Self-supervised Learning

AI602: Recent Advances in Deep Learning

Lecture 3

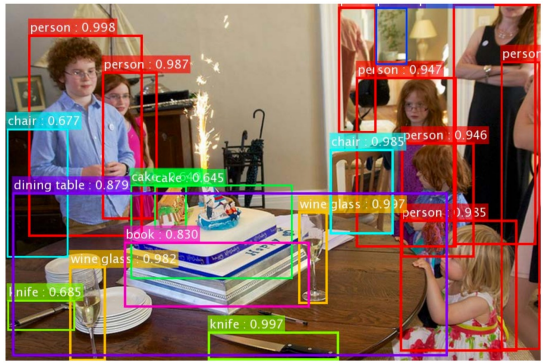
KAIST AI

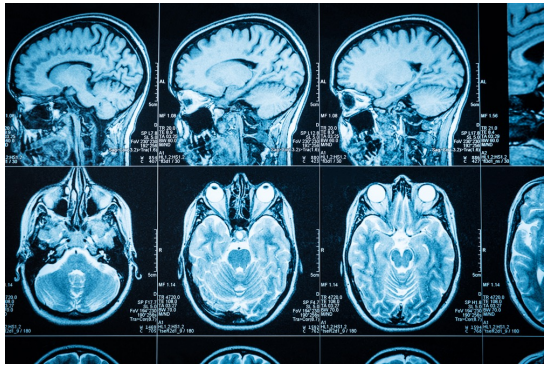
Table of Contents

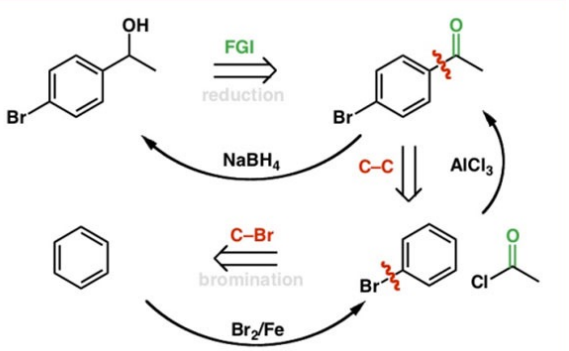
Table of Contents

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



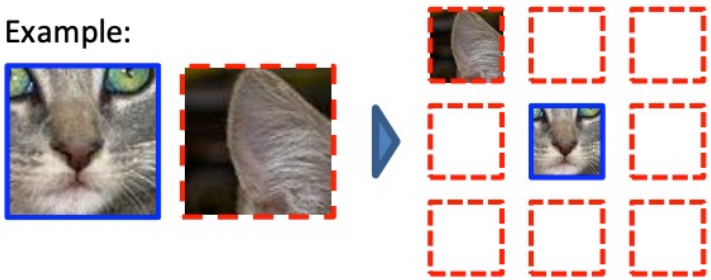


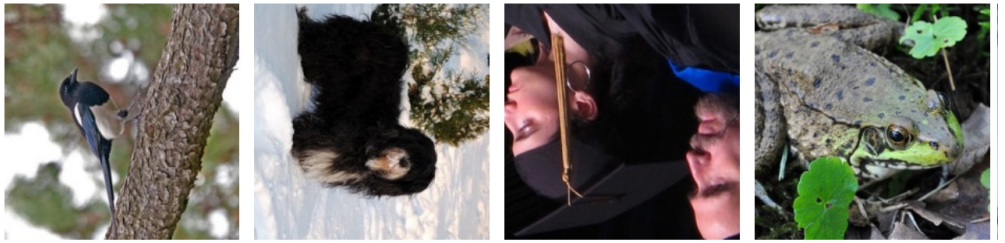


- But, **collecting unlabeled samples is extremely easy** compared to annotation

- **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 patches[1] or rotation degree[2]

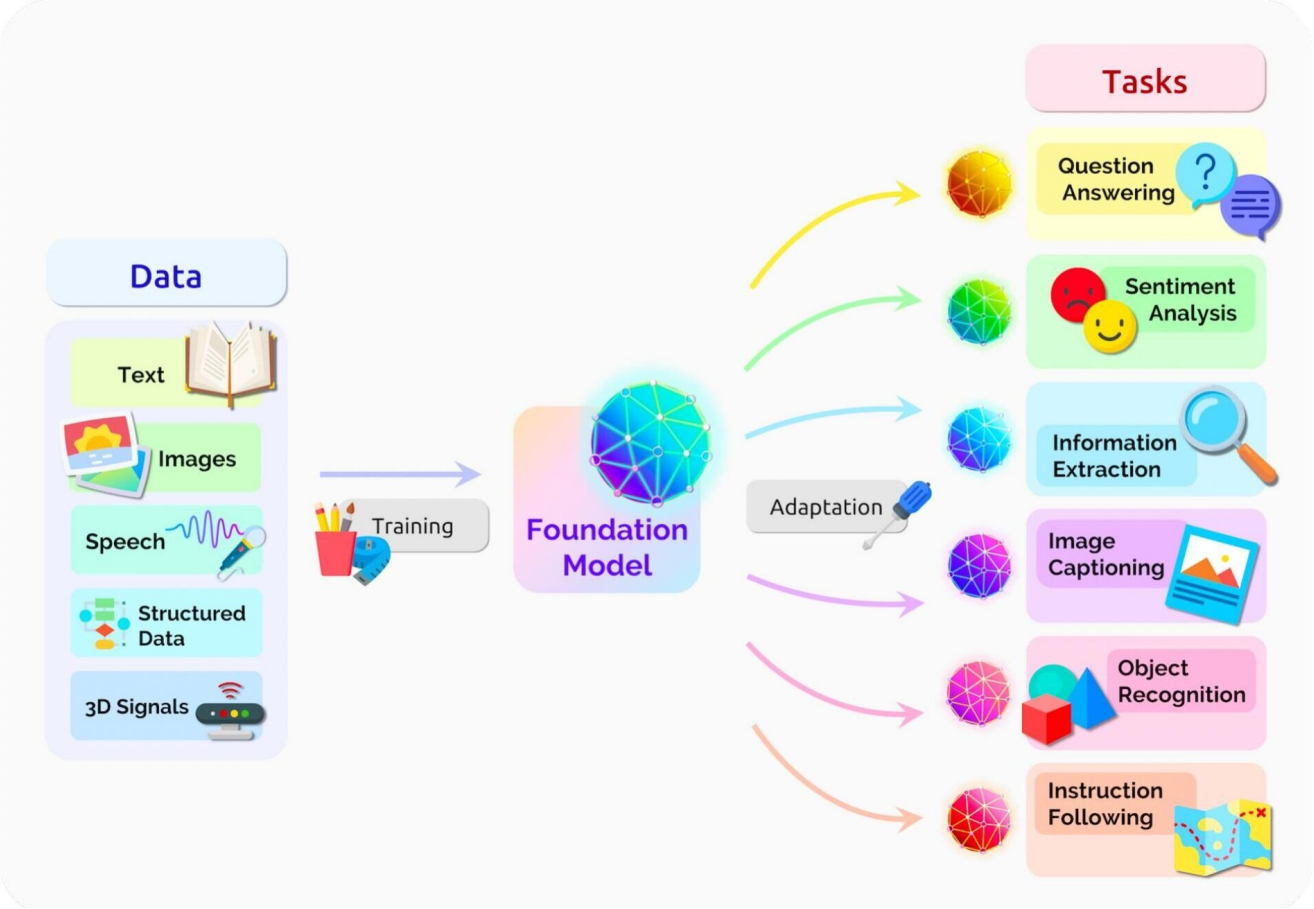




- What can we learn from self-supervised learning?
  - To predict (well-designed) self-supervision, one might **require high-level understanding of inputs**
  - E.g., we should know ⬚ is the right ear of the cat for predicting locations
  - Thus, **high-level representations could be learned w/o human-annotation**

* source : [1][Doersch et al., 2015], [2][Gidaris et al., 2018]

- **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



* source : [Bommasani et al., 2021]

Unsupervised Learning vs. Self-supervised Learning

> *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)

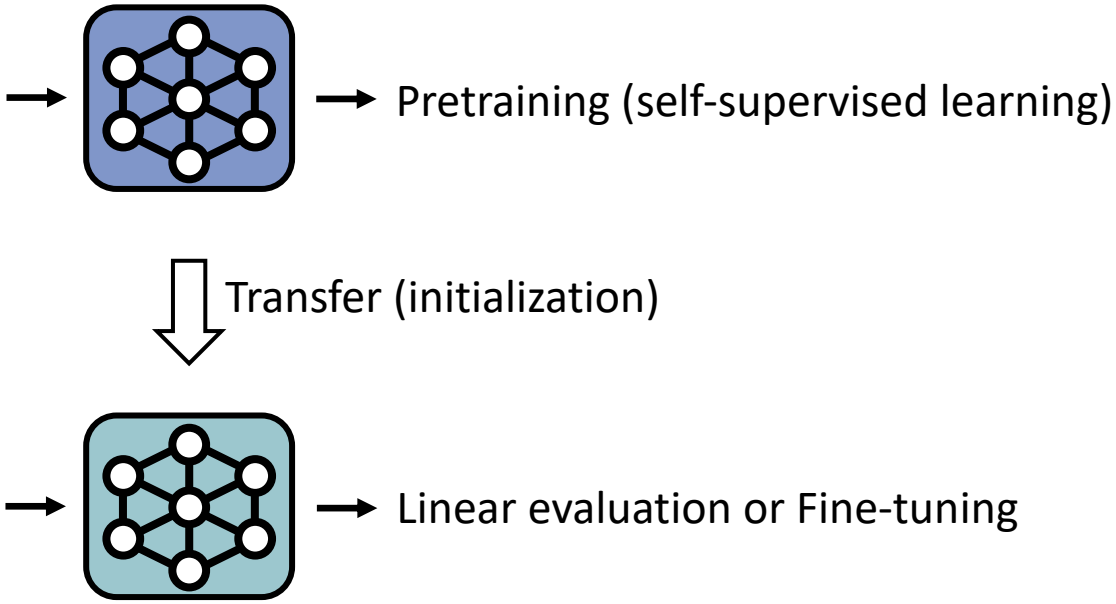
- How to define "unsupervised learning" term? (there is no answer …)
  - **Q)** We need an objective (or loss) for learning; is the objective not a (self-)supervision?
  - **Q)** Unsupervised learning ⊇ self-supervised learning?
  - **Q)** What are the purely unsupervised learning methods?
    - In classic 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)

Flowers102 (2k images)



Overview of the Lecture

- For the first part, we mainly follow the history of **"SSL for images"**
  - **2019-2021:** Contrastive Learning
    - NPID ('18), MoCo/SimCLR ('20), BYOL ('20), MoCov3/DINO ('21)
    - Similar idea was also considered in Exemplar-CNN ('14)
  - **2022-2023:** Masked Image Modeling
    - BEiT ('22), MAE ('22), data2vec ('22)
    - Similar idea was also considered in Context Encoder ('16)

- Then, we focus on the recent development of **Vision & Language SSL**
  - Image-text alignment using CLIP for transferrable visual representation
  - Fused transformer for vision-language understanding
  - Learning visual representation from frozen Large Language Models (LLMs)
  - Unifying Vision-Language pretraining

Table of Contents

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 $f($  $) \approx f($  $)$

Negative pair $f($  $) \neq f($  $)$

- **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 $f($  $) \approx f($  $)$

Negative pair $f($  $) \neq f($  $)$

- **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)

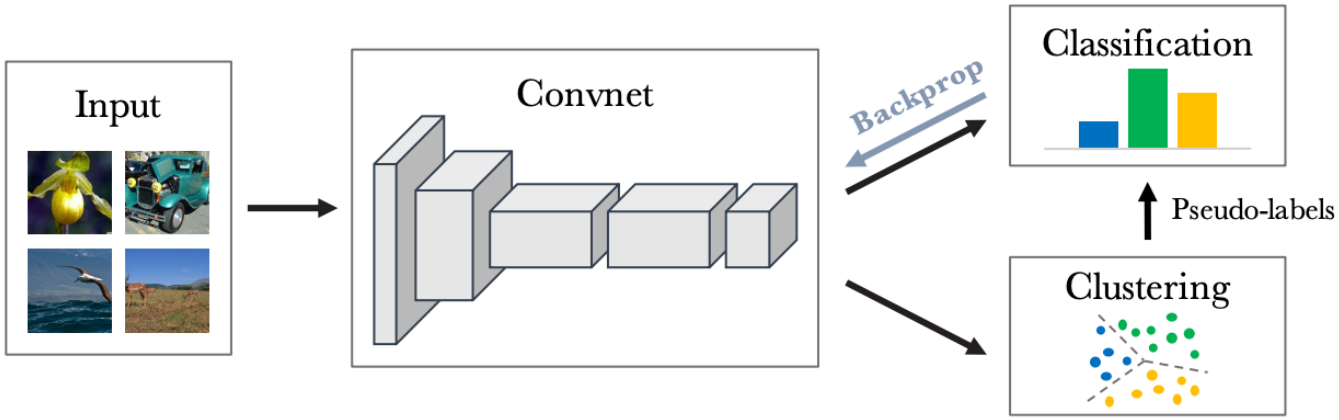
- **Instantiations of invariance-based approach**
  - Many classes of self-supervised learning can be viewed as invariance-based

  - **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

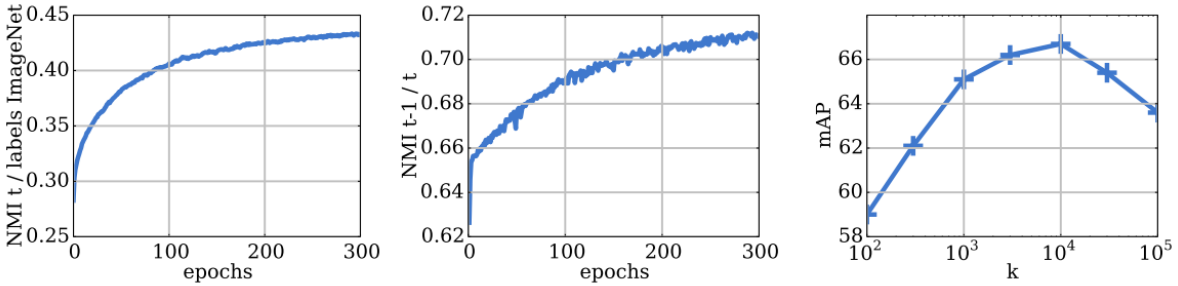


  - **Simple method:** Alternate between
    1. Clustering the features to produce pseudo-labels
    2. Updating parameters by predicting these pseudo-labels

  - How to avoid **trivial solutions**?
    - Empty cluster ⇐ feature quantization (it reassigns empty clusters)
    - Imbalanced sizes of clusters ⇐ over-sampling

* source : [Caron et al., 2018]

- **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

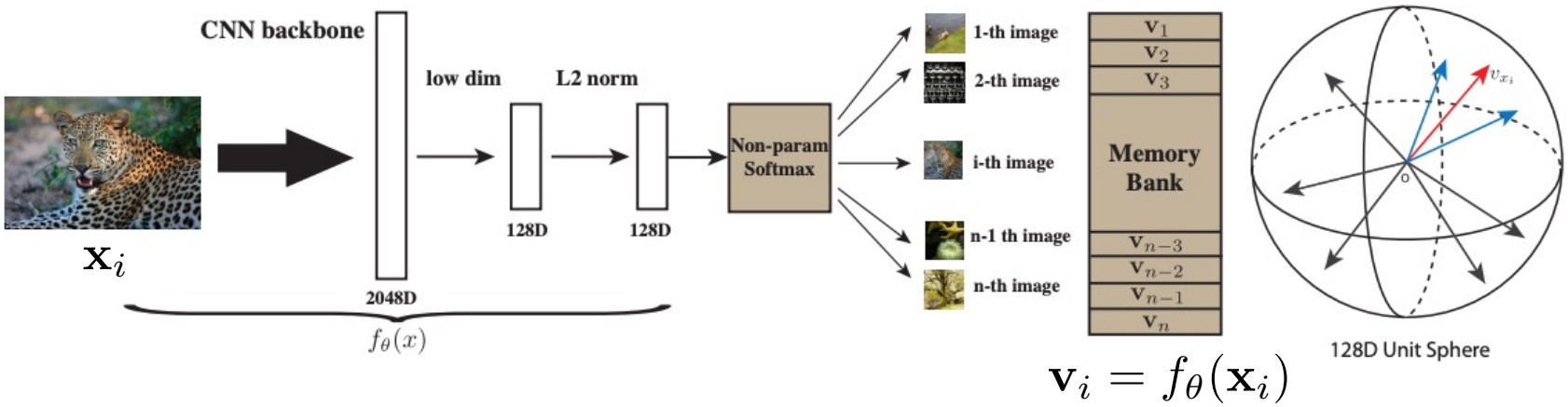


(a) Clustering quality (b) Cluster reassignment (c) Influence of k

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

* source : [Caron et al., 2018]

- **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 ⇒ $\mathbf{v}_i$ can be used directly as a class prototype

* source : [Wu et al., 2018]

- **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**

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

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

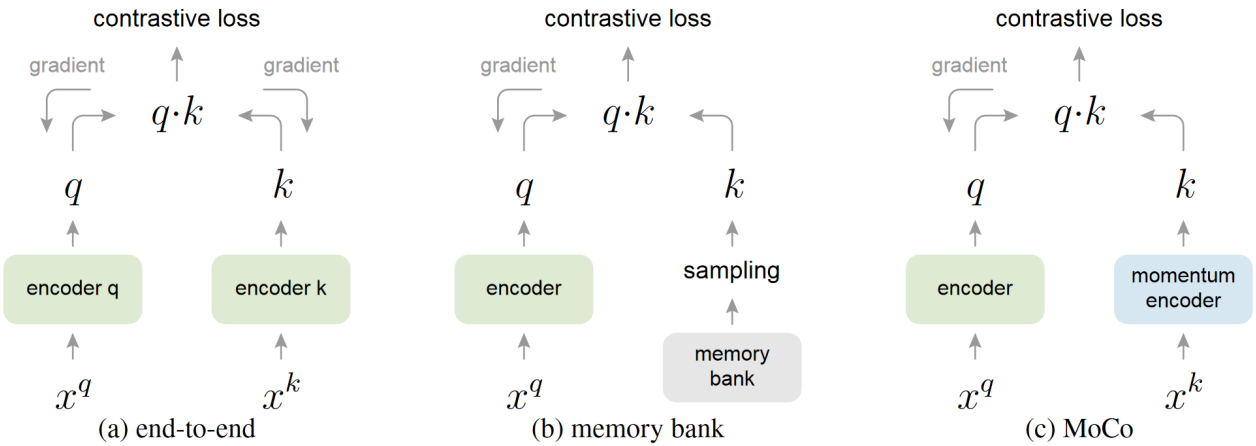
m negative samples

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

data distribution

noise distribution (uniform)

- **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**

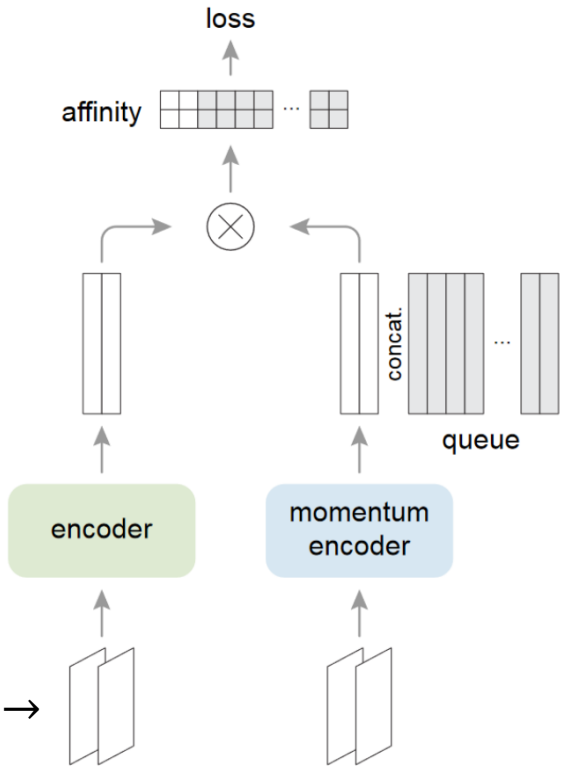


  - **Momentum encoder** increases the key representations' consistency
  - **Queue** allows us to use recent and many negative samples

* source : [He et al., 2019]

- **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

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



* source : [Chen et al., 2020]

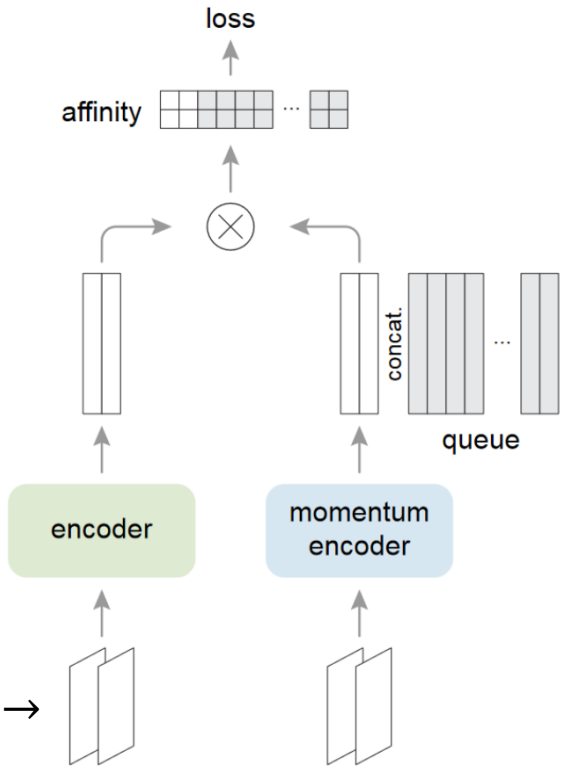
- **Momentum Contrast (MoCo)** [He et al., 2019]
  - **Key issue: the number of negatives** is very crucial in contrastive learning
  - How to resolve this issue in prior works? **Memory Bank**
    - Note: representations in the memory bank are momentum-updated
  - **MoCo's idea:** use a **momentum-updated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

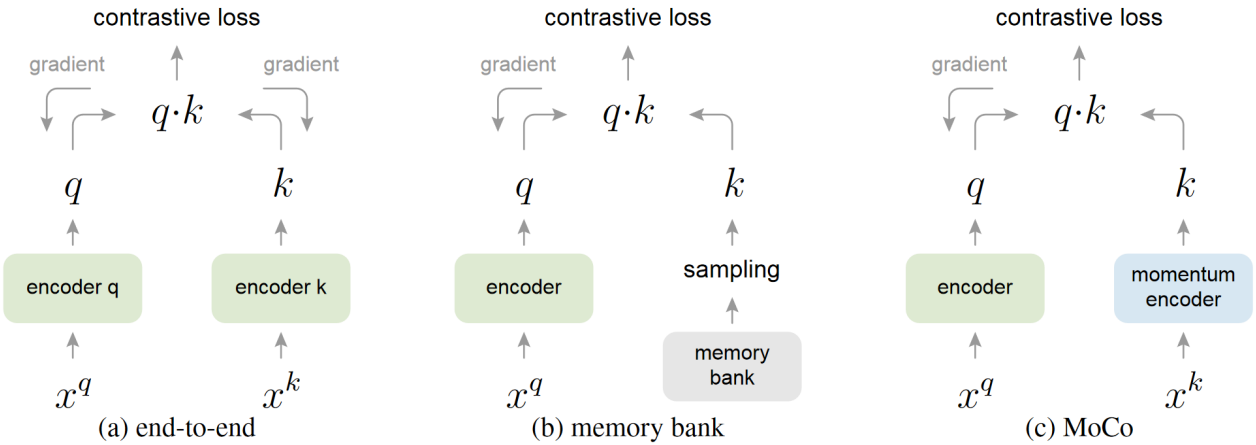
$$\theta_{\text{momentum}} \leftarrow m\theta_{\text{momentum}} + (1-m)\theta$$

    - Add the current positive keys k^+ into the queue



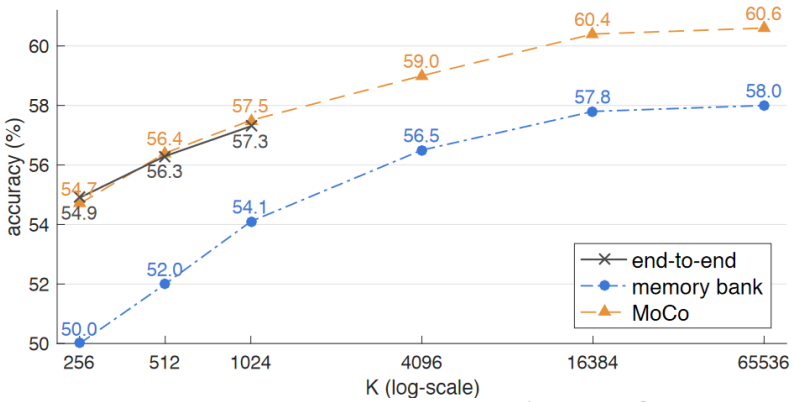
* source : [Chen et al., 2020]

- **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 (%)	*fail*	55.2	57.8	59.0	58.9



* source : [He et al., 2019]

- **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 …

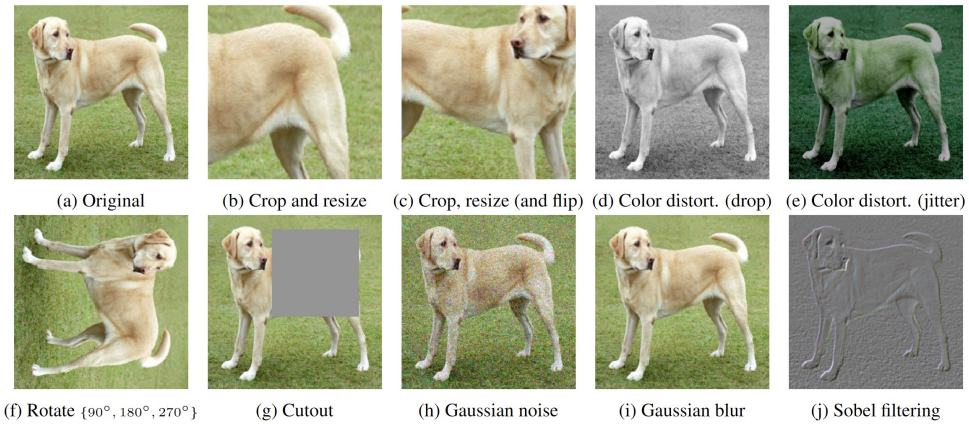
1. Strong augmentation (i.e., composition of multiple data augmentation operations)
2. A nonlinear MLP between the representation and the contrastive loss
3. Large batch sizes and longer training

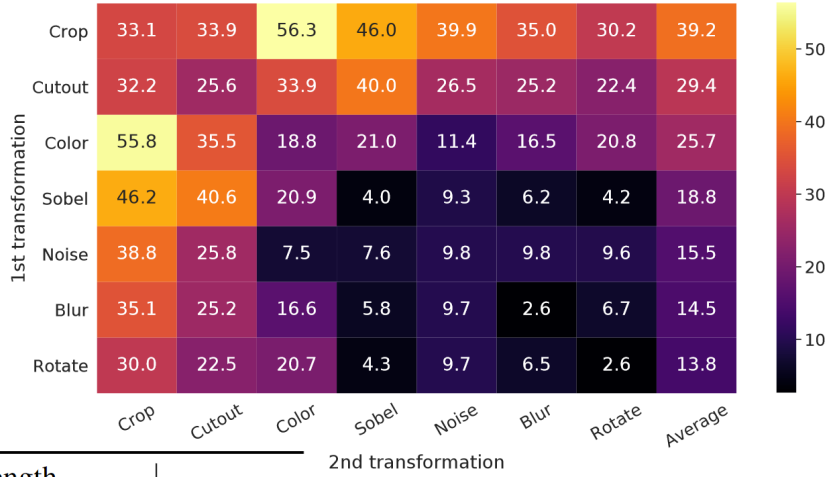
* source : https://ai.googleblog.com/2020/04/advancing-self-supervised-and-semi.html

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





Methods	Color distortion strength 1/8	1/4	1/2	1	1 (+Blur)	AutoAug
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

* source : [Chen et al., 2020]

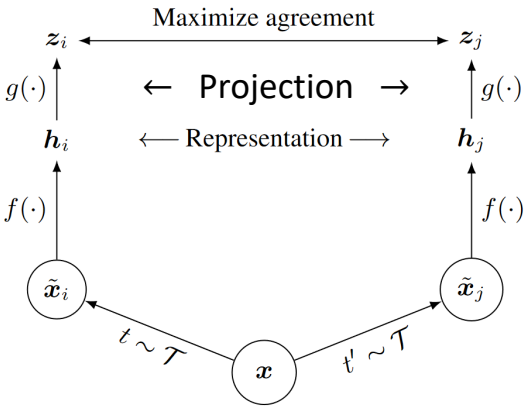
- **SimCLR** [Chen et al., 2020]
  - A **simple** framework for contrastive learning without requiring specialized architectures or a memory bank

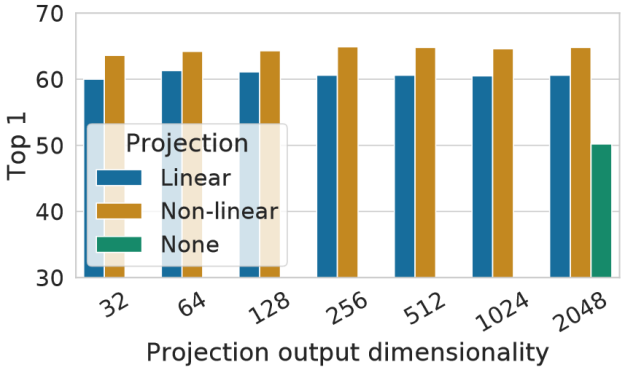
  - This paper founds that contrastive learning benefits from …

2. A nonlinear MLP between the representation and the contrastive loss
   - Contrastive learning objective learns $\mathbf{z}$ to be **invariant to augmentations**

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

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





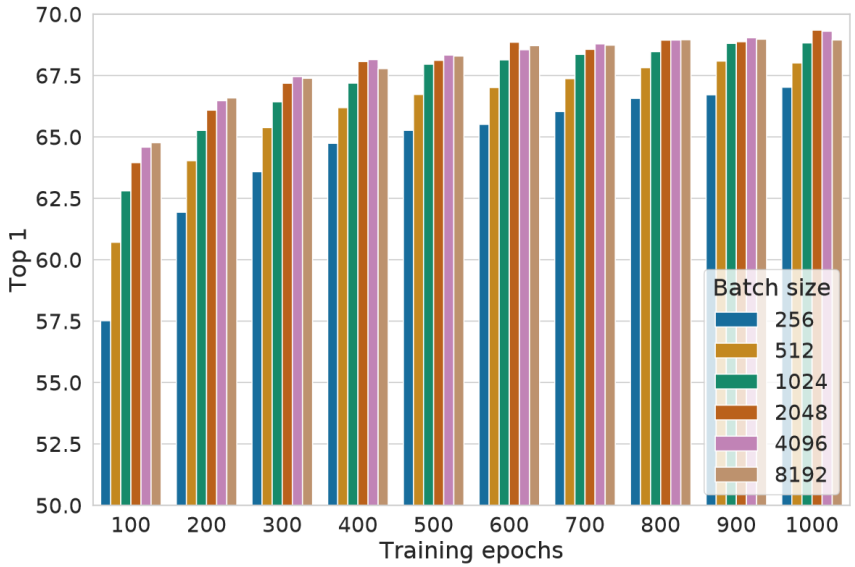
What to predict?	Random guess	Representation h	Representation $g(h)$
Color vs grayscale	80	99.3	97.4
Rotation	25	67.6	25.6
Orig. vs corrupted	50	99.5	59.6
Orig. vs Sobel filtered	50	96.6	56.3

* source : [Chen et al., 2020]

- **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 …

3. Large batch sizes and longer training



* source : [Chen et al., 2020]

- **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
Linear evaluation:												
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
Fine-tuned:												
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 (Top 5) 1%	10%
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
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
Methods using representation learning only:			
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

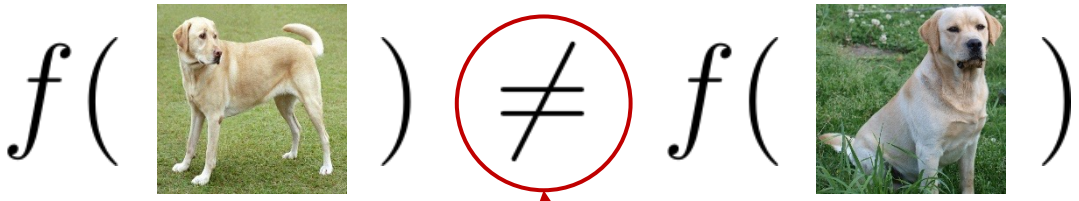
Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	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**

* source : [Chen et al., 2020]

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

Positive pair

Negative pair

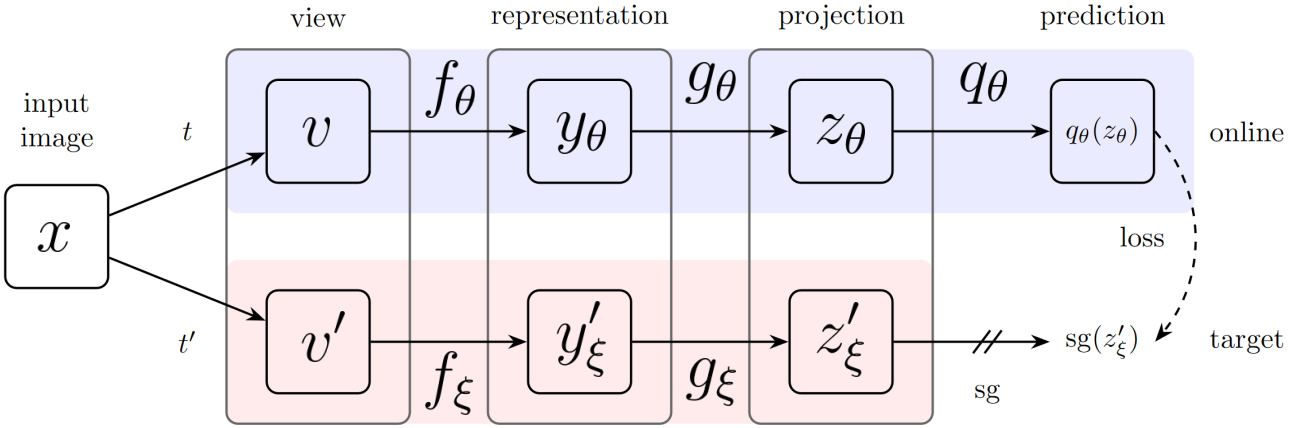


This relation might be not true

- **Q)** can we learn representations without negative samples?
- Simply minimizing $\|f(\,\,) - f(\,\,)\|$ leads to mode collapse, i.e., $\forall x, f(x) = c$

- **Next:** Positive-only approaches

- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]
  - **Idea:** directly bootstrap the representations



