global features of key frames and video clips under the **contrastive learning** → (b) different frames
(though using clip = multiple frames as positive)
 - **(Motion Prediction)** Current **visual data in a video** are used to **predict** the future **motion information** → (a) future state



- **Limitations** of invariance-based approaches
 1. **Specialized for classification**
 - Invariance-based method clusters similar data into a **single point**
 - It is effective for classifier (or linear probing), less effective for different tasks (e.g., detection or segmentation for visual domain)
 - “Dense” contrastive learning methods have thus been proposed
 2. **Nontrivial choice of positive samples**
 - Data augmentation for **non-image domain** is arguable
 - Even arguable for non-natural images (e.g., medical or fine-grained)
 3. **Less scalable for large models and datasets**
 - Contrastive learning (empirically) less merits the **scaling law**
- **Next:** more scalable and domain-agnostic approaches
 - Generation-based approaches

1. Introduction

- Overview of Self-supervised Learning (SSL)
- Evaluating Self-supervised Representation

2. SSL via Pretext Tasks

- Pretext Tasks for Vision

3. SSL via Invariance (and Contrast)

- Clustering, Consistency, Contrastive
- Choices for Positive Samples

4. SSL via Generation

- Classic Approaches
- Masked Autoencoder (e.g., BERT, MAE)
- Sequential Prediction (e.g., GPT, world model)

- **Overview of Generation-based Approaches**

- There have been a long attempts to learn representation Z from data X
- To this end, many classic ML literature designed a **probabilistic model** $p(X, Z)$
 - They are called as generative models with latent variables
- **Ancient works** (before AlexNet, 2012)
 - Early works: probabilistic PCA and latent variable models (LVM)
 - In 2006~2009, the first **deep** learning revolution have arose
 - Deep Boltzmann machines (DBM) and deep belief networks (DBN)
 - They applied “unsupervised pretraining” to train deep networks
 - Though **RBM-based** approaches was not empirically successful, they inspired early modern generative models (e.g., VAE) a lot
 - Also, **autoencoder-based** approaches (e.g., denoising autoencoder; DAE) have been proposed → modernized to BigBiGAN, MAE, etc.

- **Overview of Generation-based Approaches**

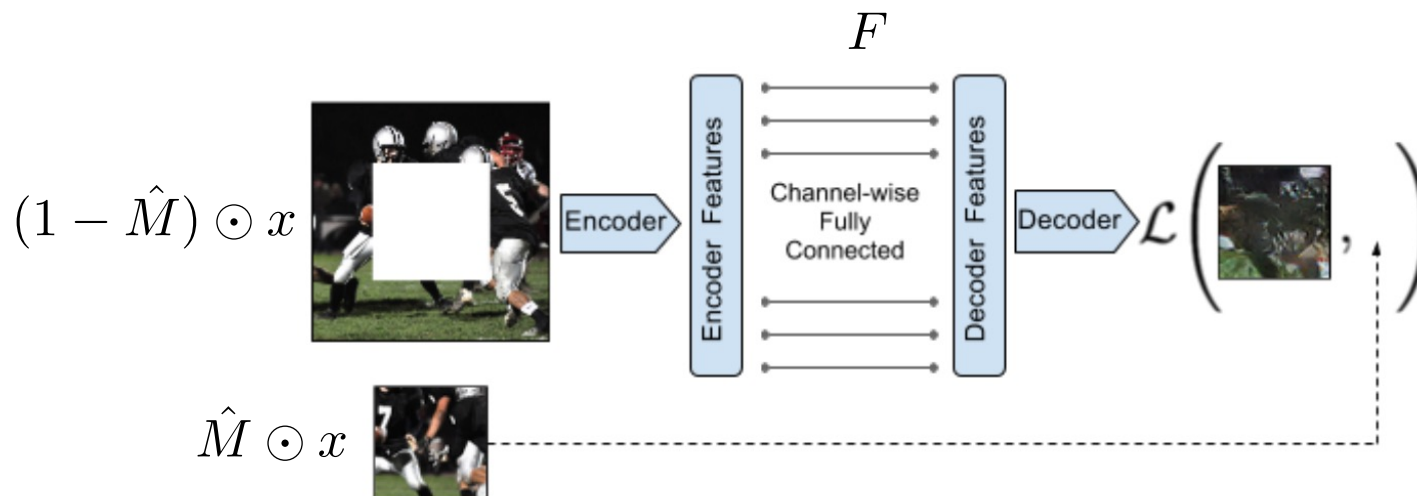
- There have been a long attempts to learn representation Z from data X
- To this end, many classic ML literature designed a **probabilistic model** $p(X, Z)$
 - They are called as generative models with latent variables
- **Classic approaches** (before contrastive learning, 2020)
 - We introduce some notable classic methods
 - Context encoder, a CNN version of masked autoencoder
 - Deep InfoMax and BigBiGAN, which were SOTA of then
- Recent methods can be categorized into **2 groups**:
 - **BERT-like approach** (or masked autoencoder)
 - Predict original X from perturbed \tilde{X} (learn $\tilde{X} \rightarrow Z \rightarrow X$ encoder)
 - **GPT-like approach** (or sequential prediction)
 - Predict future state X_{t+1} from past states $X_{1:t}$ (learn $X_{1:t} \rightarrow X_t$ decoder)

- **Context Encoder** [Pathak et al., 2016]

- **Task:** Predict the masked region using **its surrounding information**

- The auto-encoder is trained via **reconstruction loss**

$$\mathcal{L}_{\text{rec}}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2^2$$



- **Context Encoder** [Pathak et al., 2016]

- **Task:** Predict the masked region using **its surrounding information**

- The auto-encoder is trained via **reconstruction loss**

$$\mathcal{L}_{\text{rec}}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2^2$$

- With **adversarial loss**, reconstruction quality is improved further

$$\mathcal{L}_{\text{adv}} = \max_D \mathbb{E}_{x \in \mathcal{X}} \left[\log D(x) + \log(1 - D(F((1 - \hat{M}) \odot x)) \right]$$



(a) Input context



(b) Human artist



(c) Context Encoder
(L2 loss)



(d) Context Encoder
(L2 + Adversarial loss)

- **Context Encoder** [Pathak et al., 2016]

- **Task:** Predict the masked region using **its surrounding information**

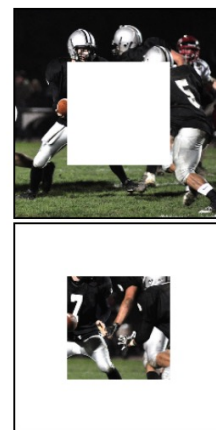
- The auto-encoder is trained via **reconstruction loss**

$$\mathcal{L}_{\text{rec}}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2^2$$

- With **adversarial loss**, reconstruction quality is improved further

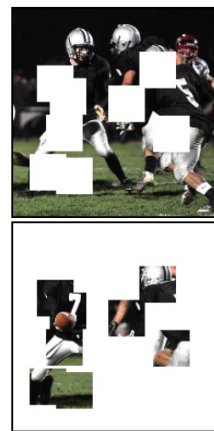
$$\mathcal{L}_{\text{adv}} = \max_D \mathbb{E}_{x \in \mathcal{X}} \left[\log D(x) + \log(1 - D(F((1 - \hat{M}) \odot x)) \right]$$

- How to **construct** the masks?



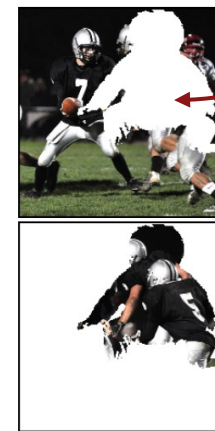
(a) Central region

<



(b) Random block

≈



(c) Random region

A segmentation mask in other dataset

- **Deep InfoMax** [Hjelm et al., 2019]

- **Idea:** Maximizing mutual information between inputs and features
- $Y = E_\psi(X)$ is the feature vector of input X where E_ψ is an embedding function
- **How to optimize mutual information?** [Donsker & Varadhan, 1983]

$$\mathcal{I}(X; Y) := \mathcal{D}_{\text{KL}}(\mathbb{J} \parallel \mathbb{M}) \geq \hat{\mathcal{I}}_w^{(\text{DV})}(X; Y) := \mathbb{E}_{\mathbb{J}}[T_w(x, y)] - \log \mathbb{E}_{\mathbb{M}}[e^{T_w(x, y)}]$$

joint distribution ↗ ↖ marginal distribution

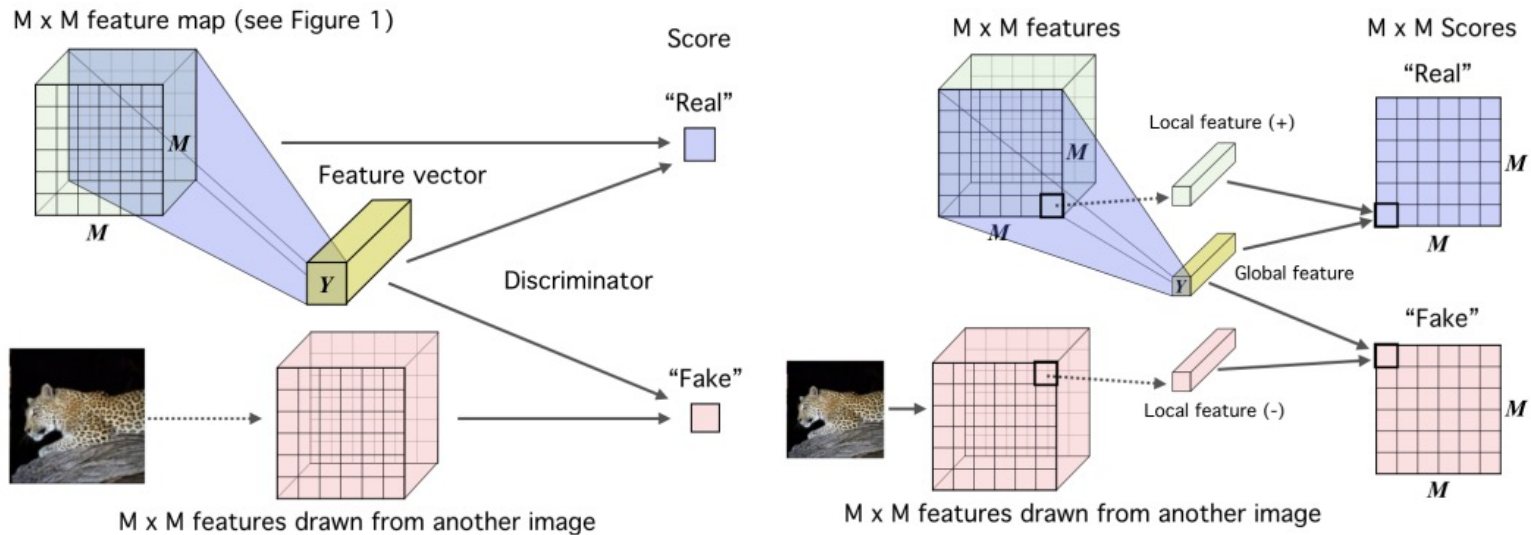
- Optimize the embedding function E_ψ and discriminator T_w simultaneously

$$\hat{w}, \hat{\psi} = \arg \max_{w, \psi} \hat{\mathcal{I}}_w(X; E_\psi(X))$$

- **Deep InfoMax** [Hjelm et al., 2019]

