

Foundation models for vision

AI602: Recent Advances in Deep Learning

Lecture 5

KAIST AI

1. Introduction

- Foundation models in vision tasks

2. Discriminative Visual Foundation Models

- Self-supervised Learning
- Image-text Contrastive Learning
- Multimodal LLM

3. Generative visual foundation models

- Text-to-Image Diffusion models
- Applications

4. Segment Anything

1. Introduction

- Foundation models in vision tasks

2. Discriminative Visual Foundation Models

- Self-supervised Learning
- Image-text Contrastive Learning
- Multimodal LLM

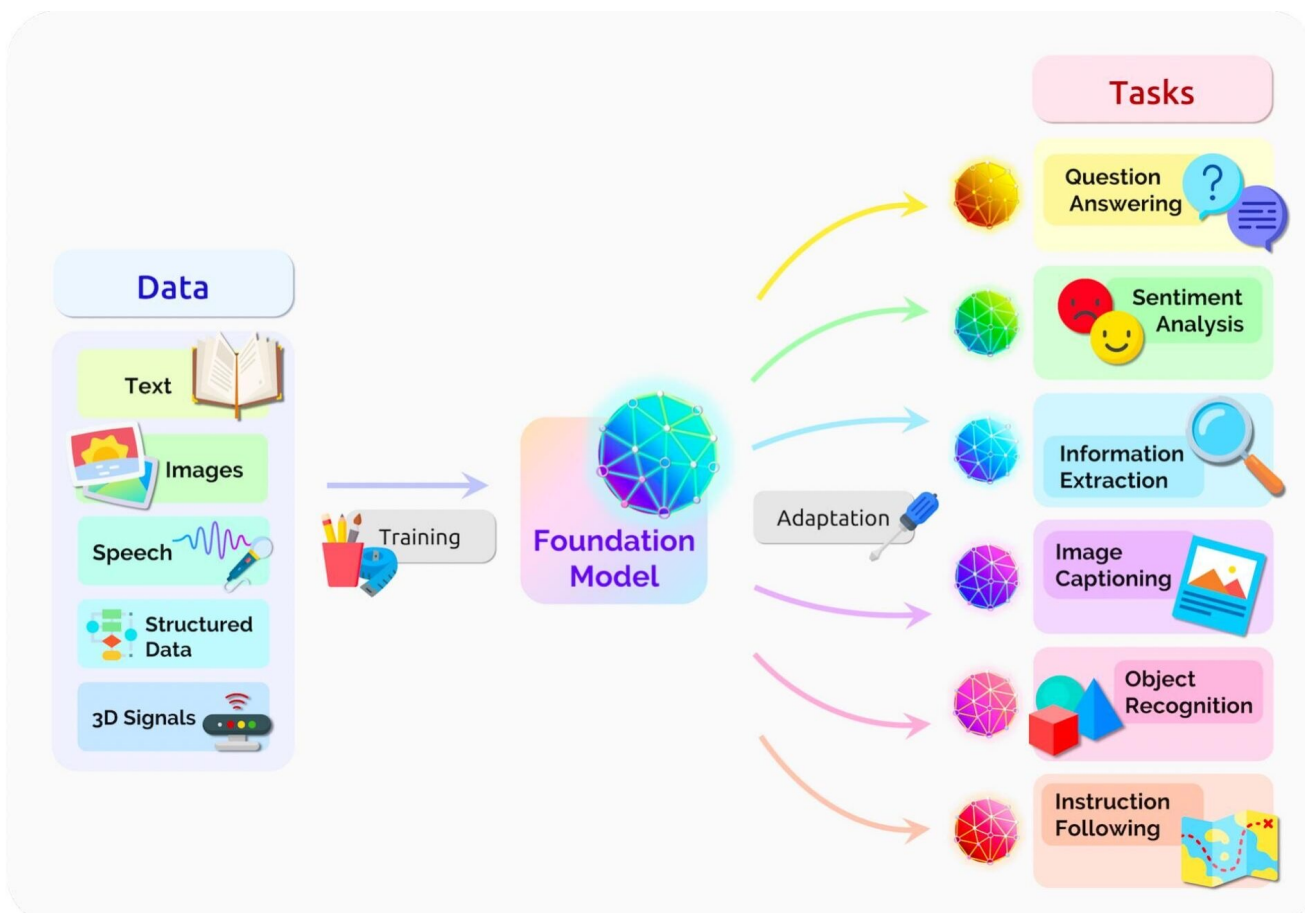
3. Generative visual foundation models

- Text-to-Image Diffusion models
- Applications

4. Segment Anything

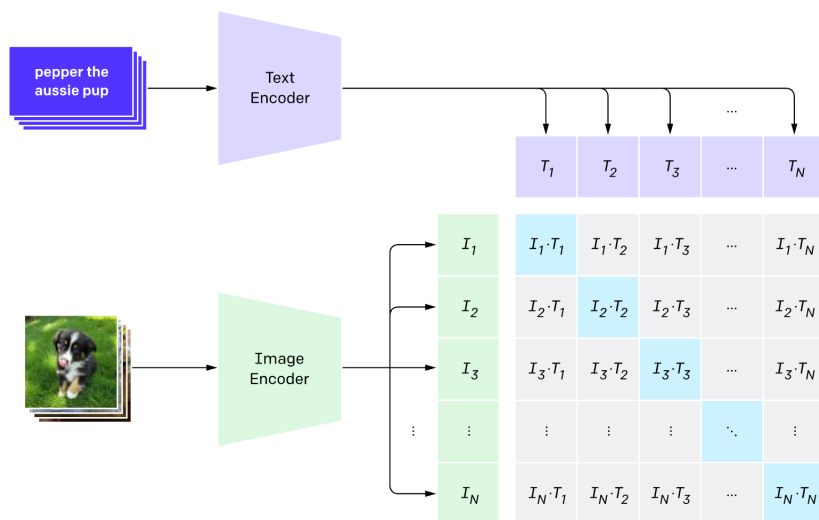
• Foundation Models for Vision

- Fixing a foundation model (e.g., trained via self-supervised learning) and only adapting a **simple task-specific model** is sufficient for many problems