Objective

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

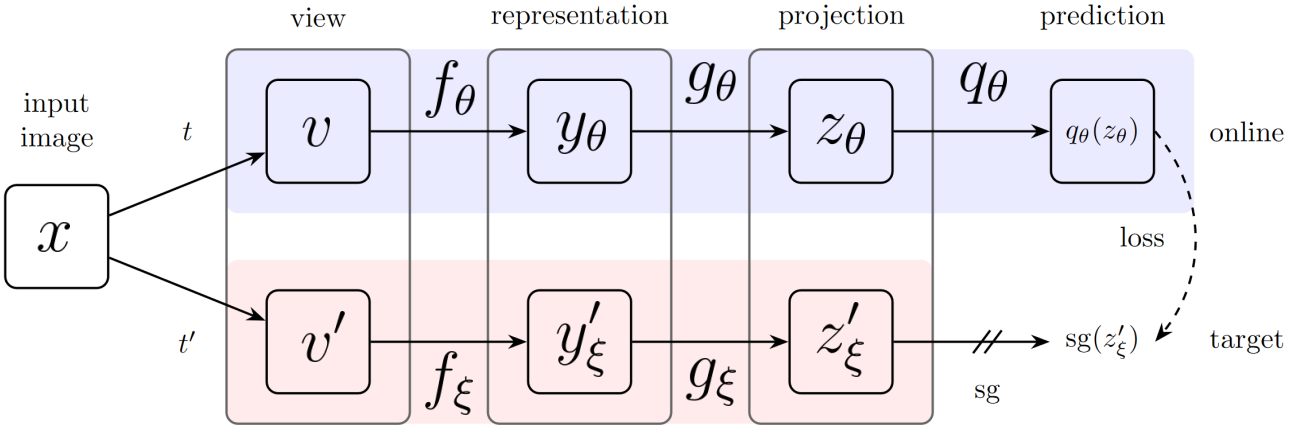
Update

$$\theta \leftarrow \text{optimizer}(\theta, \nabla_\theta \mathcal{L}_{\texttt{BYOL}})$$
$$\xi \leftarrow \tau\xi + (1-\tau)\theta$$

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

* source : [Grill et al., 2020]

- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]
  - **Idea:** directly bootstrap the representations



Objective

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

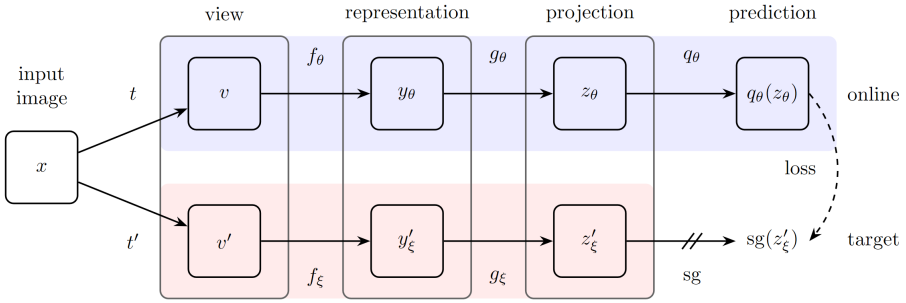
Update

$$\theta \leftarrow \text{optimizer}(\theta, \nabla_\theta \mathcal{L}_{\texttt{BYOL}})$$
$$\xi \leftarrow \tau\xi + (1-\tau)\theta$$

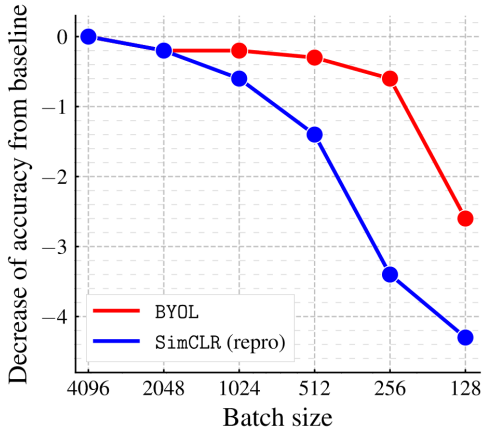
- **Q)** How does BYOL avoid the undesired collapsed solutions?
  - ξ is not updated in the direction of $\nabla_\xi \mathcal{L}_{\texttt{BYOL}}$
  - When the predictor is optimal, i.e., $q^*(z_\theta) = \mathbb{E}[z'_\xi | z_\theta]$, $\mathcal{L}_{\texttt{BYOL}} = \mathbb{E}[\sum_i \mathrm{Var}(z'_{\xi,i} | z_\theta)]$ (z'_ξ's i-th feature)
  - For any constant c, $\mathrm{Var}(z'_{\xi,i} | z_\theta) \leq \mathrm{Var}(z'_{\xi,i} | c)$ ⇒ constant equilibria is unstable

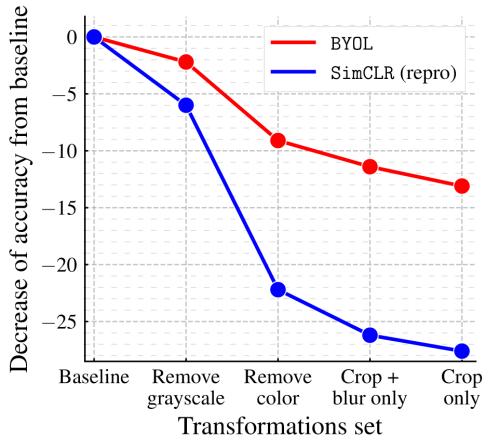
* source : [Grill et al., 2020]

- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]
  - **Idea:** directly bootstrap the representations



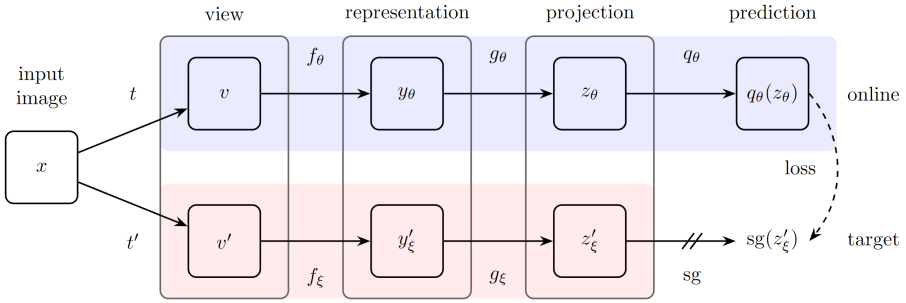
  - BYOL is **more robust** to the choice of **batch sizes** and **augmentations**



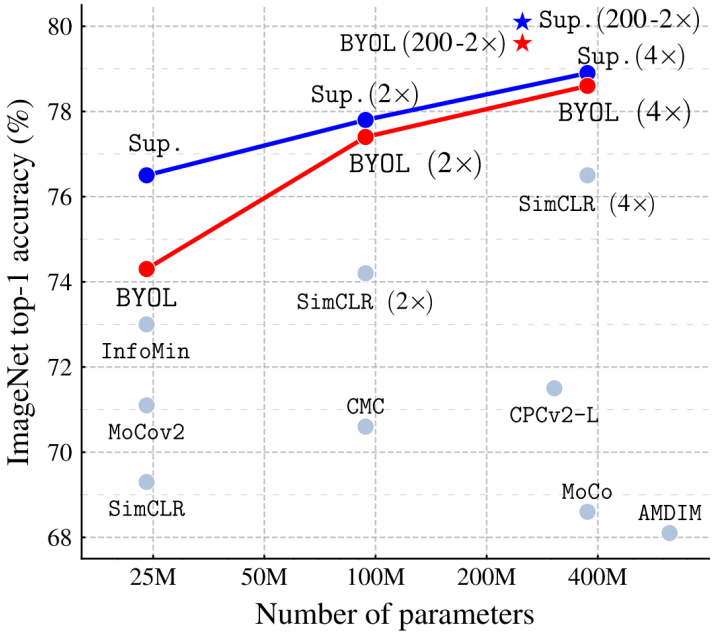


* source : [Grill et al., 2020]

- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]
  - **Idea:** directly bootstrap the representations

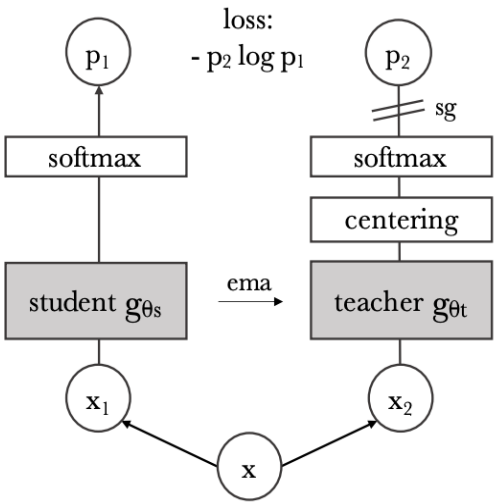


  - BYOL is **more robust** to the choice of **batch sizes** and **augmentations**
  - BYOL achieves 74.3% linear evaluation accuracy; supervised learning does 76.5%



* source : [Grill et al., 2020]

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



Objective

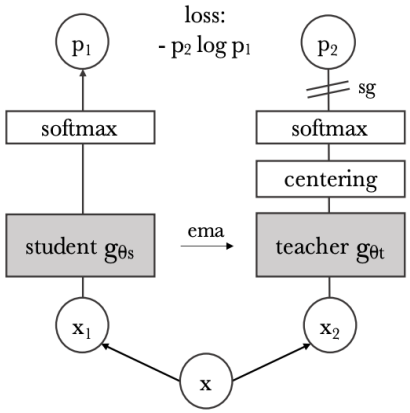
$$\mathcal{L}_{DINO} = H(P_t(x), P_s(x))$$

Update

$$\theta_s \leftarrow optimizer\left(\theta_s, \nabla_{\theta_s}\mathcal{L}_{DINO}\right)$$
$$\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$$

- **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

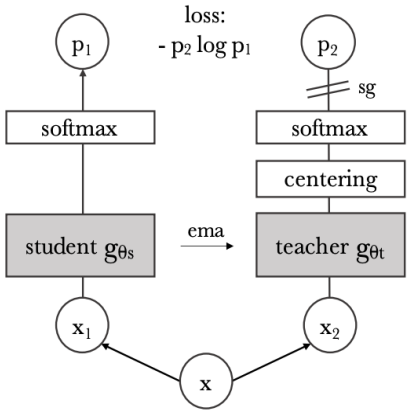


  - DINO constructs a set of views V via **multi-crop** strategy:
    - (1) global views: x_1^g, x_2^g
    - (2) local views with smaller resolution

  - 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: subtracting 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

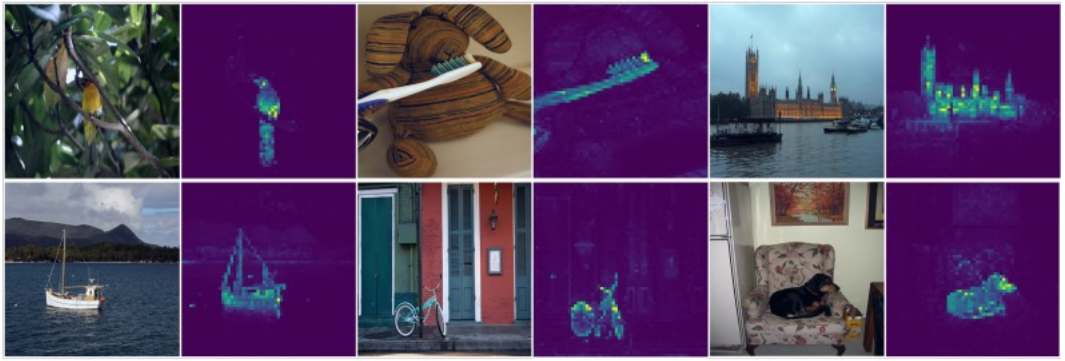
$$c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i)$$

    - 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.	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
Supervised					
ImageNet	INet	ViT-S/8	66.0	63.9	68.1
STM [48]	I/D/Y	RN50	81.8	79.2	84.3
Self-supervised					
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)

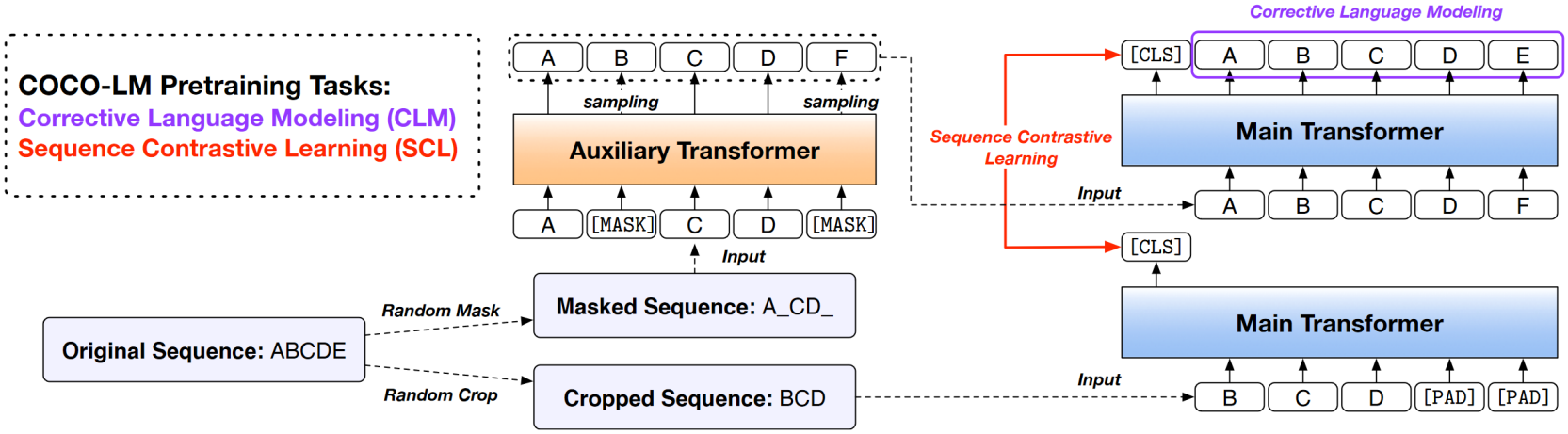
  - **Same data with different augmentation**
    - Discussed image domain before (e.g., SimCLR)
    - How about other domains (e.g., language, graph, or domain-agnostic)?

  - **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)

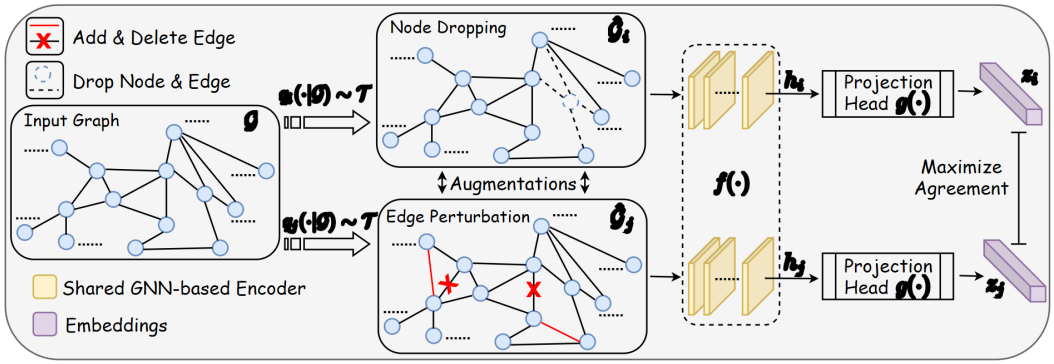
- **COCO-LM** [Meng et al., 2021]
  - **Idea:**
    - **Corrective Language Modeling**: Recover original tokens from corrupted ones
    - **Sequence Contrastive Learning** between corrupted and augmented sentences



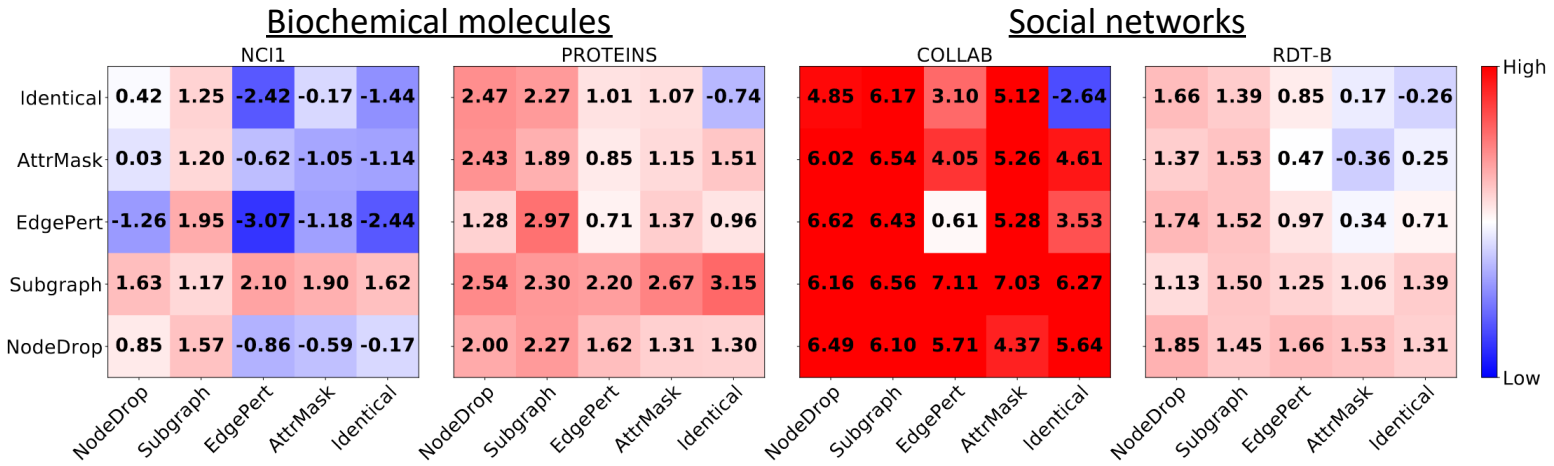
  - Both CLM and SCL improves Baseline
    - Improvements are observed on different tasks, e.g., CLM: CoLA, SCL: RTE (CoLA: grammatical validity of one sentence, RTE: relation of two sentences)

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



- **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 $v_{i,j \neq i} = 1$
    - With virtual labels, we can re-write a general contrastive loss: $\ell(x_i, v_i)$
    - Then, i-Mix loss is defined as:

$$\ell^{\text{i-Mix}}\big((x_i, v_i), (x_j, v_j); \mathcal{B}, \lambda\big) = \ell(\mathbf{Mix}(x_i, x_j; \lambda), \lambda v_i + (1-\lambda) v_j; \mathcal{B})$$

      - 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

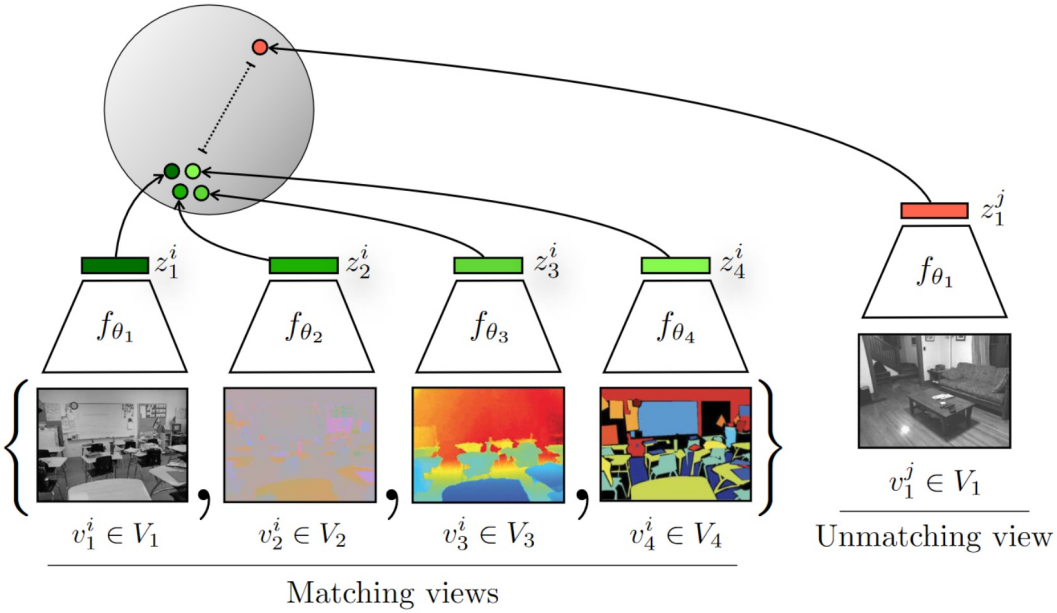
  - i-Mix consistently improves the classification accuracy on different domains

Domain	Dataset	N-pair	+ i-Mix	MoCo v2	+ i-Mix	BYOL	+ 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., video)
  - **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



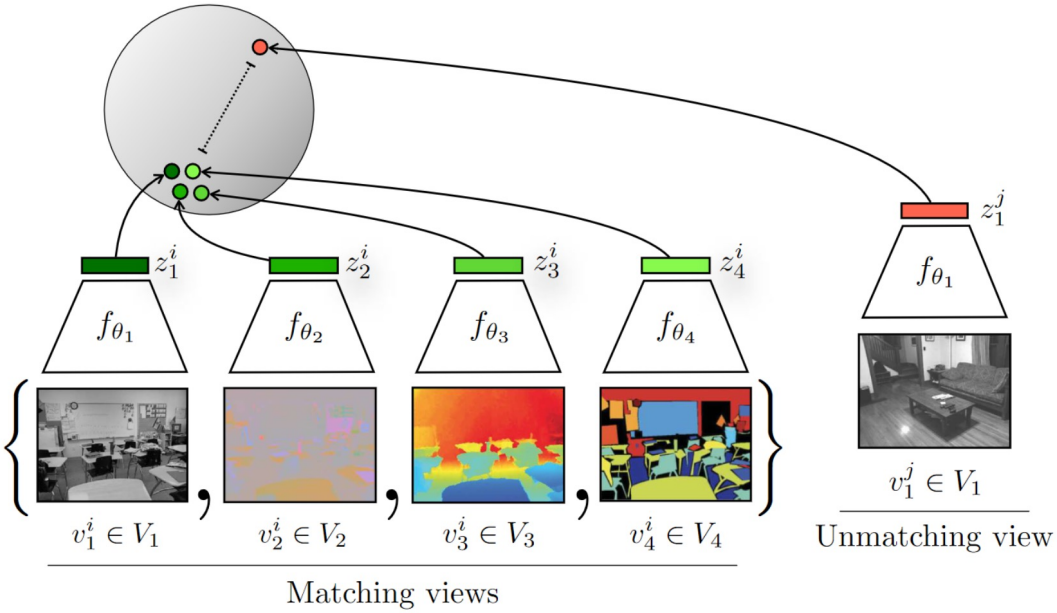
$$\mathcal{L}_{\texttt{contrast}}^{V_1,V_2} = -\mathbb{E}_{\{v_1^1, v_2^1, \ldots, v_2^{k+1}\}} \left[\log \frac{h_\theta(\{v_1^1, v_2^1\})}{\sum_{j=1}^{k+1} h_\theta(\{v_1^1, v_2^j\})} \right]$$

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

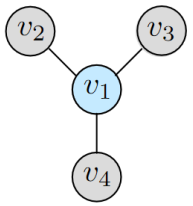
Neural network

* source : [Tian et al., 2019]

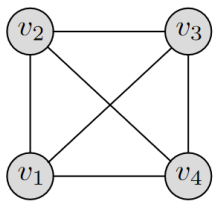
- **Contrastive Multiview Coding (CMC)** [Tian et al., 2019]
  - **Idea:** Use multiple views of the same instance as positive samples



- By minimizing $\mathcal{L}(V_1, V_2) = \mathcal{L}_{\texttt{contrast}}^{V_1,V_2} + \mathcal{L}_{\texttt{contrast}}^{V_2,V_1}$, $f_{\theta_1}(\cdot), f_{\theta_2}(\cdot)$ 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 $\mathcal{L} = \sum_{1 \le i < j \le M} \mathcal{L}(V_i, V_j)$



Core-view

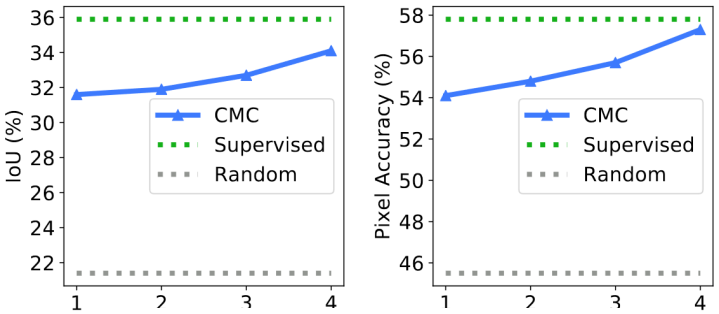


Full-graph

* source : [Tian et al., 2019]

- **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

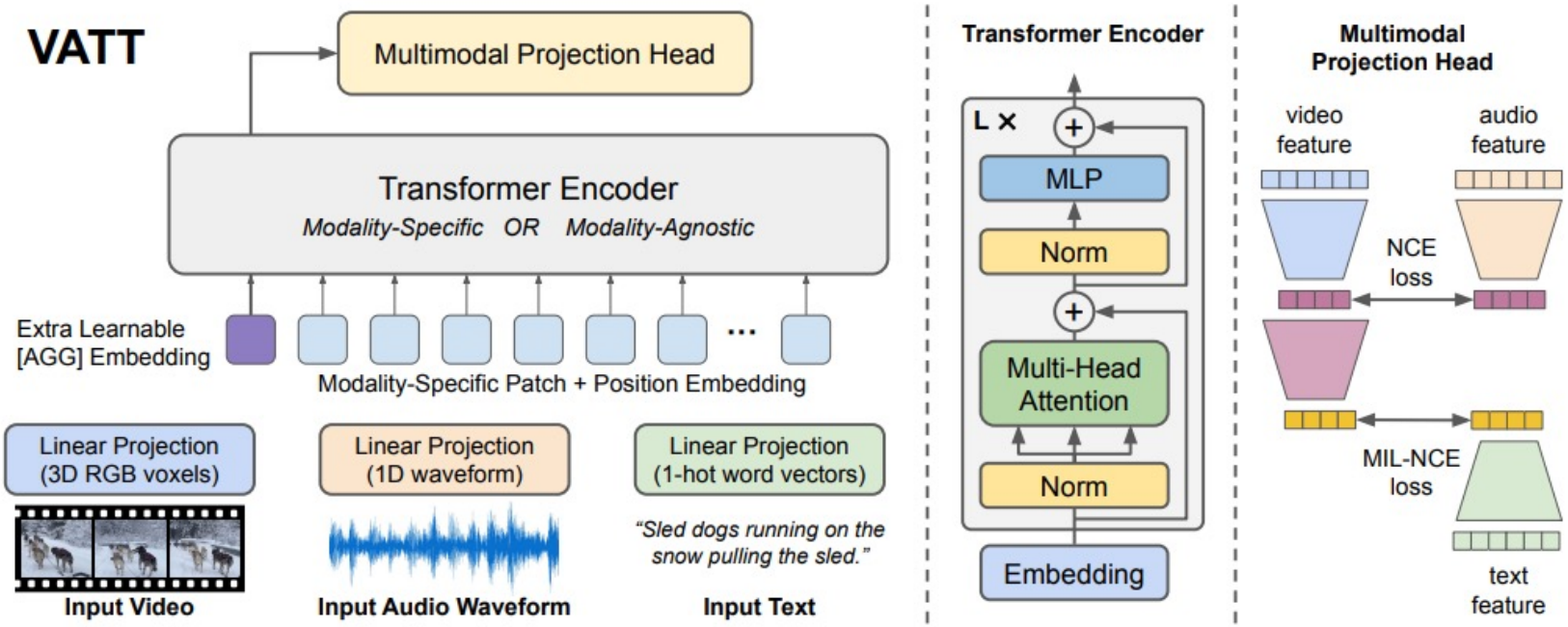


Core-view vs Full-graph

	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**

* source : [Tian et al., 2019]

- **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

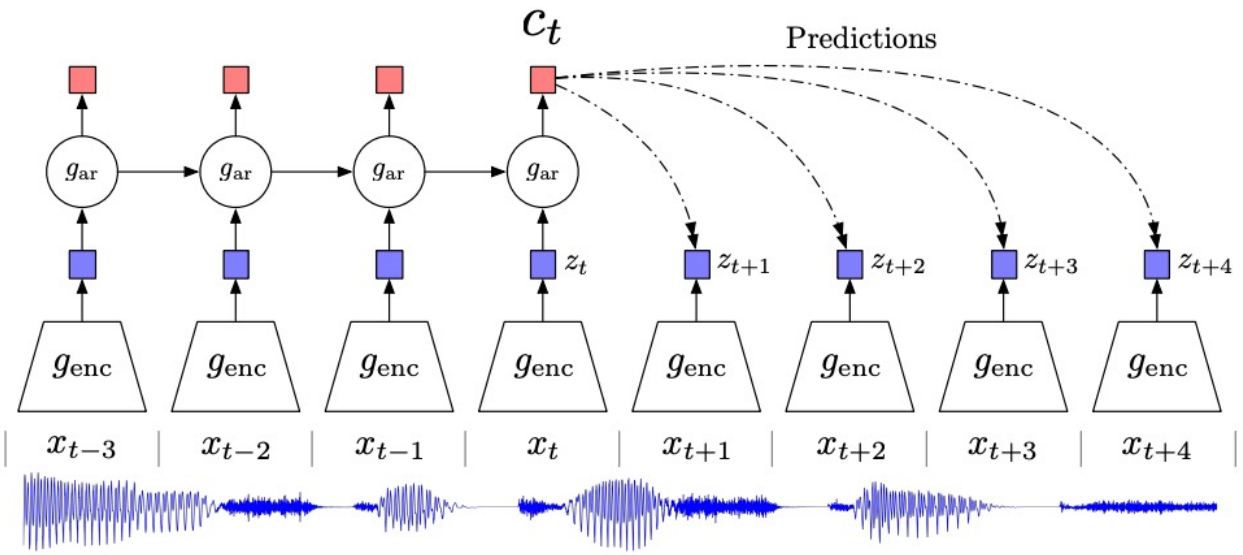
Method	Kinetics-400		Kinetics-600		Moments in Time		TFLOPs
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	
I3D [13]	71.1	89.3	71.9	90.1	29.5	56.1	-
R(2+1)D [26]	72.0	90.0	-	-	-	-	17.5
bLVNet [27]	73.5	91.2	-	-	31.4	59.3	0.84
S3D-G [96]	74.7	93.4	-	-	-	-	-
Oct-I3D+NL [20]	75.7	-	76.0	-	-	-	0.84
D3D [83]	75.9	-	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

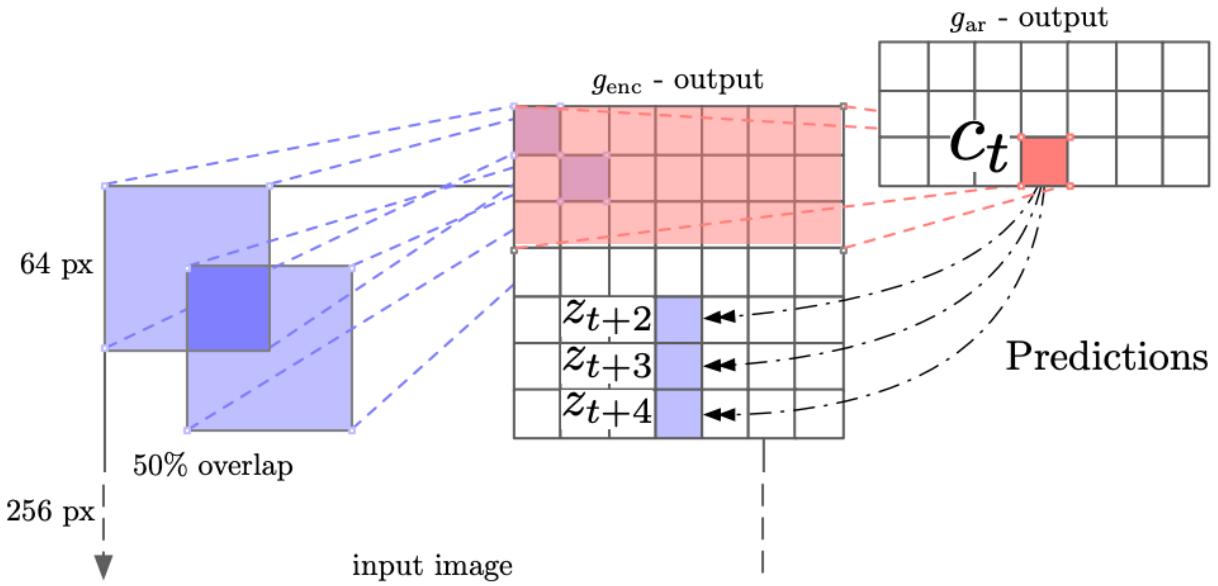
  - x_t : data at time t
  - $z_t = g_{\text{enc}}(x_t)$: high-level latent representation of x_t
  - $c_t = g_{\text{ar}}(x_1, x_2, \ldots, x_t)$: context latent representation summarizing all $z_{\leq t}$



* source : [Oord et al., 2018]

- **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
  - $c_t = g_{\text{ar}}(x_1, x_2, \ldots, x_t)$: context latent representation summarizing all $z_{\leq t}$



* source : [Oord et al., 2018]

- **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:

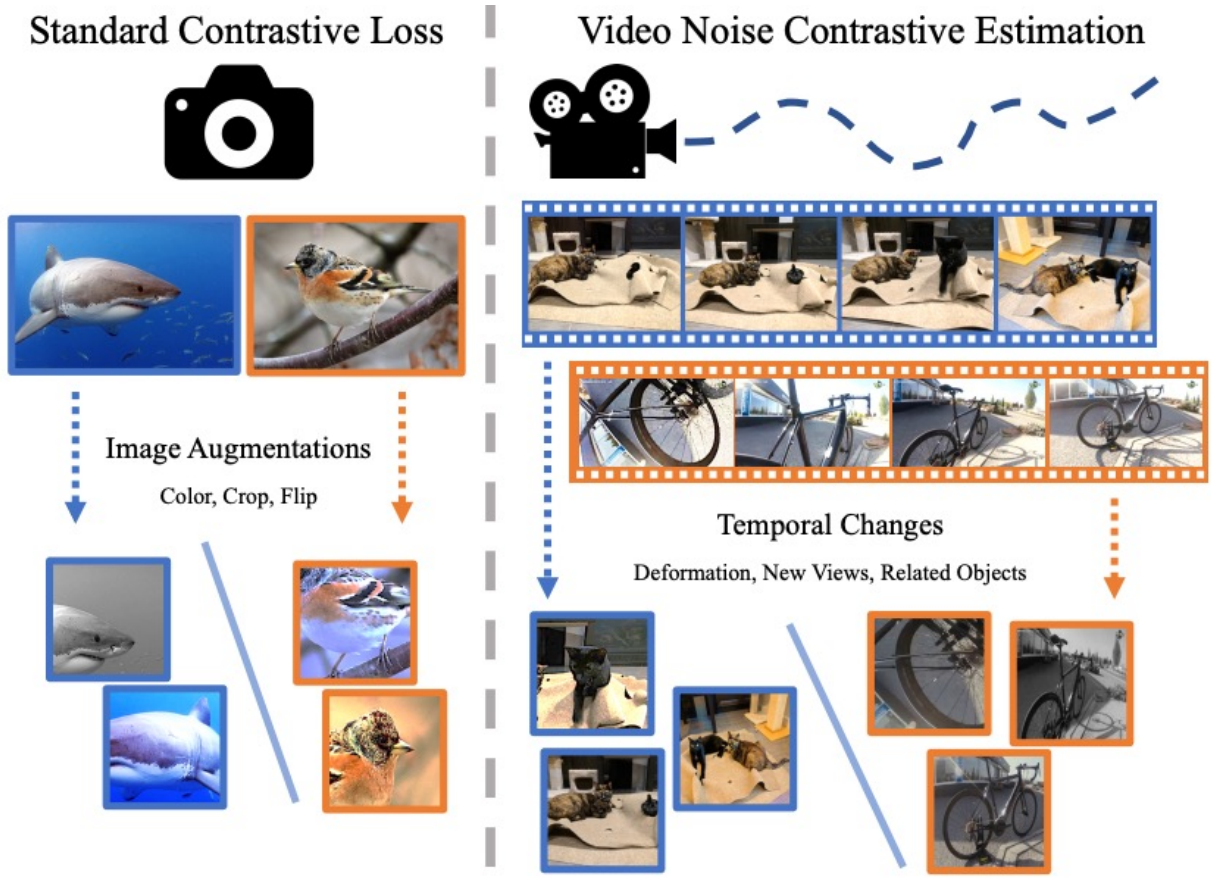
$$\mathcal{L}_N = -\mathbb{E}_X \left[\log \frac{f_k(x_{t+k}, c_t)}{\sum_x f_k(x, c_t)} \right]$$

      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

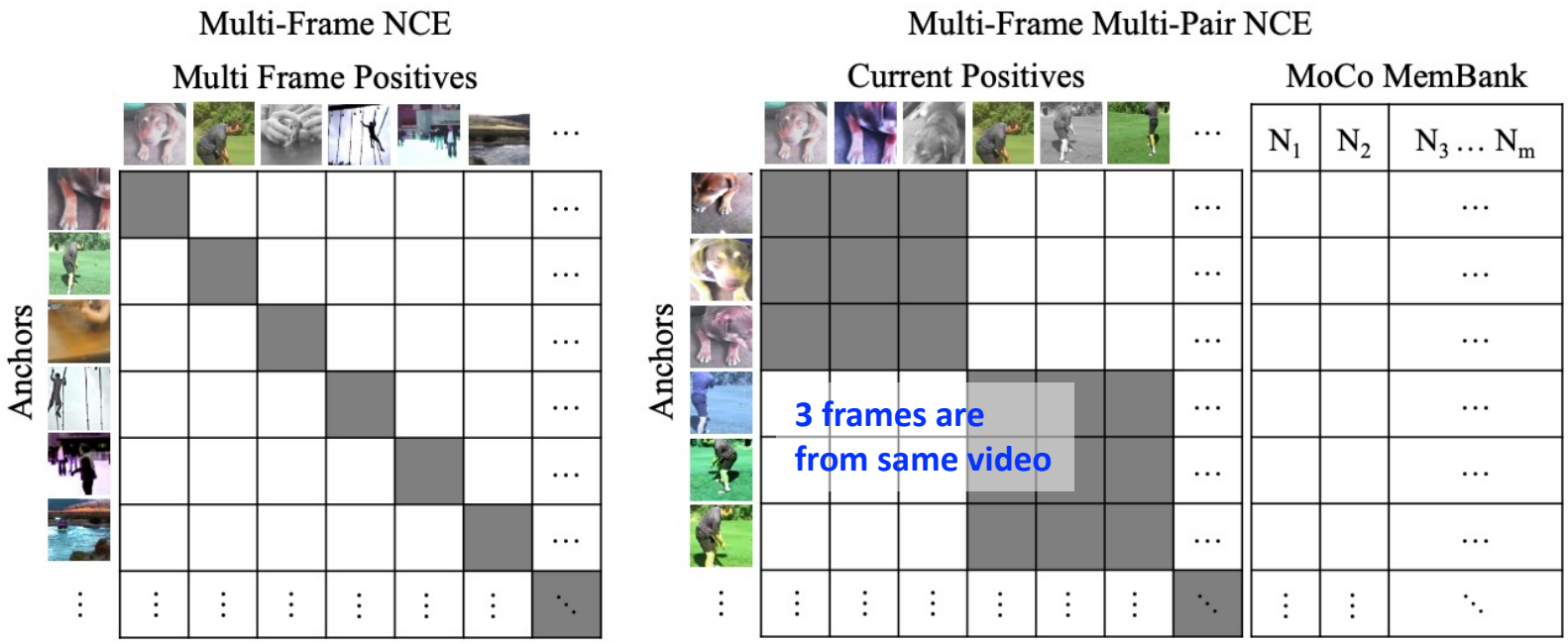
* source : [Oord et al., 2018]

- **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



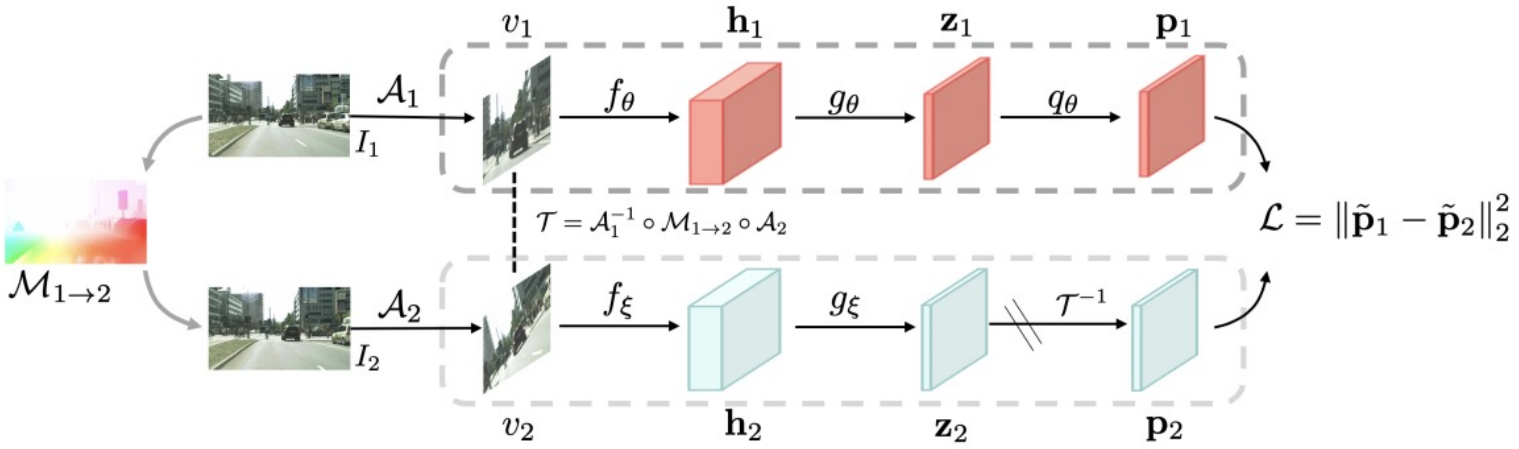
- **VINCE** [Gordon et al., 2019]
  - Data augmentations cannot tell the novel views and motions of the objects
  - Instead, use **video data** to provide 3D-aware positive views
  - Namely, use different frames from the same video as positive samples

  - 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

	Test Task				
Images Per Video	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 $\mathcal{M}_{1\to2}$ is an **optical flow**
    - Then the **spatial features** z_1, z_2 should satisfy the equivariance $z_2 = \mathcal{T}(z_1)$



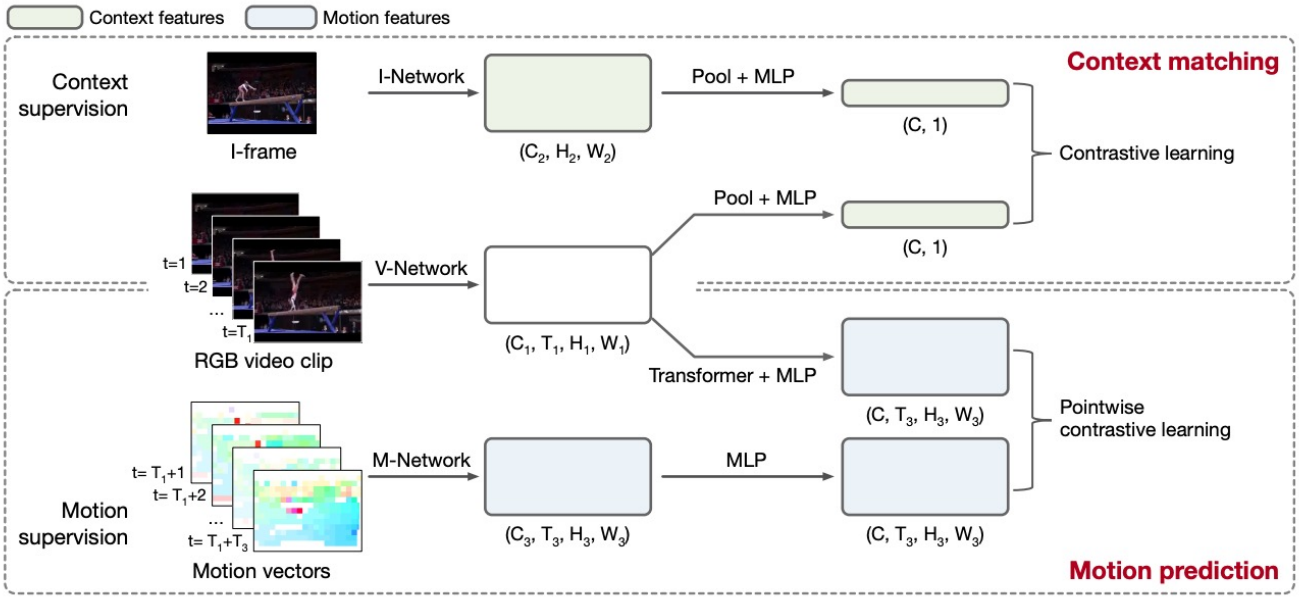
- **FlowE** [Xiong et al., 2021]
  - VINCE assumed frames from the same video are invariant
  - However, we need to consider their temporal changes

  - 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 $\mathcal{M}_{1\to2}$ 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



*source: Huang et al., Self-supervised Video Representation Learning by Context and Motion Decoupling, CVPR 2021

- **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

Table of Contents

- **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

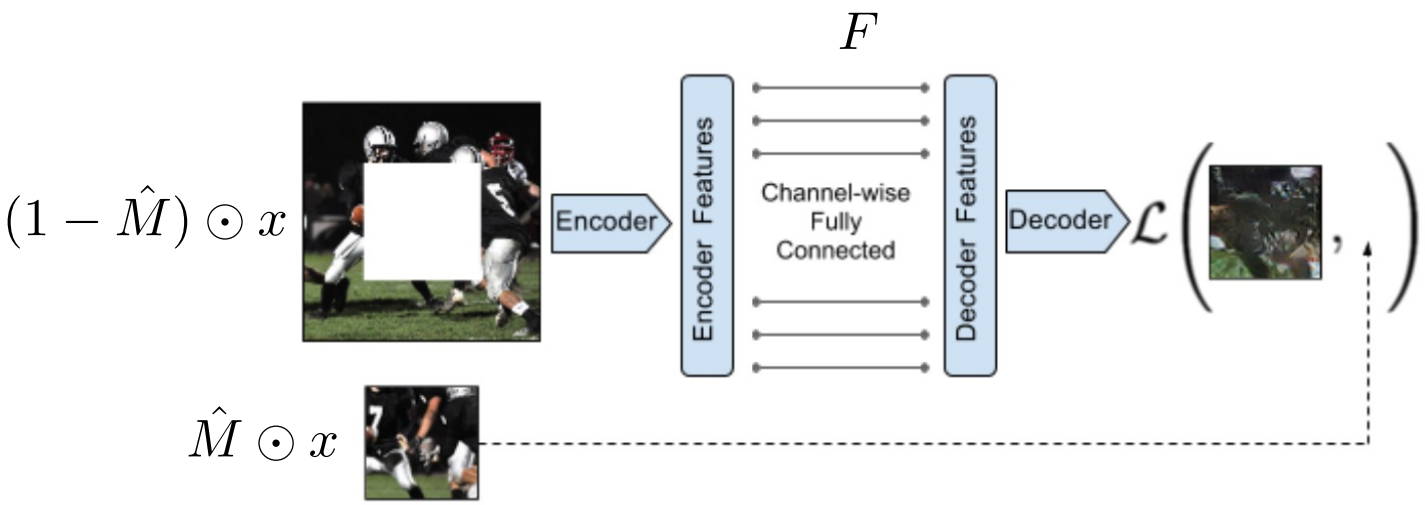
  - **Classic approaches** (before contrastive learning, 2020)
    - We introduce some notable classic methods
      - Context encoder, a CNN version of masked autoencoder
      - BigBiGAN, which were SOTA of then

  - Recent methods can be categorized into **2 groups:**
    - **BERT-like approach** (or masked autoencoder)
      - Predict original X from perturbed $\tilde{X}$ (learn $\tilde{X} \to Z \to X$ encoder)

    - **GPT-like approach** (or sequential prediction)
      - Predict future state X_{t+1} from past states $X_{1:t}$ (learn $X_{1:t} \to X_t$ decoder)

- **Context Encoder** [Pathak et al., 2016]
  - **Task:** Predict the masked region using **its surrounding information**
  - The auto-encoder is trained via **reconstruction** loss

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



* source : [Pathak et al., 2016]

- **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)

* source : [Pathak et al., 2016]

- **Context Encoder** [Pathak et al., 2016]
  - **Task:** Predict the masked region using **its surrounding information**
  - The auto-encoder is trained via **reconstruction** loss

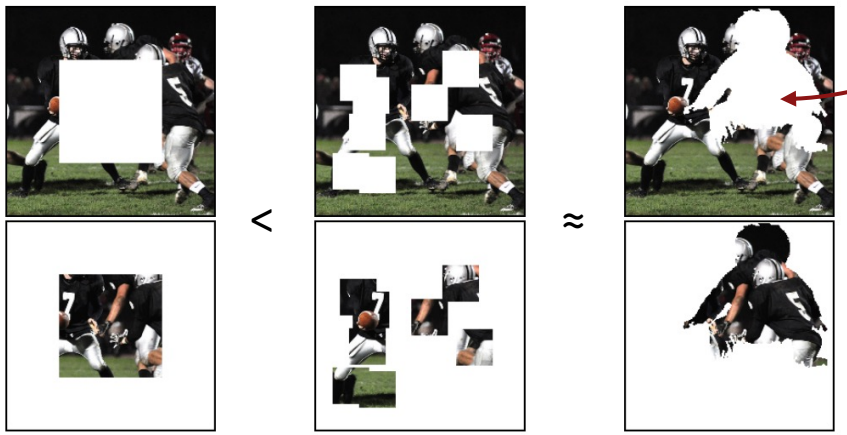
$$\mathcal{L}_{\text{rec}}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2^2$$

  - 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?

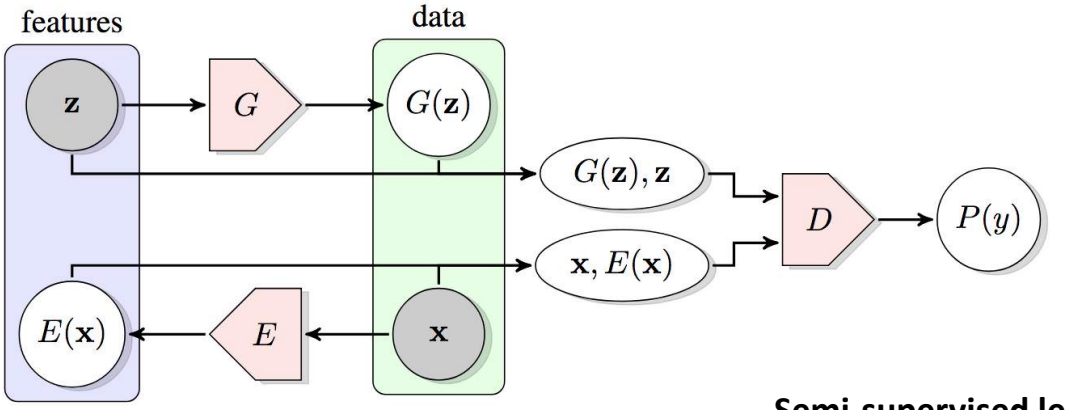
A segmentation mask in other dataset



(a) Central region (b) Random block (c) Random region

* source : [Pathak et al., 2016]

- **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

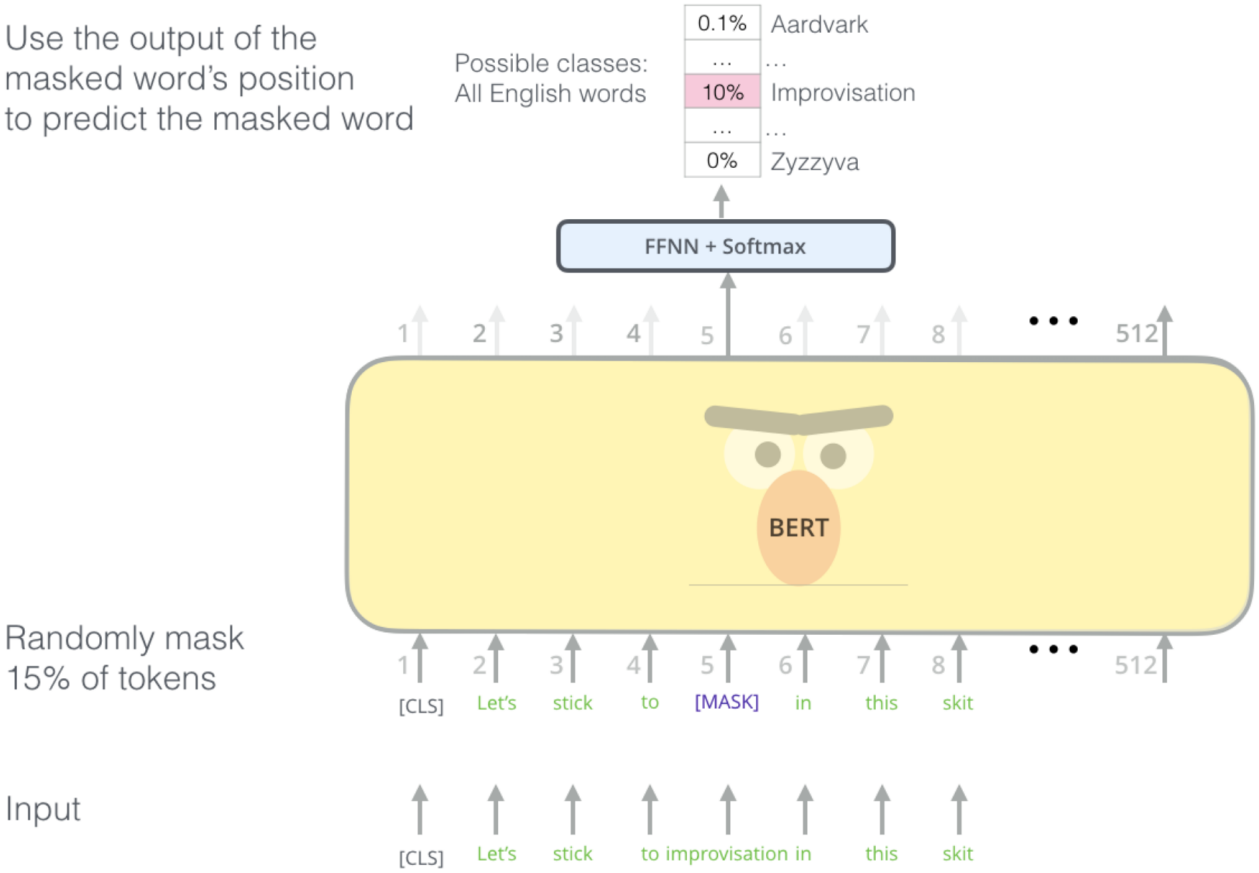
Number of labeled examples	1000	2000	4000	8000
Model	Misclassification rate			
Ladder network (Rasmus et al., 2015)			20.40	
CatGAN (Springenberg, 2015)			19.58	
GAN (feature matching) (Salimans et al., 2016)	$\mathbf{21.83 \pm 2.01}$	$\mathbf{19.61 \pm 2.09}$	$\mathbf{18.63 \pm 2.32}$	$\mathbf{17.72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

- **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 ×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 ×4	AvePool	60.8	81.4
	RevNet-50 ×4	BN+CReLU	61.3	81.9

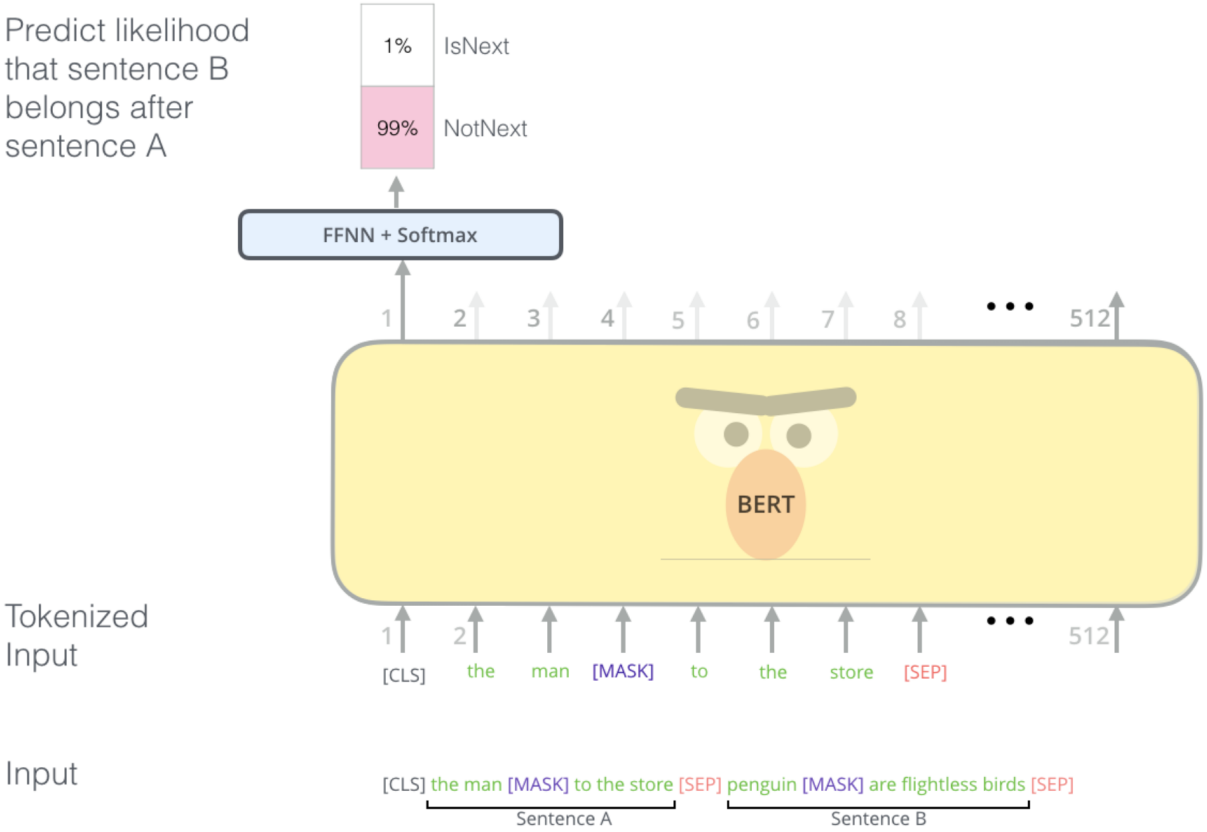
- **Overview of Generation-based Approaches**
  - There have been a long attempts to learn representation Z from data X
  - To this end, many classic ML literature designed a probabilistic model $p(X, Z)$
    - They are called as generative models with latent variables
  - **Classic approaches** (before 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 $\tilde{X} \to Z \to X$ encoder)
    - **GPT-like approach** (or sequential prediction)
      - Predict future state X_{t+1} from past states $X_{1:t}$ (learn $X_{1:t} \to X_t$ decoder)

- **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



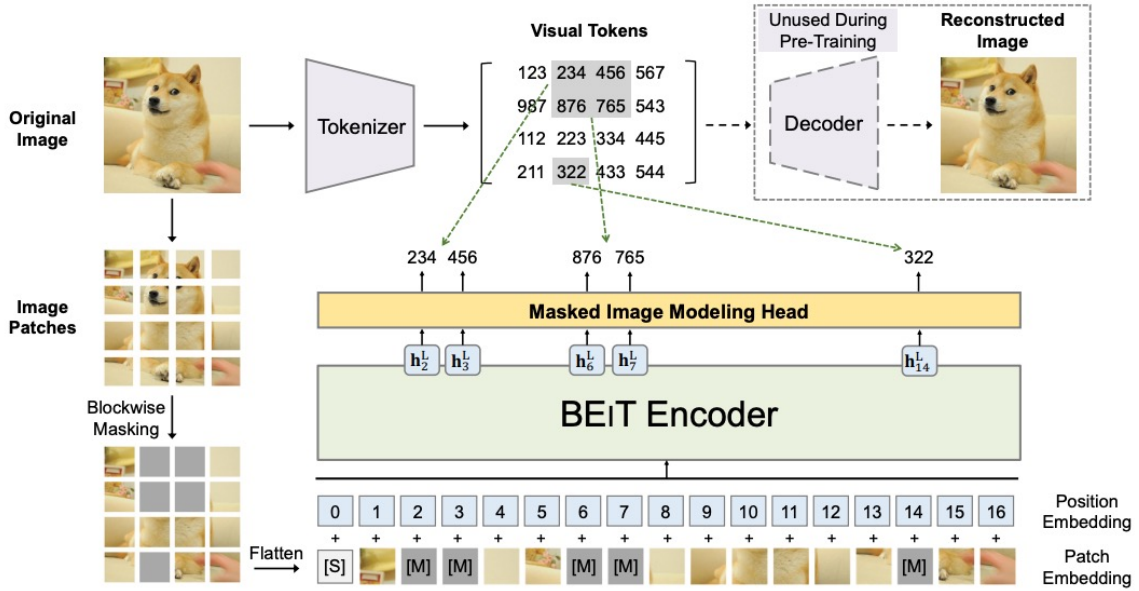
*reference: http:// http://jalammar.github.io/illustrated-bert

- **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**



*reference: http:// http://jalammar.github.io/illustrated-bert

- **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

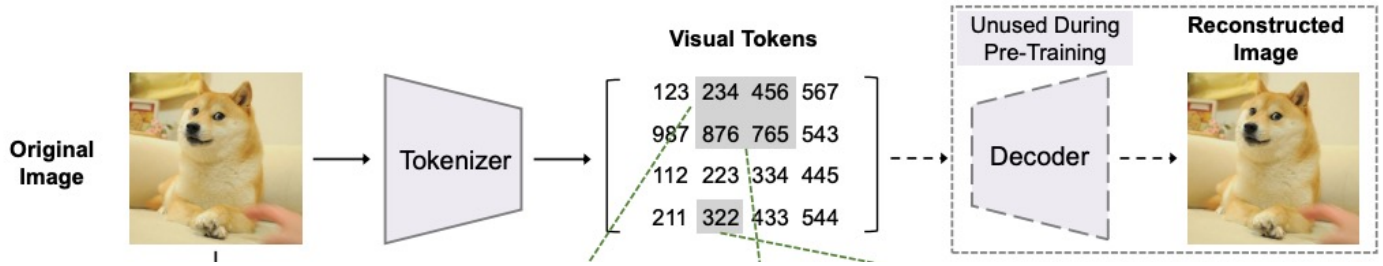


  - BEiT training procedure is consist of two stages:

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:

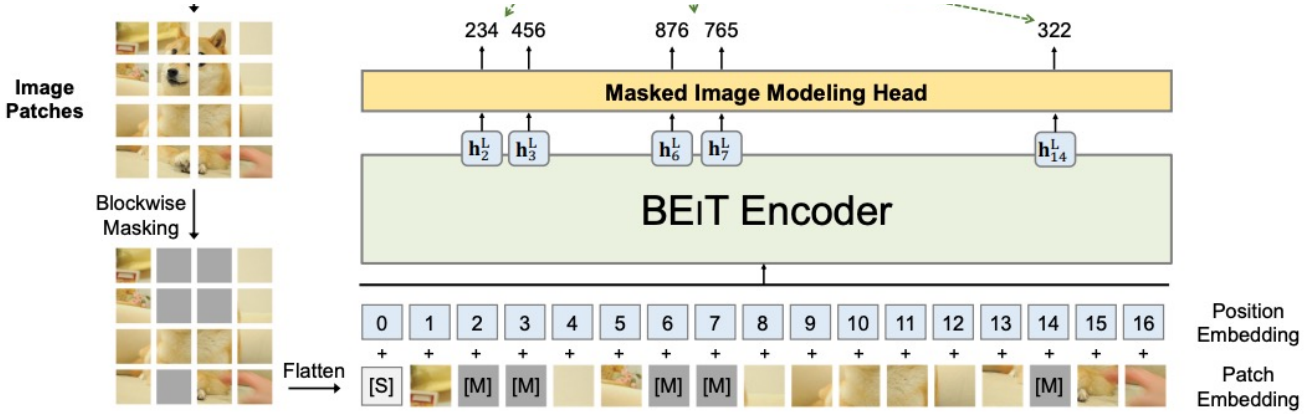
1. Learning visual tokens



- 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 whic h can assume 8192 possible values
  - The tokenizer $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ maps image image pixels into a visual codebook
  - The decoder $p_\psi(\boldsymbol{x}|\boldsymbol{z})$ 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

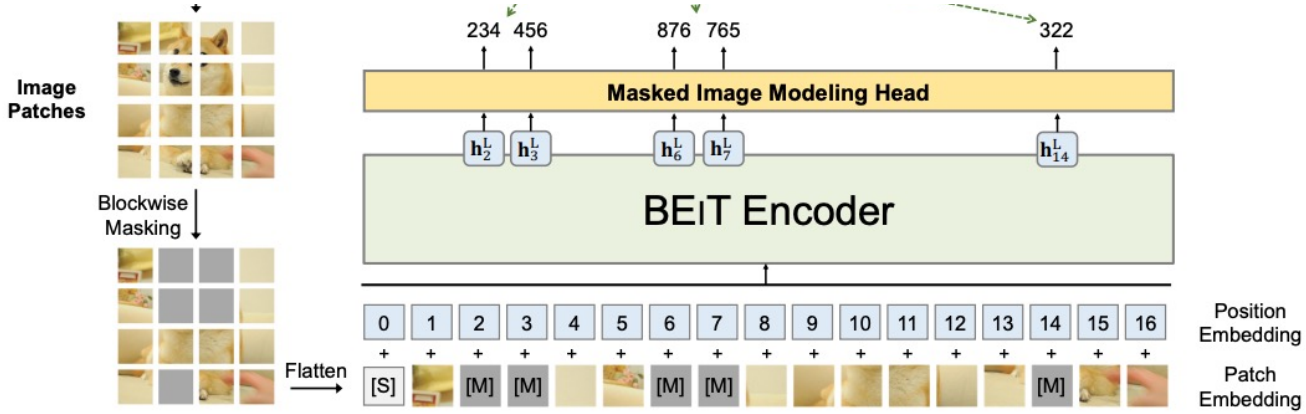


  - The standard ViT is used as the backbone network
  - Some image patches are randomly masked (approx. 40%), and then the **visual tokens that corresponds to the masked patches** are predicted
    - The objective is maximizing the log-likelihood of the correct visual tokens z_i given the corrupted image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

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

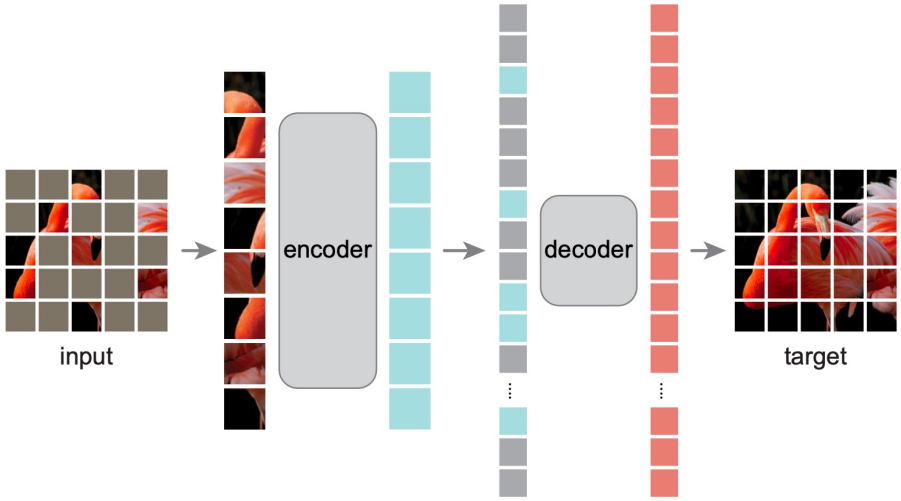
- **BEiT** [Bao et al., 2022]
  - **Task:** Masked visual tokens prediction
  - BEiT training procedure is consist of two stages:

  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**

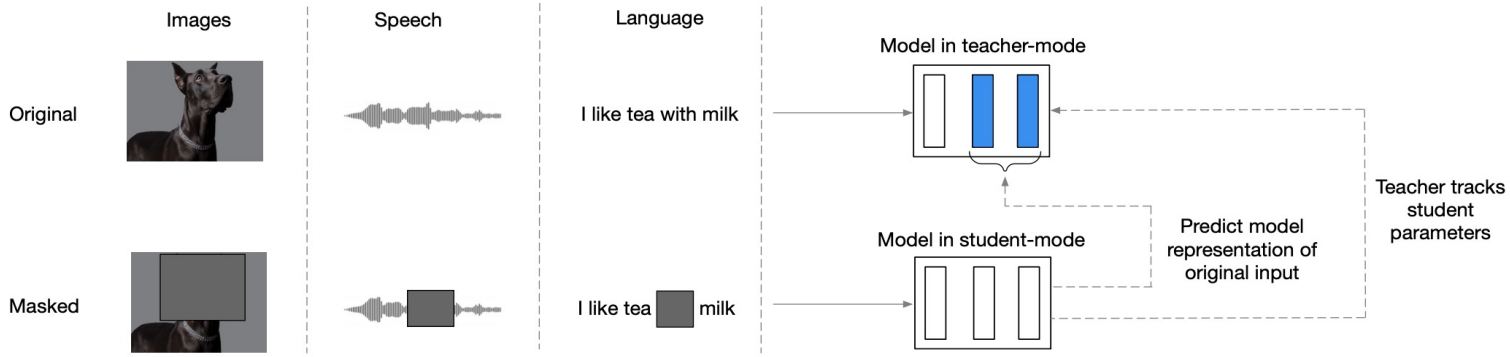
case	ratio	ft	lin
random	75	**84.9**	**73.5**
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