- **Idea:** Maximizing mutual information between inputs and features

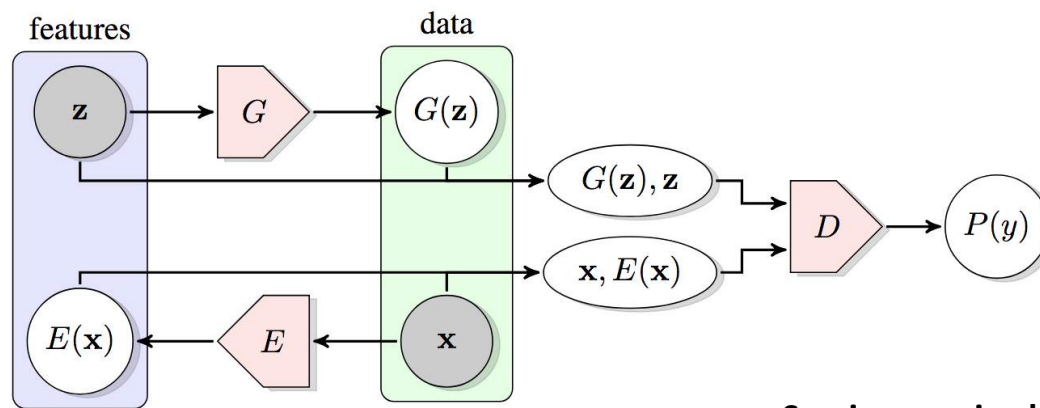
- $Y = E_{\psi}(X)$ is the feature vector of input X where E_{ψ} is an embedding function



- Instead of (left) maximizing MI between global features, **(right) doing MI between global and local features** achieves better performance

- **BigBiGAN** [Donahue et al., 2019]

- After the success of GAN for image **generation**, numerous work attempted to extend the applicability of GAN for **representation learning**
- To this end, ALI/BiGAN (2017) learned a joint distribution $p(X, Z)$ with GAN
 - ALI/BiGAN performed well on low-resolution images



Semi-supervised learning on CIFAR-10

Number of labeled examples	1000	2000	4000	8000
Model	Misclassification rate			
Ladder network (Rasmus et al., 2015)			20.40	
CatGAN (Springenberg, 2015)			19.58	
GAN (feature matching) (Salimans et al., 2016)	21.83 ± 2.01	19.61 ± 2.09	18.63 ± 2.32	17.72 ± 1.82
ALI (ours, no feature matching)	19.98 ± 0.89	19.09 ± 0.44	17.99 ± 1.62	17.05 ± 1.49

- **BigBiGAN** [Donahue et al., 2019]

- After the success of GAN for image **generation**, numerous work attempted to extend the applicability of GAN for **representation learning**
- To this end, ALI/BiGAN (2017) learned a joint distribution $p(X, Z)$ with GAN
 - Leveraging the power of BigGAN on high-resolution image generation, BigBiGAN achieved SOTA representation learning performance
 - It was the SOTA before the dominance of contrastive learning
 - Cf. ContraD (2021) combined BigBiGAN and contrastive learning

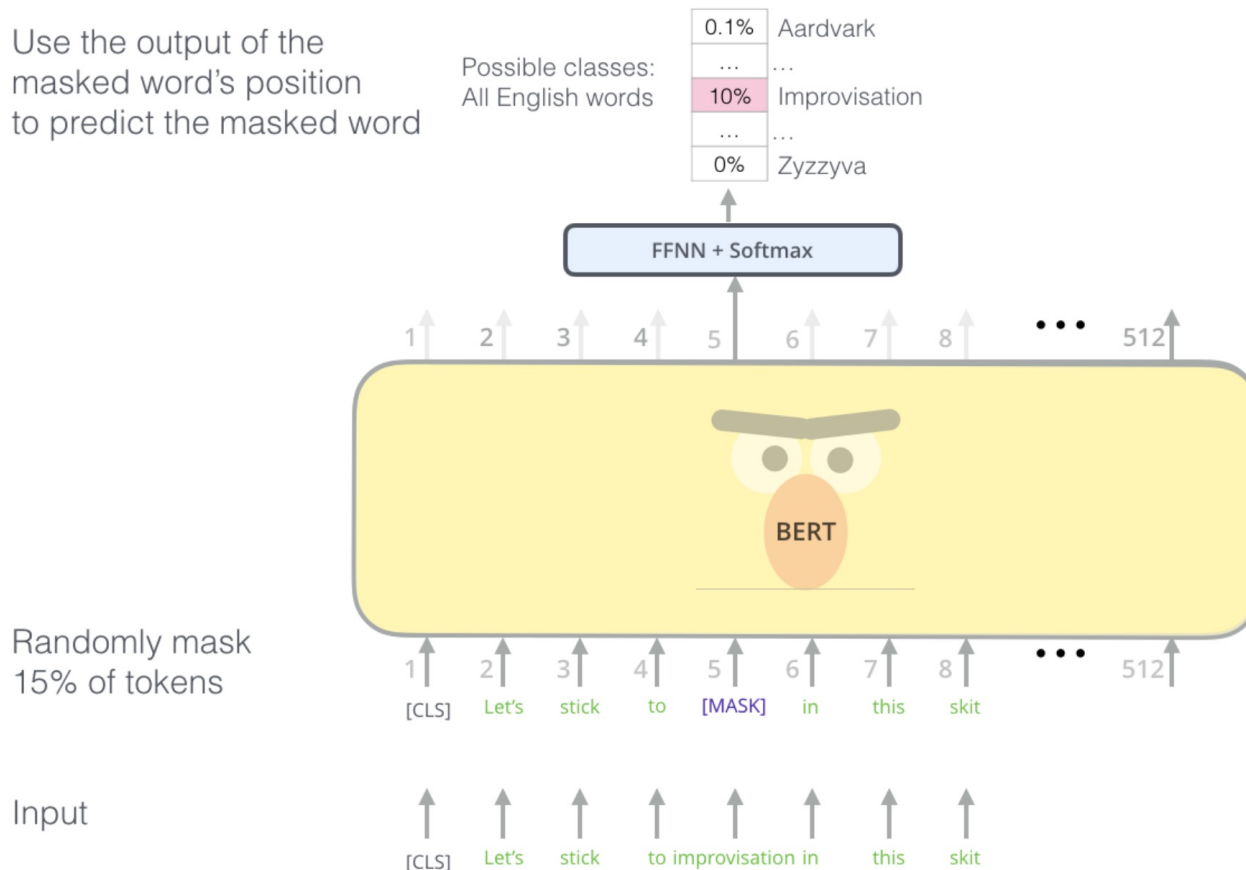
Method	Architecture	Feature	Top-1	Top-5
BiGAN [7, 42]	AlexNet	Conv3	31.0	-
SS-GAN [4]	ResNet-19	Block6	38.3	-
Motion Segmentation (MS) [30, 6]	ResNet-101	AvePool	27.6	48.3
Exemplar (Ex) [8, 6]	ResNet-101	AvePool	31.5	53.1
Relative Position (RP) [5, 6]	ResNet-101	AvePool	36.2	59.2
Colorization (Col) [41, 6]	ResNet-101	AvePool	39.6	62.5
Combination of MS+Ex+RP+Col [6]	ResNet-101	AvePool	-	69.3
CPC [39]	ResNet-101	AvePool	48.7	73.6
Rotation [11, 24]	RevNet-50 $\times 4$	AvePool	55.4	-
Efficient CPC [17]	ResNet-170	AvePool	61.0	83.0
BigBiGAN (ours)	ResNet-50	AvePool	55.4	77.4
	ResNet-50	BN+CReLU	56.6	78.6
	RevNet-50 $\times 4$	AvePool	60.8	81.4
	RevNet-50 $\times 4$	BN+CReLU	61.3	81.9

- **Overview of Generation-based Approaches**

- There have been a long attempts to learn representation Z from data X
- To this end, many classic ML literature designed a probabilistic model $p(X, Z)$
 - They are called as generative models with latent variables
- **Classic approaches** (before contrastive learning, 2020)
 - We introduce some notable classic methods
 - Context encoder, a CNN version of masked autoencoder
 - Deep InfoMax and BigBiGAN, which were SOTA of then
- Recent methods can be categorized into **2 groups**:
 - **BERT-like approach** (or masked autoencoder)
 - Predict original X from perturbed \tilde{X} (learn $\tilde{X} \rightarrow Z \rightarrow X$ encoder)
 - **GPT-like approach** (or sequential prediction)
 - Predict future state X_{t+1} from past states $X_{1:t}$ (learn $X_{1:t} \rightarrow X_t$ decoder)

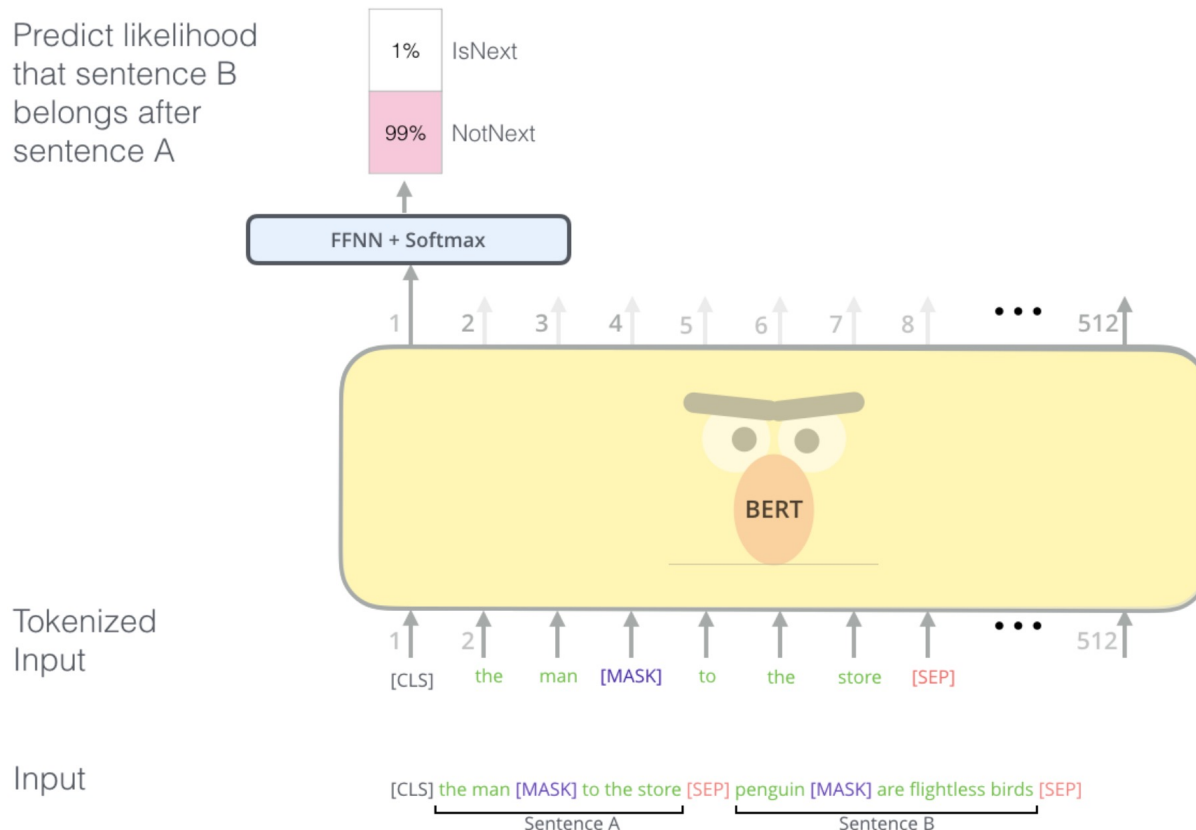
SSL via Generation – Masked Autoencoder

- **BERT** [Devlin et al., 2018]
 - As **encoders get bidirectional context**, language modeling **can't be used anymore**
 - Instead, **masked language modeling** is used for pre-training
 - Replace some fraction of words (15%) in the input, then predict these words