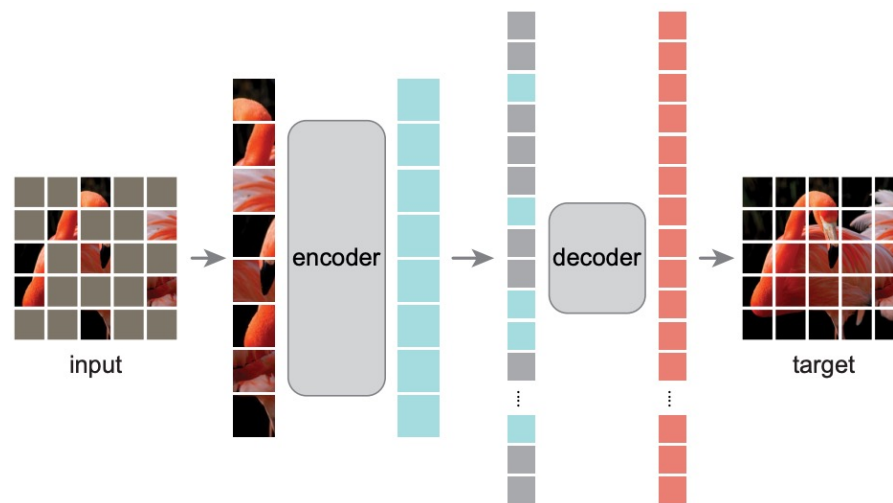


• Foundation Models for Vision

- Fixing a foundation model (e.g., trained via self-supervised learning) and only adapting a **simple task-specific model** is sufficient for many problems
- This lecture will cover following foundation models for vision
 - Discriminative models (e.g., self-supervised models, CLIP)



CLIP [Radford et al., '21]



MAE [He et al., '21]

• Foundation Models for Vision

- Fixing a foundation model (e.g., trained via self-supervised learning) and only adapting a **simple task-specific model** is sufficient for many problems
- This lecture will cover following foundation models for vision
 - Discriminative models (e.g., self-supervised models, CLIP)
 - Generative models (e.g., text-to-image diffusion models)

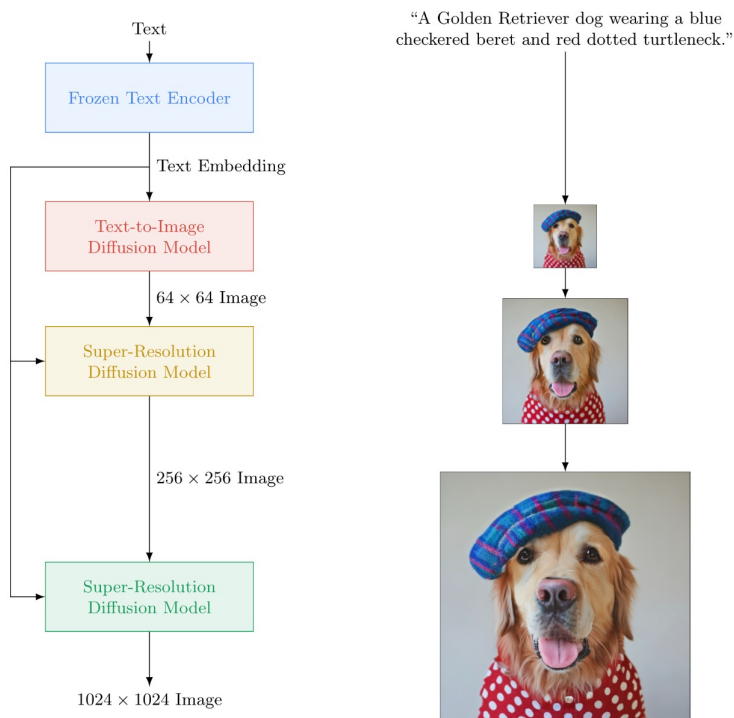
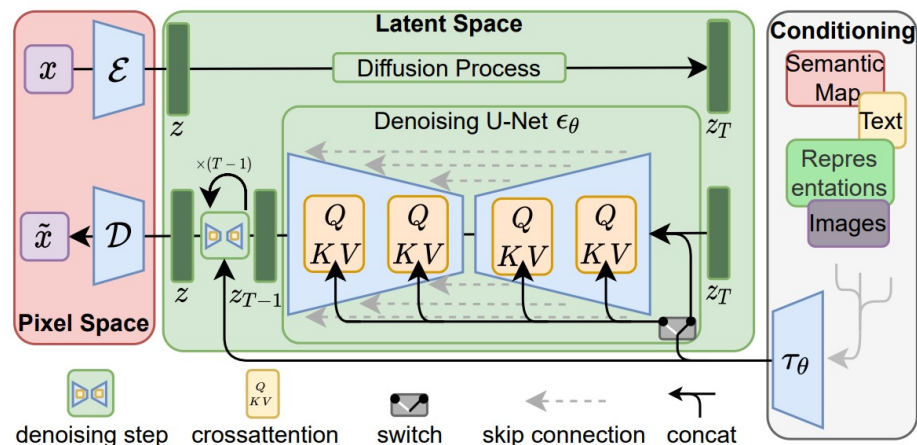


Imagen [Sahria et al., '22]



Latent Diffusion [Rombach et al., '21]

- **Foundation Models for Vision**

- Fixing a foundation model (e.g., trained via self-supervised learning) and only adapting a **simple task-specific model** is sufficient for many problems
- This lecture will cover following foundation models for vision
 - Discriminative models (e.g., self-supervised models, CLIP)
 - Generative models (e.g., text-to-image diffusion models)
 - Vision-specific models (e.g., Segment Anything (SAM),



Segment Anything [Meta AI, '22]

- **Foundation Models for Vision**

- Fixing a foundation model (e.g., trained via self-supervised learning) and only adapting a **simple task-specific model** is sufficient for many problems
- This lecture will cover following foundation models for vision
 - Discriminative models (e.g., self-supervised models, CLIP)
 - Generative models (e.g., text-to-image diffusion models)
 - Vision-specific models (e.g., Segment Anything (SAM))
- In specific, this lecture will answer (or at least hint) to the following questions:
 - How to train foundation models?
 - What are the zero-shot capabilities of foundation models?
 - How to exploit foundation models on specific tasks?