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

random 75% block 50% grid 75%

(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

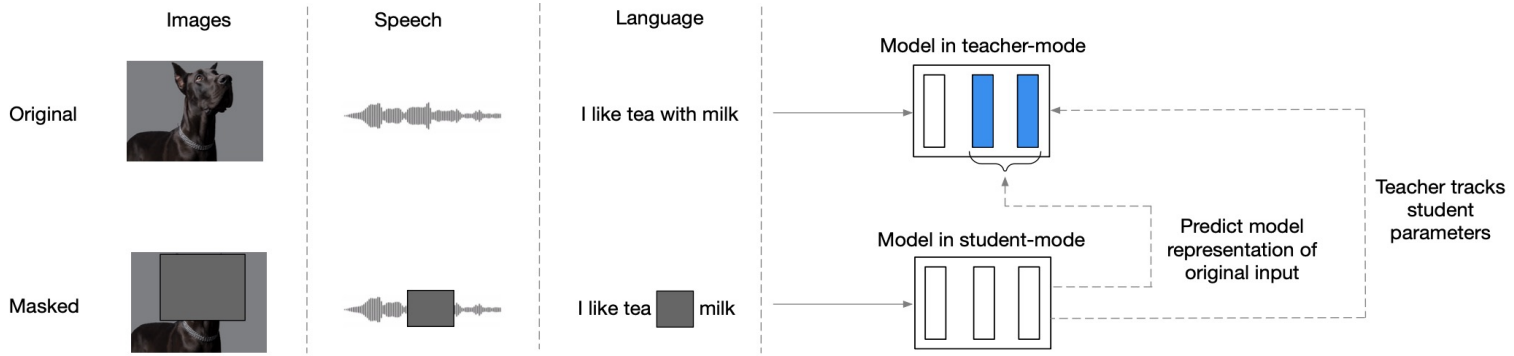
- **data2vec** [Baevski et al., 2022]
  - data2vec is a framework for **general self-supervised learning** for images, speech, and text where the **learning objective is identical in each modality**



  - **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
      - $\widehat{a_t^l}$: the normalized output of block l at time-step t
      - Training target: $y_t = \frac{1}{K}\sum_{l=L-K+1}^{L} \widehat{a_t^l}$
    - The objective is smooth-L1 loss between y_t and the prediction $f_t(x)$ at t:

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

- **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
Multiple models		
BEiT (Bao et al., 2021)	83.2	85.2
PeCo (Dong et al., 2022)	84.5	86.5
Single models		
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

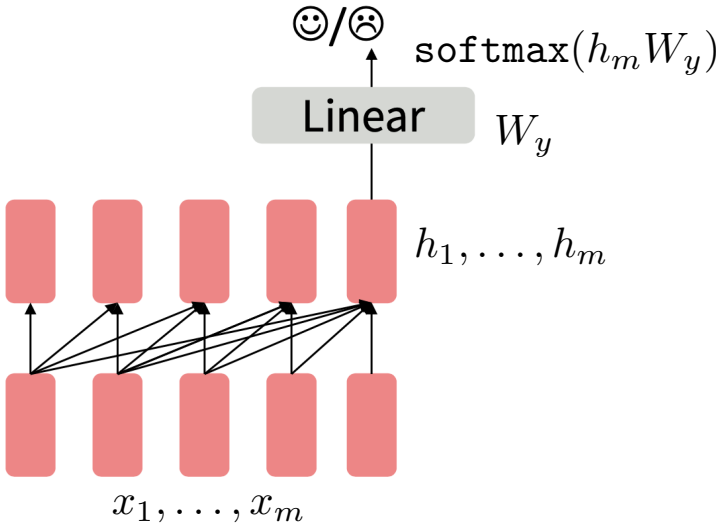
NLP

- **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 $\tilde{X} \to Z \to X$ encoder)
    - **GPT-like approach** (or sequential prediction)
      - Predict future state X_{t+1} from past states $X_{1:t}$ (learn $X_{1:t} \to X_t$ decoder)

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

$$\arg\max_{\theta} \log p(\boldsymbol{x}) = \sum_{n} p_\theta(x_n | x_1, \ldots, x_{n-1})$$

  - **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



*reference: http://web.stanford.edu/class/cs224n/

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

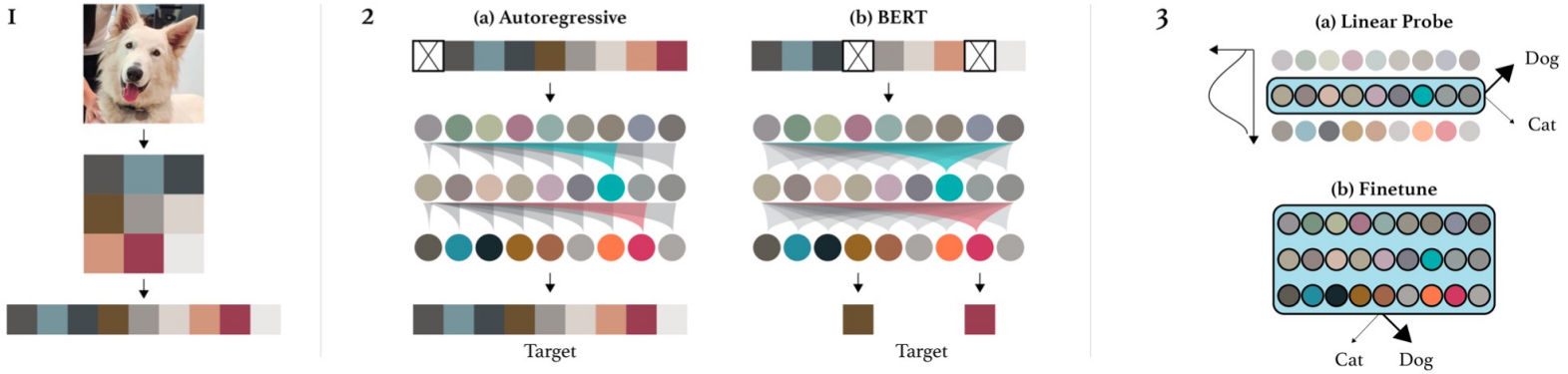


Figure 1. An overview of our approach. First, we pre-process raw images by resizing to a low resolution and reshaping into a 1D sequence. We then chose one of two pre-training objectives, auto-regressive next pixel prediction or masked pixel prediction. Finally, we evaluate the representations learned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

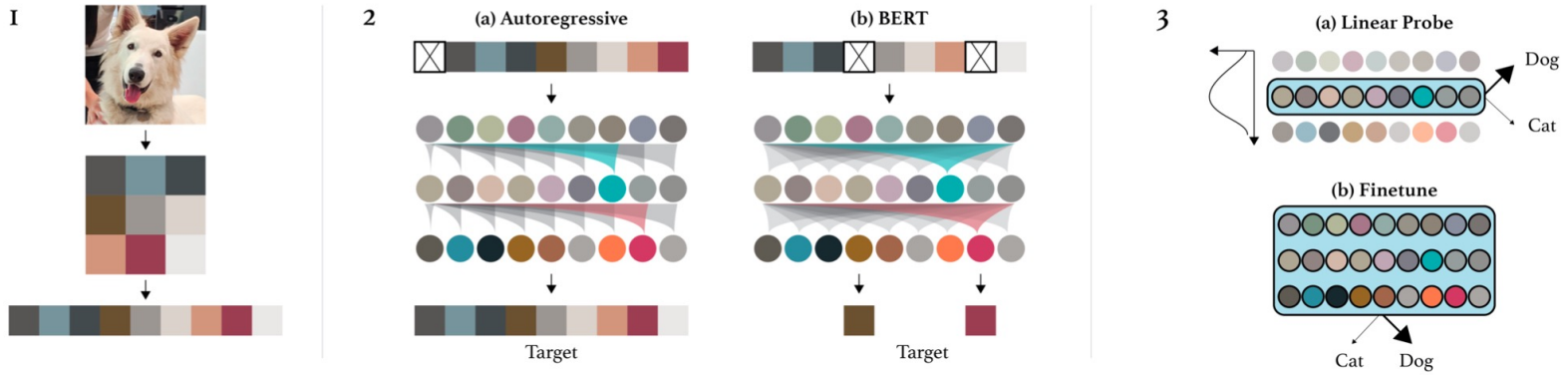
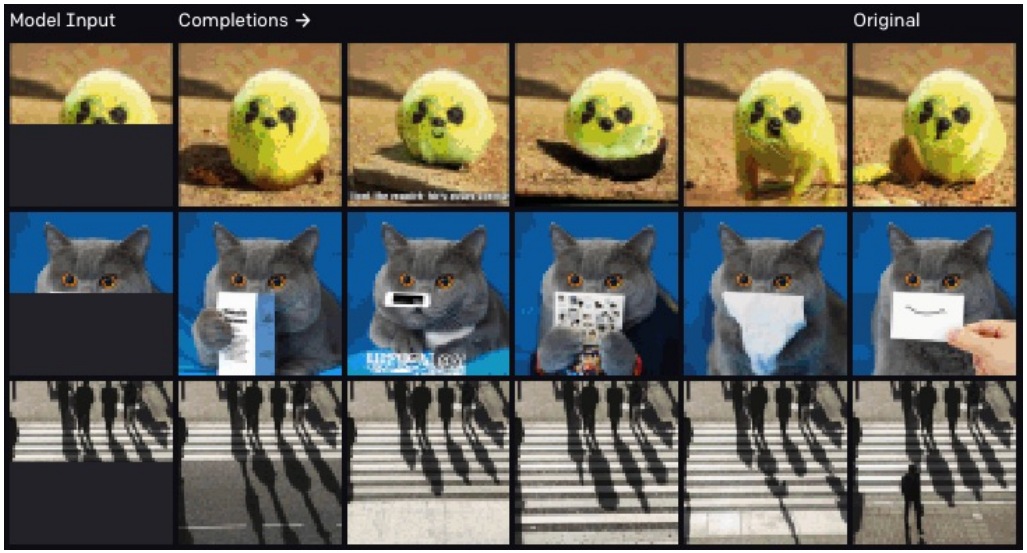


Figure 1. An overview of our approach. First, we pre-process raw images by resizing to a low resolution and reshaping into a 1D sequence. We then chose one of two pre-training objectives, auto-regressive next pixel prediction or masked pixel prediction. Finally, we evaluate the representations learned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

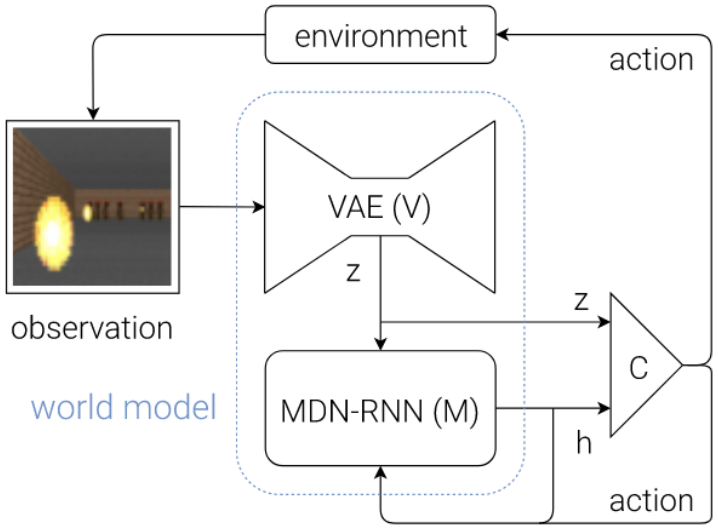
  - **Input data format:** 9-bit color palette
    - iGPT down-samples an image into one of 32 × 32, 48 × 48, or 64 × 64 RGB data
    - iGPT clusters all (R, G, B) values in training dataset using k-means with k=512, which is 9-bit color palette
      - It further **reduces input sequence length** 3 times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

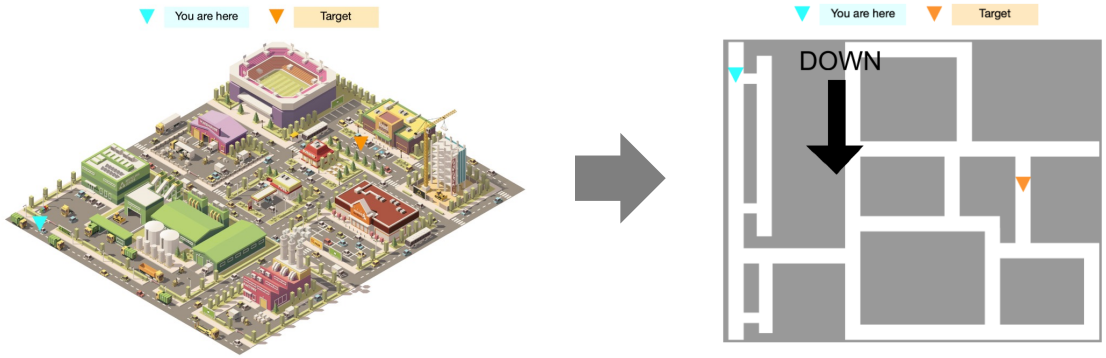


- **World Model**
  - Recall that it is similar to the **CPC objective** in the SSL via Invariance section
    - **Generation:** Predict the target X_{t+1} directly
    - **Contrastive:** Find the positive X_{t+1} from negative samples X'_{t+1}
    - One can interchange them arbitrarily ⇒ **Q.** Which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



*reference: https://jacobbuckman.com/2019-10-25-three-paradigms-of-reinforcement-learning/

Interim Summary

- We discussed **2 types** of self-supervised learning
  1. **Invariance:** Maximize MI of representations of positive samples
  2. **Generation:** Maximize MI of representation and (perturbed) data

  - 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!
  - 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, etc.
  - Recent works discover that visual SSL is also effective for the **planning** tasks
- Now, we'll focus on recent trends on multimodal SSL methods

Table of Contents

Multimodal SSL: Overview

- There have been a long attempts to learn vision-language (VL) models
  - Different objective, different methods have been studied
- We discuss **four approaches** in achieving various VL representations
  1. Image-text alignment using CLIP for transferrable visual representation
     - Enables zero-shot classification & high robustness
  2. Fused transformer for vision-language understanding
     - Better vision-language understanding tasks, e.g., Visual Question Answering
  3. Learning visual representation from frozen Large Language Models (LLMs)
     - Leveraging the power of LLMs for visual in-context learning
  4. Unifying Vision-Language pretraining
     - Learning Vision-Language model from scratch for all tasks

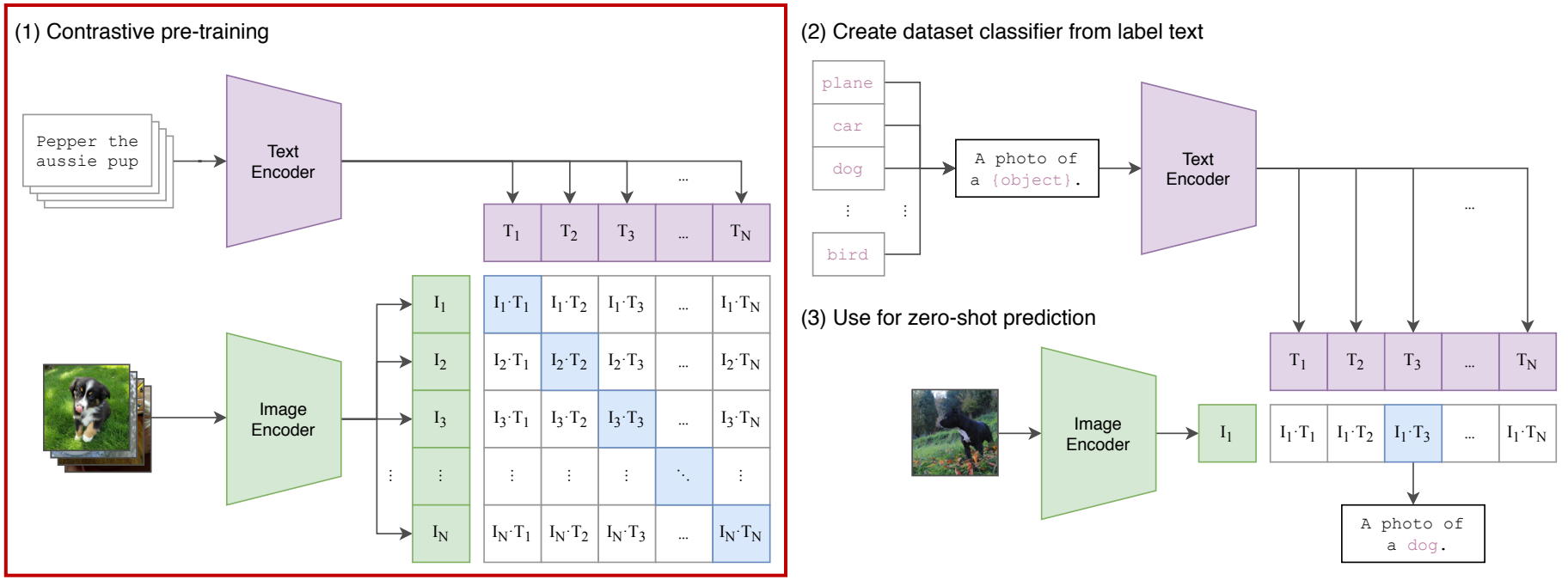
Multimodal SSL: Overview

- There have been a long attempts to learn vision-language (VL) models
  - Different objective, different methods have been studied
- We discuss **four approaches** in achieving various VL representations
  1. Image-text alignment using CLIP for transferrable visual representation
     - Enables zero-shot classification & high robustness
  2. Fused transformer for vision-language understanding
     - Better vision-language understanding tasks, e.g., Visual Question Answering
  3. Learning visual representation from frozen Large Language Models (LLMs)
     - Leveraging the power of LLMs for visual in-context learning
  4. Unifying Vision-Language pretraining
     - Learning Vision-Language model from scratch for all tasks

CLIP [Radford et al., 2020]

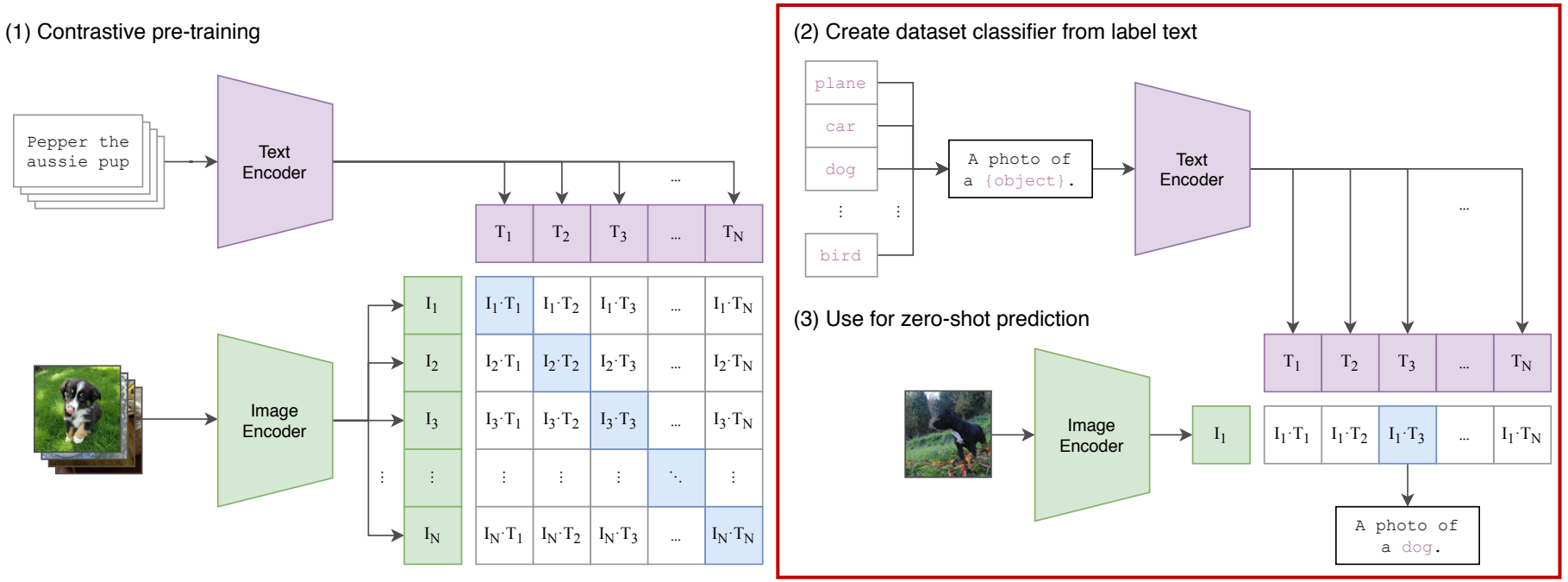
- Simple contrastive learning between **image** and **text** embeddings
- Trained on large-scale web image-text pairs

$$L_{\text{CLIP}} = -\frac{1}{2N}\sum_{i=1}^{N}\log\frac{\exp(I_i \cdot T_i)}{\sum_{j=1}^{N}\exp(I_i \cdot T_j)} - \frac{1}{2N}\sum_{j=1}^{N}\log\frac{\exp(I_j \cdot T_j)}{\sum_{i=1}^{N}\exp(I_i \cdot T_j)}$$



CLIP [Radford et al., 2020]

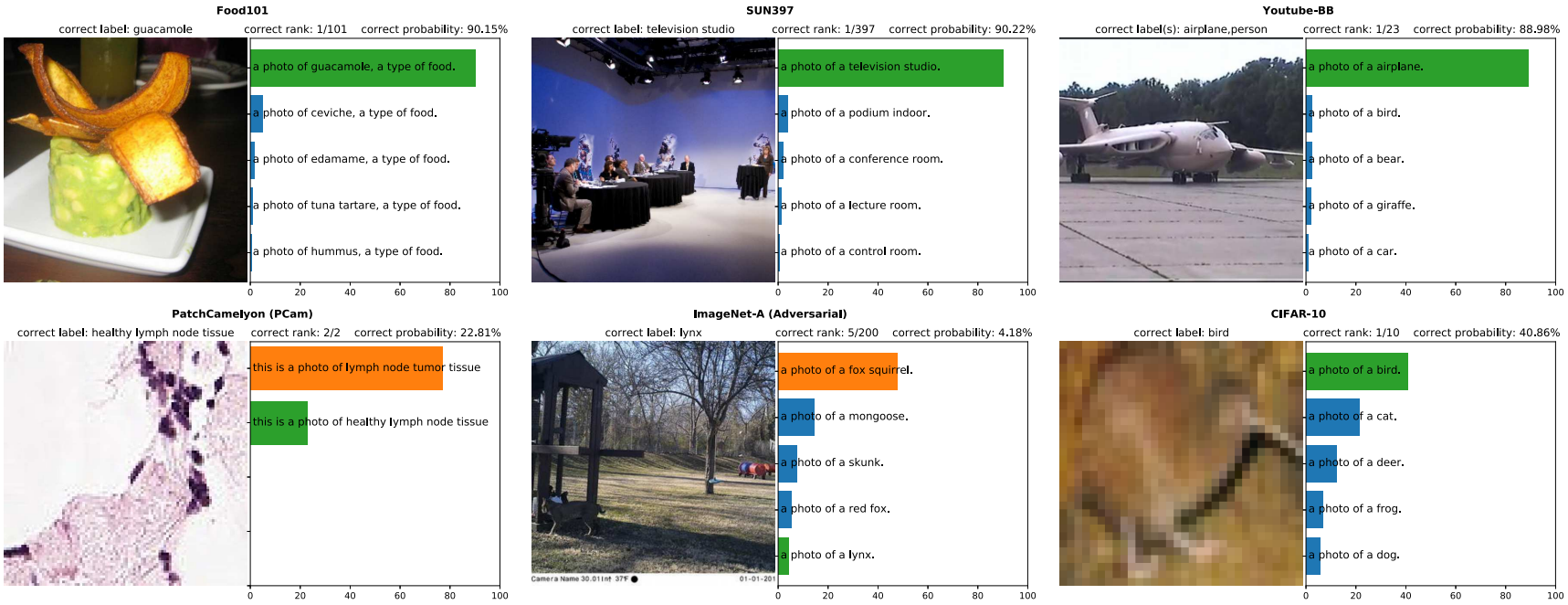
- Zero-shot transfer
  - Transfer learning without seeing the images or labels
  - **Prompt Engineering**: ”A photo of a [MASK]”
  - Choose class that maximizes similarity with respect to image



CLIP: Contrastive Language-Image Pre-training

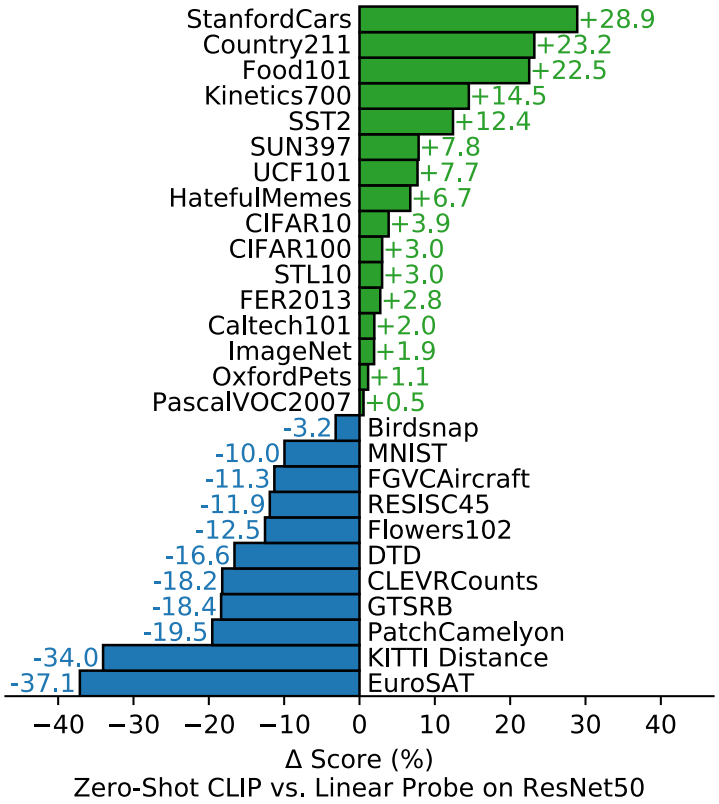
CLIP [Radford et al., 2020]

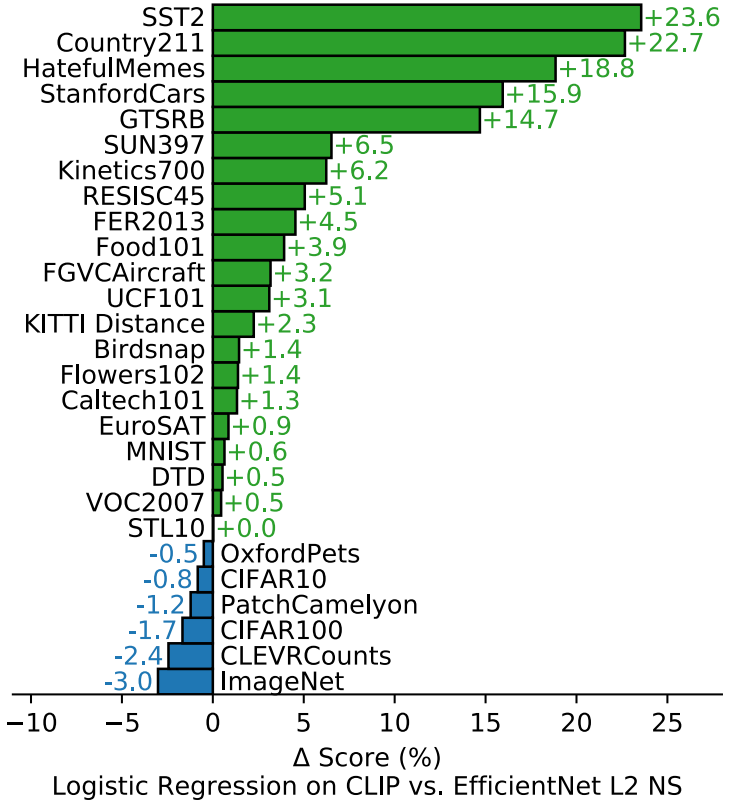
- Zero-shot transfer
  - Transfer learning without seeing the images or labels
  - **Prompt Engineering**: ”A photo of a [MASK]”
  - Choose class that maximizes similarity with respect to image



CLIP [Radford et al., 2020]

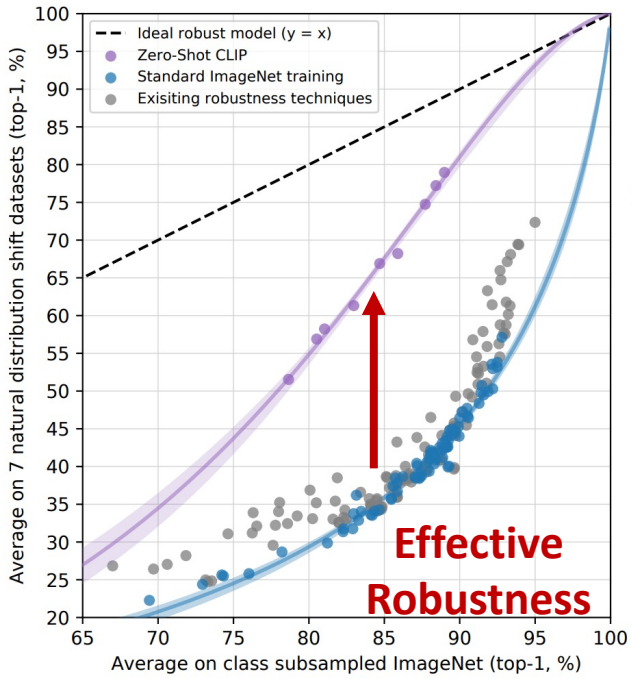
- A zero-shot CLIP classifier shows a competitive performance with a fully supervised linear classifier fitted on ResNet-50 features
- Linear-probing with CLIP image features outperform the best ImageNet model





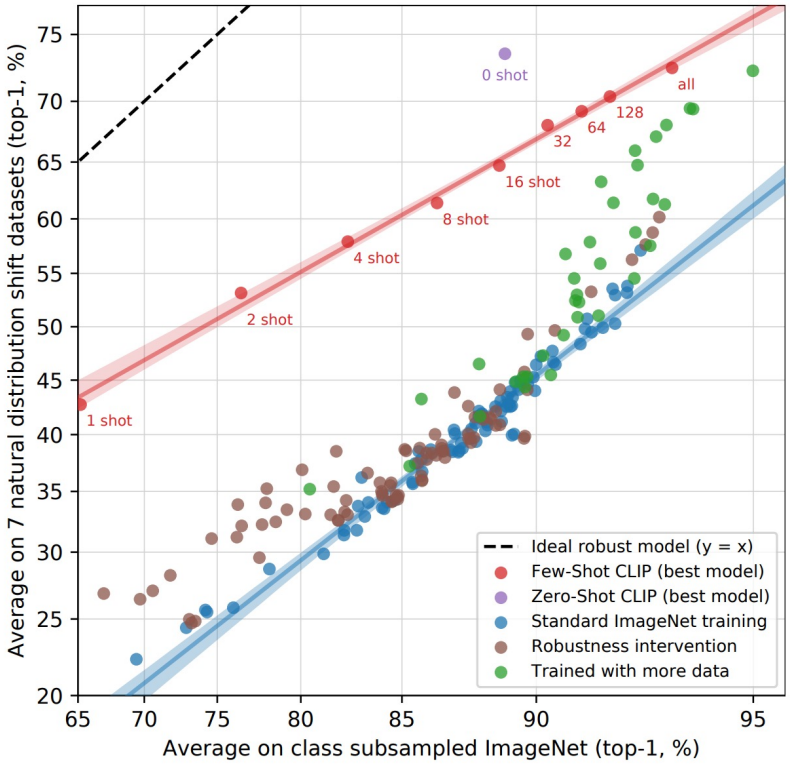
CLIP [Radford et al., 2020]

- Zero-shot CLIP classifier is more robust to natural **distributional shift**
  - Interestingly, [Ilharco et al., 2021] show that CLIP have high **effective robustness** even at small scale



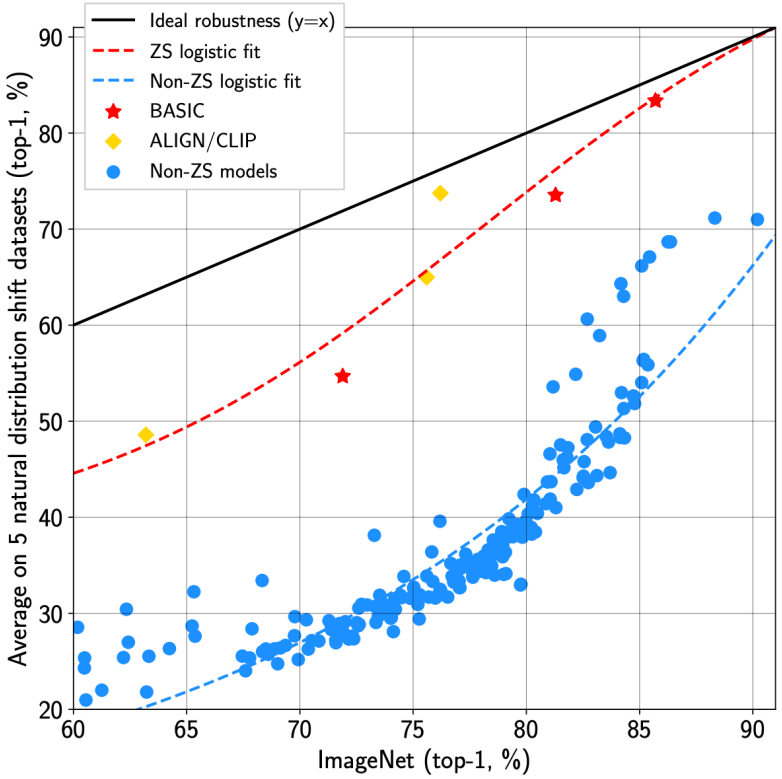
CLIP [Radford et al., 2020]

- Zero-shot CLIP classifier is more robust to natural **distributional shift**
  - Interestingly, [Ilharco et al., 2021] show that CLIP have high **effective robustness** even at small scale
- Few-shot CLIP classifier also shows high effective robustness, but less than zero-shot CLIP classifier