SSL via Generation – Masked Autoencoder

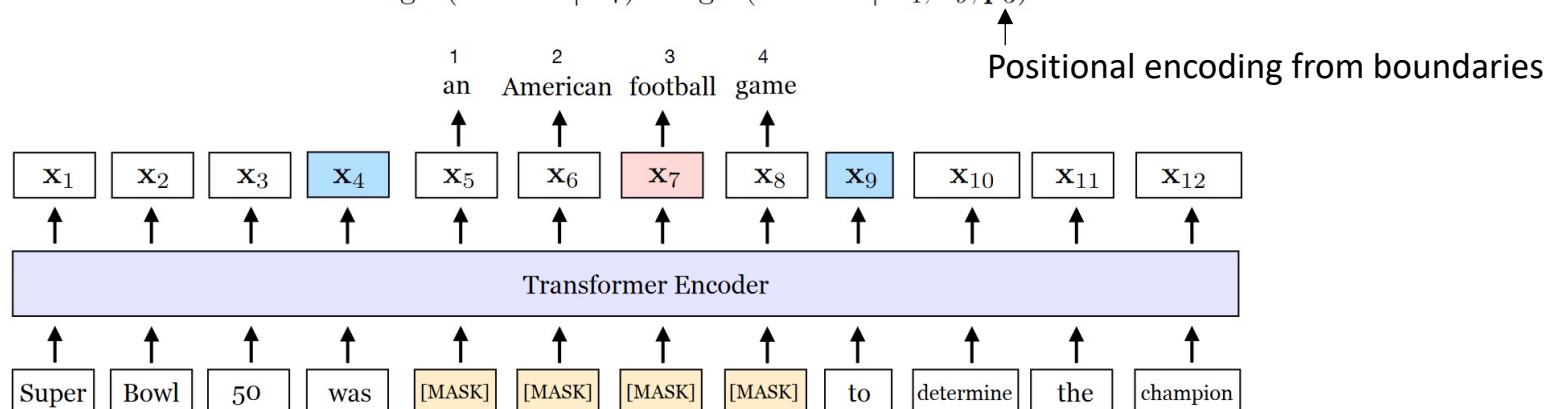
- **BERT** [Devlin et al., 2018]
 - As **encoders get bidirectional context**, language modeling **can't be used anymore**
 - Instead, **masked language modeling** is used for pre-training
 - Additionally, **next sentence prediction** (NSP) task is used for pre-training
 - Decide whether two input sentences are **consecutive or not**



- **SpanBERT** [Joshi et al., 2019]
 - **Recap:** BERT selects [MASK] tokens at uniformly random
 - **Idea:** mask contiguous random spans rather than random tokens

$$\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football})$$

$$= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)$$



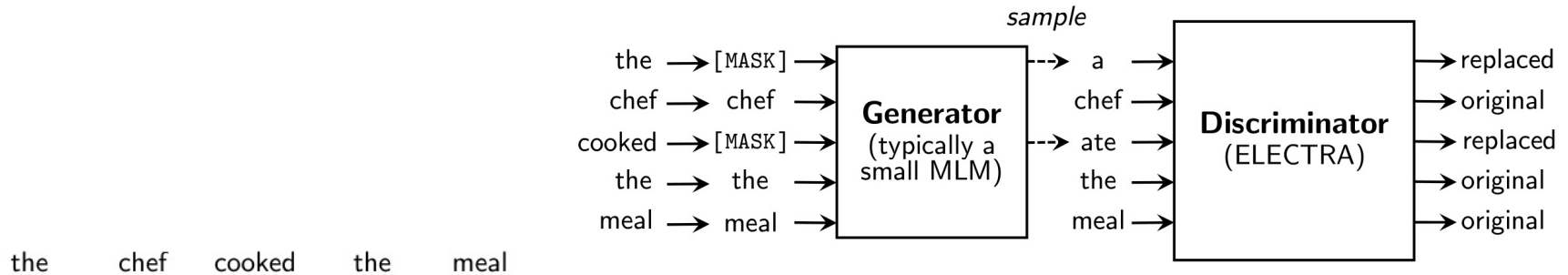
- **Span Boundary Objective (SBO)** encourages model to store **span-level information** at the boundary tokens

	SQuAD 2.0	NewsQA	TriviaQA	Coref	MNLI-m	QNLI	GLUE (Avg)
Span Masking (2seq) + NSP	85.4	73.0	78.8	76.4	87.0	93.3	83.4
Span Masking (1seq)	86.7	73.4	80.0	76.3	87.3	93.8	83.8
Span Masking (1seq) + SBO	86.8	74.1	80.3	79.0	87.6	93.9	84.0

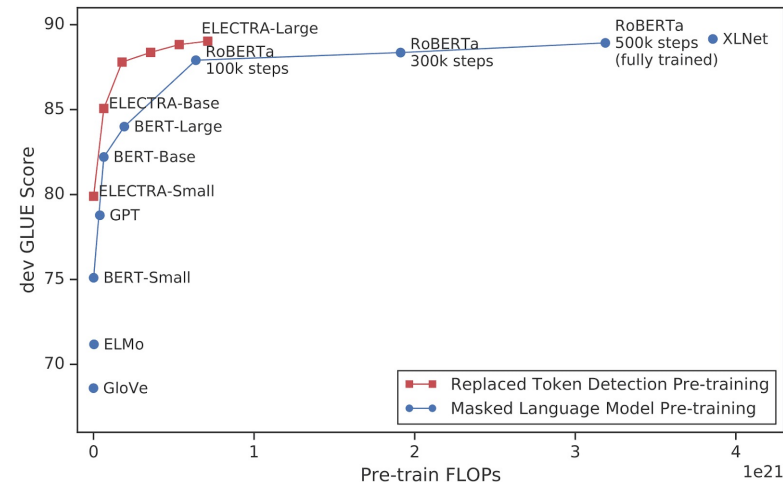
SSL via Generation – Masked Autoencoder

- **ELECTRA** [Clark et al., 2020]
 - **Idea:** Replaced Token Detection (RTD) inspired by GANs

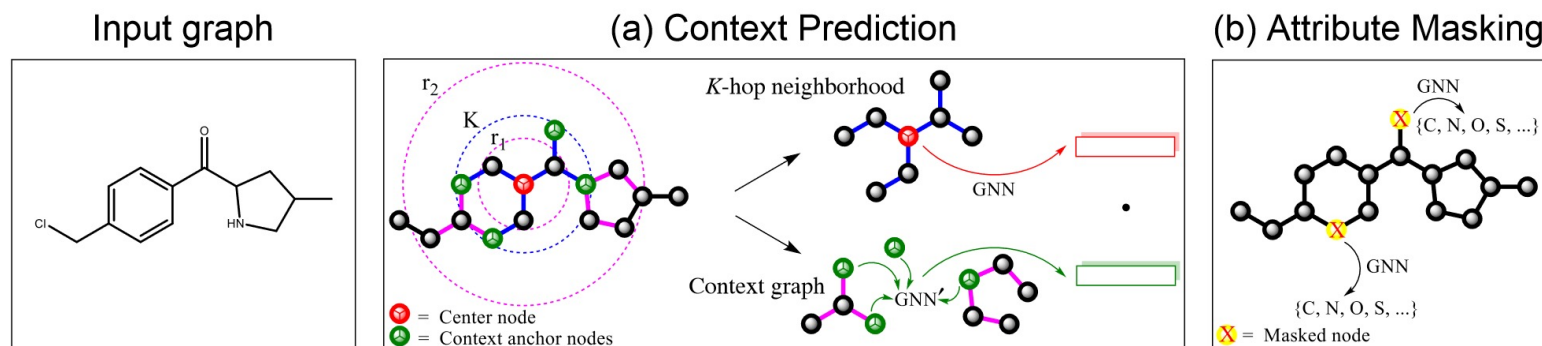
Replaced Token Detection



- **Generator** is trained with MLM
 - Generator can be trained to fool discriminator adversarially, but it is worse than MLM training
- **Discriminator** is trained to predict whether each token is replaced one or not
- Both are **jointly trained**, and discriminator will be used for downstream tasks



- **Strategies for Pre-training Graph Neural Networks** [Hu et al., 2020]
 - **Note.** This paper also uses well-known properties of molecules as supervision
 - **Idea:**
 - **Context Prediction:** predict surrounding graph structures
 - **Attribute Masking:** predict masked node attributes like MLM

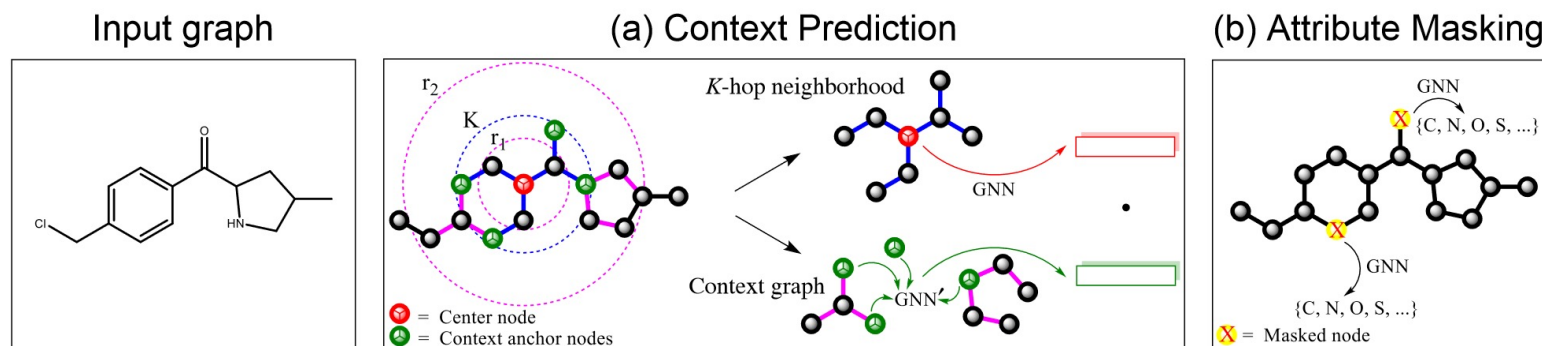


- **Details of Context Prediction:**
 - $h_v^{(K)}$: Node embedding of **center node** v with K-hop neighborhood
 - c_v^G : Average of embeddings of **context anchor nodes**
 - Learning with negative sampling:

$$\sigma \left(h_v^{(K)\top} c_{v'}^{G'} \right) \approx \mathbf{1}[v = v' \wedge G = G']$$