We are interested in visual representations that extract high-level semantics which can be applied to various **downstream tasks** such as

- Supervised learning (e.g., classification, detection)
- Unsupervised learning (e.g., clustering, metric learning)
- Modular component for multimodal understanding (e.g., image-text retrieval, visual question answering)

Scaling model and data size is key recipe in training foundation models:

- The loss function must be designed to be scalable and stable
- The data should be curated to remove bias or noisy label
- Computation efficiency to lower the training cost

First, we introduce self-supervised learning (SSL) methods:

- Invariance based methods such as contrastive learning
- Masked image modeling (MIM)

Second, we will cover image-text contrastive methods (i.e., CLIP):

- Training data perspective of CLIP
- Training objective perspective of CLIP

Lastly, we will cover combination of visual foundation models with language models for vision-language multimodal understanding (i.e, multimodal LLM)

1. Introduction

- Foundation models in vision tasks

2. Discriminative Visual Foundation Models

- Self-supervised Learning
- Image-text Contrastive Learning
- Multimodal LLM

3. Generative visual foundation models



- Text-to-Image Diffusion models
- Applications

4. Segment Anything



Core idea of invariance-based learning:

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

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



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



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

Core idea of invariance-based learning:

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

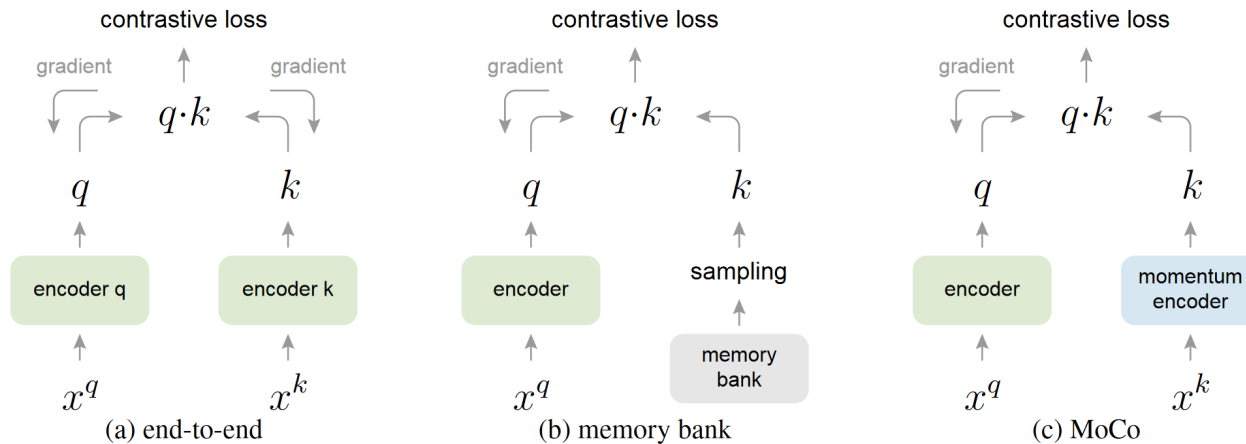
Positive pair $f\left(\text{img}_1\right) \approx f\left(\text{img}_2\right)$

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

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

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

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

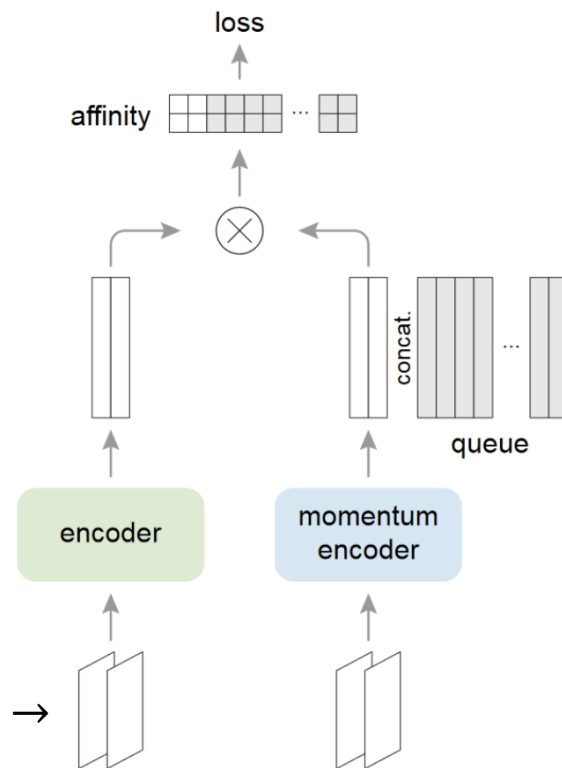


- **Momentum encoder** increases the **key representations' consistency**
- **Queue** allows us to use **recent and many negative** samples

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

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

Randomly augmented samples →



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

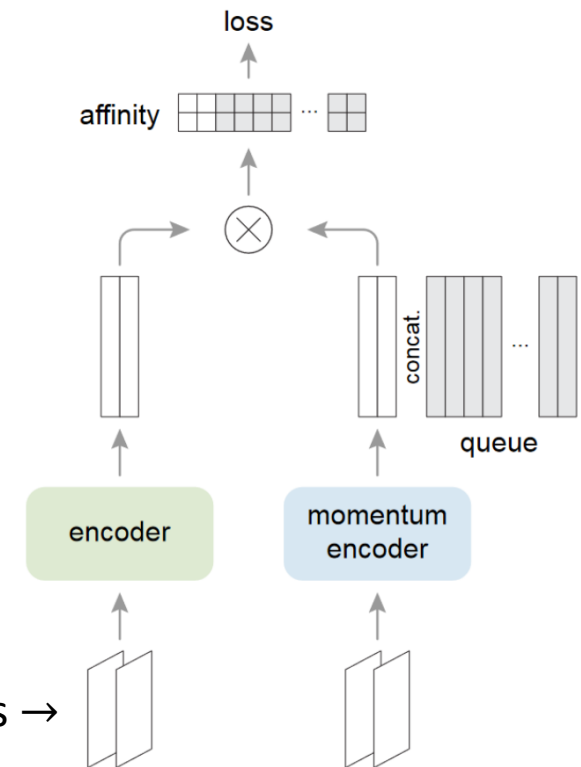
- **MoCo's idea:** use a **momentum-updated encoder** and maintain a **queue**