Follow-up studies showed scaling dataset size improves performance

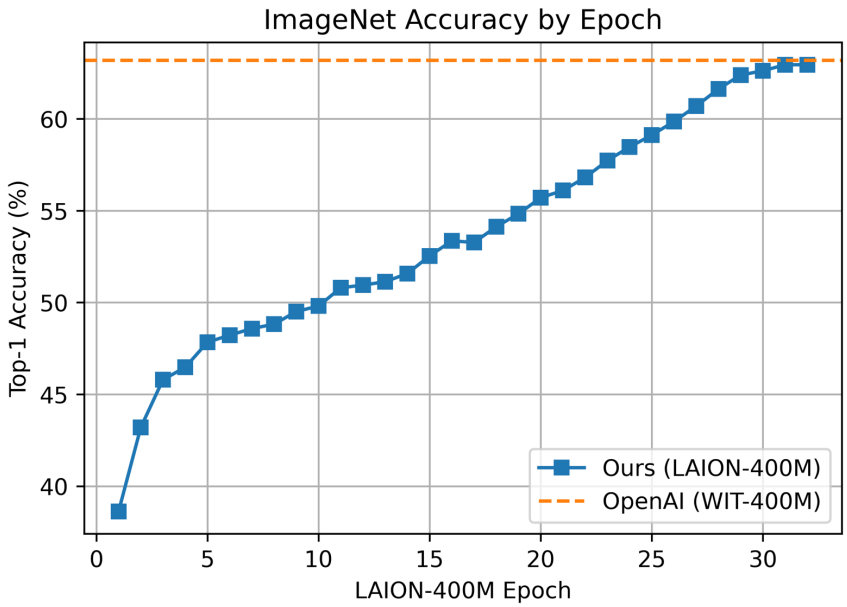
- CLIP uses carefully filtered **400M** image-text pairs from web
- **ALIGN** [Jia et al., 2020] collected noisy **1.8B** image-text pairs to scale CLIP
- **BASIC** [Pham et al., 2021] used **6.6B** image-text pairs with bigger model size

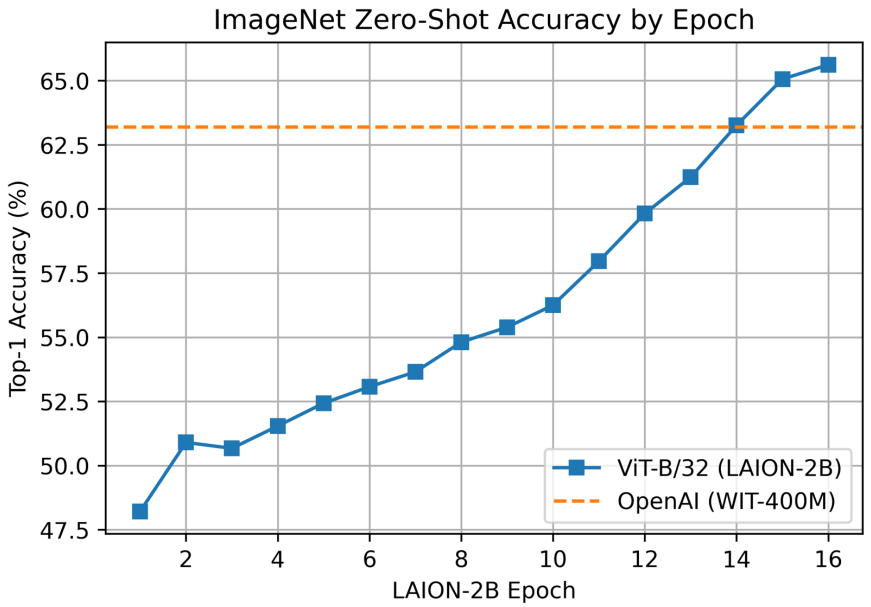


Open-source Implementation

However, those datasets and implementations are not publicly available

- **OpenCLIP** [Ilharco et al., 2021]: open-source implementation of CLIP
- **LAION** [Schuhmann et al., 2022]: publicly available **400M** & **5B** size of dataset that shows competitive results of CLIP



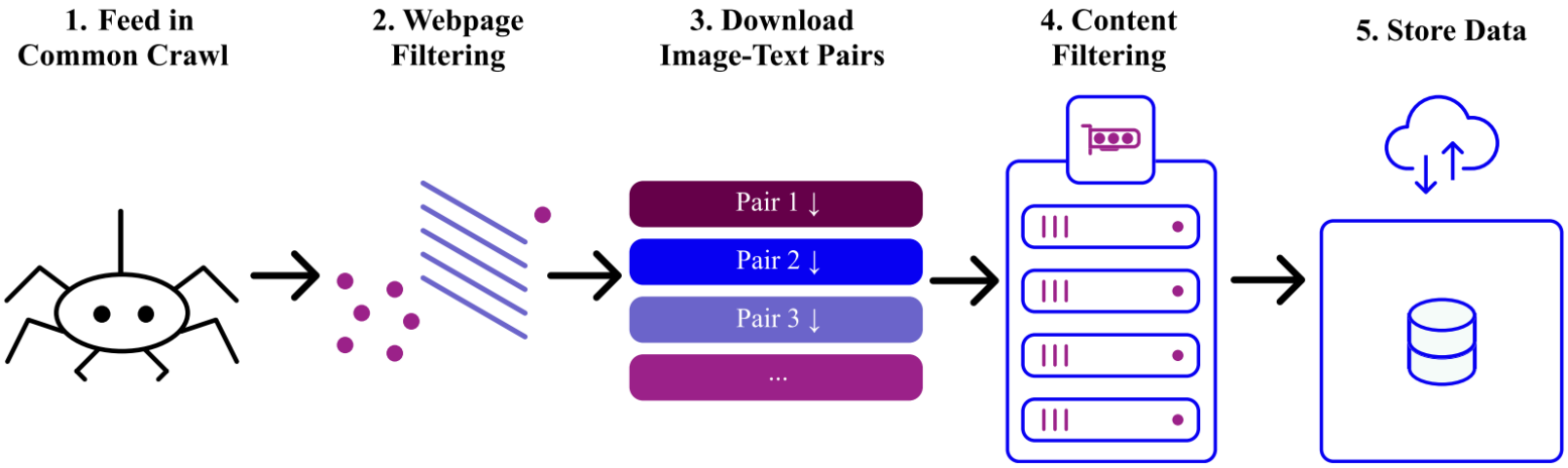


Trained with OpenCLIP on LAION 400M & 2B datasets

Open-source Implementation

However, those datasets and implementations are not publicly available

- **OpenCLIP** [Ilharco et al., 2021]: open-source implementation of CLIP
- **LAION** [Schuhmann et al., 2022]: publicly available **400M** & **5B** size of dataset that shows competitive results of CLIP
  - They used pre-trained CLIP features to filter the dataset

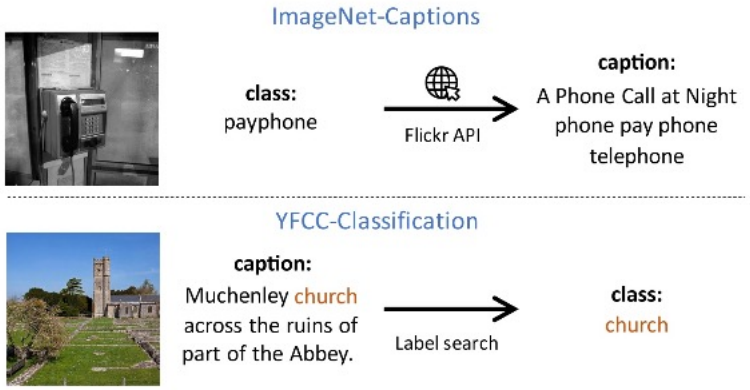


Data acquisition pipeline of LAION

Motivation: What causes CLIP's unprecedented robustness?

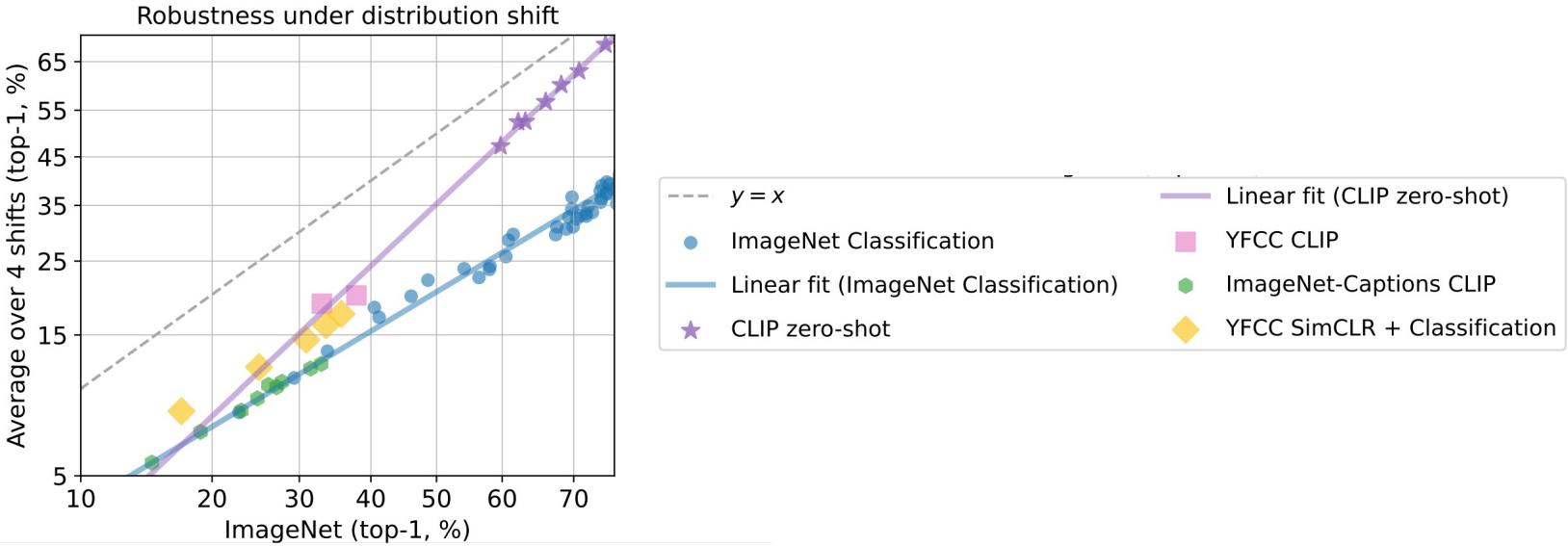
- [Fang et al., 2022] examined following sources of CLIP
  1. Size of training dataset
  2. Distribution of training data
  3. Language supervision at training
  4. Prompt-tuning as test-time
  5. Contrastive learning objectives

- For systematic study, they considered two datasets
  - **ImageNet-Captions:** Captions for ImageNet dataset to do CLIP
  - **YFCC-Classification:** Labeled YFCC dataset to do original training



Dataset Design and Distributional Robustness

- **Size of training dataset do not affect effective robustness**
  - CLIP on YFCC shows similar effective robustness as original CLIP
- **CLIP model is not robust than classification models on same dataset**
  - CLIP on ImageNet-Caption does not show high effective robustness
    - It follows the trend of other ImageNet models
  - SimCLR on labeled YFCC shows similar effective robustness as YFCC CLIP
- **YFCC CLIP follows the trend of original CLIP model**
  - Data distribution affects the effective robustness!

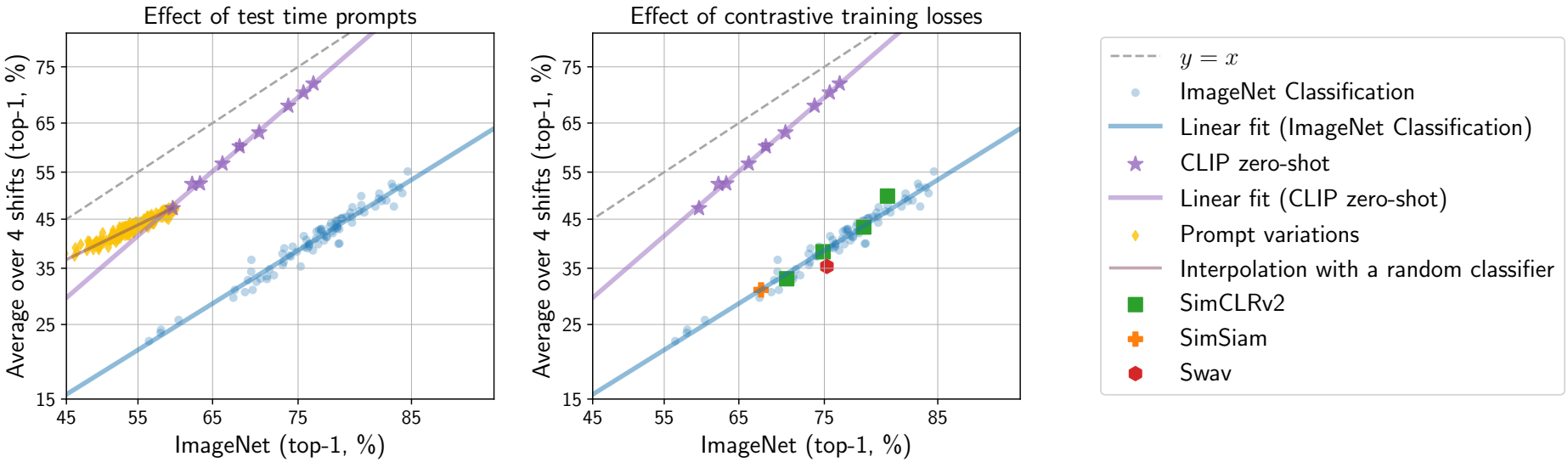


Motivation: What causes CLIP's unprecedented robustness?

- [Fang et al., 2022] examined following sources of CLIP
  1. ~~Size of training dataset~~
  2. Distribution of training data
  3. ~~Language supervision at training~~
  4. Prompt-tuning as test-time
  5. Contrastive learning objectives

- **Prompt-tuning does not have correlation on effective robustness**
  - Prompt variation act as interpolation with a random classifier

- **Various contrastive learning methods do not affect effective robustness**
  - SwAV [Caron et al., 2020], SimSiam [Chen et al., 2021], SimCLR v2 [Chen et al., 2021] on ImageNet dataset follows the trend on ImageNet models



Motivation: What causes CLIP's unprecedented robustness?

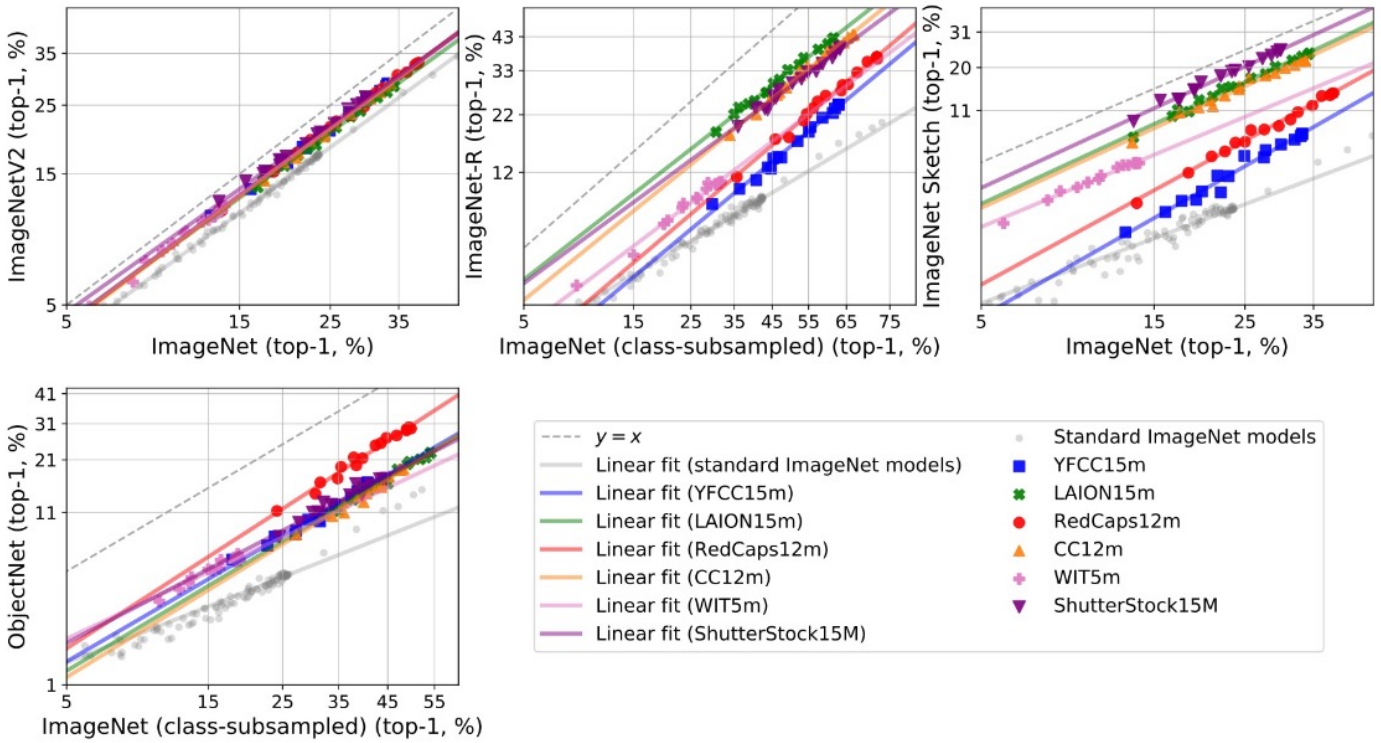
- [Fang et al., 2022] examined following sources of CLIP
  1. ~~Size of training dataset~~
  2. Distribution of training data
  3. ~~Language supervision at training~~
  4. ~~Prompt tuning as test-time~~
  5. ~~Contrastive learning objectives~~

- Conclusion
  - The effective robustness of CLIP is not from language supervision
  - The choice of **training data distribution** matters in effective robustness

  - **But then, how to choose the training dataset?**

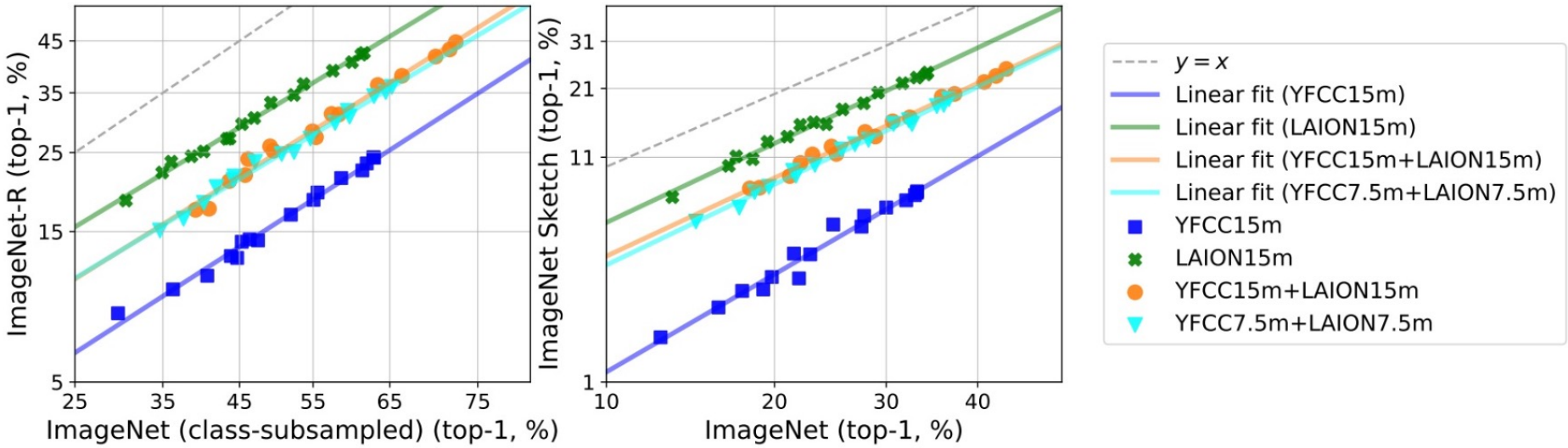
Motivation: Why don't we simply gather all image-text pairs for training data?

- [Nguyen et al., 2022] claimed that simply merging dataset is not an option!
  - **Distributional robustness is determined by the training data distribution**
    - 6 image-text datasets by web-crawling: YFCC, LAION, Conceptual Captions (CC), RedCaps, Shutterstock and WIT
    - For each shift, the level of robustness vary by the choice of dataset



Motivation: Why don't we simply gather all image-text pairs for training data?

- [Nguyen et al., 2022] claimed that simply merging dataset is not an option!
  - Distributional robustness is determined by the training data distribution
    - 6 image-text datasets by web-crawling: YFCC, LAION, Conceptual Captions (CC), RedCaps, Shutterstock and WIT
    - For each shift, the level of robustness vary by the choice of dataset
  - **The robustness of a mixed dataset is not additive**
    - Effective robustness of mixed dataset interpolates between that of two datasets
    - Robustness(YFCC) < Robustness(YFCC+LAION) < Robustness(LAION)



Motivation: Why don't we simply gather all image-text pairs for training data?

- [Nguyen et al., 2022] claimed that simply merging dataset is not an option!
  - Distributional robustness is determined by the training data distribution
    - 6 image-text datasets by web-crawling: YFCC, LAION, Conceptual Captions (CC), RedCaps, Shutterstock and WIT
    - For each shift, the level of robustness vary by the choice of dataset
  - The robustness of a mixed dataset is not additive
    - ImageNet accuracy increases by mixing dataset
    - Robustness(YFCC) < Robustness(YFCC+LAION) < Robustness(LAION)
  - However, this does not give us how to choose effective dataset for CLIP
  - Their theoretical analysis show that **filtering with pretrained model** is beneficial
    - E.g., LAION filters image-text pairs by using pre-trained CLIP

Multimodal SSL: Overview

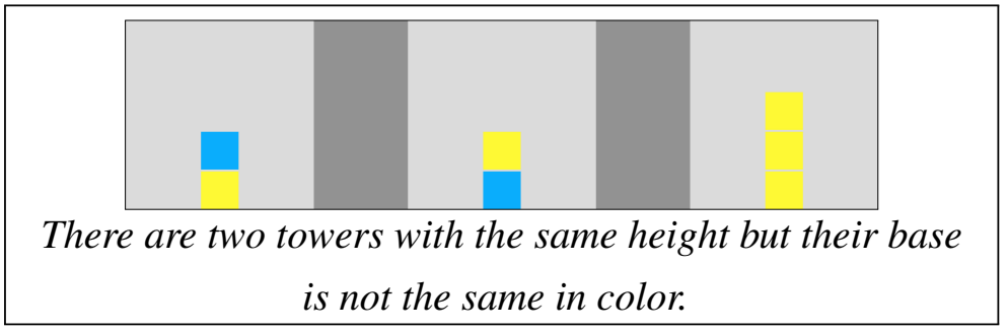
- There have been a long attempts to learn vision-language (VL) models
  - Different objective, different methods have been studied
- We discuss **four approaches** in achieving various VL representations
  1. Image-text alignment using CLIP for transferrable visual representation
     - Enables zero-shot classification & high robustness
  2. Fused transformer for vision-language understanding
     - Better vision-language understanding tasks, e.g., Visual Question Answering
  3. Learning visual representation from frozen Large Language Models (LLMs)
     - Leveraging the power of LLMs for visual in-context learning
  4. Unifying Vision-Language pretraining
     - Learning Vision-Language model from scratch for all tasks

Vision-Language Understanding Benchmarks

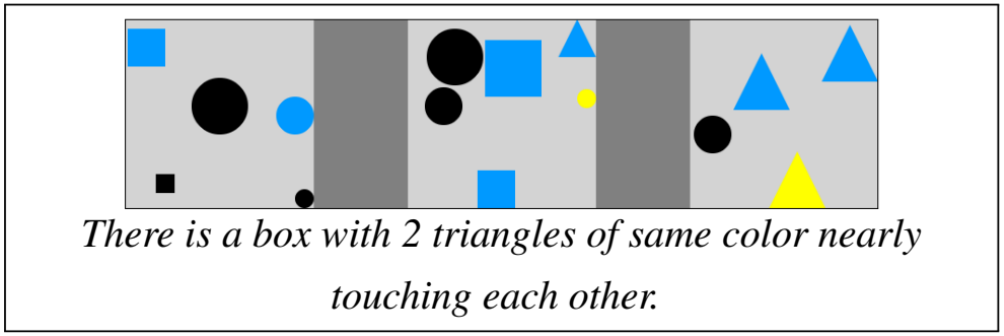
- So far, we've considered the performance on vision-only tasks, e.g., ImageNet classification
- Concurrently, many **Vision-Language Pretrained (VLP)** models are studied to do better on vision-language understanding tasks, e.g.,
  - **Visual Question Answering (VQA) [Goyal et al., 2017]**

Vision-Language Pretraining Models

- So far, we've considered the performance on vision-only tasks, e.g., ImageNet classification
- Concurrently, many **Vision-Language Pretrained (VLP)** models are studied to do better on vision-language understanding tasks, e.g.,
  - Visual Question Answering (VQA) [Goyal et al., 2017]
  - **Natural Language Visual Reasoning (NLVR) [Suhr et al., 2018]**



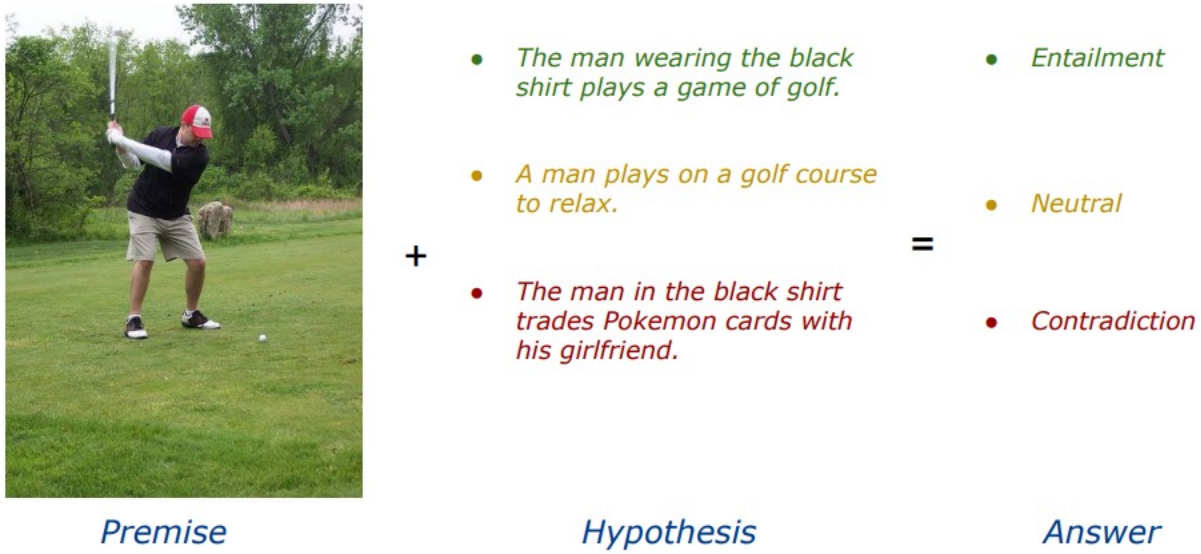
There are two towers with the same height but their base is not the same in color.



There is a box with 2 triangles of same color nearly touching each other.

Vision-Language Pretraining Models

- So far, we've considered the performance on vision-only tasks, e.g., ImageNet classification
- Concurrently, many **Vision-Language Pretrained (VLP)** models are studied to do better on vision-language understanding tasks, e.g.,
  - Visual Question Answering (VQA) [Goyal et al., 2017]
  - Natural Language Visual Reasoning (NLVR) [Suhr et al., 2018]
  - **Visual-Entailment (SNLI-VE) [Xie et al., 2019]**



Vision-Language Pretraining Models

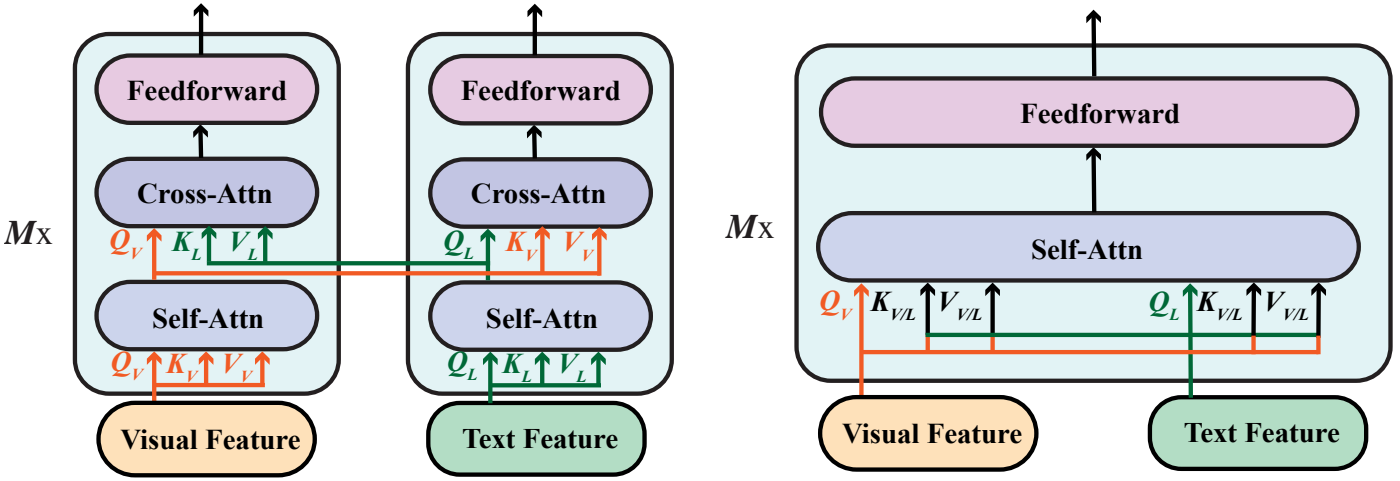
- So far, we've considered the performance on vision-only tasks, e.g., ImageNet classification
- Concurrently, many **Vision-Language Pretrained (VLP)** models are studied to do better on vision-language understanding tasks, e.g.,
  - Visual Question Answering (VQA) [Goyal et al., 2017]
  - Natural Language Visual Reasoning (NLVR) [Suhr et al., 2017]
  - Visual-Entailment (SNLI-VE) [Xie et al., 2019]

- Here, we follow the history of the VLP models by following:
  - **Development of visual encoder architectures**
    - Object detector -> CNN -> Vision Transformer (ViT)
  - **Multimodality fusion mechanism**
    - Co-attention and Merged-Attention
  - **Pre-training objectives**
    - Image-Text Matching, Masked Language Modeling, Masked Image Modeling

Earlier works on Vision-Language Pretraining Models

Earlier works focused on fusing visual and text features using attention

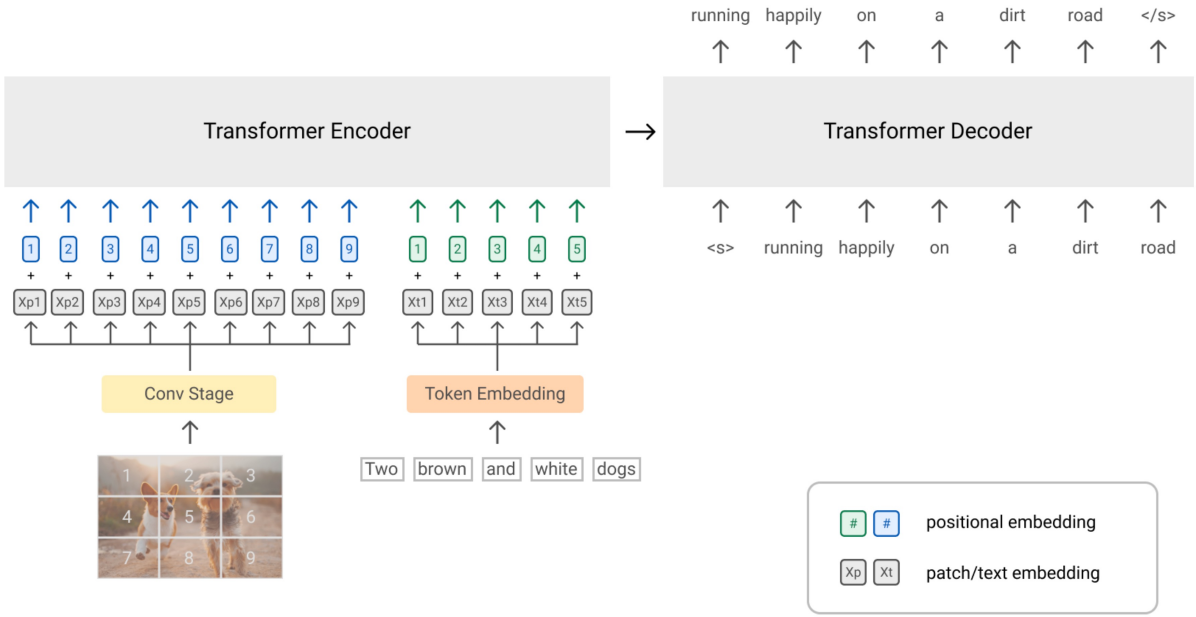
- **Co-attention**: transformer fuse vision and language encoder outputs independently
  - ViLBERT [Lu et al., 2019], LXMERT [Tan & Bansal, 2019]
- **Merged attention**: fuse image patches and text features into unified transformer
  - VisualBERT [Li et al., 2020], VL-BERT [Su et al., 2019], UNITER [Chen et al., 2020]
  - OSCAR [Li et al., 2020] uses object tags as inputs additionally
  - VinVL uses 3-way contrastive loss for VQA and image-text matching
- Pretrained (and frozen) **object detectors** (e.g., Faster R-CNN) are used for visual features



(a) Co-attention model. **(b)** Merged attention model.