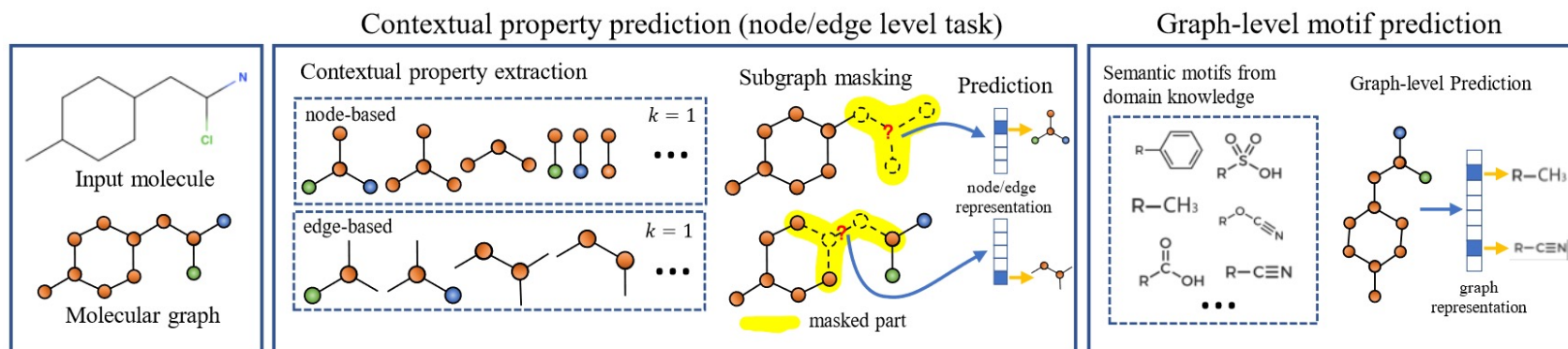
- This is similar to contrastive learning

- **Strategies for Pre-training Graph Neural Networks** [Hu et al., 2020]
 - **Note.** This paper also uses well-known properties of molecules as supervision
 - **Idea:**
 - **Context Prediction:** predict surrounding graph structures
 - **Attribute Masking:** predict masked node attributes like MLM



Dataset		BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE	Average
# Molecules		2039	7831	8575	1427	1478	93087	41127	1513	/
# Binary prediction tasks		1	12	617	27	2	17	1	1	/
Pre-training strategy		Out-of-distribution prediction (scaffold split)								
Graph-level	Node-level									
–	–	65.8 ±4.5	74.0 ±0.8	63.4 ±0.6	57.3 ±1.6	58.0 ±4.4	71.8 ±2.5	75.3 ±1.9	70.1 ±5.4	67.0
–	Infomax	68.8 ±0.8	75.3 ±0.5	62.7 ±0.4	58.4 ±0.8	69.9 ±3.0	75.3 ±2.5	76.0 ±0.7	75.9 ±1.6	70.3
–	EdgePred	67.3 ±2.4	76.0 ±0.6	64.1 ±0.6	60.4 ±0.7	64.1 ±3.7	74.1 ±2.1	76.3 ±1.0	79.9 ±0.9	70.3
–	AttrMasking	64.3 ±2.8	76.7 ±0.4	64.2 ±0.5	61.0 ±0.7	71.8 ±4.1	74.7 ±1.4	77.2 ±1.1	79.3 ±1.6	71.1
–	ContextPred	68.0 ±2.0	75.7 ±0.7	63.9 ±0.6	60.9 ±0.6	65.9 ±3.8	75.8 ±1.7	77.3 ±1.0	79.6 ±1.2	70.9
Supervised	–	68.3 ±0.7	77.0 ±0.3	64.4 ±0.4	62.1 ±0.5	57.2 ±2.5	79.4 ±1.3	74.4 ±1.2	76.9 ±1.0	70.0
Supervised	Infomax	68.0 ±1.8	77.8 ±0.3	64.9 ±0.7	60.9 ±0.6	71.2 ±2.8	81.3 ±1.4	77.8 ±0.9	80.1 ±0.9	72.8
Supervised	EdgePred	66.6 ±2.2	78.3 ±0.3	66.5 ±0.3	63.3 ±0.9	70.9 ±4.6	78.5 ±2.4	77.5 ±0.8	79.1 ±3.7	72.6
Supervised	AttrMasking	66.5 ±2.5	77.9 ±0.4	65.1 ±0.3	63.9 ±0.9	73.7 ±2.8	81.2 ±1.9	77.1 ±1.2	80.3 ±0.9	73.2
Supervised	ContextPred	68.7 ±1.3	78.1 ±0.6	65.7 ±0.6	62.7 ±0.8	72.6 ±1.5	81.3 ±2.1	79.9 ±0.7	84.5 ±0.7	74.2

- **GROVER** [Rong et al., 2020]
 - **Note.** This paper also propose an architecture for molecules
 - **Idea:**
 - **Contextual Property Prediction** by masking subgraph like SpanBERT
 - Subgraph categories are extracted from data: k-hop sub-graphs
 - **Graph-level Motif Prediction** (multi-label classification) with domain knowledge
 - Motifs can be obtained by a program (RDKit)



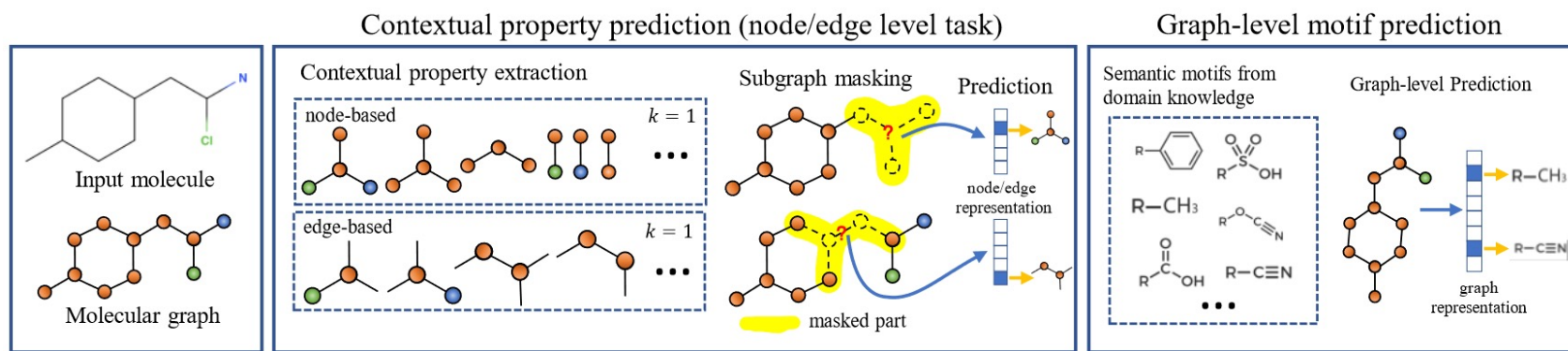
- GROVER** [Rong et al., 2020]

- Note.** This paper also propose an architecture for molecules

- Idea:**

- Contextual Property Prediction** by masking subgraph like SpanBERT

- Graph-level Motif Prediction** (multi-label classification) with domain knowledge

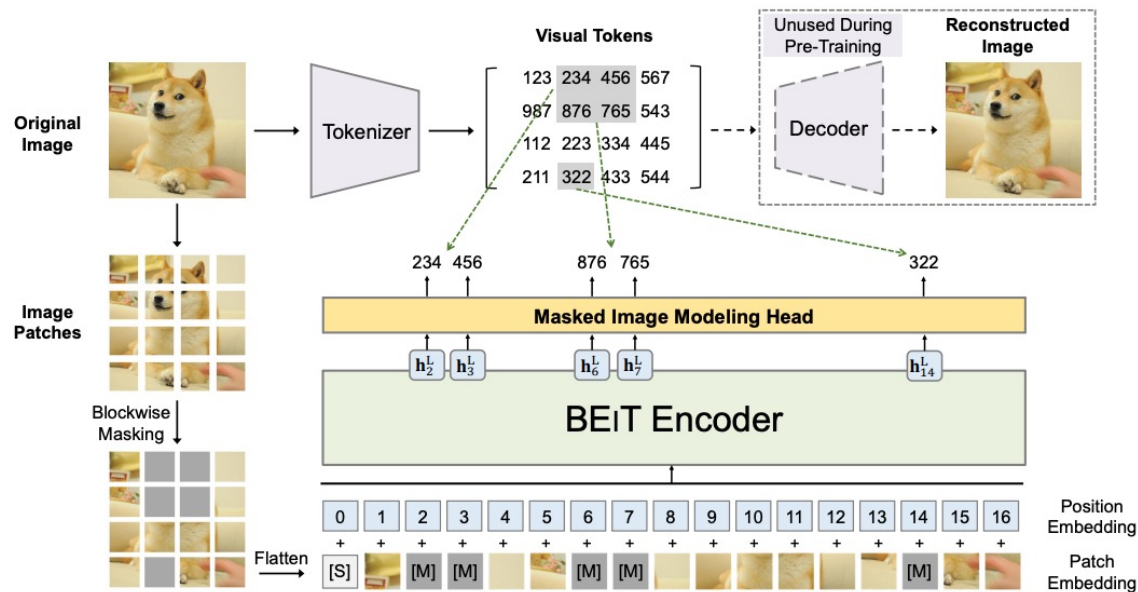


- Pretraining improves GNNs in various downstream tasks

	GROVER	No Pretrain	Abs. Imp.
BBBP (2039)	0.940	0.911	+0.029
SIDER (1427)	0.658	0.624	+0.034
ClinTox (1478)	0.944	0.884	+0.060
BACE (1513)	0.894	0.858	+0.036
Tox21 (7831)	0.831	0.803	+0.028
ToxCast (8575)	0.737	0.721	+0.016
Average	0.834	0.803	+0.038

- **BEiT** [Bao et al., 2022]

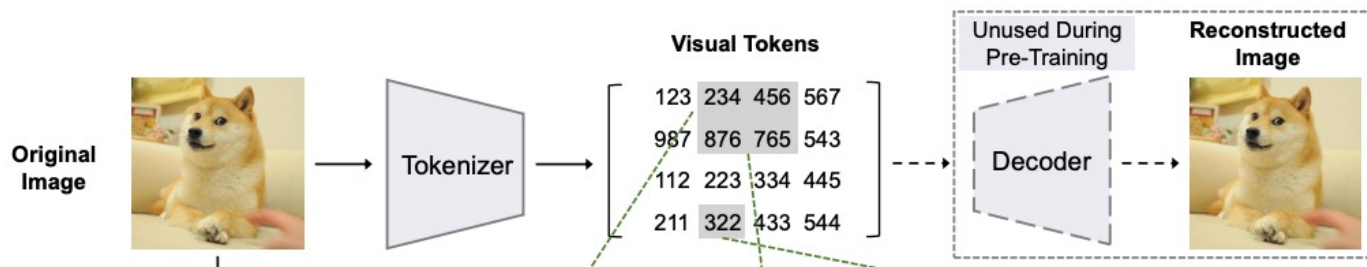
- **Task:** Masked visual tokens prediction
 - Similar to BERT in NLP, BEiT randomly masks image patches and trains to **recover the visual tokens** of masked patches (instead of the raw pixels)
 - Visual token: a discretized vocabulary for the image patch



- BEiT training procedure is consist of two stages:
 1. Learning visual tokens
 2. Masked image modeling

- **BEiT** [Bao et al., 2022]
 - **Task:** Masked visual tokens prediction
 - BEiT training procedure is consist of two stages:

1. Learning visual tokens

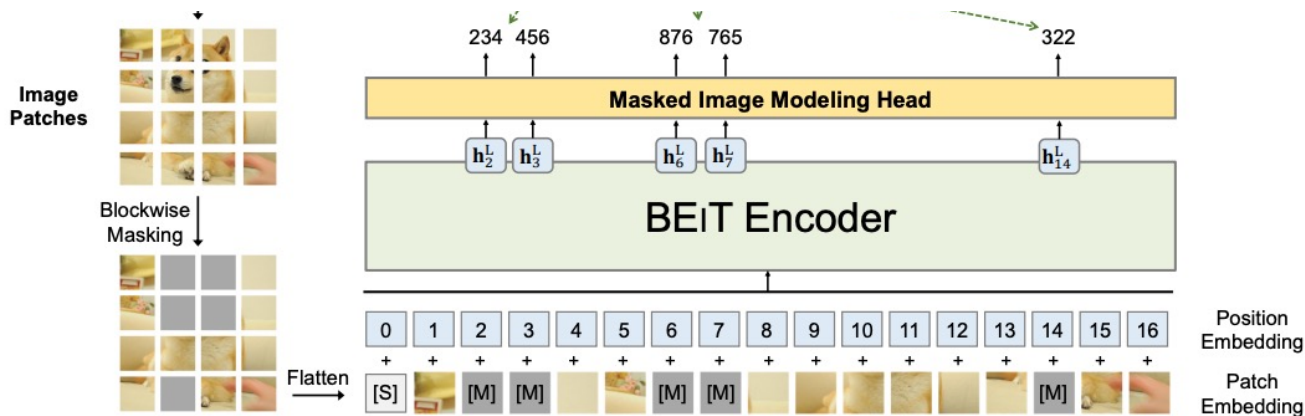


- In this stage, a **discrete variational autoencoder (dVAE)** is trained to represent each 224×224 image into a 14×14 grid of **discrete image tokens**, each element of which can assume 8192 possible values
 - The tokenizer $q_{\phi}(\mathbf{z}|\mathbf{x})$ maps image pixels into a visual codebook
 - The decoder $p_{\psi}(\mathbf{x}|\mathbf{z})$ learns to reconstruct the input image

- **BEiT** [Bao et al., 2022]

- **Task:** Masked visual tokens prediction
- BEiT training procedure is consist of two stages:

2. Masked Image Modeling



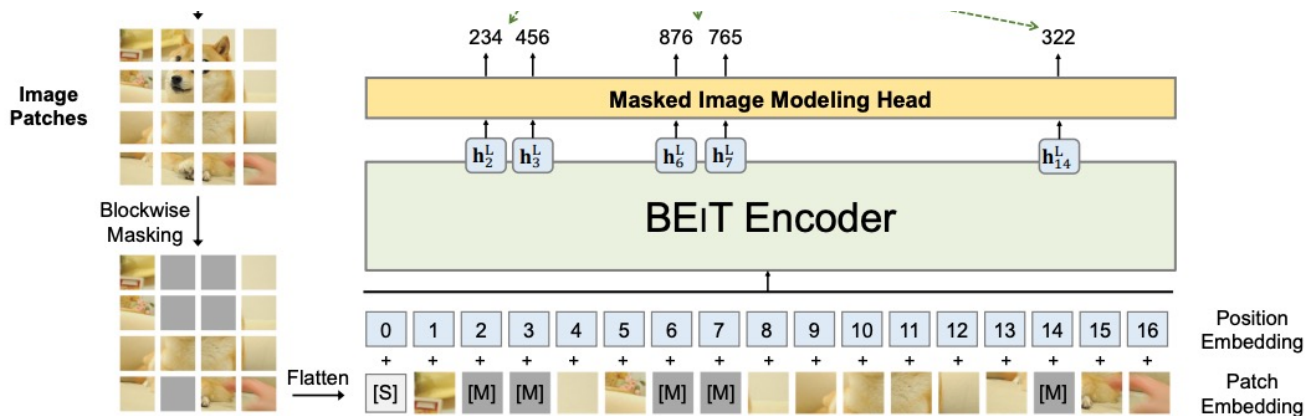
- The standard ViT is used as the backbone network
- Some image patches are randomly masked (approx. 40%), and then the **visual tokens that corresponds to the masked patches** are predicted
 - The objective is maximizing the log-likelihood of the correct visual tokens z_i given the corrupted image $x^{\mathcal{M}}$ with the masked positions \mathcal{M}

$$\max_{x \in \mathcal{D}} \sum \mathbb{E}_{\mathcal{M}} \left[\sum_{i \in \mathcal{M}} \log p_{\text{MIM}}(z_i | x^{\mathcal{M}}) \right]$$

- **BEiT** [Bao et al., 2022]

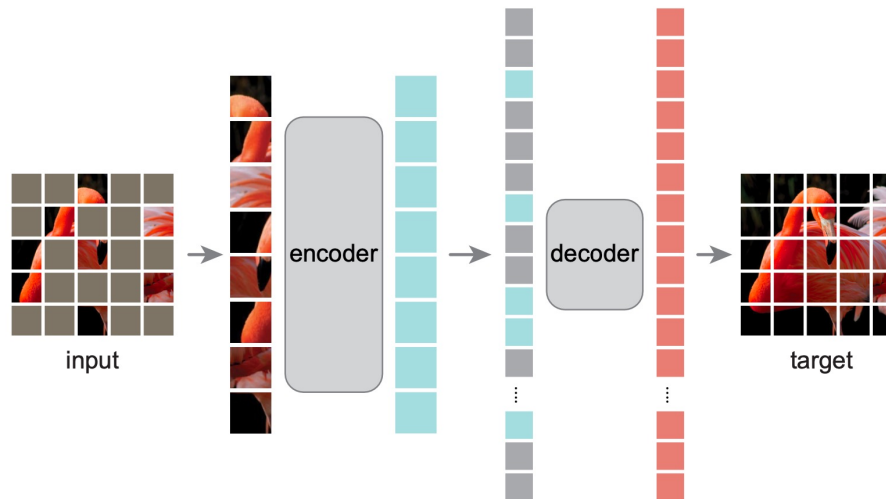
- **Task:** Masked visual tokens prediction
- BEiT training procedure is consist of two stages:

2. Masked Image Modeling



- During masked image modeling, **block-wise masking strategy** is used
 - A block with the minimum number of patches to 16 is masked
 - Repeat masking until obtaining enough masked patches (total 40% of patches)

- **MAE** [He et al., 2022]
 - **Task:** Predicting the **pixel** values for each masked patch
 - Objective: MSE loss of masked patches



- **Key components:**
 - High masking ratio (75%):
 - BERT masks 15% of tokens, MAE needs higher masking ratio
 - Asymmetric encoder-decoder architecture:
 - MAE allows to train very large transformer encoder by using the lightweight decoder => it significantly reduces the pre-training time

- **MAE** [He et al., 2022]
 - **Task:** Predicting the **pixel** values for each masked patch
 - **Asymmetric encoder-decoder architecture:** MAE uses the **lightweight decoder**

blocks	ft	lin
1	84.8	65.5
2	84.9	70.0
4	84.9	71.9
8	84.9	73.5
12	84.4	73.3

(a) **Decoder depth.** A deep decoder can improve linear probing accuracy.

dim	ft	lin
128	84.9	69.1
256	84.8	71.3
512	84.9	73.5
768	84.4	73.1
1024	84.3	73.1

(b) **Decoder width.** The decoder can be narrower than the encoder (1024-d).

- The decoder depth is less influential for improving fine-tuning
 - Only a single transformer block decoder can perform strongly with fine-tuning
- MAE decoder uses the decoder with 8 blocks and a width of 512-d, which has 9% FLOPs per token vs. ViT-L

- **MAE** [He et al., 2022]
 - **Task:** Predicting the **pixel** values for each masked patch
 - **Other intriguing properties of MAE**

case	ft	lin	FLOPs
encoder w/ [M]	84.2	59.6	3.3×
encoder w/o [M]	84.9	73.5	1×

(c) **Mask token.** An encoder without mask tokens is more accurate and faster (Table 2).

case	ft	lin
pixel (w/o norm)	84.9	73.5
pixel (w/ norm)	85.4	73.9
PCA	84.6	72.3
dVAE token	85.3	71.6

(d) **Reconstruction target.** Pixels as reconstruction targets are effective.

case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	84.9	73.5
crop + color jit	84.3	71.9

(e) **Data augmentation.** Our MAE works with minimal or no augmentation.

(c) MAE skips the mask token [M] in the encoder and apply it later in the decoder

- It is more accurate and decreases the computation time

(d) Predicting pixels with *per-patch* normalization improves accuracy

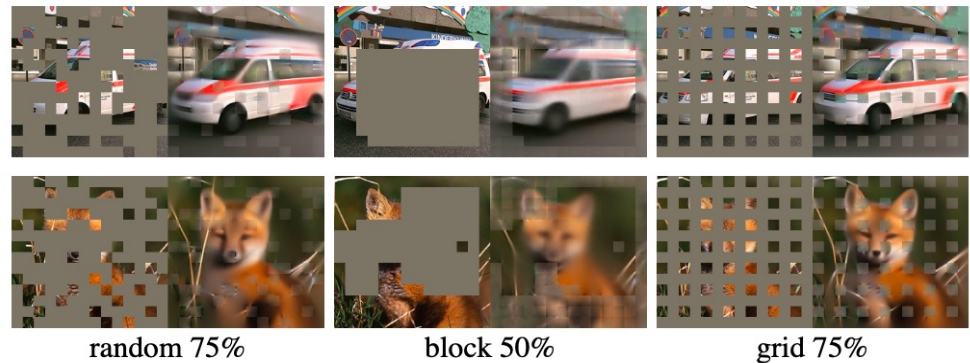
(e) MAE works well using cropping-only augmentation

- MAE behaves decently even if using no data augmentation

- **MAE** [He et al., 2022]
 - **Task:** Predicting the **pixel** values for each masked patch
 - **Other intriguing properties of MAE**

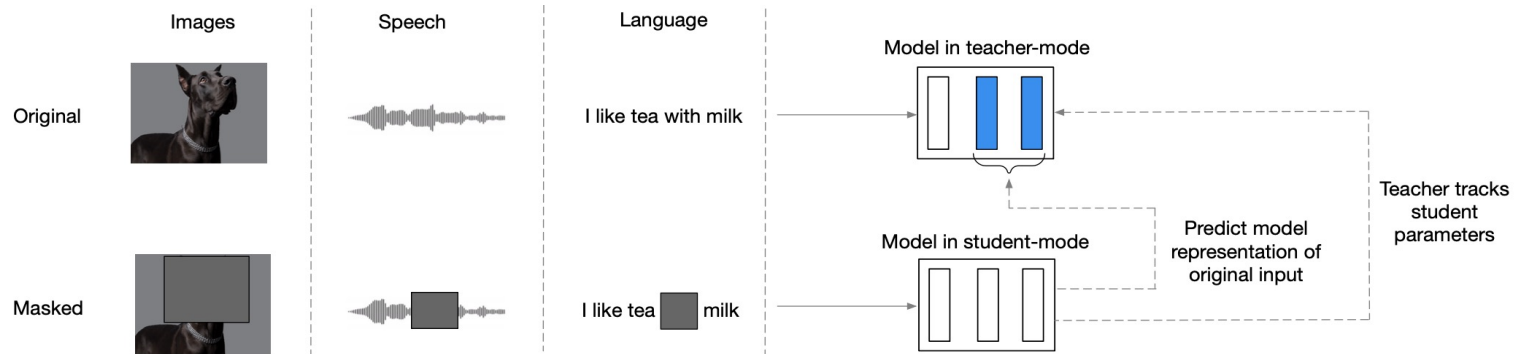
case	ratio	ft	lin
random	75	84.9	73.5
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

(f) **Mask sampling.** Random sampling works the best. See Figure 6 for visualizations.