- MoCo also optimizes contrastive learning objective

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

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

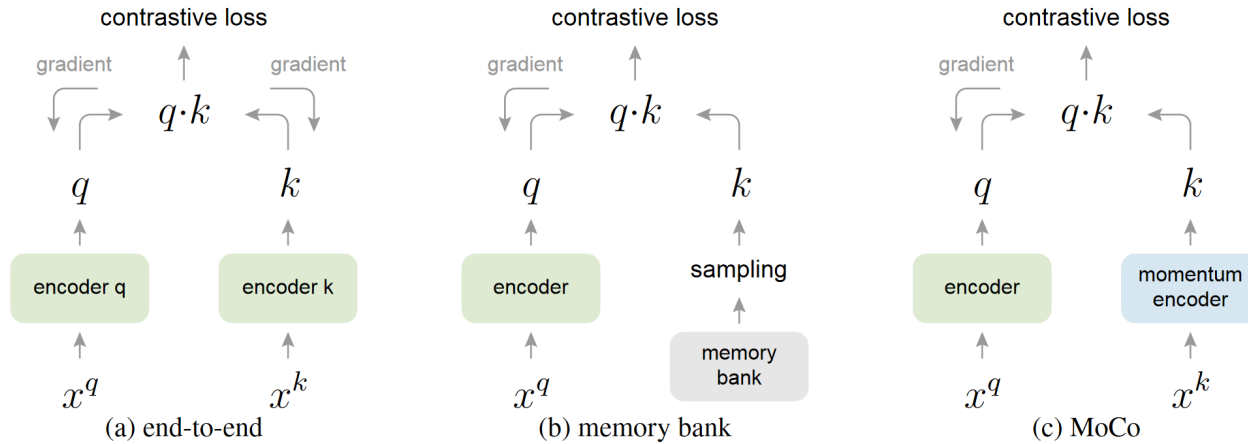
$$\theta_{\text{momentum}} \leftarrow m\theta_{\text{momentum}} + (1 - m)\theta$$
 - Add the current positive keys k^+ into the queue



Randomly augmented samples →

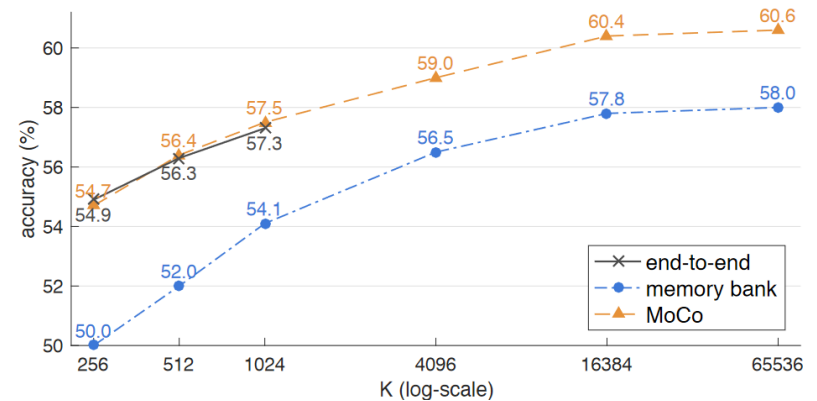
- **Momentum Contrast (MoCo)** [He et al., 2019]

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



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

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



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

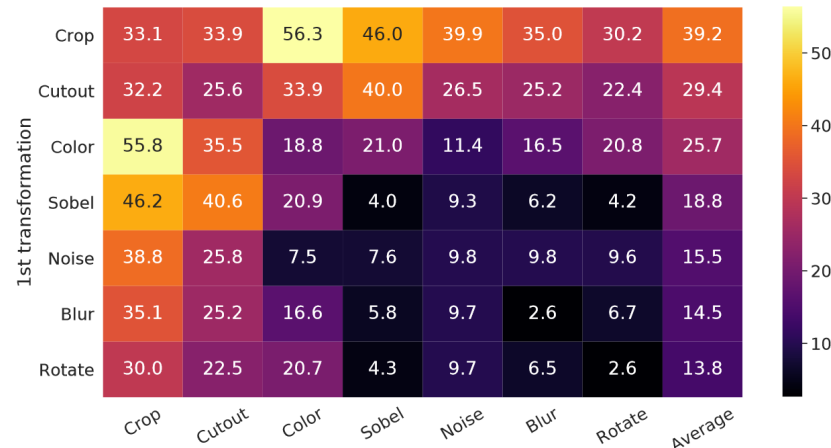
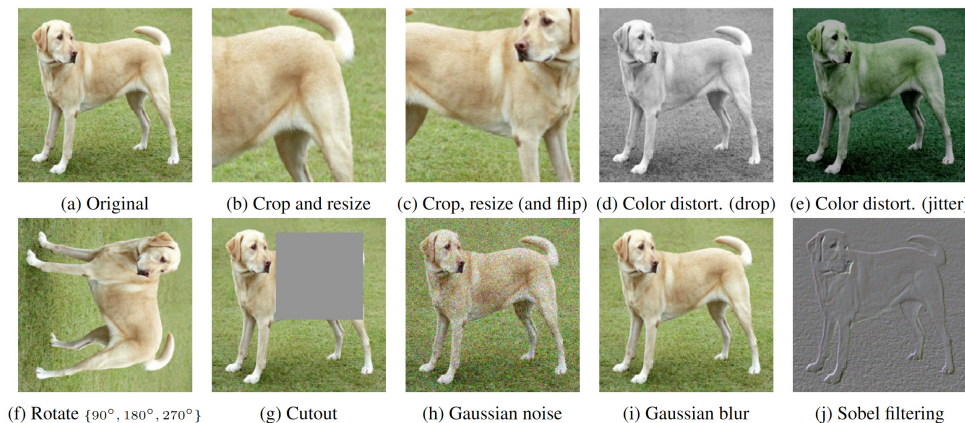
- **SimCLR** [Chen et al., 2020]