End-to-end pretraining with **CNN visual encoder**

- **PixelBERT** [Huang et al., 2020] uses CNN based visual encoder and sentence encoder, and fed to transformer via cross-modality alignment
- **SimVLM** [Wang et al., 2021] uses CNN and text token embedding along with encoder-decoder architecture
- **MDETR** [Kamath et al., 2021] uses CNN and RoBERTa for image and text feature extraction, and pass to transformer with Image-Text-Box annotated data

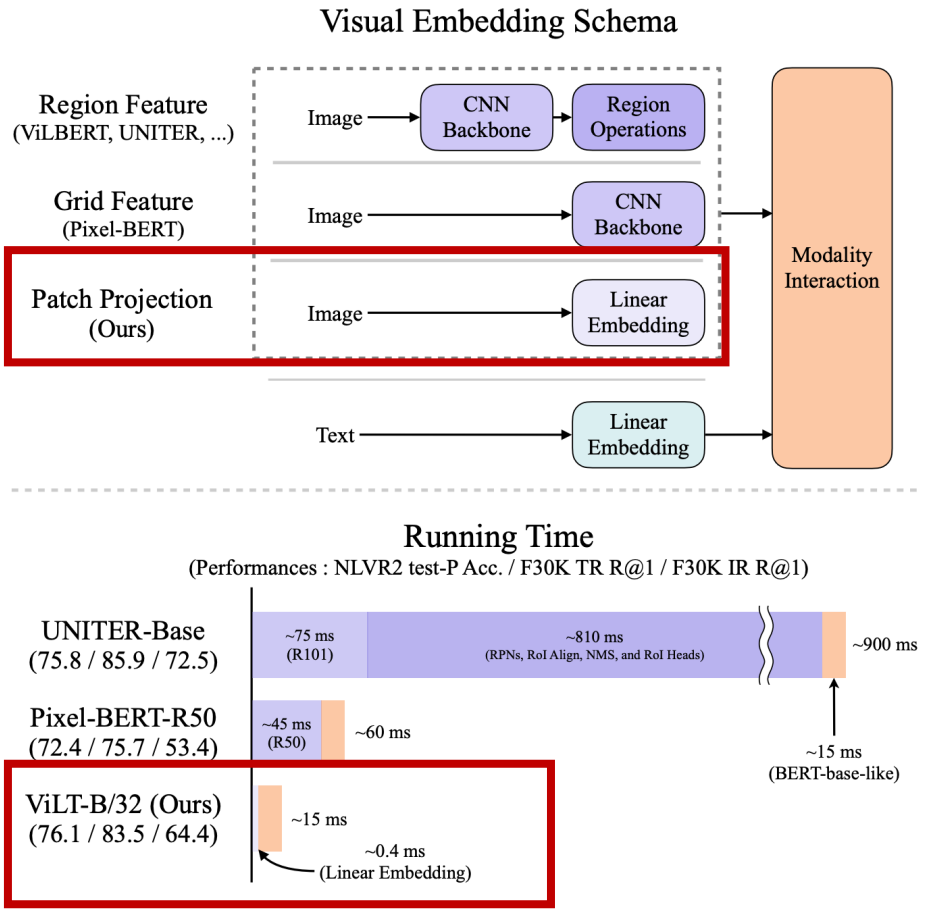


SimVLM architecture

ViLT: Vision-Language Transformer

Incorporating Vision Transformers (ViT) [Dosovitskiy et al., 2021] for VLP models

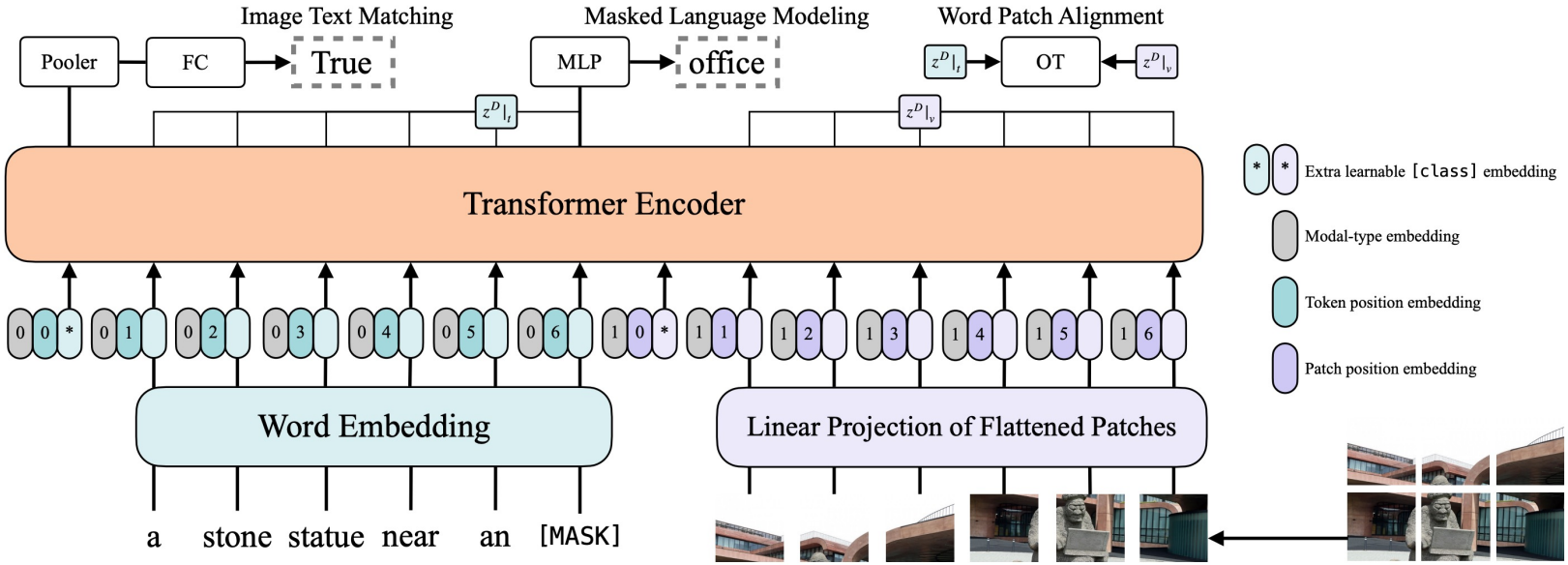
- **Vision-Language Transformer (ViLT)** [Kim et al., 2021]
  - Minimal VLP models for efficiency and expressive power



ViLT: Vision-Language Transformer

Incorporating Vision Transformers (ViT) [Dosovitskiy et al., 2021] for VLP models

- **Vision-Language Transformer (ViLT)** [Kim et al., 2021]
  - Minimal VLP models for efficiency and expressive power
  - Image patches and Text tokens are fed into **unified transformer encoder**
  - Pretraining objectives
    - Image Text Matching (ITM)
    - Masked Language Modeling (MLM)
    - Word-Patch Alignment (WPA): use optimal transport to align words & patches

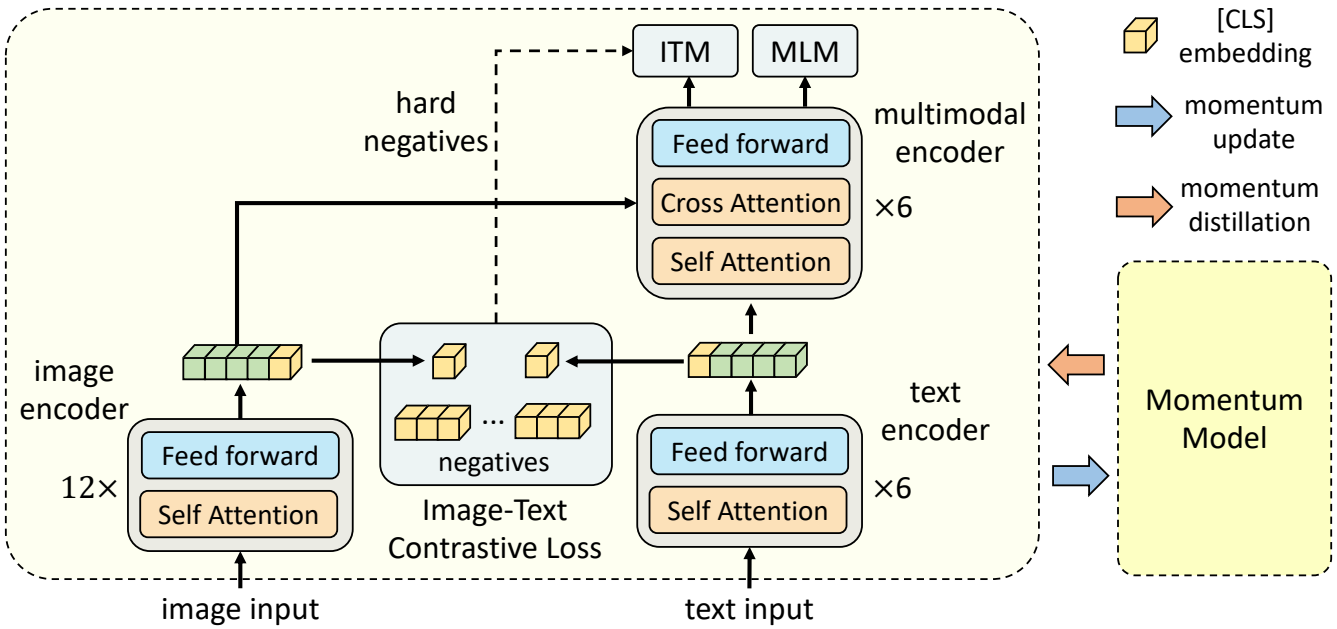


Align Before Fuse: VLP model with Momentum Distillation

Motivation: Image features and word tokens may not be aligned

Align Before Fuse (ALBEF) [Li et al., 2021]

- Use additional **multimodal encoder** to fuse information
  - Image-Text Contrastive Loss (i.e., CLIP) to align unimodal representation before fusion
  - ITM and MLM loss to learn multimodal interactions between image and text

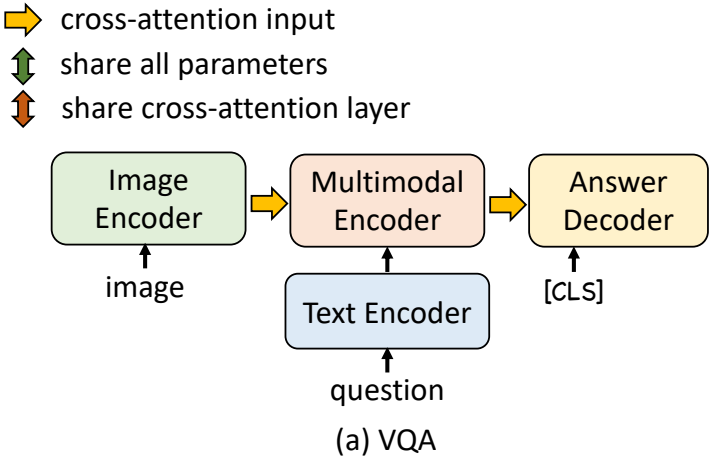


Align Before Fuse: VLP model with Momentum Distillation

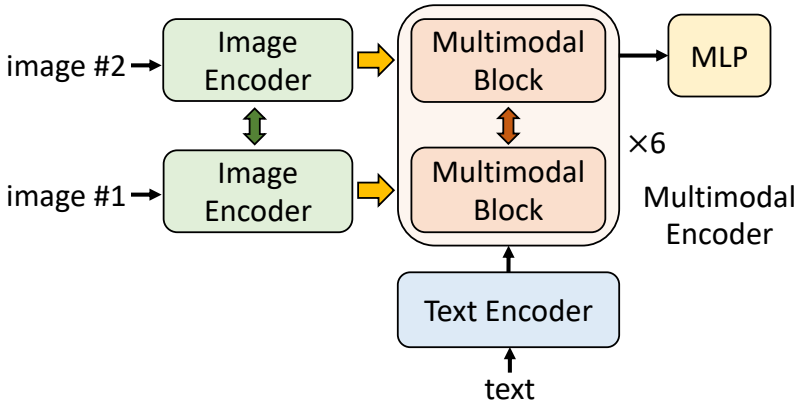
Motivation: Image features and word tokens may not be aligned

Align Before Fuse (ALBEF) [Li et al., 2021]

- Use additional **multimodal encoder** to fuse information
  - Image-Text Contrastive Loss (i.e., CLIP) to align unimodal representation before fusion
  - ITM and MLM loss to learn multimodal interactions between image and text
- **Momentum Distillation** to deal with noisy image-text pair
- Application to VQA and NLVR tasks:



(a) VQA



(b) NLVR^2

Align Before Fuse: VLP model with Momentum Distillation

- ALBEF achieved SOTA in various VL tasks (VQA, NLVR, SNLI-VE)

Method	VQA		NLVR^2		SNLI-VE	
	test-dev	test-std	dev	test-P	val	test
VisualBERT [13]	70.80	71.00	67.40	67.00	-	-
VL-BERT [10]	71.16	-	-	-	-	-
LXMERT [1]	72.42	72.54	74.90	74.50	-	-
12-in-1 [12]	73.15	-	-	78.87	-	76.95
UNITER [2]	72.70	72.91	77.18	77.85	78.59	78.28
VL-BART/T5 [54]	-	71.3	-	73.6	-	-
ViLT [21]	70.94	-	75.24	76.21	-	-
OSCAR [3]	73.16	73.44	78.07	78.36	-	-
VILLA [8]	73.59	73.67	78.39	79.30	79.47	79.03
ALBEF (4M)	74.54	74.70	80.24	80.50	80.14	80.30
ALBEF (14M)	**75.84**	**76.04**	**82.55**	**83.14**	**80.80**	**80.91**

Vision-Language Understanding Tasks

Align Before Fuse: VLP model with Momentum Distillation

- ALBEF achieved SOTA in various VL tasks (VQA, NLVR, SNLI-VE)
- Also, it outperforms other methods in image-text retrieval
  - In both zero-shot and fine-tuned cases

Method	# Pre-train Images	Flickr30K (1K test set)						MSCOCO (5K test set)					
		TR			IR			TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
UNITER	4M	87.3	98.0	99.2	75.6	94.1	96.8	65.7	88.6	93.8	52.9	79.9	88.0
VILLA	4M	87.9	97.5	98.8	76.3	94.2	96.8	-	-	-	-	-	-
OSCAR	4M	-	-	-	-	-	-	70.0	91.1	95.5	54.0	80.8	88.5
ALIGN	1.2B	95.3	99.8	100.0	84.9	97.4	98.6	77.0	93.5	96.9	59.9	83.3	89.8
ALBEF	4M	94.3	99.4	99.8	82.8	96.7	98.4	73.1	91.4	96.0	56.8	81.5	89.2
ALBEF	14M	**95.9**	**99.8**	**100.0**	**85.6**	**97.5**	**98.9**	**77.6**	**94.3**	**97.2**	**60.7**	**84.3**	**90.5**

Fine-tuned

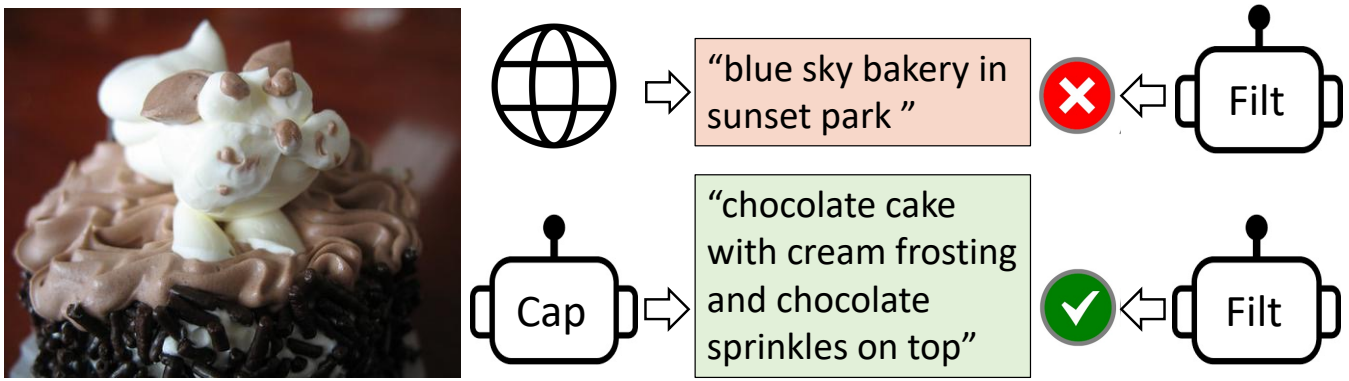
Method	# Pre-train Images	Flickr30K (1K test set)					
		TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10
UNITER [2]	4M	83.6	95.7	97.7	68.7	89.2	93.9
CLIP [6]	400M	88.0	98.7	99.4	68.7	90.6	95.2
ALIGN [7]	1.2B	88.6	98.7	99.7	75.7	93.8	96.8
ALBEF	4M	90.5	98.8	99.7	76.8	93.7	96.7
ALBEF	14M	**94.1**	**99.5**	**99.7**	**82.8**	**96.3**	**98.1**

Zero-shot

Motivation: Image-Text pairs from web are noisy

Bootstrapping Language-Image Pretraining (BLIP) [Li et al., 2022]

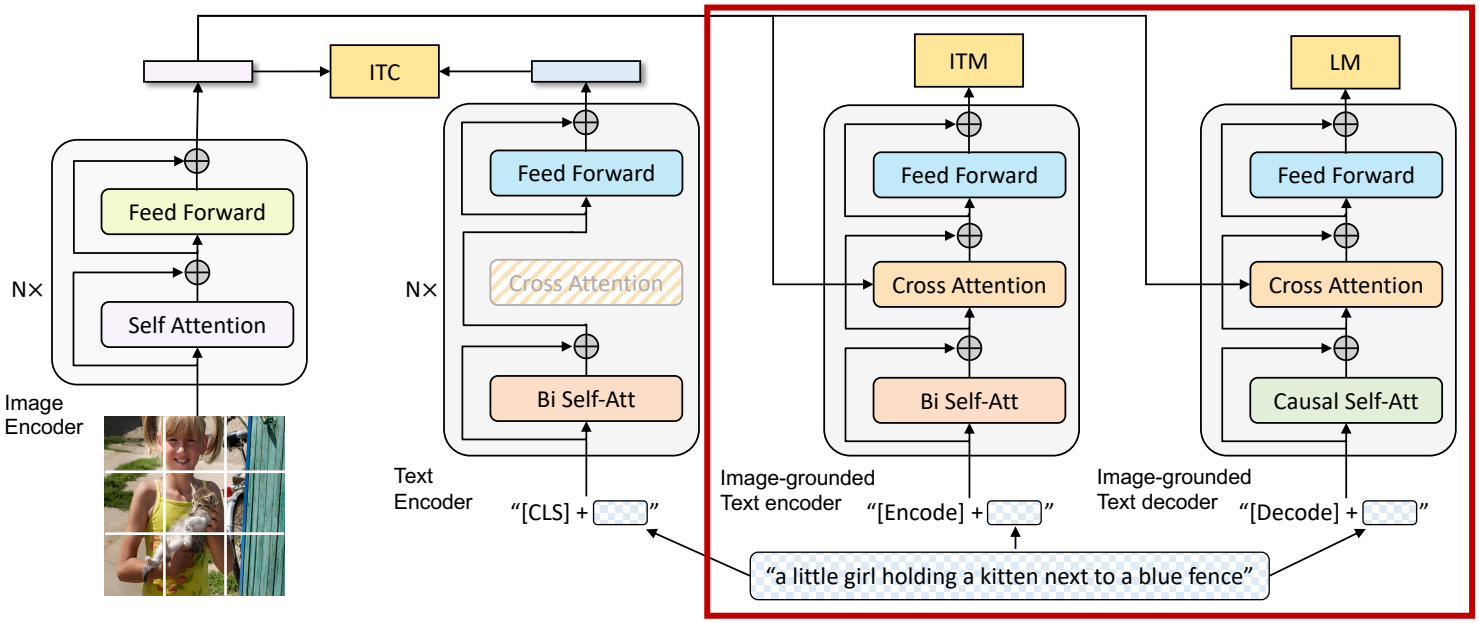
- **Key idea**: learn Captioner and Filter to bootstrap dataset
  - Captioner: generate synthetic caption
  - Filter: filter out noisy image-text pairs



Motivation: Image-Text pairs from web are noisy

Bootstrapping Language-Image Pretraining (BLIP) [Li et al., 2022]

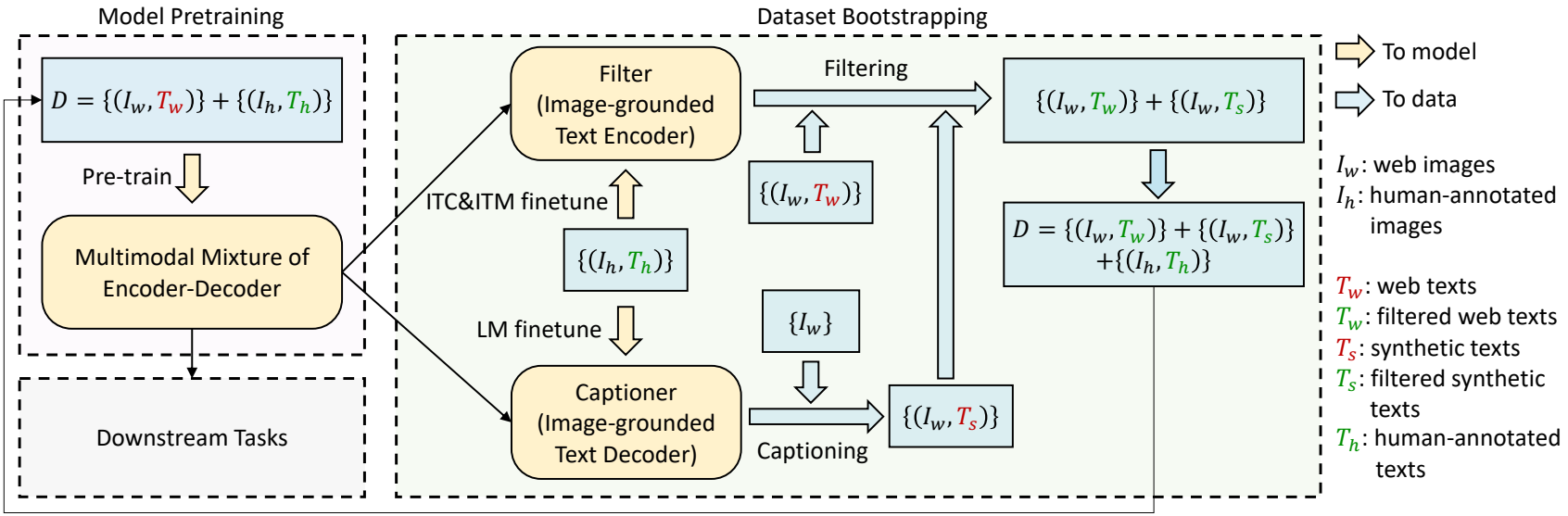
- Multimodal mixture of Encoder-Decoder (MED)
  - Contrastive loss for unimodal encoder outputs
  - **Image-grounded Text Encoder & Decoder**



Motivation: Image-Text pairs from web are noisy

Bootstrapping Language-Image Pretraining (BLIP) [Li et al., 2022]

- Multimodal mixture of Encoder-Decoder (MED)
- **CapFilt**: Bootstrapping noisy web data through Captioner and Filter
  - **Captioner** (using Image-grounded Decoder): generate synthetic captions
  - **Filter** (using Image-grounded Encoder): remove noisy captions



- Effect of bootstrapping with **CapFilt**
  - Using both Captioner and Filter consistently improves image-text retrieval

Pre-train dataset	Bootstrap C	Bootstrap F	Vision backbone	Retrieval-FT (COCO) TR@1	Retrieval-FT (COCO) IR@1	Retrieval-ZS (Flickr) TR@1	Retrieval-ZS (Flickr) IR@1	Caption-FT (COCO) B@4	Caption-FT (COCO) CIDEr	Caption-ZS (NoCaps) CIDEr	Caption-ZS (NoCaps) SPICE
COCO+VG +CC+SBU (14M imgs)	✗	✗	ViT-B/16	78.4	60.7	93.9	82.1	38.0	127.8	102.2	13.9
	✗	✓$_B$		79.1	61.5	94.1	82.8	38.1	128.2	102.7	14.0
	✓$_B$	✗		79.7	62.0	94.4	83.6	38.4	128.9	103.4	14.2
	✓$_B$	✓$_B$		80.6	63.1	94.8	84.9	38.6	129.7	105.1	14.4
COCO+VG +CC+SBU +LAION (129M imgs)	✗	✗	ViT-B/16	79.6	62.0	94.3	83.6	38.8	130.1	105.4	14.2
	✓$_B$	✓$_B$		81.9	64.3	96.0	85.0	39.4	131.4	106.3	14.3
	✓$_L$	✓$_L$		81.2	64.1	96.0	85.5	39.7	133.3	109.6	14.7
	✗	✗	ViT-L/16	80.6	64.1	95.1	85.5	40.3	135.5	112.5	14.7
	✓$_L$	✓$_L$		82.4	65.1	96.7	86.7	40.4	136.7	113.2	14.8

BLIP: Bootstrapping Language-Image Pretraining

- Effect of bootstrapping with **CapFilt**
- BLIP achieves SOTA in Image-Text Retrieval

Method	Pre-train # Images	COCO (5K test set)						Flickr30K (1K test set)					
		TR			IR			TR			IR		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
UNITER (Chen et al., 2020)	4M	65.7	88.6	93.8	52.9	79.9	88.0	87.3	98.0	99.2	75.6	94.1	96.8
VILLA (Gan et al., 2020)	4M	-	-	-	-	-	-	87.9	97.5	98.8	76.3	94.2	96.8
OSCAR (Li et al., 2020)	4M	70.0	91.1	95.5	54.0	80.8	88.5	-	-	-	-	-	-
UNIMO (Li et al., 2021b)	5.7M	-	-	-	-	-	-	89.4	98.9	99.8	78.0	94.2	97.1
ALIGN (Jia et al., 2021)	1.8B	77.0	93.5	96.9	59.9	83.3	89.8	95.3	99.8	100.0	84.9	97.4	98.6
ALBEF (Li et al., 2021a)	14M	77.6	94.3	97.2	60.7	84.3	90.5	95.9	99.8	100.0	85.6	97.5	98.9
BLIP	14M	80.6	95.2	97.6	63.1	85.3	91.1	96.6	99.8	**100.0**	87.2	97.5	98.8
BLIP	129M	**81.9**	95.4	97.8	**64.3**	85.7	91.5	**97.3**	**99.9**	**100.0**	87.3	97.6	**98.9**
$\text{BLIP}_{\text{CapFilt-L}}$	129M	81.2	**95.7**	**97.9**	64.1	**85.8**	**91.6**	97.2	**99.9**	**100.0**	**87.5**	**97.7**	**98.9**
$\text{BLIP}_{\text{ViT-L}}$	129M	82.4	95.4	97.9	65.1	86.3	91.8	97.4	99.8	99.9	87.6	97.7	99.0

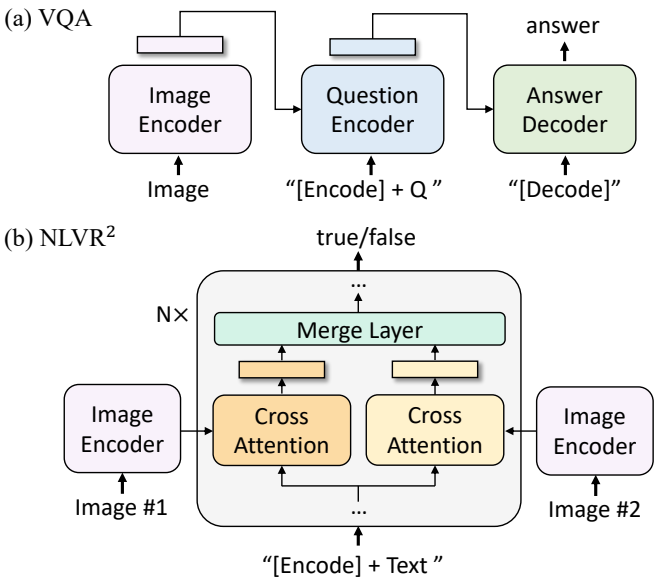
BLIP: Bootstrapping Language-Image Pretraining

- Effect of bootstrapping with **CapFilt**
- BLIP achieves SOTA in Image-Text Retrieval
- BLIP achieves comparable performance with SOTA in Image Captioning

Method	Pre-train #Images	NoCaps validation in-domain C	in-domain S	near-domain C	near-domain S	out-domain C	out-domain S	overall C	overall S	COCO Caption Karpathy test B@4	COCO Caption Karpathy test C
Enc-Dec (Changpinyo et al., 2021)	15M	92.6	12.5	88.3	12.1	94.5	11.9	90.2	12.1	-	110.9
VinVL† (Zhang et al., 2021)	5.7M	103.1	14.2	96.1	13.8	88.3	12.1	95.5	13.5	38.2	129.3
$\text{LEMON}_{\text{base}}$† (Hu et al., 2021)	12M	104.5	14.6	100.7	14.0	96.7	12.4	100.4	13.8	-	-
$\text{LEMON}_{\text{base}}$† (Hu et al., 2021)	200M	107.7	14.7	106.2	14.3	107.9	13.1	106.8	14.1	**40.3**	**133.3**
BLIP	14M	111.3	15.1	104.5	14.4	102.4	13.7	105.1	14.4	38.6	129.7
BLIP	129M	109.1	14.8	105.8	14.4	105.7	13.7	106.3	14.3	39.4	131.4
$\text{BLIP}_{\text{CapFilt-L}}$	129M	**111.8**	**14.9**	**108.6**	**14.8**	**111.5**	**14.2**	**109.6**	**14.7**	39.7	**133.3**
$\text{LEMON}_{\text{large}}$† (Hu et al., 2021)	200M	116.9	15.8	113.3	15.1	111.3	14.0	113.4	15.0	40.6	135.7
$\text{SimVLM}_{\text{huge}}$ (Wang et al., 2021)	1.8B	113.7	-	110.9	-	115.2	-	112.2	-	40.6	143.3
$\text{BLIP}_{\text{ViT-L}}$	129M	114.9	15.2	112.1	14.9	115.3	14.4	113.2	14.8	40.4	136.7

BLIP: Bootstrapping Language-Image Pretraining

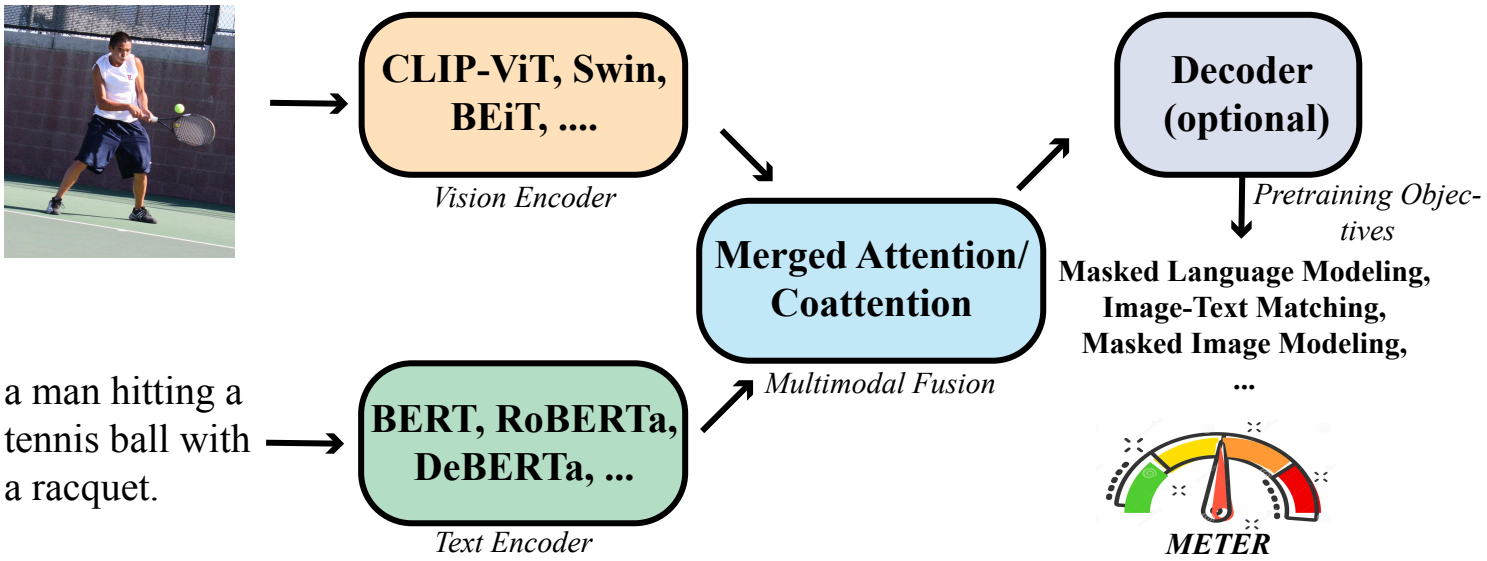
- Effect of bootstrapping with **CapFilt**
- BLIP achieves SOTA in Image-Text Retrieval
- BLIP achieves comparable performance with SOTA in Image Captioning
- BLIP shows strong empirical performance on various VL understanding tasks



Method	Pre-train #Images	VQA test-dev	VQA test-std	NLVR^2 dev	NLVR^2 test-P
LXMERT	180K	72.42	72.54	74.90	74.50
UNITER	4M	72.70	72.91	77.18	77.85
VL-T5/BART	180K	-	71.3	-	73.6
OSCAR	4M	73.16	73.44	78.07	78.36
SOHO	219K	73.25	73.47	76.37	77.32
VILLA	4M	73.59	73.67	78.39	79.30
UNIMO	5.6M	75.06	75.27	-	-
ALBEF	14M	75.84	76.04	82.55	83.14
SimVLM$_{\text{base}}$†	1.8B	77.87	78.14	81.72	81.77
BLIP	14M	77.54	77.62	**82.67**	82.30
BLIP	129M	78.24	78.17	82.48	**83.08**
BLIP$_{\text{CapFilt-L}}$	129M	**78.25**	**78.32**	82.15	82.24

Multimodal End-to-end TransformER (METER) [Dou et al., 2022]

- Extensive study on design of end-to-end transformer for VLP
- Three components that METER considered:
  - Architecture of Vision & Text encoders
  - Multimodal fusion method
  - Pretraining objectives



Exp 1. Effect of Vision & Text encoders without VLP

- Impact of **Text Encoders**
  - No significant difference between pretrained text encoders
  - **Pretrained text encoders** are better than embedding only
- Impact of **Vision Encoders**
  - Impact of **vision encoders** vary more than text encoders
- Better unimodal task performance (e.g. ImageNet Acc or MNLI) does not guarantee better VL performance

Text Enc.	VQAv2 Acc.	VE Acc.	IR R@1	TR R@1	SQuAD EM	MNLI Acc.
Emb-only	67.13	74.85	49.06	68.20	-	-
ELECTRA	69.22	76.57	41.80	58.30	86.8	**88.8**
CLIP	69.31	75.37	**54.96**	**73.80**	-	-
DeBERTa	69.40	**76.74**	51.50	67.70	**87.2**	**88.8**
BERT	69.56	76.27	49.60	66.60	76.3	84.3
RoBERTa	69.69	76.53	49.86	68.90	84.6	87.6
ALBERT	**69.94**	76.20	52.20	68.70	86.4	87.9

Impact of Text Encoders

Vision Encoder	VQAv2	VE	IR	TR	ImageNet
Dis. DeiT B-384/16	67.84	76.17	34.84	52.10	85.2
BEiT B-224/16	68.45	75.28	32.24	59.80	85.2
DeiT B-384/16	68.92	75.97	33.38	50.90	82.9
ViT B-384/16	69.09	76.35	40.30	59.80	83.97
CLIP B-224/32	69.69	76.53	49.86	68.90	-
VOLO 4-448/32	71.44	76.42	40.90	61.40	**86.8**
CaiT M-384/32	71.52	76.62	38.96	61.30	86.1
CLIP B-224/16	71.75	77.54	**57.64**	**76.90**	-
Swin B-384/32	**72.38**	**77.65**	52.30	69.50	86.4

Impact of Vision Encoders

Exp 2. Effect of Vision & Text encoders with VLP

- Using pretrained Vision & Text encoder is better
- Impact of Multimodal Fusion Module
  - **Co-attention** is better than Merged-attention(**contradict** to previous region-based VLP)
- Impact of Decoder
  - Using **Only-Encoder** is better than Encoder-Decoder
  - But decoder can be used for image captioning, e.g. BLIP