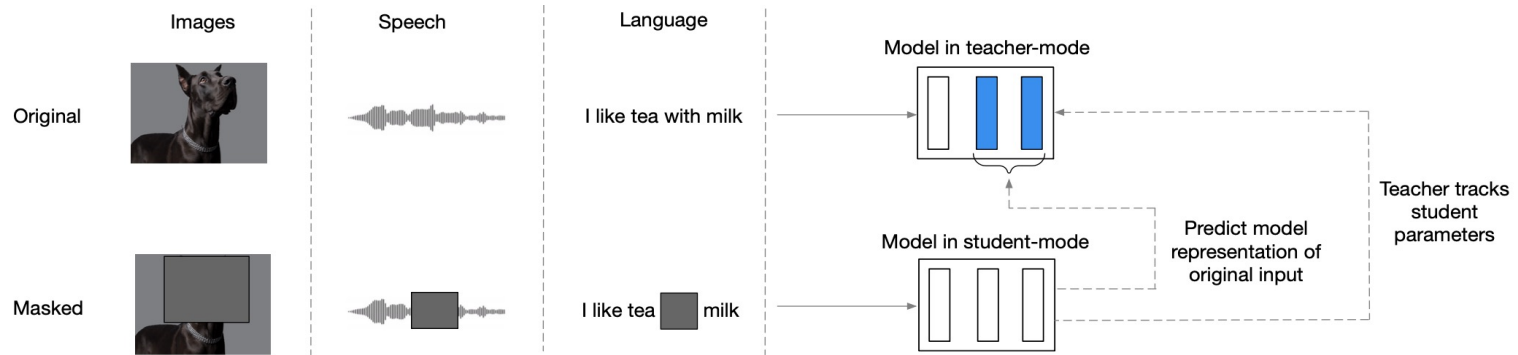
- (f) Random patch masking is better than block-wise and grid-wise sampling
- Block-wise sampling: Removes large random blocks
 - Grid-wise sampling: Keeps one of every four patches

- **data2vec** [Baevski et al., 2022]
 - data2vec is a framework for **general self-supervised learning** for images, speech, and text where the **learning objective is identical in each modality**



- **Modality-unified algorithm:**
 - 1) Build representations of the full input data with the teacher model
 - The teacher is an exponentially decaying average of the student
 - 2) Encode the masked version of the input sample with the student model and predict the representations of original input
- Modality-specified data processing and masking strategies are used

- **data2vec** [Baevski et al., 2022]
 - data2vec is a framework for **general self-supervised learning** for images, speech, and text where the **learning objective is identical in each modality**



- The objective is **predicting the representation for time-steps** which are masked
 - data2vec uses the standard transformer architecture
 - Training targets are the output of the top K blocks of the teach network
 - \hat{a}_t^l : the normalized output of block l at time-step t
 - Training target: $y_t = \frac{1}{K} \sum_{l=L-K+1}^L \hat{a}_t^l$
 - The objective is smooth-L1 loss between y_t and the prediction $f_t(x)$ at t :

$$\mathcal{L}(y_t, f_t(x)) = \begin{cases} \frac{1}{2}(y_t - f_t(x))^2 / \beta & |y_t - f_t(x)| \leq \beta \\ (|y_t - f_t(x)| - \frac{1}{2}\beta) & \text{otherwise} \end{cases}$$

- **data2vec** [Baevski et al., 2022]
 - data2vec is a framework for **general self-supervised learning** for images, speech, and text where the **learning objective is identical in each modality**
 - **Modality-specified data processing and masking strategy**
 - **Image processing**
 - (Input embed) Embed images of 224×224 pixels as patches of 16×16 pixel
 - (Masking) Apply BEiT masking strategy with 60% masking ratio
 - **Speech processing**
 - (Input embed) Sample with 16kHz then forward seven temporal convolutions
 - (Masking) Mask 49% of all time-steps
 - **NLP processing**
 - (Input embed) The input data is tokenized using a byte-pair encoding (BPE)
 - (Masking) Apply BERT masking strategy to 15% of uniformly selected tokens
 - 80% are replaced by a learned mask token, [M]
 - 10% are left unchanged
 - 10% are replaced by randomly selected vocabulary token

• data2vec [Baevski et al., 2022]

- data2vec shows a new state of the art or competitive performance to predominant approaches on three domains
 - Vision task: ImageNet classification
 - Speech task: Word error rate (smaller is better) on the Librispeech dataset
 - NLP task: GLEU benchmark

Table 1. Computer vision: top-1 validation accuracy on ImageNet-1K with ViT-B and ViT-L models. data2vec ViT-B was trained for 800 epochs and ViT-L for 1,600 epochs. We distinguish between individual models and setups composed of multiple models (BEiT/PeCo train separate visual tokenizers and PeCo also distills two MoCo-v3 models).

	ViT-B	ViT-L
<i>Multiple models</i>		
BEiT (Bao et al., 2021)	83.2	85.2
PeCo (Dong et al., 2022)	84.5	86.5
<i>Single models</i>		
MoCo v3 (Chen et al., 2021b)	83.2	84.1
DINO (Caron et al., 2021)	82.8	-
MAE (He et al., 2021)	83.6	85.9
SimMIM (Xie et al., 2021)	83.8	-
iBOT (Zhou et al., 2021)	83.8	-
MaskFeat (Wei et al., 2021)	84.0	85.7
data2vec	84.2	86.6

Vision

Table 2. Speech processing: word error rate on the Librispeech test-other test set when fine-tuning pre-trained models on the Libri-light low-resource labeled data setups (Kahn et al., 2020) of 10 min, 1 hour, 10 hours, the clean 100h subset of Librispeech and the full 960h of Librispeech. Models use the 960 hours of audio from Librispeech (LS-960) as unlabeled data. We indicate the language model used during decoding (LM). Results for all dev/test sets and other LMs can be found in the supplementary material (Table 5).

	Unlabeled data	LM	Amount of labeled data				
			10m	1h	10h	100h	960h
wav2vec 2.0 (Baevski et al., 2020b)	LS-960	4-gram	15.6	11.3	9.5	8.0	6.1
HuBERT (Hsu et al., 2021)	LS-960	4-gram	15.3	11.3	9.4	8.1	-
WavLM (Chen et al., 2021a)	LS-960	4-gram	-	10.8	9.2	7.7	-
data2vec	LS-960	4-gram	12.3	9.1	8.1	6.8	5.5

Speech

Table 3. Natural language processing: GLUE results on the development set for single-task fine-tuning of individual models. For MNLI we report accuracy on both the matched and unmatched dev sets, for MRPC and QQP, we report the unweighted average of accuracy and F1, for STS-B the unweighted average of Pearson and Spearman correlation, for CoLA we report Matthews correlation and for all other tasks we report accuracy. BERT Base results are from Wu et al. (2020) and our baseline is RoBERTa re-trained in a similar setup as BERT. We also report results with wav2vec 2.0 style masking of spans of four BPE tokens with no unmasked tokens or random targets.

	MNLI	QNLI	RTE	MRPC	QQP	STS-B	CoLA	SST	Avg.
BERT (Devlin et al., 2019)	84.0/84.4	89.0	61.0	86.3	89.1	89.5	57.3	93.0	80.7
Baseline (Liu et al., 2019)	84.1/83.9	90.4	69.3	89.0	89.3	88.9	56.8	92.3	82.5
data2vec	83.2/83.0	90.9	67.0	90.2	89.1	87.2	62.2	91.8	82.7
+ wav2vec 2.0 masking	82.8/83.4	91.1	69.9	90.0	89.0	87.7	60.3	92.4	82.9

NLP

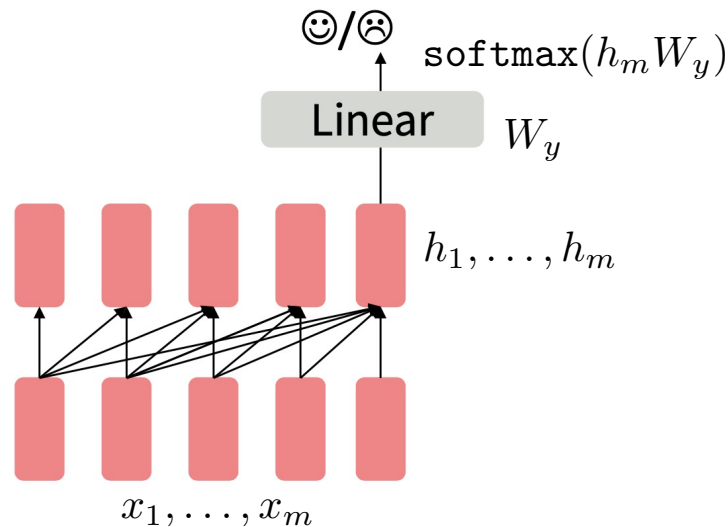
- **Overview of Generation-based Approaches**

- There have been a long attempts to learn representation Z from data X
- To this end, many classic ML literature designed a probabilistic model $p(X, Z)$
 - They are called as generative models with latent variables
- **Classic approaches** (before contrastive learning, 2020)
 - We introduce some notable classic methods
 - Context encoder, a CNN version of masked autoencoder
 - Deep InfoMax and BigBiGAN, which were SOTA of then
- Recent methods can be categorized into **2 groups**:
 - **BERT-like approach** (or masked autoencoder)
 - Predict original X from perturbed \tilde{X} (learn $\tilde{X} \rightarrow Z \rightarrow X$ encoder)
 - **GPT-like approach** (or sequential prediction)
 - Predict future state X_{t+1} from past states $X_{1:t}$ (learn $X_{1:t} \rightarrow X_t$ decoder)

- **GPT** [Radford et al., 2018]

$$\arg \max_{\theta} \log p(\mathbf{x}) = \sum_n p_{\theta}(x_n | x_1, \dots, x_{n-1})$$

- **Pre-training** by language modeling over 7000 unique books (**unlabeled data**)
 - Contains long spans of contiguous text, for learning long-distance dependencies
- **Fine-tuning** by training a classifier with target task-specific **labeled data**
 - Classifier is added on the final transformer block's last word's hidden state



- **iGPT** [Chen et al., 2020]
 - **Task:** Auto-regressively predict pixels, without incorporating 2D structure of image

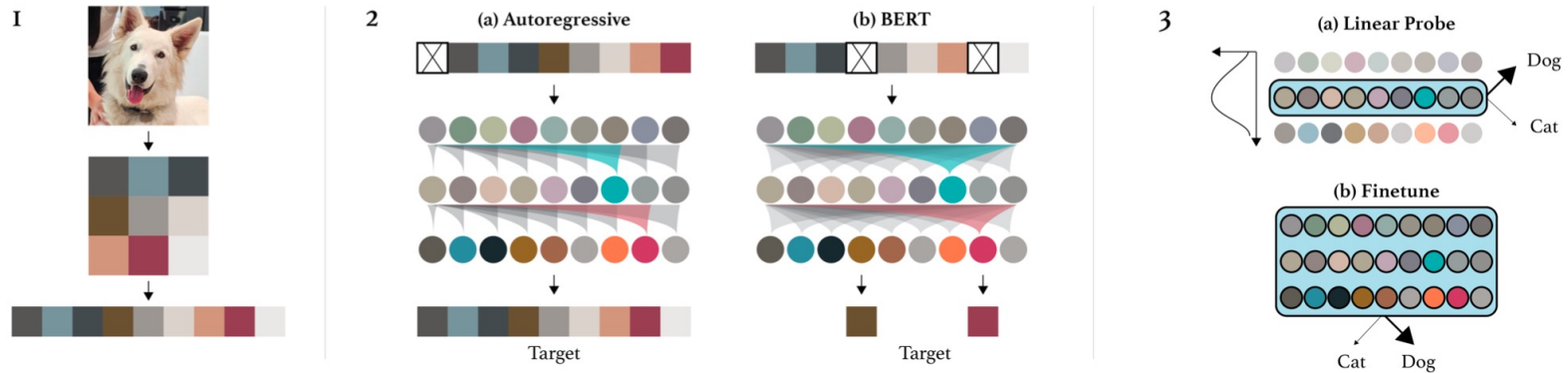


Figure 1. An overview of our approach. First, we pre-process raw images by resizing to a low resolution and reshaping into a 1D sequence. We then chose one of two pre-training objectives, auto-regressive next pixel prediction or masked pixel prediction. Finally, we evaluate the representations learned by these objectives with linear probes or fine-tuning.