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

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

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

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



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

2nd transformation

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

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

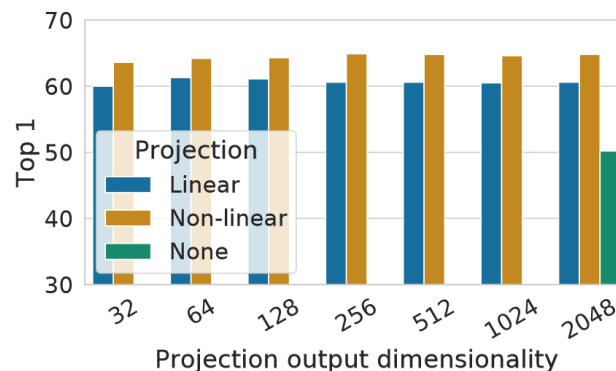
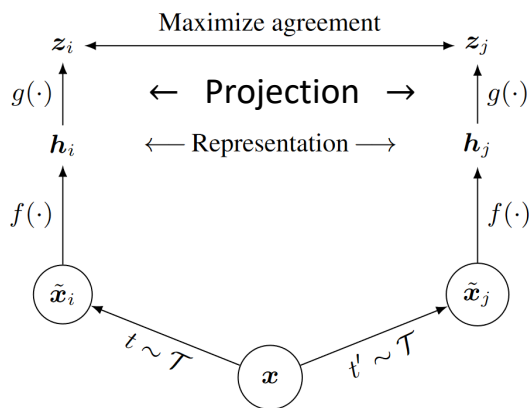
- This paper finds that contrastive learning benefits from ...

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

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

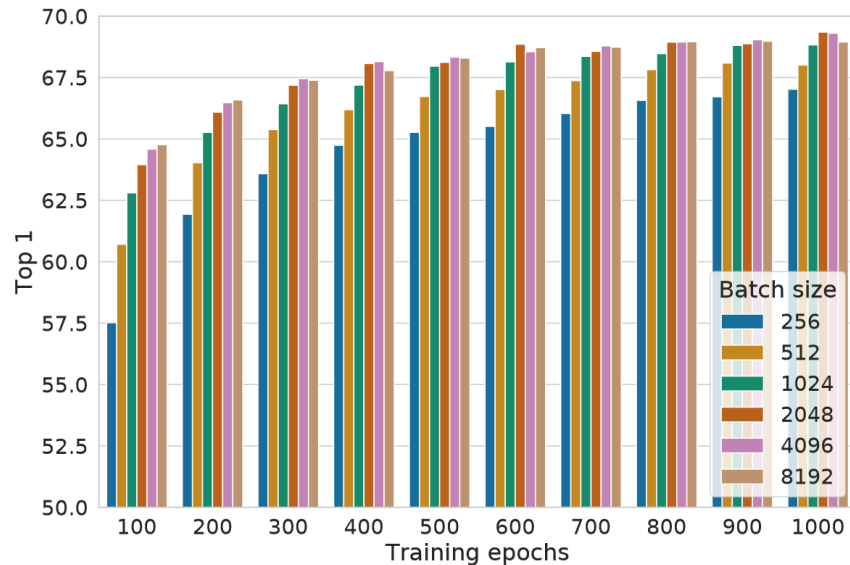
$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$

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



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

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



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

Fine-grained image classification tasks

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

Semi-supervised learning in ImageNet

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

Linear evaluation in ImageNet

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

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

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

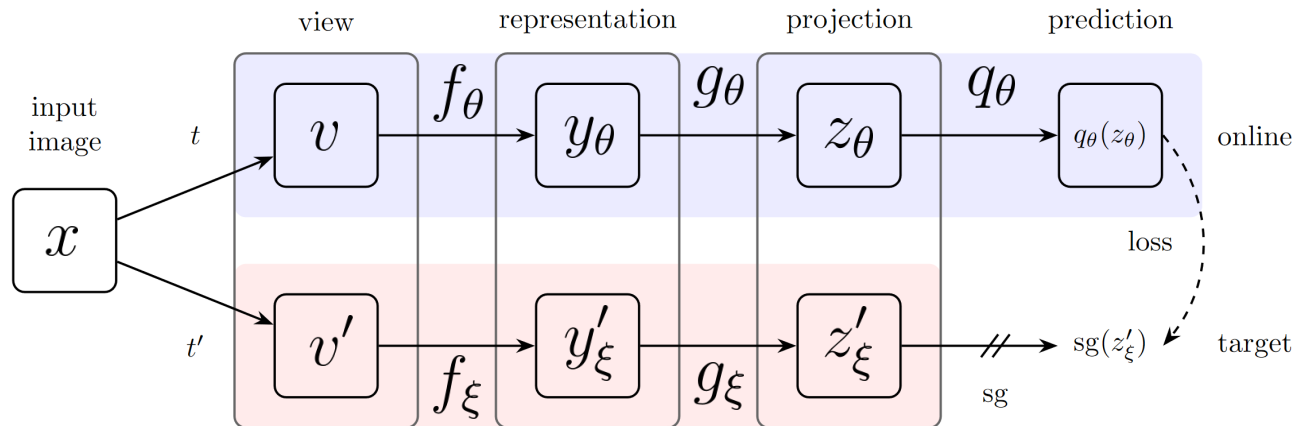
Negative pair $f\left(\text{img}_3\right) \neq f\left(\text{img}_4\right)$

This relation might be not true

- **Q)** can we learn representations without negative samples?
- Simply minimizing $\|f(\text{img}_1) - f(\text{img}_2)\|$ leads to mode collapse, i.e., $\forall x, f(x) = c$
- **Next:** Positive-only approaches