Text Enc.	Vision Enc.	VQAv2	Flickr-ZS IR	Flickr-ZS TR
Emb-only	CLIP-32	73.99	60.32	74.10
BERT	CLIP-32	74.98	66.08	78.10
	CLIP-16	76.70	74.52	87.20
RoBERTa	CLIP-32	74.67	65.50	76.60
	CLIP-16	**77.19**	**76.64**	**89.60**
	Swin	76.43	71.68	85.30

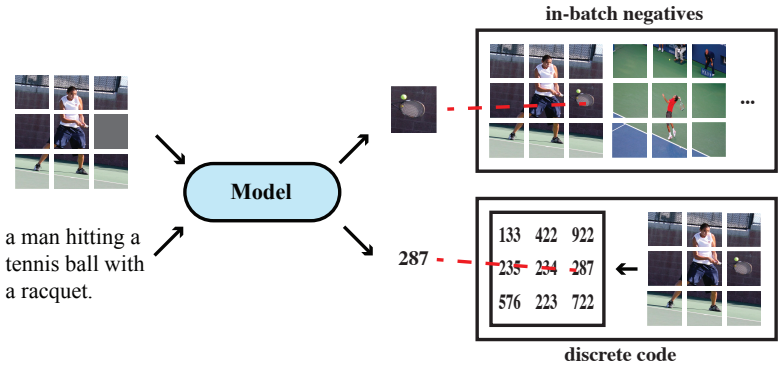
Fine-tuning with VLP

Fusion	Decoder	VQAv2	Flickr-ZS IR	Flickr-ZS TR
Merged attention	✗	74.00	57.46	73.10
Co-attention	✗	**74.98**	**66.08**	**78.10**
Co-attention	✓	74.73	48.96	71.60

Effect of attention fusion & decoder

Exp 3. Effect of Pretraining Objectives

- MLM + ITM helps VLP
- Masked Image Modeling is not helpful in VLP
  - Masked Patch Classification with In-batch Negatives (i.e., Contrastive loss)
  - Masked Patch Classification with Discrete Code (i.e., VQ-VAE, DALL-E)



Masked Image Modeling

Pre-training Objectives	VQAv2	Flickr-ZS IR	Flickr-ZS TR
MLM	74.19	-	-
ITM	72.63	53.74	71.00
MLM+ITM	**74.98**	**66.08**	**78.10**
MLM+ITM + MIM (In-batch Negatives)	74.01	62.12	76.90
MLM+ITM + MIM (Discrete Code)	74.21	59.80	76.30

Effect of Pretraining Objectives

Final METER setup

- **METER-Swin: RoBERTa-Base + Swin Transformer + Co-attention**
- **METER-CLIP: RoBERTa-Base + CLIP-ViT-Base + Co-attention**

Model	VQAv2		$NLVR^2$		SNLI-VE		Flickr-ZS					
	test-dev	test-std	dev	test	dev	test	IR@1	IR@5	IR@10	TR@1	TR@5	TR@10
Pre-trained with >10M images												
ALBEF (14M) [29]	75.84	76.04	82.55	83.14	80.80	80.91	82.8	96.3	98.1	94.1	99.5	99.7
$SimVLM_{BASE}$ (1.8B) [58]	77.87	78.14	81.72	81.77	84.20	84.15	-	-	-	-	-	-
$SimVLM_{HUGE}$ (1.8B) [58]	80.03	80.34	84.53	85.15	86.21	86.32	-	-	-	-	-	-
Pre-trained with <10M images												
$UNITER_{LARGE}$ [6]	73.82	74.02	79.12	79.98	79.39	79.38	68.74	89.20	93.86	83.60	95.70	97.70
$VILLA_{LARGE}$ [14]	74.69	74.87	79.76	81.47	80.18	80.02	-	-	-	-	-	-
$UNIMO_{LARGE}$ [31]	75.06	75.27	-	-	**81.11**	80.63	-	-	-	-	-	-
$VinVL_{LARGE}$ [65]	76.52	76.60	**82.67**	**83.98**	-	-	-	-	-	-	-	-
PixelBERT [20]	74.45	74.55	76.5	77.2	-	-	-	-	-	-		
CLIP-ViL (ResNet50x4) [48]	76.48	76.70	-	-	80.61	80.20	-	-	-	-	-	-
ViLT [65]	71.26	-	75.70	76.13	-	-	55.0	82.5	89.8	73.2	93.6	96.5
Visual Parsing [60]	74.00	74.17	77.61	78.05	-	-	-	-	-	-	-	-
ALBEF (4M) [29]	74.54	74.70	80.24	80.50	80.14	80.30	76.8	93.7	96.7	90.5	**98.8**	**99.7**
METER-$Swin_{BASE}$	76.43	76.42	82.23	82.47	80.61	80.45	71.68	91.80	95.30	85.30	97.70	99.20
METER-CLIP-ViT_{BASE}	**77.68**	**77.64**	82.33	83.05	80.86	**81.19**	**79.60**	**94.96**	**97.28**	**90.90**	98.30	99.50

Performance on VL understanding tasks

Final METER setup

- **METER-Swin: RoBERTa-Base + Swin Transformer + Co-attention**
- **METER-CLIP: RoBERTa-Base + CLIP-ViT-Base + Co-attention**

Model	Flickr						COCO					
	IR@1	IR@5	IR@10	TR@1	TR@5	TR@10	IR@1	IR@5	IR@10	TR@1	TR@5	TR@10
Pre-trained with >10M images												
ALBEF (14M) [29]	85.6	97.5	98.9	95.9	99.8	100.0	60.7	84.3	90.5	77.6	94.3	97.2
Pre-trained with <10M images												
UNITER$_{\text{LARGE}}$ [6]	75.56	94.08	96.76	87.30	98.00	99.20	52.93	79.93	87.95	65.68	88.56	93.76
VILLA$_{\text{LARGE}}$ [14]	76.26	94.24	96.84	87.90	97.50	98.80	-	-	-	-	-	-
UNIMO$_{\text{LARGE}}$ [31]	78.04	94.24	97.12	89.40	98.90	99.80	-	-	-	-	-	-
VinVL$_{\text{LARGE}}$ [65]	-	-	-	-	-	-	**58.8**	**83.5**	**90.3**	75.4	92.9	96.2
PixelBERT [20]	71.5	92.1	95.8	87.0	98.9	99.5	50.1	77.6	86.2	63.6	87.5	93.6
ViLT [65]	64.4	88.7	93.8	83.5	96.7	98.6	42.7	72.9	83.1	61.5	86.3	92.7
Visual Parsing [60]	73.5	93.1	96.4	87.0	98.4	99.5	-	-	-	-	-	-
ALBEF (4M) [29]	**82.8**	**96.7**	**98.4**	**94.3**	99.4	99.8	56.8	81.5	89.2	73.1	91.4	96.0
METER-Swin$_{\text{BASE}}$	79.02	95.58	98.04	92.40	99.00	99.50	54.85	81.41	89.31	72.96	92.02	96.26
METER-CLIP-ViT$_{\text{BASE}}$	82.22	96.34	**98.36**	**94.30**	**99.60**	**99.90**	57.08	82.66	90.07	**76.16**	**93.16**	**96.82**

Performance on Image-Text retrieval

Multimodal SSL: Overview

- There have been a long attempts to learn vision-language (VL) models
  - Different objective, different methods have been studied
- We discuss **four approaches** in achieving various VL representations
  1. Image-text alignment using CLIP for transferrable visual representation
     - Enables zero-shot classification & high robustness
  2. Fused transformer for vision-language understanding
     - Better vision-language understanding tasks, e.g., Visual Question Answering
  3. Learning visual representation from frozen Large Language Models (LLMs)
     - Leveraging the power of LLMs for visual in-context learning
  4. Unifying Vision-Language pretraining
     - Learning Vision-Language model from scratch for all tasks

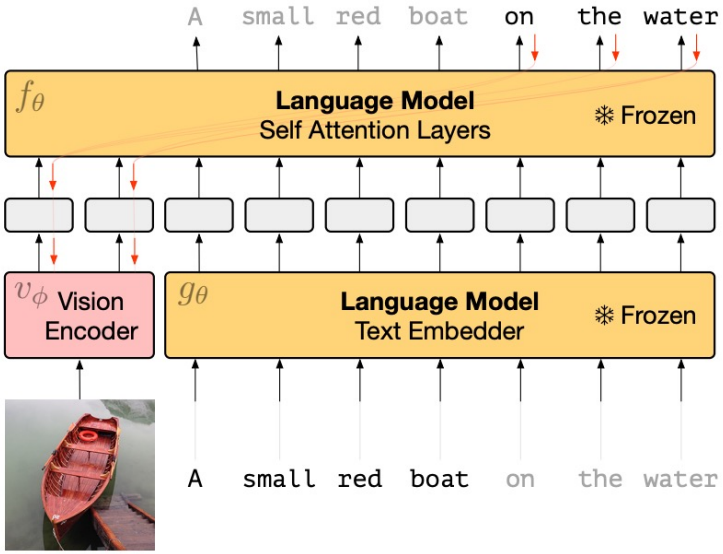
Frozen: Multimodal Few-Shot Learning with Frozen Language Models

Frozen [Tsimpoukelli et al., 2020]

- Large Language Models (LLMs) are effective few-shot learners [Brown et al., 2020]
- How we can leverage the ability of LLMs for visual few-shot learning?

Frozen [Tsimpoukelli et al., 2020]

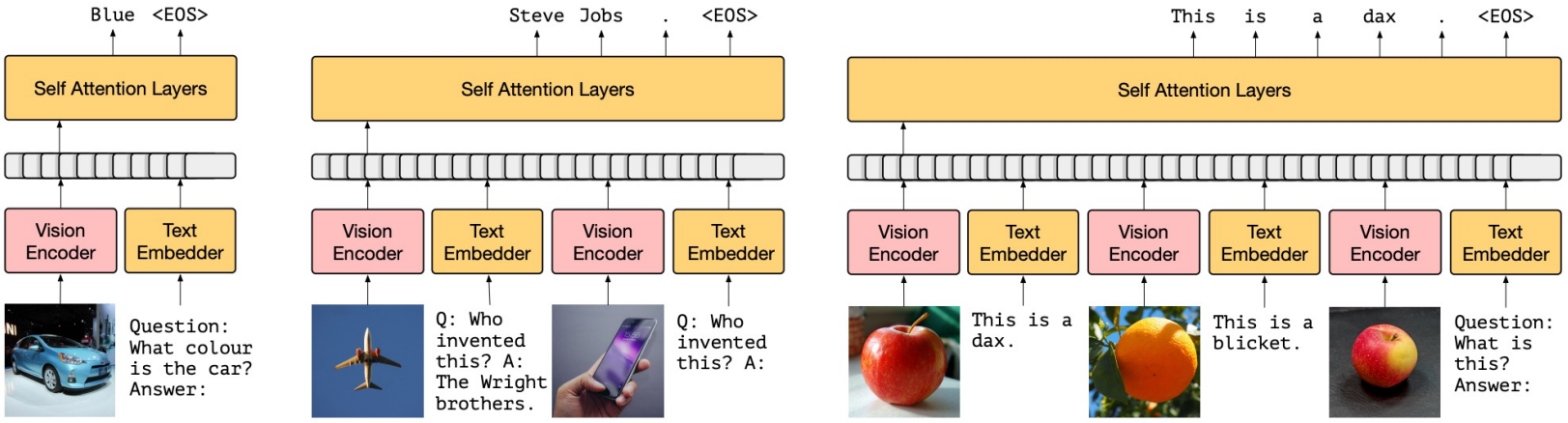
- Large Language Models (LLMs) are effective few-shot learners [Brown et al., 2020]
- How we can leverage the ability of LLMs for visual few-shot learning?
- Given pretrained vision encoder (e.g., ResNet50) and LLMs (e.g., GPT-3), Frozen only updates vision encoder for image-text alignment
  - Use linear mapping to embed into LLM
  - Autoregressive captioning loss is used for training



Frozen: Multimodal Few-Shot Learning with Frozen Language Models

Frozen [Tsimpoukelli et al., 2020]

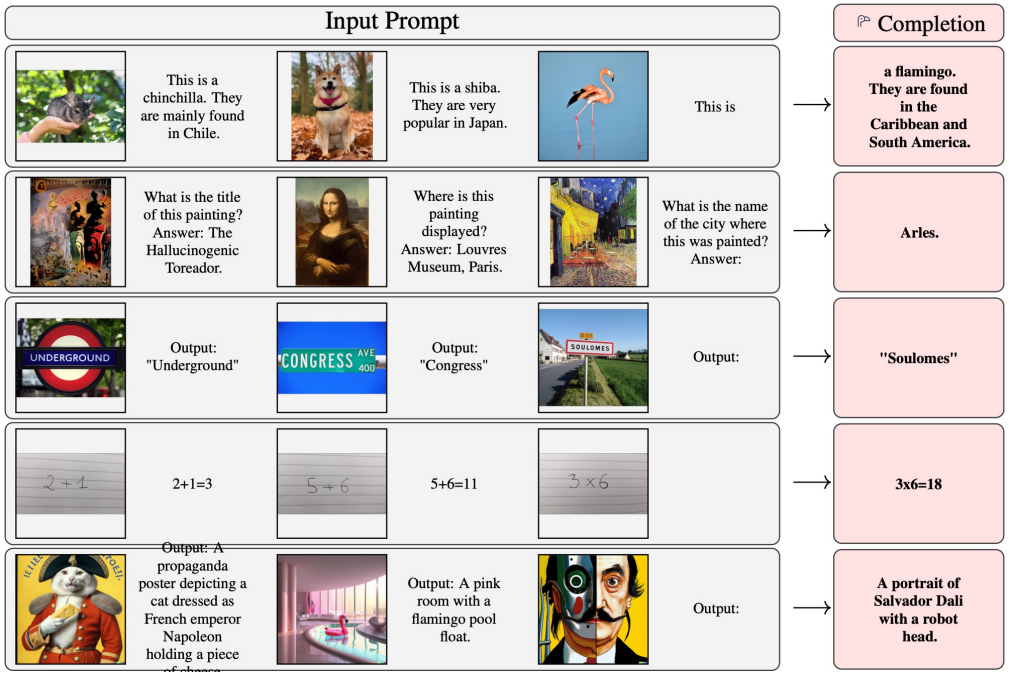
- Large Language Models (LLMs) are effective few-shot learners [Brown et al., 2020]
- How we can leverage the ability of LLMs for visual few-shot learning?
- Given pretrained vision encoder (e.g., ResNet50) and LLMs (e.g., GPT-3), Frozen only updates vision encoder for image-text alignment
  - Use linear mapping to embed into LLM
  - Autoregressive captioning loss is used for training
- Frozen enables few-shot classification as well as few-shot VQA



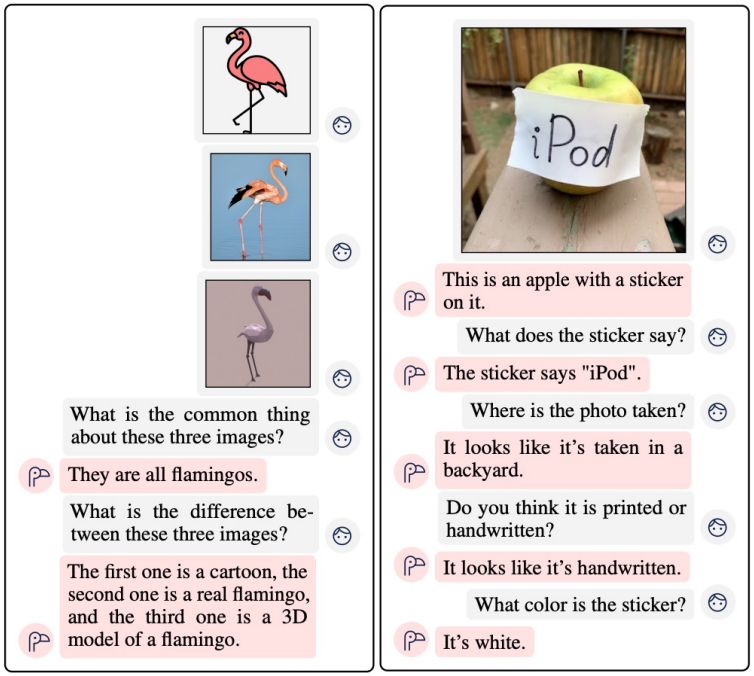
(a) **0-shot VQA** (b) **1-shot outside-knowledge VQA** (c) **Few-shot image classification**

Flamingo [Alayrac et al., 2022]

- Better VL models for few-shot learning by
  - Bridging pre-trained vision-only and language-only models
  - Can handle sequences of arbitrary visual and textual data
  - Seamlessly ingest images or videos as inputs



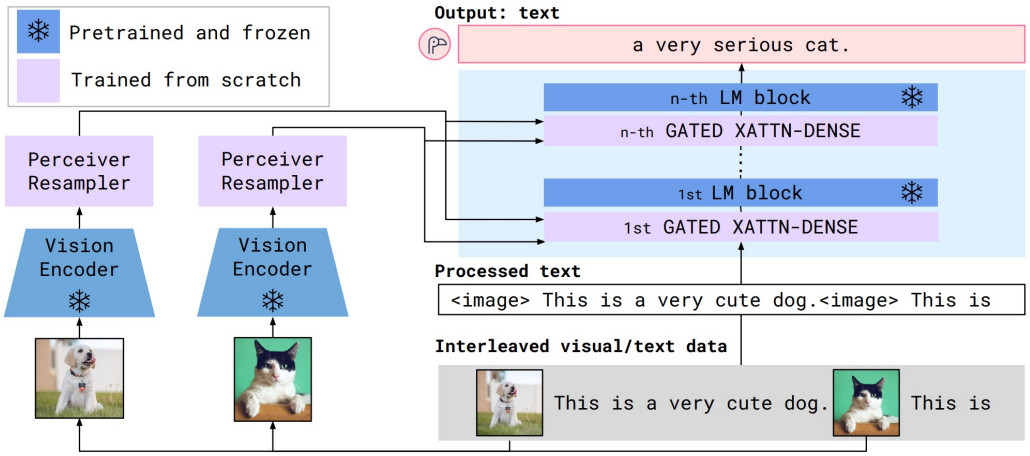
Multimodal In-Context Learning



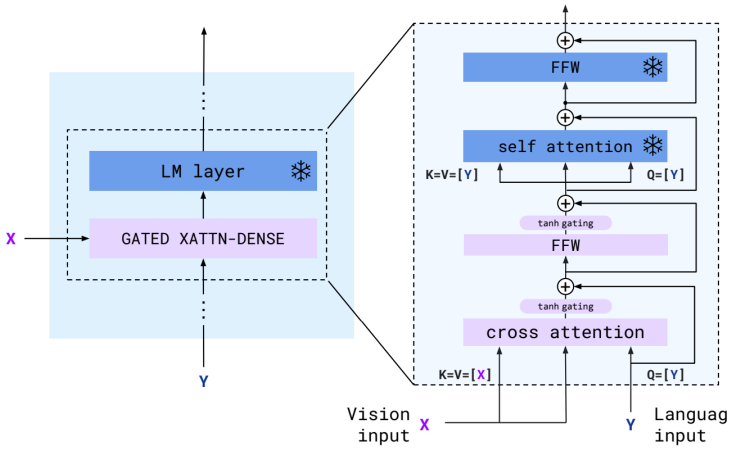
Multimodal visual dialogue

Flamingo [Alayrac et al., 2022]

- Better pretrained vision and language model
  - Vision encoder pretrained from CLIP-like objective with more data
  - Used 1.4B, 7B, 70B Chinchilla model for LLM
  - New **Perceiver-Resampler** module for vision-language alignment
  - Gated Cross-attention dense **(GATED XATTN-DENSE)** layers for vision-language fusion



Perceiver-Resampler Architecture



GATED-XATTN-DENSE layer

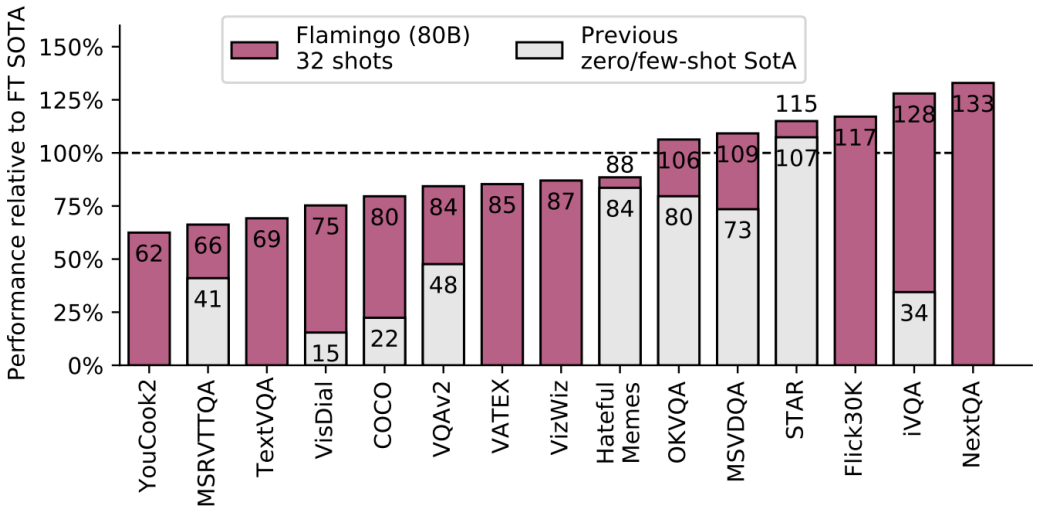
Flamingo [Alayrac et al., 2022]

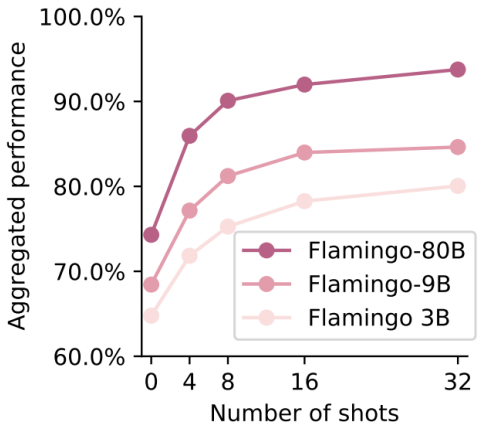
- MultiModal MassiveWeb (M3W) dataset – Mixture of datasets
  - Extract text and images from HTML of 43M webpages
  - Special tokens: Use `<image>` token to determine locations of images and `<EOC>` prior to image and end of document
  - Also use 1.8B image-text pairs from ALIGN and 27M video-text pairs
  - Use autoregressive captioning loss, weighted per dataset

$$\sum_{m=1}^{M} \lambda_m \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}_m} \left[-\sum_{\ell=1}^{L} \log p(y_\ell | y_{<\ell}, x_{\leq\ell}) \right]$$

Flamingo [Alayrac et al., 2022]

- Flamingo outperforms (6 out of 16) existing SOTA fine-tuned models with no fine-tuning



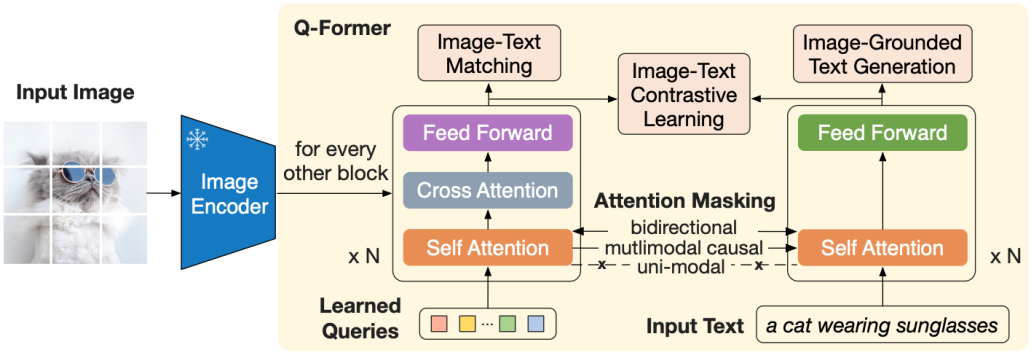


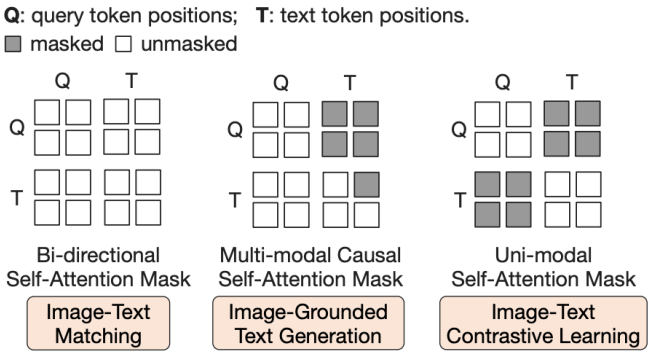
- When fine-tuned, it achieves SOTA various tasks

Method	VQAV2		COCO	VATEX	VizWiz		MSRVTTQA	VisDial		YouCook2	TextVQA		HatefulMemes
	test-dev	test-std	test	test	test-dev	test-std	test	valid	test-std	valid	valid	test-std	test seen
32 shots	67.6	-	113.8	65.1	49.8	-	31.0	56.8	-	86.8	36.0	-	70.0
Fine-tuned	**82.0**	**82.1**	138.1	**84.2**	**65.7**	**65.4**	**47.4**	61.8	59.7	118.6	**57.1**	54.1	**86.6**
SotA	81.3† [133]	81.3† [133]	**149.6**† [119]	81.4† [153]	57.2† [65]	60.6† [65]	46.8 [51]	**75.2** [79]	**75.4**† [123]	**138.7** [132]	54.7 [137]	**73.7** [84]	84.6† [152]

BLIP-2 [Li et al., 2023]

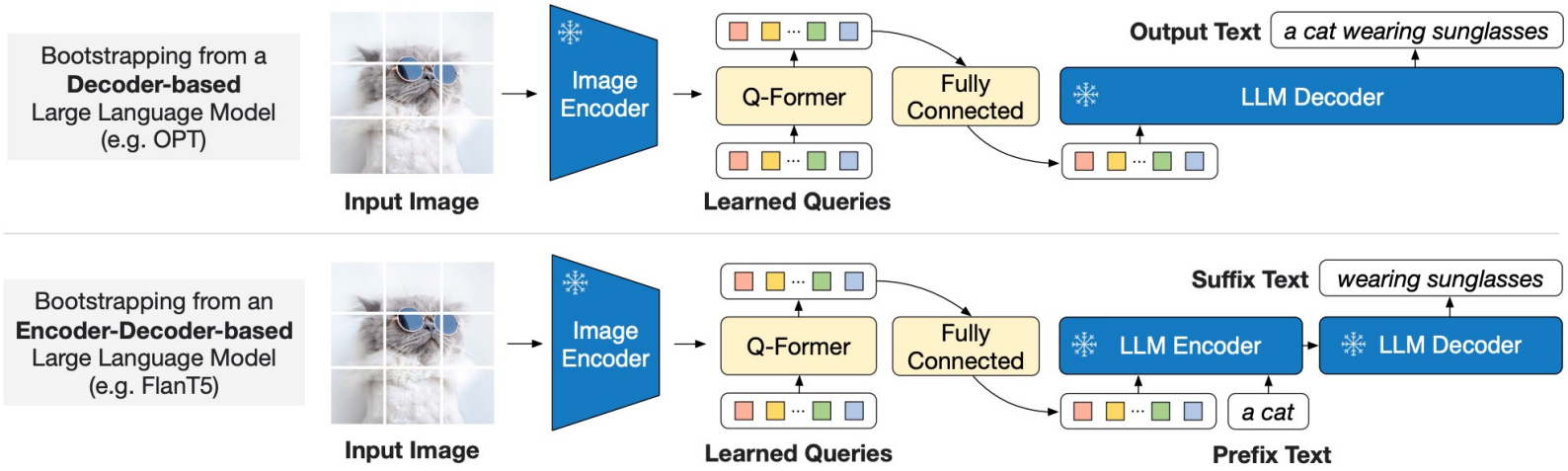
- Lighter approach for aligning pretrained vision encoder and LLM for VL tasks
- Propose two-stage alignment using **Q-former**
  - **Stage 1**: Representation learning with Q-former
    - Q-former: BERT initialized transformer that encodes visual information given query
    - Various learning objectives used
      - Image-Text Matching (binary classification loss)
      - Image-Text Contrastive Learning (i.e., CLIP loss)
      - Image-grounded text generation (i.e., captioning loss)





BLIP-2 [Li et al., 2023]

- Lighter approach for aligning pretrained vision encoder and LLM for VL tasks
- Propose two-stage alignment using **Q-former**
  - **Stage 1**: Representation learning with Q-former
  - **Stage 2**: Bootstrapping with Frozen LLM
    - Can be applied to both decoder-based / encoder-decoder-based LLM



BLIP-2 [Li et al., 2023]

- BLIP-2 achieves SOTA on zero-shot VL tasks

Models	#Trainable Params	Open-sourced?	**Visual Question Answering** VQAv2 (test-dev) VQA acc.	**Image Captioning** NoCaps (val) CIDEr	SPICE	**Image-Text Retrieval** Flickr (test) TR@1	IR@1
BLIP (Li et al., 2022)	583M	✓	-	113.2	14.8	96.7	86.7
SimVLM (Wang et al., 2021b)	1.4B	✗	-	112.2	-	-	-
BEIT-3 (Wang et al., 2022b)	1.9B	✗	-	-	-	94.9	81.5
Flamingo (Alayrac et al., 2022)	10.2B	✗	56.3	-	-	-	-
BLIP-2	188M	✓	**65.0**	**121.6**	**15.8**	**97.6**	**89.7**

Models	#Trainable Params	#Total Params	VQAv2 val	VQAv2 test-dev	OK-VQA test	GQA test-dev
VL-T5$_{\text{no-vqa}}$	224M	269M	13.5	-	5.8	6.3
FewVLM (Jin et al., 2022)	740M	785M	47.7	-	16.5	29.3
Frozen (Tsimpoukelli et al., 2021)	40M	7.1B	29.6	-	5.9	-
VLKD (Dai et al., 2022)	406M	832M	42.6	44.5	13.3	-
Flamingo3B (Alayrac et al., 2022)	1.4B	3.2B	-	49.2	41.2	-
Flamingo9B (Alayrac et al., 2022)	1.8B	9.3B	-	51.8	44.7	-
Flamingo80B (Alayrac et al., 2022)	10.2B	80B	-	56.3	**50.6**	-
BLIP-2 ViT-L OPT$_{2.7B}$	104M	3.1B	50.1	49.7	30.2	33.9
BLIP-2 ViT-G OPT$_{2.7B}$	107M	3.8B	53.5	52.3	31.7	34.6
BLIP-2 ViT-G OPT$_{6.7B}$	108M	7.8B	54.3	52.6	36.4	36.4
BLIP-2 ViT-L FlanT5$_{XL}$	103M	3.4B	62.6	62.3	39.4	44.4
BLIP-2 ViT-G FlanT5$_{XL}$	107M	4.1B	63.1	63.0	40.7	44.2
BLIP-2 ViT-G FlanT5$_{XXL}$	108M	12.1B	**65.2**	**65.0**	45.9	**44.7**

BLIP-2 [Li et al., 2023]

- BLIP-2 achieves SOTA on zero-shot VL tasks
- Also it achieves SOTA on image-text retrieval tasks, outperforming various dual encoder-based (e.g., CLIP) or fusion-encoder based models

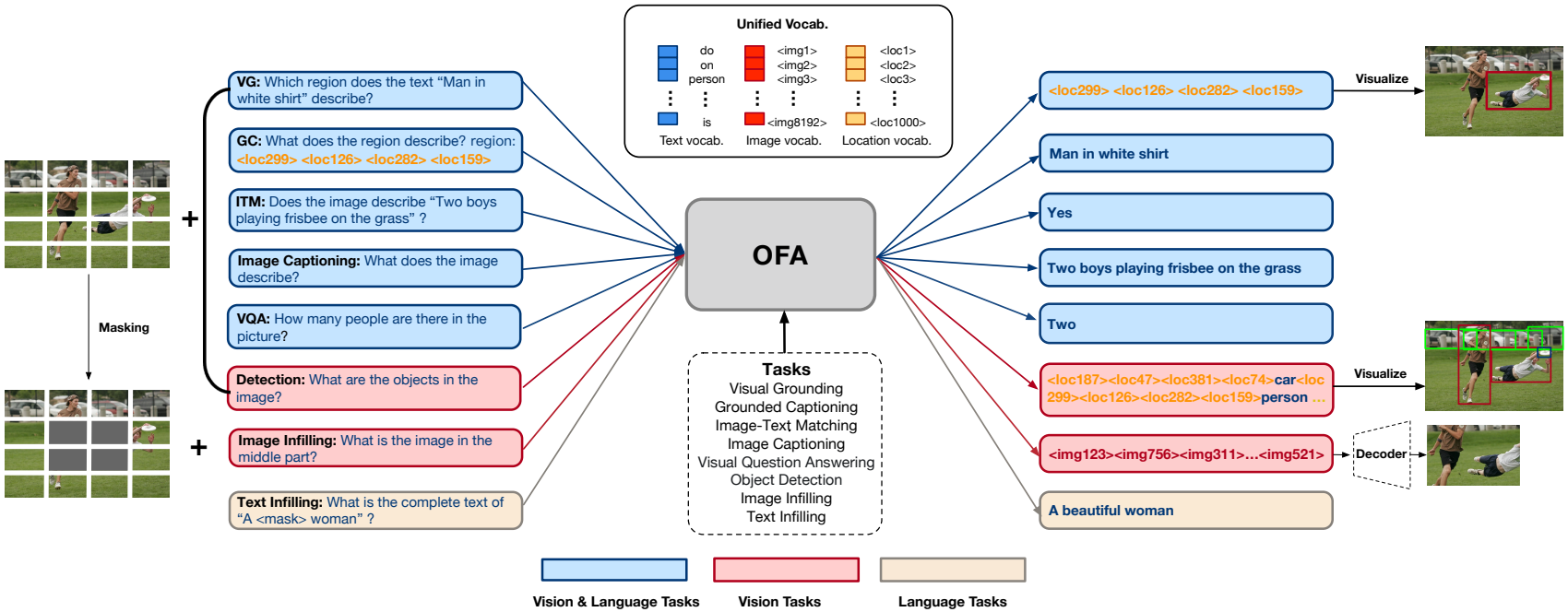
Model	#Trainable Params	Flickr30K Zero-shot (1K test set)						COCO Fine-tuned (5K test set)					
		Image → Text			Text → Image			Image → Text			Text → Image		
		R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Dual-encoder models													
CLIP (Radford et al., 2021)	428M	88.0	98.7	99.4	68.7	90.6	95.2	-	-	-	-	-	-
ALIGN (Jia et al., 2021)	820M	88.6	98.7	99.7	75.7	93.8	96.8	77.0	93.5	96.9	59.9	83.3	89.8
FILIP (Yao et al., 2022)	417M	89.8	99.2	99.8	75.0	93.4	96.3	78.9	94.4	97.4	61.2	84.3	90.6
Florence (Yuan et al., 2021)	893M	90.9	99.1	-	76.7	93.6	-	81.8	95.2	-	63.2	85.7	-
BEIT-3(Wang et al., 2022b)	1.9B	94.9	99.9	**100.0**	81.5	95.6	97.8	84.8	96.5	98.3	67.2	**87.7**	**92.8**
Fusion-encoder models													
UNITER (Chen et al., 2020)	303M	83.6	95.7	97.7	68.7	89.2	93.9	65.7	88.6	93.8	52.9	79.9	88.0
OSCAR (Li et al., 2020)	345M	-	-	-	-	-	-	70.0	91.1	95.5	54.0	80.8	88.5
VinVL (Zhang et al., 2021)	345M	-	-	-	-	-	-	75.4	92.9	96.2	58.8	83.5	90.3
Dual encoder + Fusion encoder reranking													
ALBEF (Li et al., 2021)	233M	94.1	99.5	99.7	82.8	96.3	98.1	77.6	94.3	97.2	60.7	84.3	90.5
BLIP (Li et al., 2022)	446M	96.7	**100.0**	**100.0**	86.7	97.3	98.7	82.4	95.4	97.9	65.1	86.3	91.8
BLIP-2 ViT-L	474M	96.9	**100.0**	**100.0**	88.6	97.6	**98.9**	83.5	96.0	98.0	66.3	86.5	91.8
BLIP-2 ViT-G	1.2B	**97.6**	**100.0**	**100.0**	**89.7**	**98.1**	**98.9**	**85.4**	**97.0**	**98.5**	**68.3**	**87.7**	92.6