- Similar to NLP domain, iGPT considers two **pre-training** objectives:
 - Auto-regressive modeling (like GPT)
 - BERT objective
- When **fine-tuning**, iGPT average pool all tokens in a sequence and use it as a feature vector, then learn a projection layer

- **iGPT** [Chen et al., 2020]

- **Task:** Auto-regressively predict pixels, without incorporating 2D structure of image

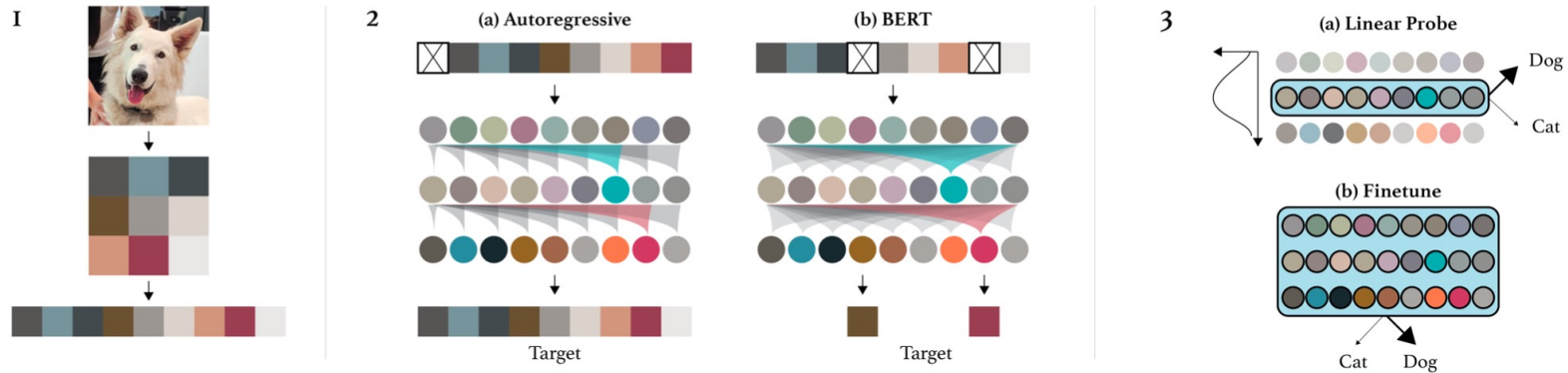
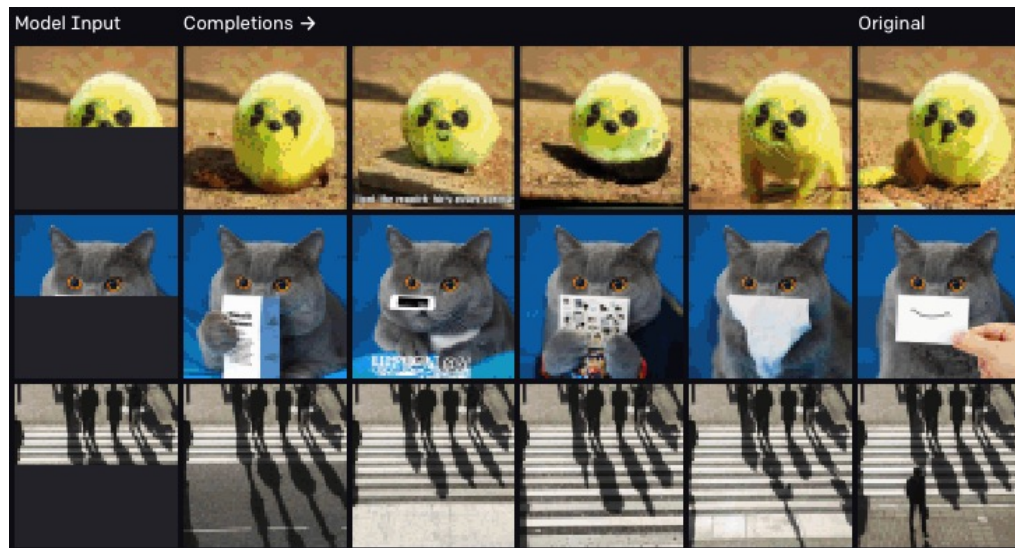


Figure 1. An overview of our approach. First, we pre-process raw images by resizing to a low resolution and reshaping into a 1D sequence. We then chose one of two pre-training objectives, auto-regressive next pixel prediction or masked pixel prediction. Finally, we evaluate the representations learned by these objectives with linear probes or fine-tuning.

- **Input data format: 9-bit color palette**
 - iGPT down-samples an image into one of 32×32 , 48×48 , or 64×64 RGB data
 - iGPT clusters all (R, G, B) values in training dataset using k-means with $k=512$, which is 9-bit color palette
 - It further **reduces input sequence length** 3 times
 - It also **discretizes** the input data and output target

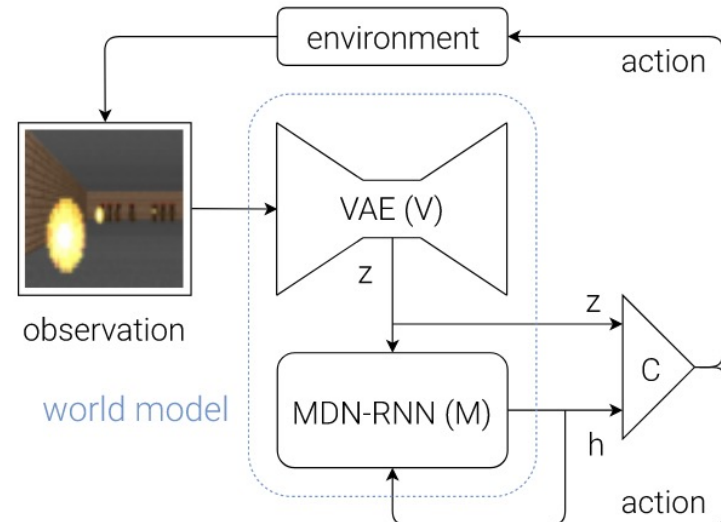
- **iGPT** [Chen et al., 2020]
 - **Task:** Auto-regressively predict pixels, without incorporating 2D structure of image
 - iGPT is not only successful for (conditional) image **generation**, but also show notable **representation learning** performance (Comparable with SimCLR)



Model	Acc	Unsup Transfer	Sup Transfer
CIFAR-10			
ResNet-152	94		✓
SimCLR	95.3	✓	
iGPT-L	96.3	✓	
CIFAR-100			
ResNet-152	78.0		✓
SimCLR	80.2	✓	
iGPT-L	82.8	✓	
STL-10			
AMDIM-L	94.2	✓	
iGPT-L	95.5	✓	

• World Model

- Autoregressive modeling can be also applied for more complex domains such as **video** or **action-conditioned videos** (called “transition model”)
- Recurrent world model [Ha & Schmidhuber, 2018]:
 - **Encoder** and **decoder** that converts data X_t to representation Z_t
 - **Transition model** that predicts action-conditioned future $Z_{t+1} = f(Z_t, A_t)$
- **Objective:** Given trajectory $\{X_{1:t}, A_{1:t}\}$, the model (a) encodes them to $Z_{1:t}$, (b) predict Z_{t+1} with transition model, and (c) decode X_{t+1}
- The learned model can be utilized for **visual planning** (for both training and inference)

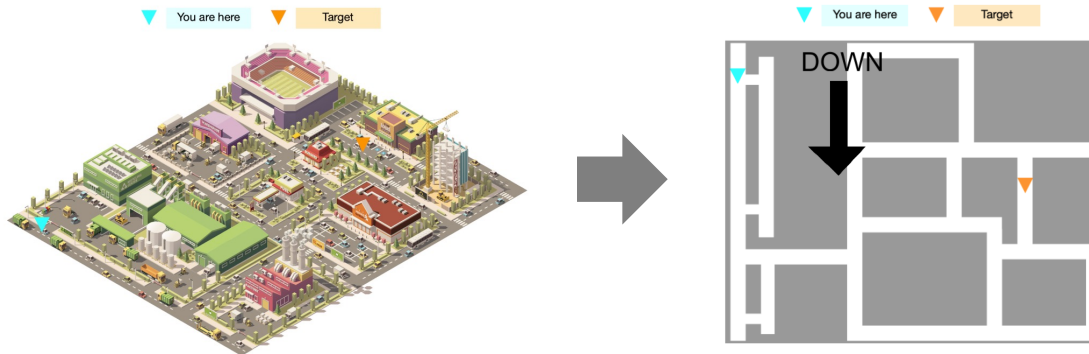


- **World Model**

- Recall that it is similar to the **CPC objective** in the SSL via Invariance section
 - **Generation:** Predict the target X_{t+1} directly
 - **Contrastive:** Find the positive X_{t+1} from negative samples X'_{t+1}
 - One can interchange them arbitrarily \Rightarrow **Q.** Which one is better?