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

- **Idea:** directly bootstrap the representations



Objective

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

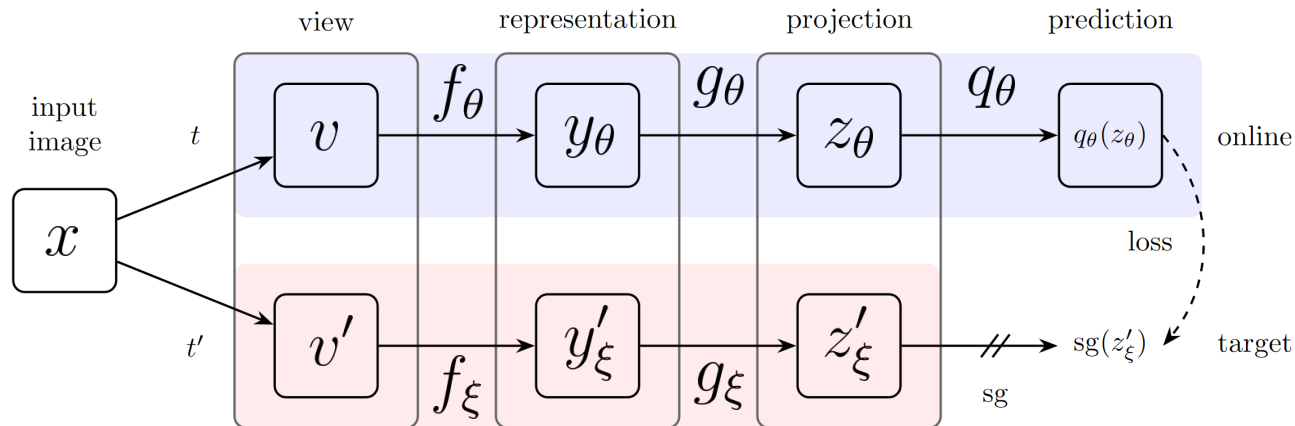
Update

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

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

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

- **Idea:** directly bootstrap the representations



Objective

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

Update

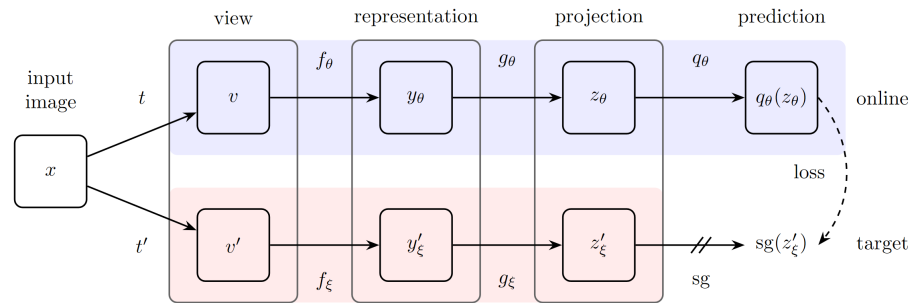
$$\begin{aligned} \theta &\leftarrow \text{optimizer}(\theta, \nabla_\theta \mathcal{L}_{\text{BYOL}}) \\ \xi &\leftarrow \tau \xi + (1 - \tau) \theta \end{aligned}$$

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

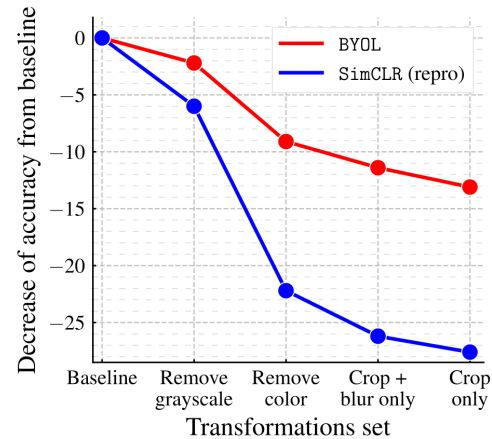
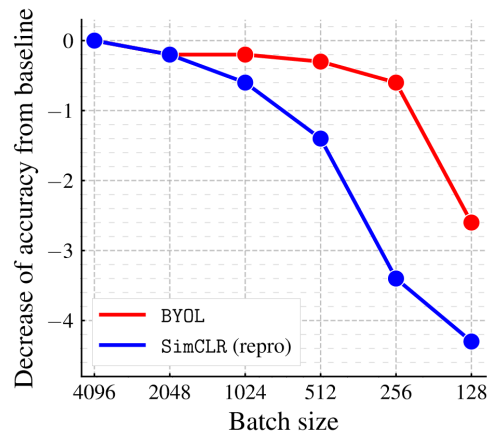
- ξ is not updated in the direction of $\nabla_\xi \mathcal{L}_{\text{BYOL}}$
- When the predictor is optimal, i.e., $q^*(z_\theta) = \mathbb{E}[z'_\xi | z_\theta]$, $\mathcal{L}_{\text{BYOL}} = \mathbb{E}[\sum_i \text{Var}(z'_{\xi,i} | z_\theta)]$ z'_ξ 's i -th feature \searrow
- For any constant c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

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

- **Idea:** directly bootstrap the representations

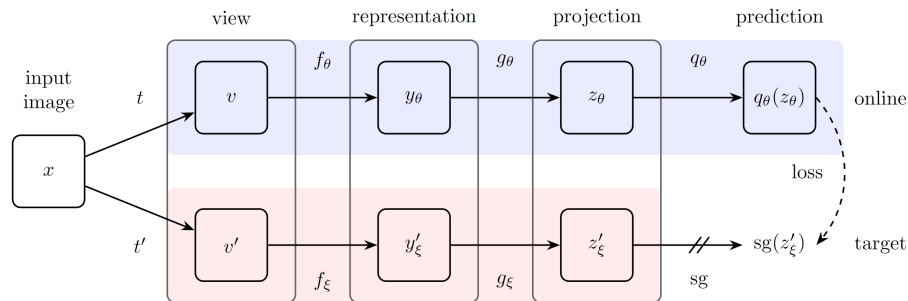


- BYOL is **more robust** to the choice of **batch sizes** and **augmentations**

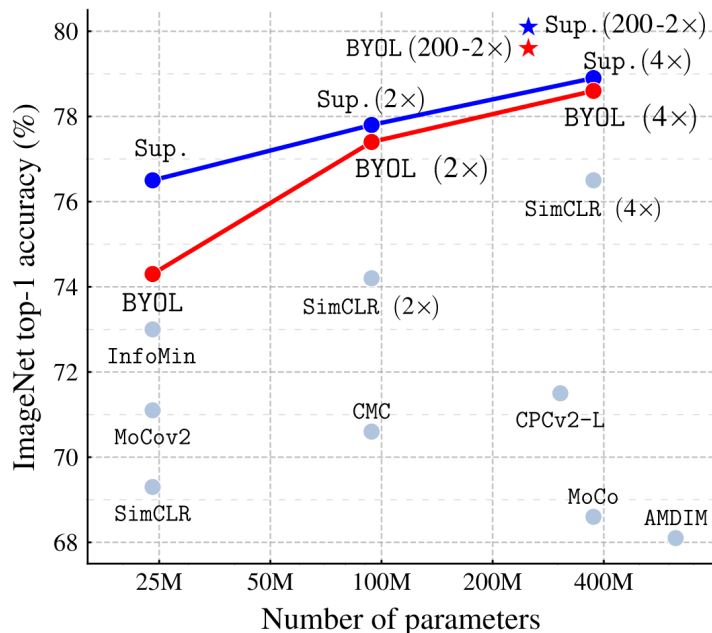


- **Bootstrap You Own Latent (BYOL)** [Grill et al., 2020]