Multimodal SSL: Overview

- There have been a long attempts to learn vision-language (VL) models
  - Different objective, different methods have been studied
- We discuss **four approaches** in achieving various VL representations
  1. Image-text alignment using CLIP for transferrable visual representation
     - Enables zero-shot classification & high robustness
  2. Fused transformer for vision-language understanding
     - Better vision-language understanding tasks, e.g., Visual Question Answering
  3. Learning visual representation from frozen Large Language Models (LLMs)
     - Leveraging the power of LLMs for visual in-context learning
  4. Unifying Vision-Language pretraining
     - Learning Vision-Language model from scratch for all tasks

One For ALL (OFA) [Wang et al., 2022]

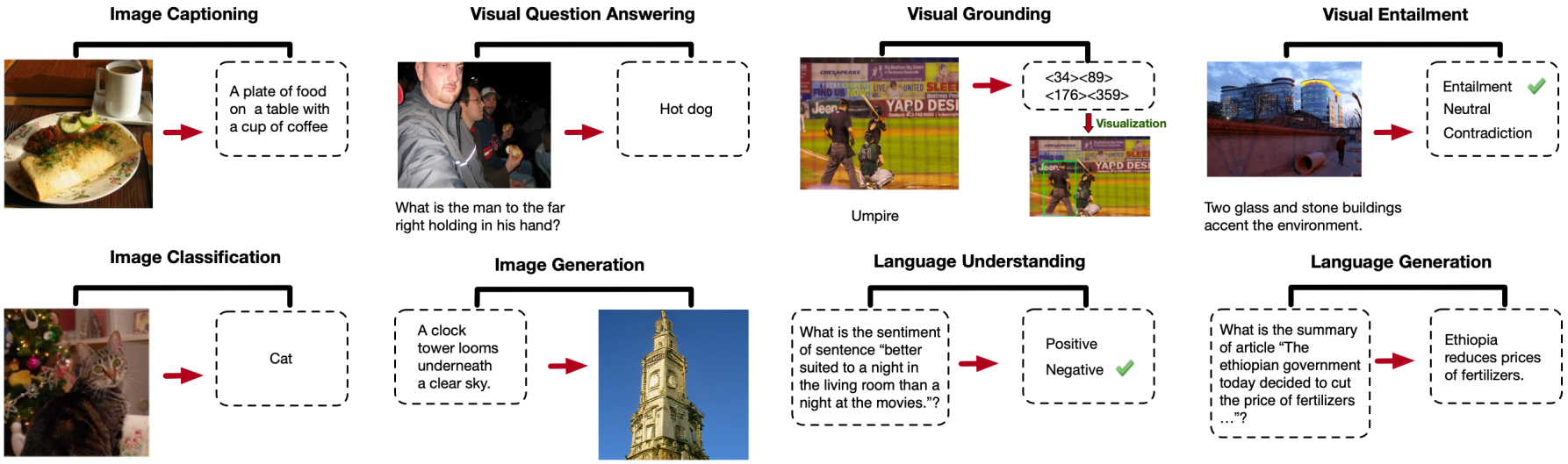
- A **unified transformer** model for various modalities (image, text, location)
- Pretrain with all Vision, Language, and Vision-Language tasks
  - Incorporate all available images (with labels), texts, image-text pairs



One For ALL (OFA) [Wang et al., 2022]

- A **unified transformer** model for various modalities (image, text, location)
- Pretrain with all Vision, Language, and Vision-Language tasks
  - Incorporate all available images (with labels), texts, image-text pairs

- OFA supports various unimodal and multimodal tasks with decent performance



One For ALL (OFA) [Wang et al., 2022]

- A **unified transformer** model for various modalities (image, text, location)
- Pretrain with all Vision, Language, and Vision-Language tasks
  - Incorporate all available images (with labels), texts, image-text pairs

- OFA supports various **multimodal** and unimodal tasks with decent performance

Model	VQA		SNLI-VE	
	test-dev	test-std	dev	test
UNITER [14]	73.8	74.0	79.4	79.4
OSCAR [15]	73.6	73.8	-	-
VILLA [16]	74.7	74.9	80.2	80.0
VL-T5 [56]	-	70.3	-	-
VinVL [17]	76.5	76.6	-	-
UNIMO [46]	75.0	75.3	81.1	80.6
ALBEF [69]	75.8	76.0	80.8	80.9
METER [70]	77.7	77.6	80.9	81.2
VLMo [48]	79.9	80.0	-	-
SimVLM [22]	80.0	80.3	86.2	86.3
Florence [23]	80.2	80.4	-	-
OFA_{Tiny}	70.3	70.4	85.3	85.2
$\text{OFA}_{\text{Medium}}$	75.4	75.5	86.6	87.0
OFA_{Base}	78.0	78.1	89.3	89.2
$\text{OFA}_{\text{Large}}$	80.3	80.5	90.3	90.2
OFA	**82.0**	**82.0**	**91.0**	**91.2**

VL Understanding

Model	Cross-Entropy Optimization				CIDEr Optimization			
	BLEU@4	METEOR	CIDEr	SPICE	BLEU@4	METEOR	CIDEr	SPICE
VL-T5 [56]	34.5	28.7	116.5	21.9	-	-	-	-
OSCAR [15]	37.4	30.7	127.8	23.5	41.7	30.6	140.0	24.5
UNICORN [57]	35.8	28.4	119.1	21.5	-	-	-	-
VinVL [17]	38.5	30.4	130.8	23.4	41.0	31.1	140.9	25.2
UNIMO [46]	39.6	-	127.7	-	-	-	-	-
LEMON [71]	41.5	30.8	139.1	24.1	42.6	31.4	145.5	25.5
SimVLM [22]	40.6	**33.7**	143.3	**25.4**	-	-	-	-
OFA_{Tiny}	35.9	28.1	119.0	21.6	38.1	29.2	128.7	23.1
$\text{OFA}_{\text{Medium}}$	39.1	30.0	130.4	23.2	41.4	30.8	140.7	24.8
OFA_{Base}	41.0	30.9	138.2	24.2	42.8	31.7	146.7	25.8
$\text{OFA}_{\text{Large}}$	42.4	31.5	142.2	24.5	43.6	32.2	150.7	26.2
OFA	**43.9**	31.8	**145.3**	24.8	**44.9**	**32.5**	**154.9**	**26.6**

Image Captioning

One For ALL (OFA) [Wang et al., 2022]

- A **unified transformer** model for various modalities (image, text, location)
- Pretrain with all Vision, Language, and Vision-Language tasks
  - Incorporate all available images (with labels), texts, image-text pairs

- OFA supports various multimodal and **unimodal** tasks with decent performance

Model	SST-2	RTE	MRPC	QQP	MNLI	QNLI
Multimodal Pretrained Baseline Models						
VisualBERT [38]	89.4	56.6	71.9	89.4	81.6	87.0
UNITER [14]	89.7	55.6	69.3	89.2	80.9	86.0
VL-BERT [8]	89.8	55.7	70.6	89.0	81.2	86.3
VilBERT [13]	90.4	53.7	69.0	88.6	79.9	83.8
LXMERT [40]	90.2	57.2	69.8	75.3	80.4	84.2
Uni-Perceiver [61]	90.2	64.3	86.6	87.1	81.7	89.9
SimVLM [22]	90.9	63.9	75.2	90.4	83.4	88.6
FLAVA [60]	90.9	57.8	81.4	90.4	80.3	87.3
UNIMO [46]	96.8	-	-	-	89.8	-
Natural-Language-Pretrained SOTA Models						
BERT [2]	93.2	70.4	88.0	91.3	86.6	92.3
RoBERTa [28]	96.4	86.6	90.9	92.2	90.2	93.9
XLNet [25]	**97.0**	85.9	90.8	92.3	90.8	94.9
ELECTRA [82]	96.9	88.0	90.8	92.4	90.9	95.0
DeBERTa [83]	96.8	88.3	**91.9**	92.3	**91.1**	**95.3**
Ours						
OFA	96.6	**91.0**	91.7	**92.5**	90.2	94.8

Language tasks

Model	Top-1 Acc.
EfficientNet-B7 [89]	84.3
ViT-L/16 [6]	82.5
DINO [90]	82.8
SimCLR v2 [32]	82.9
MoCo v3 [35]	84.1
$BEiT_{384}$-L/16 [36]	**86.3**
MAE-L/16 [37]	85.9
OFA	85.6

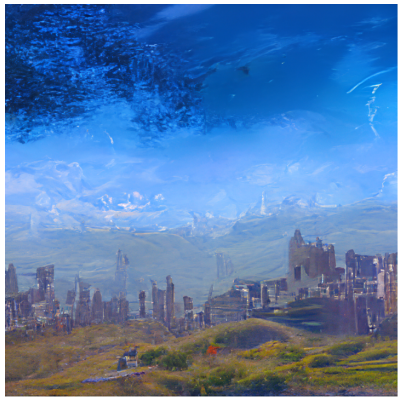
Image classification task

- Also, OFA can generate image from text
  - Showing better performance than DALLE, NUWA

Model	FID↓	CLIPSIM↑	IS↑
DALLE [50]	27.5	-	17.9
CogView [51]	27.1	33.3	18.2
GLIDE [77]	12.2	-	-
Unifying [78]	29.9	30.9	-
NÜWA [52]	12.9	34.3	27.2
OFA	**10.5**	**34.4**	**31.1**

OFA: Unifying Architectures, Tasks and Modalities

- Also, OFA can generate image from text
  - Showing better performance than DALLE, NUWA



An eagle view of a magic city.

A pathway to a temple with sakura trees in full bloom, HD.

A beautiful painting of native forest landscape photography, HD.

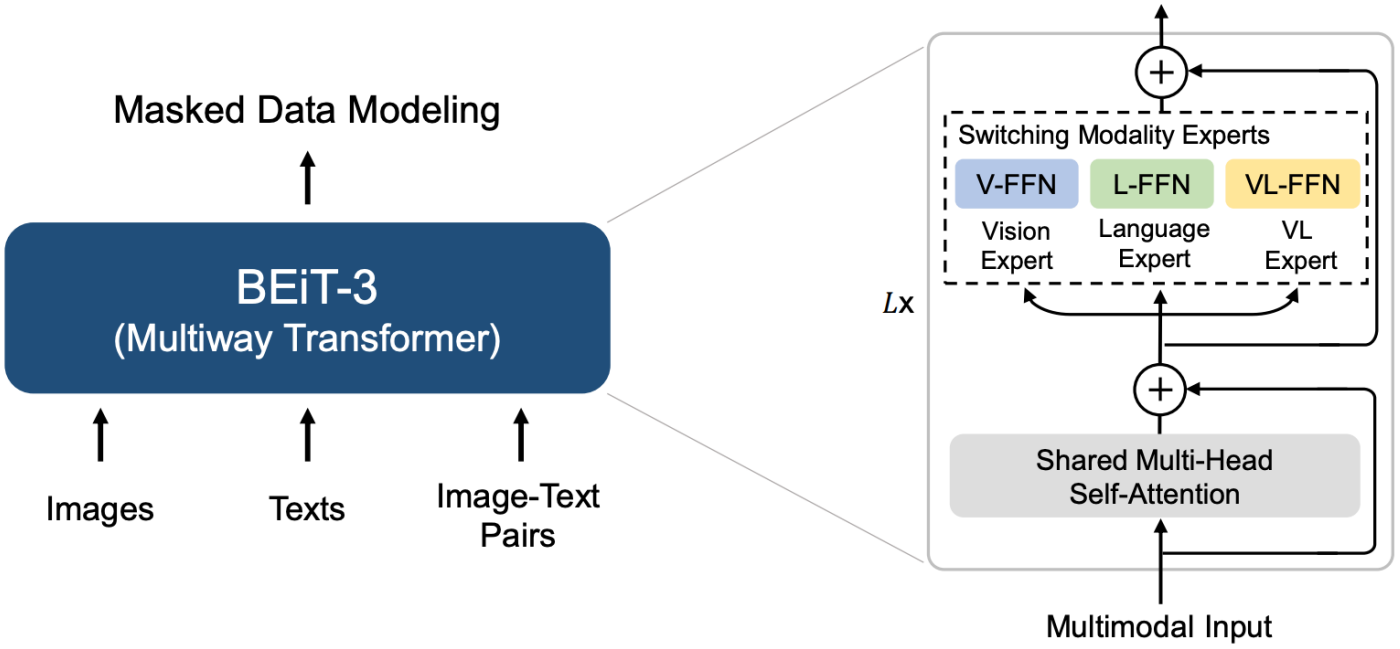
An art painting of a soldier, in the style of cyperpunk.

The golden palace of the land of clouds.

Rustic interior of an alchemy shop.

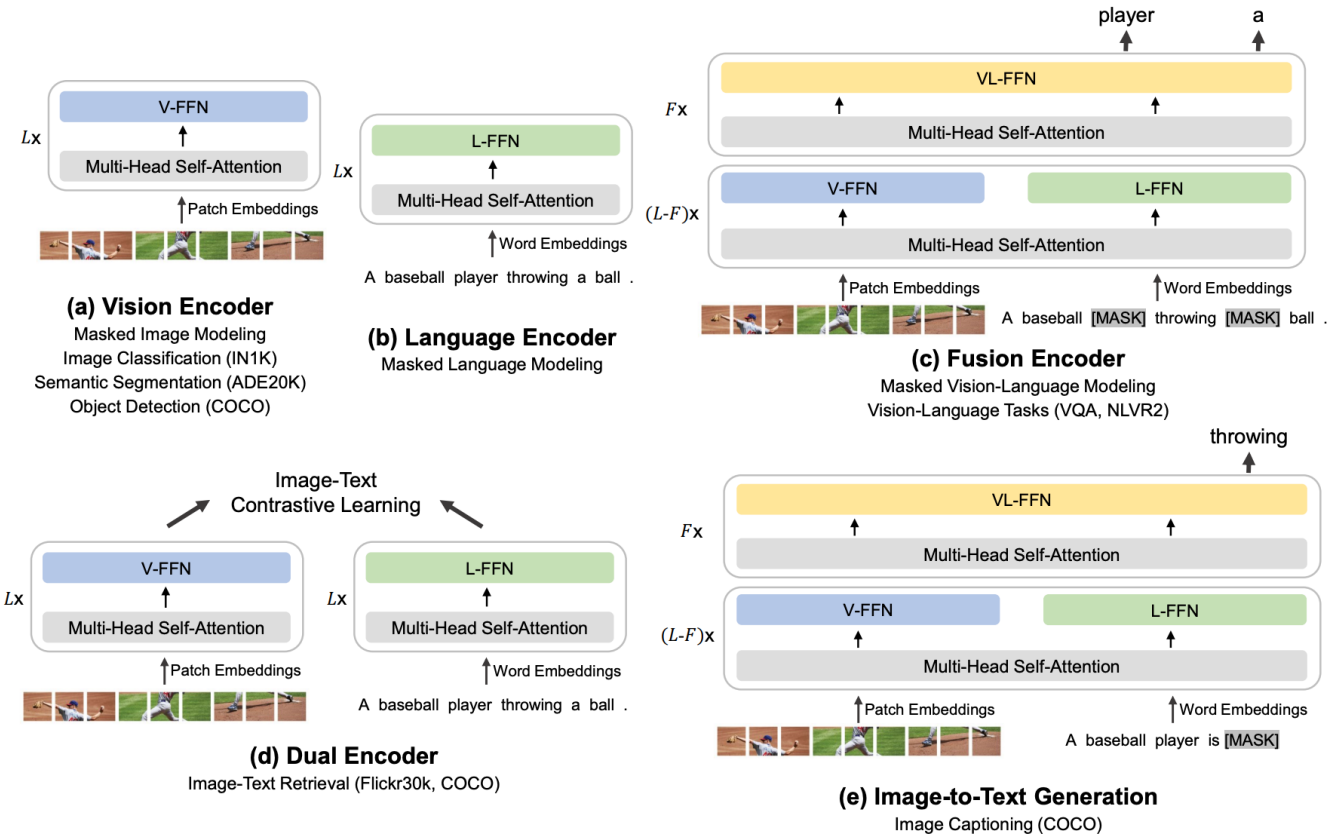
BEiT v3 [Wang et al., 2022]

- The Big Convergence
  - Unification of architecture to Transformer
  - Pretraining task based on masked data modeling
- Scaling up the vision, language, vision-language transformers with masked data modeling pretraining
  - Pretraining with interleaved image and texts, and image-caption pairs



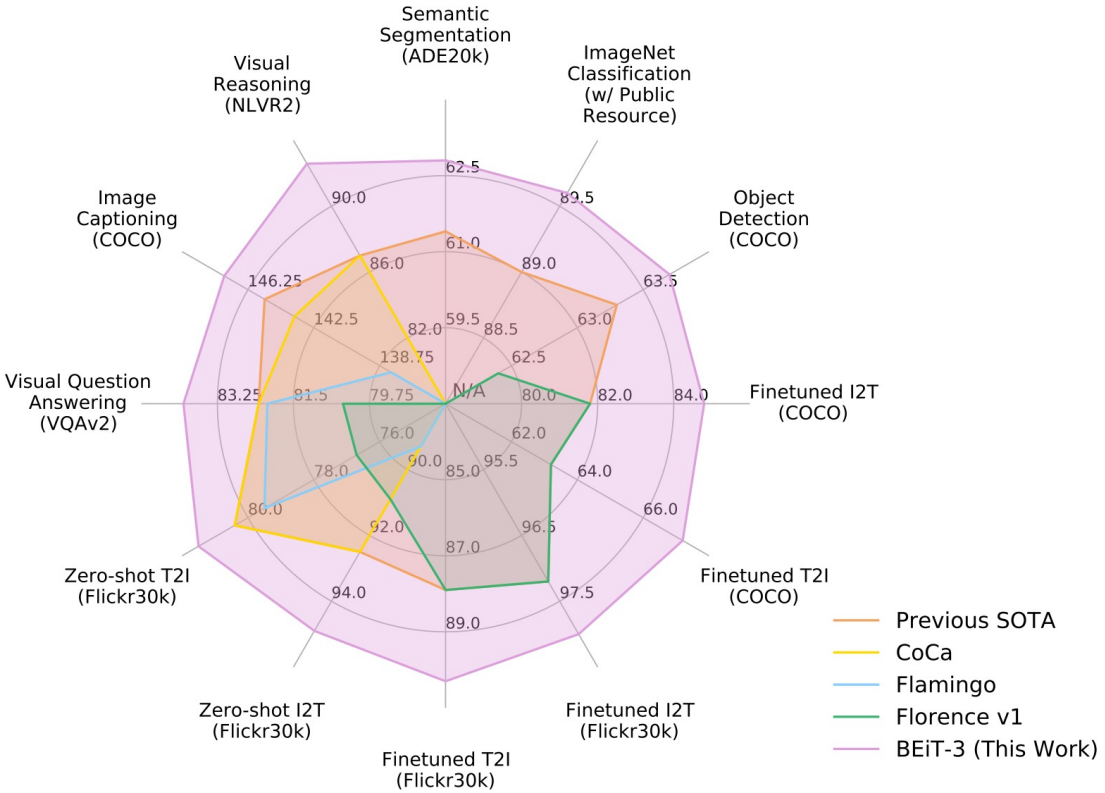
BEiT v3 [Wang et al., 2022]

- Scaling up the vision, language, vision-language transformers with masked data modeling pretraining
- BEiT v3 can be adapted to various vision and vision-language tasks



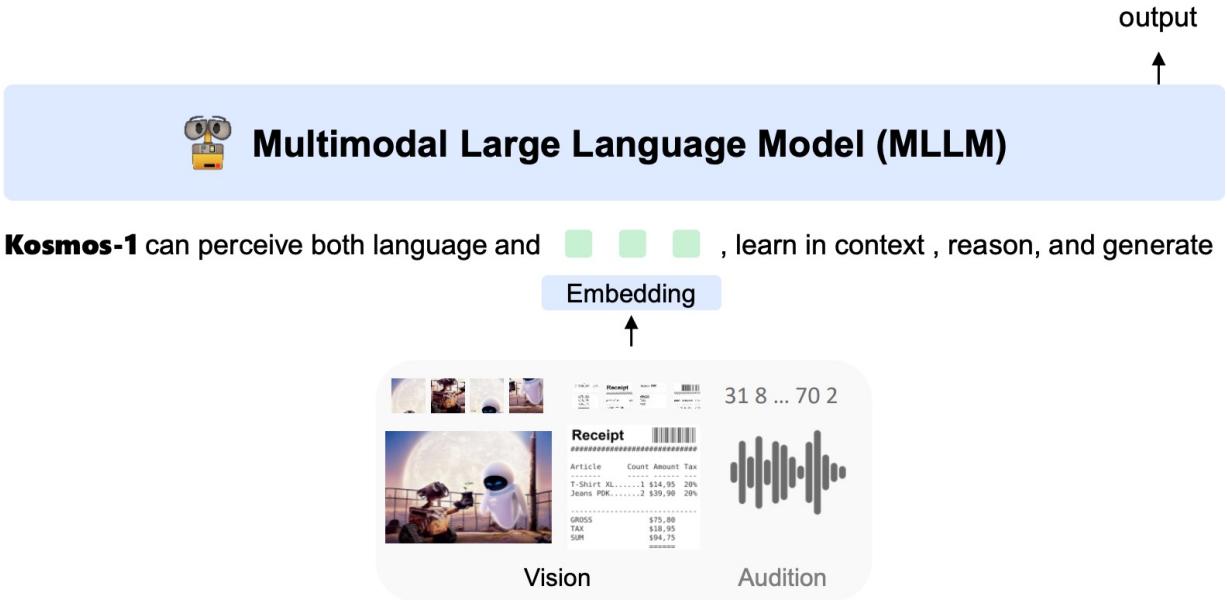
BEiT v3 [Wang et al., 2022]

- As a result, BEiT v3 achieves SOTA performance on various tasks including
  - Vision tasks: Classification, Segmentation, Detection
  - Vision-Language tasks: Retrieval, VQA, Visual Reasoning



KOSMOS-1 [Huang et al., 2023]

- LLMs are great at following instructions and learn in context
- **KOSMOS-1** is **Multimodal Large Language Model (MLLM)** that can perceive general modalities that follows instruction and learn in context
  - Language models as general-purpose interfaces: other modalities are embedded into language models
  - As LLMs, KOSMOS-1 conduct instruction tuning for better human alignment



KOSMOS-1 [Huang et al., 2023]

- Capabilities of KOSMOS-1
  - Zero-shot / Few-shot multimodal learning, outperforming Flamingo

Model	COCO		Flickr30k	
	CIDEr	SPICE	CIDEr	SPICE
ZeroCap	14.6	5.5	-	-
VLKD	58.3	13.4	-	-
FewVLM	-	-	31.0	10.0
METALM	82.2	15.7	43.4	11.7
Flamingo-3B*	73.0	-	60.6	-
Flamingo-9B*	79.4	-	61.5	-
KOSMOS-1 (1.6B)	**84.7**	**16.8**	**67.1**	**14.5**

Zero-shot Image Captioning

Model	VQAv2	VizWiz
Frozen	29.5	-
VLKDViT-B/16	38.6	-
METALM	41.1	-
Flamingo-3B*	49.2	28.9
Flamingo-9B*	**51.8**	28.8
KOSMOS-1 (1.6B)	51.0	**29.2**

Zero-shot Visual Question Answering

Model	COCO			Flickr30k		
	$k=2$	$k=4$	$k=8$	$k=2$	$k=4$	$k=8$
Flamingo-3B	-	85.0	90.6	-	72.0	71.7
Flamingo-9B	-	93.1	**99.0**	-	72.6	**73.4**
KOSMOS-1 (1.6B)	**99.6**	**101.7**	96.7	**70.0**	**75.3**	68.0

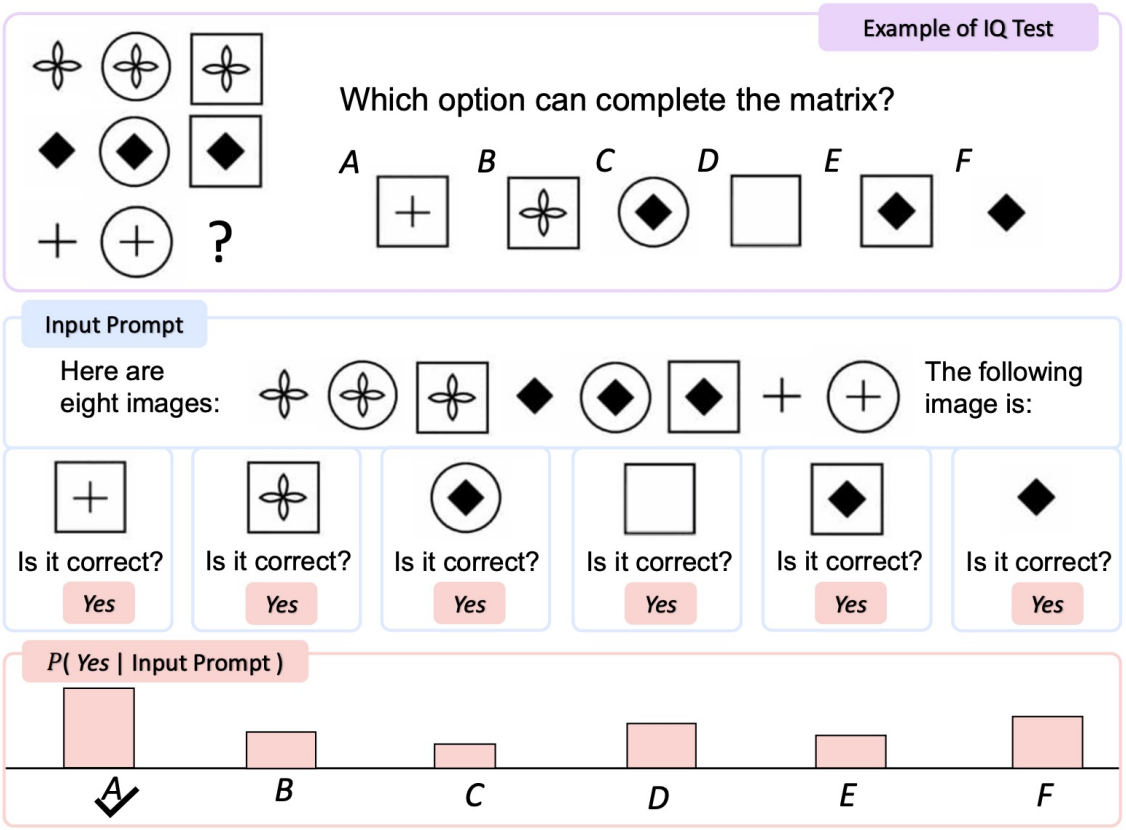
Few-shot Image Captioning

Model	VQAv2			VizWiz		
	$k=2$	$k=4$	$k=8$	$k=2$	$k=4$	$k=8$
Frozen	-	38.2	-	-	-	-
METALM	-	45.3	-	-	-	-
Flamingo-3B	-	53.2	55.4	-	34.4	38.4
Flamingo-9B	-	**56.3**	**58.0**	-	34.9	**39.4**
KOSMOS-1 (1.6B)	**51.4**	51.8	51.4	**31.4**	**35.3**	39.0

Few-shot Visual Question Answering

KOSMOS-1 [Huang et al., 2023]

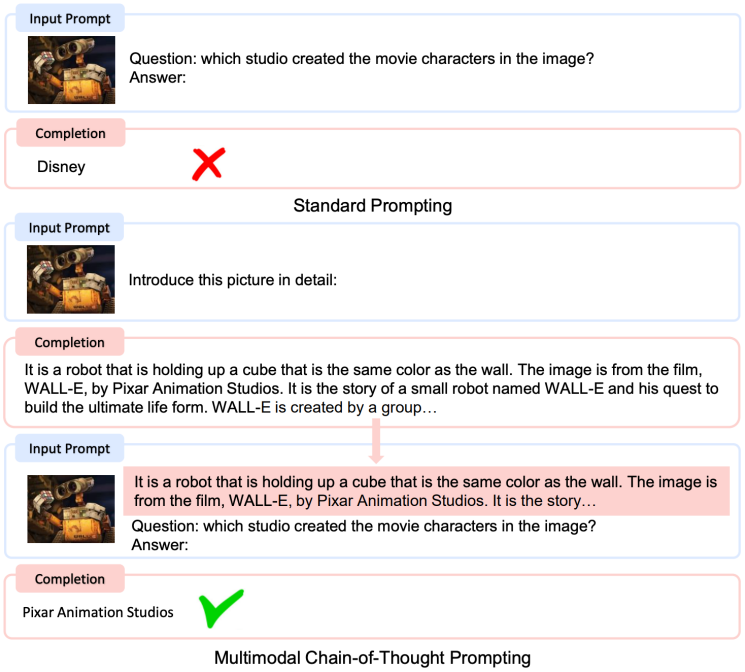
- Capabilities of KOSMOS-1
  - Zero-shot / Few-shot multimodal learning, outperforming Flamingo
  - Nonverbal visual reasoning: Raven IQ-Test
    - First attempt of DNN model for IQ-test, but still large gap between human



Method	Accuracy
Random Choice	17%
KOSMOS-1	22%
w/o language-only instruction tuning	**26%**

KOSMOS-1 [Huang et al., 2023]

- Capabilities of KOSMOS-1
  - Zero-shot / Few-shot multimodal learning, outperforming Flamingo
  - Nonverbal visual reasoning: Raven IQ-Test
  - Prompting for Image classification: KOSMOS-1 benefits from Chain-of-Thought (CoT) prompting for visual recognition task



Models	Accuracy
CLIP ViT-B/32	59.6
CLIP ViT-B/16	59.8
CLIP ViT-L/14	64.0
KOSMOS-1	67.1
w/ multimodal CoT prompting	**72.9**

Effect of prompting on Rendered SST2 dataset

GPT-4 [OpenAI, 2023]

- The new GPT-4 supports visual prompts
  - It supports various input; natural images, documents, OCR, etc.

User What is funny about this image? Describe it panel by panel.

Source: hmmm (Reddit)

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

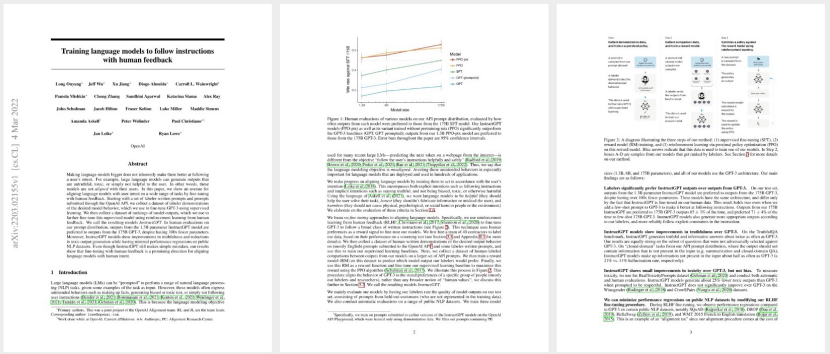
Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

Image captioning / VQA

User Below is part of the InstuctGPT paper. Could you read and summarize it to me?



Training language models to follow instructions with human feedback

GPT-4 The InstructGPT paper focuses on training large language models to follow instructions with human feedback. The authors note that making language models larger doesn't inherently make them better at following a user's intent. Large models can generate outputs that are untruthful, toxic, or simply unhelpful.

To address this issue, the authors fine-tune language models on a wide range of tasks using human feedback. They start with a set of labeler-written prompts and responses, then collect a dataset of labeler demonstrations of the desired model behavior. They fine-tune GPT-3 using supervised learning and then use reinforcement learning from human feedback to further fine-tune the model. The resulting model, called InstructGPT, shows improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets.

Document understanding

GPT-4 [OpenAI, 2023]

- The new GPT-4 supports visual prompts
  - It supports various input; natural images, documents, OCR, etc.
- It shows comparable performance on various vision-language understanding tasks
  - However, the model is unknown (e.g., architecture, training method, data used)

Benchmark	GPT-4 Evaluated few-shot	Few-shot SOTA	SOTA Best external model (includes benchmark-specific training)
VQAv2 VQA score (test-dev)	77.2% 0-shot	67.6% Flamingo 32-shot	84.3% PaLI-17B
TextVQA VQA score (val)	78.0% 0-shot	37.9% Flamingo 32-shot	71.8% PaLI-17B
ChartQA Relaxed accuracy (test)	78.5%[A]	-	58.6% Pix2Struct Large
AI2 Diagram (AI2D) Accuracy (test)	78.2% 0-shot	-	42.1% Pix2Struct Large
DocVQA ANLS score (test)	88.4% 0-shot (pixel-only)	-	88.4% ERNIE-Layout 2.0
Infographic VQA ANLS score (test)	75.1% 0-shot (pixel-only)	-	61.2% Applica.ai TILT
TVQA Accuracy (val)	87.3% 0-shot	-	86.5% MERLOT Reserve Large
LSMDC Fill-in-the-blank accuracy (test)	45.7% 0-shot	31.0% MERLOT Reserve 0-shot	52.9% MERLOT

Summary

- We discussed four approaches in vision-language pretraining
  1. Image-text alignment using CLIP for transferrable visual representation
  2. Fused transformer for vision-language understanding
  3. Learning visual representation from frozen Large Language Models (LLMs)
  4. Unifying Vision-Language pretraining

  - The development of LLMs have affected the visual representation learning
    - Zero-shot / Few-shot classification as instruction / in-context learning
    - Scaling up foundation models showing emergent capabilities
  - But still, understanding and designing better learning method is important problem
    - Scaling is not the only way to improve performance!

References

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

References

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 forLanguage 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

References

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

References

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