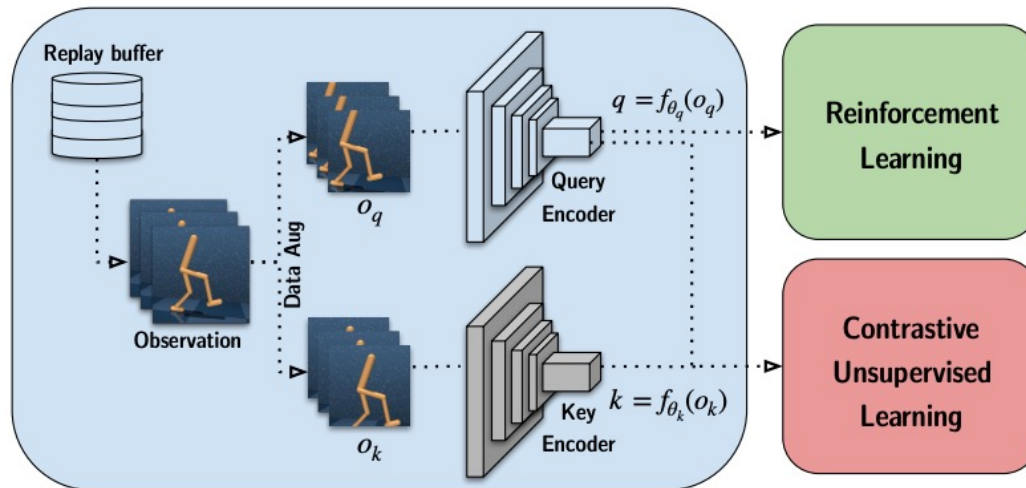
- **World Model**

- Recall that it is similar to the **CPC objective** in the SSL via Invariance section
 - **Generation:** Predict the target X_{t+1} directly
 - **Contrastive:** Find the positive X_{t+1} from negative samples X'_{t+1}
 - One can interchange them arbitrarily \Rightarrow **Q**. Which one is better?
- Contrastive structured world model (C-SWM) [Kipf et al., 2020]:
 - **Generation** objective distracts the model by focusing on **low-level** styles
 - **Contrastive** objective more focus on **high-level** semantics
- Learning a proper **invariance** is also essential for planning!
- Contrastive learning (Z projects low-level styles from X) can be beneficial



• Representation for Visual Planning

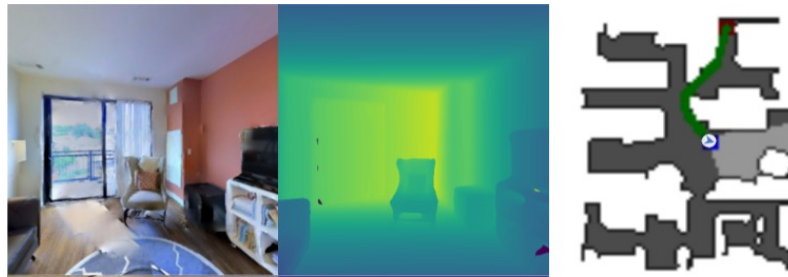
- Inspired the success of **SSL on visual domain**, some works attempted to leverage the techniques (e.g., MoCo, MAE) for learning **world models**



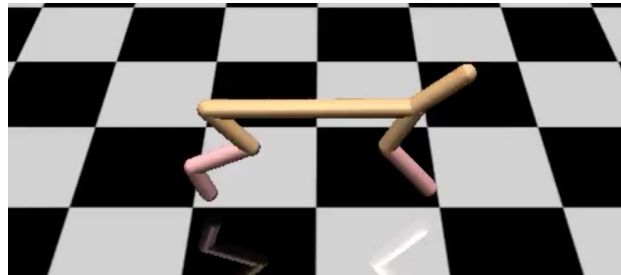
- Recent works found that transferring the **SSL models from image/video datasets** (e.g., ImageNet pre-trained MoCo) is also effective for visual planning tasks
 - PVR¹ and MVP² uses **frozen visual backbone** (MoCo and MAE) to extract representation, and apply IL/RL techniques upon the representation
 - APV³ **fine-tunes video models** to learn action-conditional world models

- **Representation for Visual Planning**

- PVR¹ and MVP² uses **frozen visual backbone** (MoCo and MAE) to extract representation, and apply IL/RL techniques upon the representation
 - Similar to the vision tasks (e.g., semantic vs. dense representation tasks), the **appropriate backbone** depends on the **control task**
 - **Navigation** (e.g., Habitat) → semantic representation

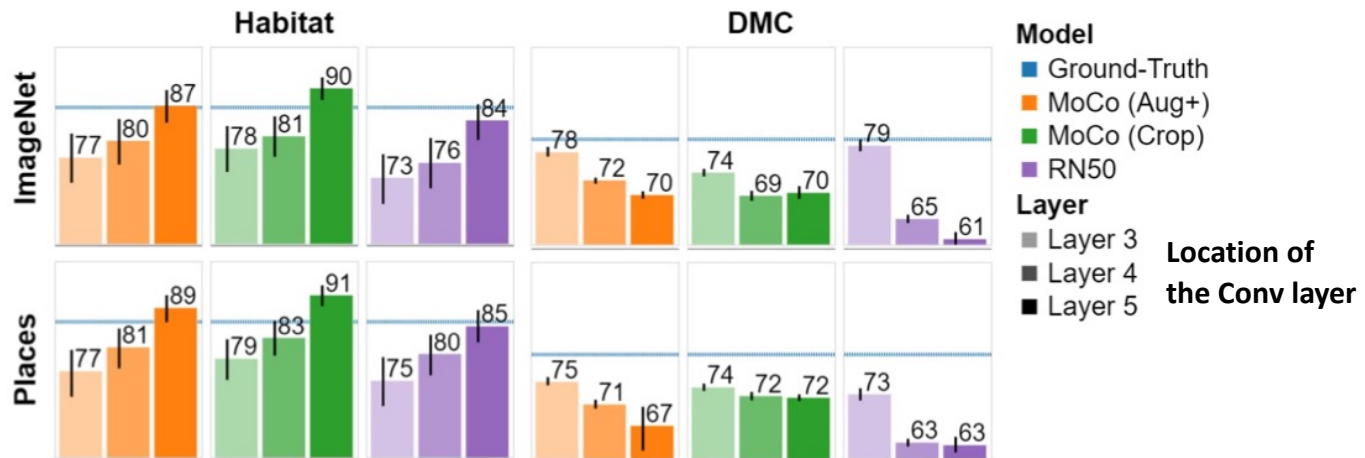


- **Low-level control** (e.g., MuJoCo/DMC) → dense representation



• Representation for Visual Planning

- PVR¹ and MVP² uses **frozen visual backbone** (MoCo and MAE) to extract representation, and apply IL/RL techniques upon the representation
 - Similar to the vision tasks (e.g., semantic vs. dense representation tasks), the **appropriate backbone** depends on the **control task**
 - **Navigation** (e.g., Habitat) → semantic representation
 - **Low-level control** (e.g., MuJoCo/DMC) → dense representation
- Intuitively, **MoCo** is more effective for the **navigation** task
 - Also, using Conv in late layers is more beneficial for navigation



- We discussed **3 types** of self-supervised learning
 1. **Pretext task:** Maximize MI of representation and pretext label
 2. **Invariance:** Maximize MI of representations of positive samples
 3. **Generation:** Maximize MI of representation and (perturbed) data
- (3) **Generation-based** approach is currently the most **promising direction**
 - BERT/MAE for encoder, and GPT for encoder-decoder models
 - Large-scale & multimodal foundation models are being stronger!
- (2) **invariance-based** method is still effective at learning **semantic tasks**
 - Leverage the additional prior knowledge of positive samples
 - Thus, one may need to choose an appropriate backbone for the task
- Self-supervised learning have shown its effectiveness on **various domains**
 - Image, video, language, audio, graph, tabular, and **multimodal** domains
 - Recent works discover that visual SSL is also effective for the **planning** tasks

SSL via Pretext Tasks

[Doersch et al., 2015] Unsupervised Visual Representation Learning by Context Prediction, ICCV 2015

[Noroozi & Favaro, 2016] Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles, ECCV 2016

[Kim et al., 2019] Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles, AAAI 2019

[Zhang et al., 2016] Colorful Image Colorization, ECCV 2016

[Gidaris et al., 2018] Unsupervised Representation Learning by Predicting Image Rotations, ICLR 2018

SSL via Invariance (and Contrast)

- [Caron et al., 2018] Deep Clustering for Unsupervised Learning of Visual Features, ECCV 2018
- [Wu et al., 2018] Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination, CVPR 2018
- [He et al., 2020] Momentum Contrast for Unsupervised Visual Representation Learning, CVPR 2020
- [Chen et al., 2020] A Simple Framework for Contrastive Learning of Visual Representations, ICML 2020
- [Grill et al., 2020] Bootstrap your own latent: A new approach to self-supervised Learning, NeurIPS 2020
- [Caron et al., 2021] Emerging Properties in Self-Supervised Vision Transformers, ICCV 2021
- [Meng et al., 2021] COCO-LM: Correcting and Contrasting Text Sequences for Language Model Pretraining, 2021
- [You et al., 2020] Graph Contrastive Learning with Augmentations, NeurIPS 2020
- [Hassani & Khasahmadi, 2020] Contrastive Multi-View Representation Learning on Graphs, ICML 2020
- [Lee et al., 2021] i-Mix: A Domain-Agnostic Strategy for Contrastive Representation Learning, ICLR 2021
- [Tian et al., 2020] Contrastive Multiview Coding, ECCV 2020
- [Radford et al., 2021] Learning Transferable Visual Models From Natural Language Supervision, ICML 2021
- [Akbari et al., 2021] VATT: Transformers for Multimodal Self-Supervised Learning from Raw Video, Audio and Text, NeurIPS 2021
- [Oord et al., 2018] Representation Learning with Contrastive Predictive Coding, 2018
- [Gordon et al., 2019] Watching the World Go By: Representation Learning from Unlabeled Videos, 2019
- [Xiong et al., 2021] Self-Supervised Representation Learning from Flow Equivariance, ICCV 2021
- [Huang et al., 2021] Self-supervised Video Representation Learning by Context and Motion Decoupling, CVPR 2021

SSL via Generation

- [Pathak et al., 2016] Context Encoders: Feature Learning by Inpainting, CVPR 2016
- [Hjelm et al., 2019] Learning deep representations by mutual information estimation and maximization, ICLR 2019
- [Donahue et al., 2019] Large Scale Adversarial Representation Learning, NeurIPS 2019
- [Hjelm et al., 2019] Learning deep representations by mutual information estimation and maximization, ICLR 2019
- [Devlin et al., 2019] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL 2019
- [Joshi et al., 2019] SpanBERT: Improving Pre-training by Representing and Predicting Spans, TACL 2019
- [Clark et al., 2020] ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, ICLR 2020
- [Hu et al., 2020] Strategies for Pre-training Graph Neural Networks, ICLR 2020
- [Rong et al., 2020] Self-Supervised Graph Transformer on Large-Scale Molecular Data, NeurIPS 2020
- [Bao et al., 2022] BEiT: BERT Pre-Training of Image Transformers, ICLR 2022
- [He et al., 2022] Masked Autoencoders Are Scalable Vision Learners, CVPR 2022
- [Baevski et al., 2022] data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language, 2022
- [Radford et al., 2018] Language Models are Unsupervised Multitask Learners, 2018
- [Chen et al., 2020] Generative Pretraining from Pixels, ICML 2020

SSL for Visual Planning

[Ha & Schmidhuber, 2018] Recurrent World Models Facilitate Policy Evolution, NeurIPS 2018

[Kipf et al., 2020] Contrastive Learning of Structured World Models, ICLR 2020

[Srinivas et al., 2020] CURL: Contrastive Unsupervised Representations for Reinforcement Learning, ICML 2020

[Stooke et al., 2021] Decoupling Representation Learning from Reinforcement Learning, ICML 2021

[Parisi et al., 2022] The Unsurprising Effectiveness of Pre-Trained Vision Models for Control, 2022

[Xiao et al., 2022] Masked Visual Pre-training for Motor Control, 2022

[Seo et al., 2022] Reinforcement Learning with Action-Free Pre-Training from Videos, 2022