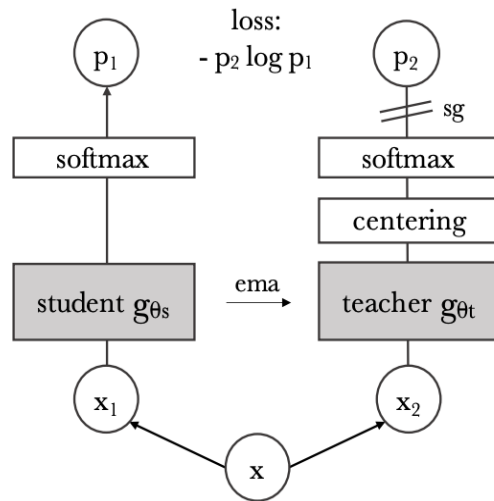
- **Idea:** directly bootstrap the representations



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



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



Objective

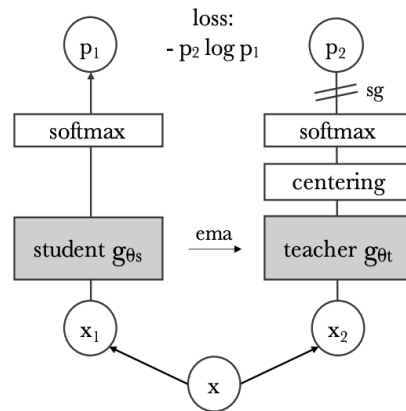
$$\mathcal{L}_{DINO} = H(P_t(x), P_s(x))$$

Update

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

- **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

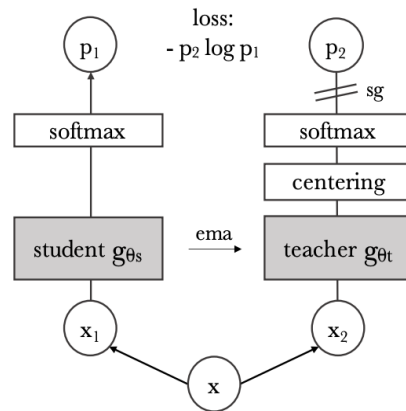
- **DINO** [Caron et al., 2021]
 - **Idea**: representation learning via self knowledge-distillation



- 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: adding a bias term c to the teacher

$$g_t(x) \leftarrow g_t(x) + c$$

- The center c is updated with an exponential moving average

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

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

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

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

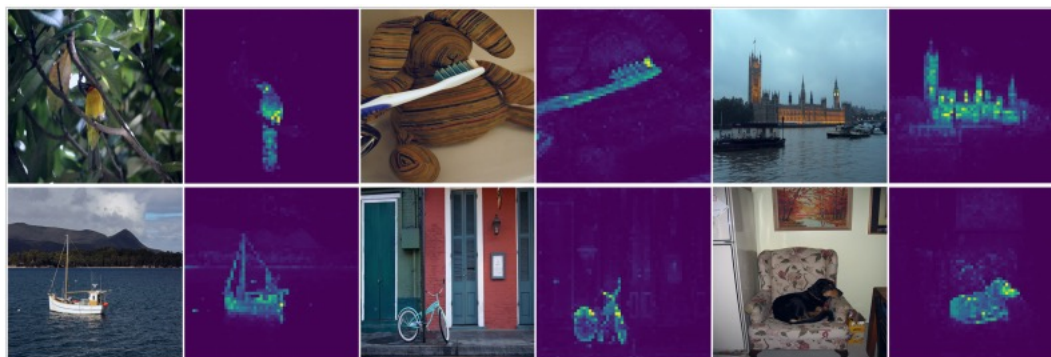
Method	Arch.	Param.	im/s	Linear	k-NN
Supervised	RN50	23	1237	79.3	79.3
SCLR [12]	RN50	23	1237	69.1	60.7
MoCov2 [15]	RN50	23	1237	71.1	61.9
InfoMin [67]	RN50	23	1237	73.0	65.3
BarlowT [81]	RN50	23	1237	73.2	66.0
OBoW [27]	RN50	23	1237	73.8	61.9
BYOL [30]	RN50	23	1237	74.4	64.8
DCv2 [10]	RN50	23	1237	75.2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

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

Comparison across architectures

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

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



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

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

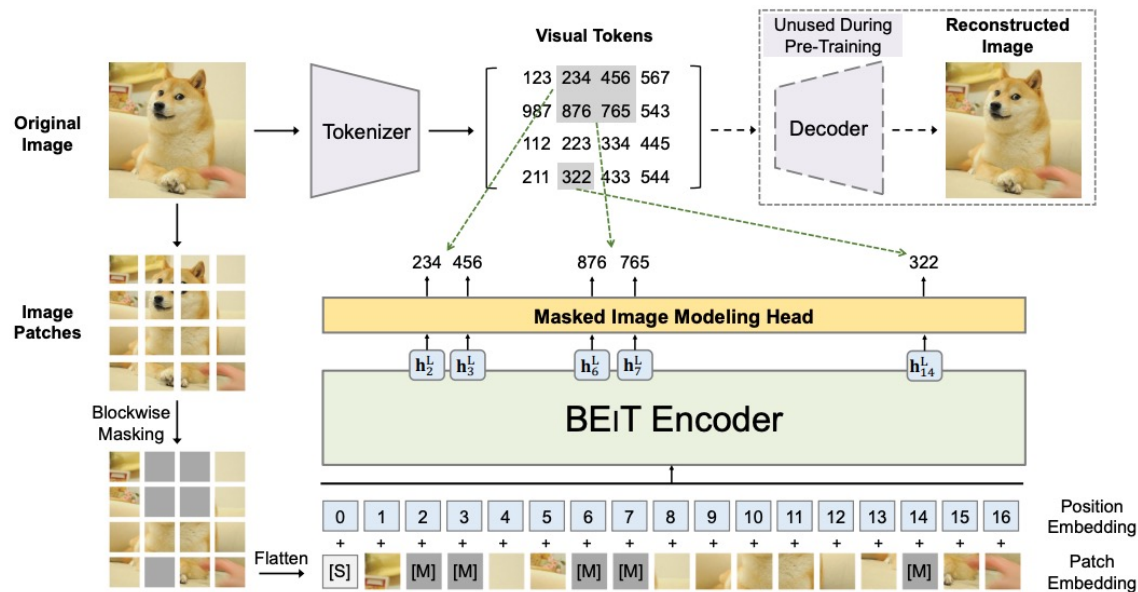
Video instance segmentation on top of self-supervised feature

Masked Image Modeling

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

- **Task:** Masked visual tokens prediction

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



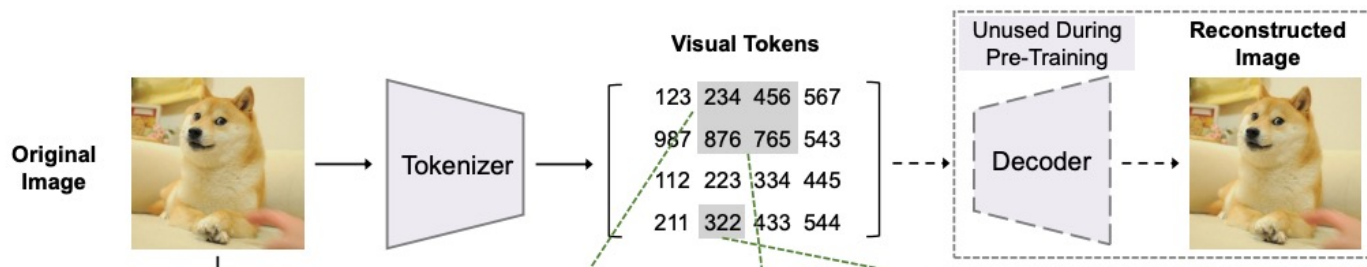
- BEiT training procedure is consist of two stages:

1. Learning visual tokens
2. Masked image modeling

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

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

1. Learning visual tokens



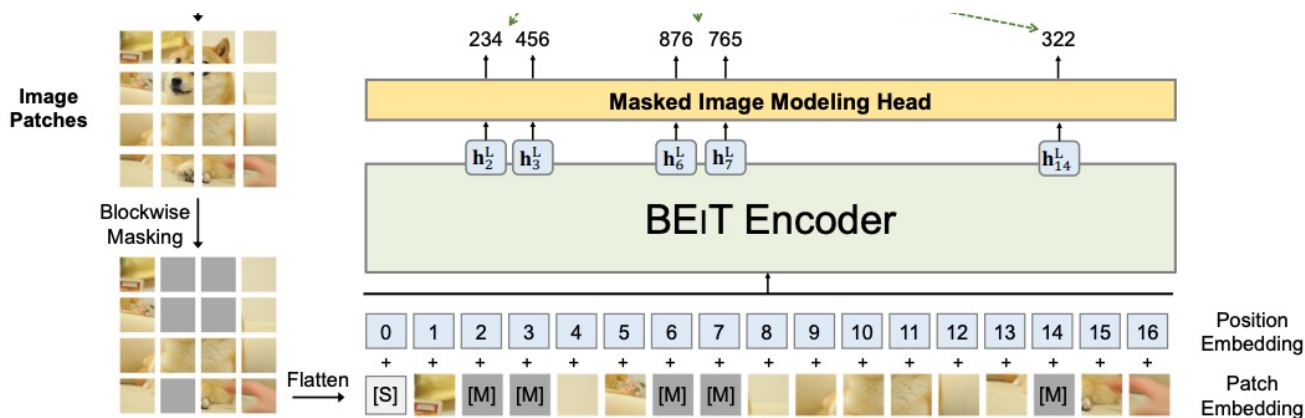
- In this stage, a **discrete variational autoencoder (dVAE)** is trained to represent each 224×224 image into a 14×14 grid of **discrete image tokens**, each element of which can assume 8192 possible values
 - The tokenizer $q_{\phi}(\mathbf{z}|\mathbf{x})$ maps image image pixels into a visual codebook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

Masked Image Modeling

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

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

2. Masked Image Modeling



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

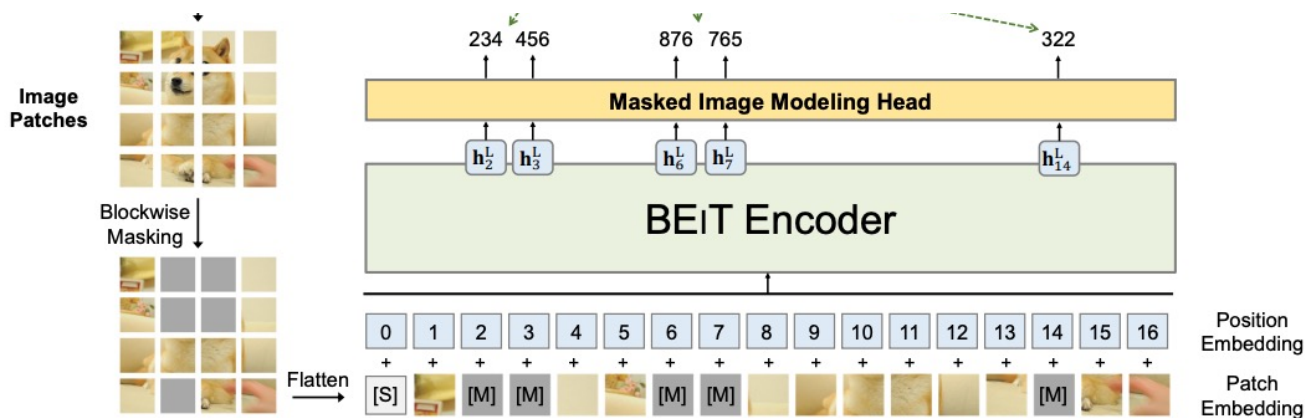
$$\max_{x \in \mathcal{D}} \sum \mathbb{E}_{\mathcal{M}} \left[\sum_{i \in \mathcal{M}} \log p_{\text{MIM}}(z_i | x^{\mathcal{M}}) \right]$$

Masked Image Modeling

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

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

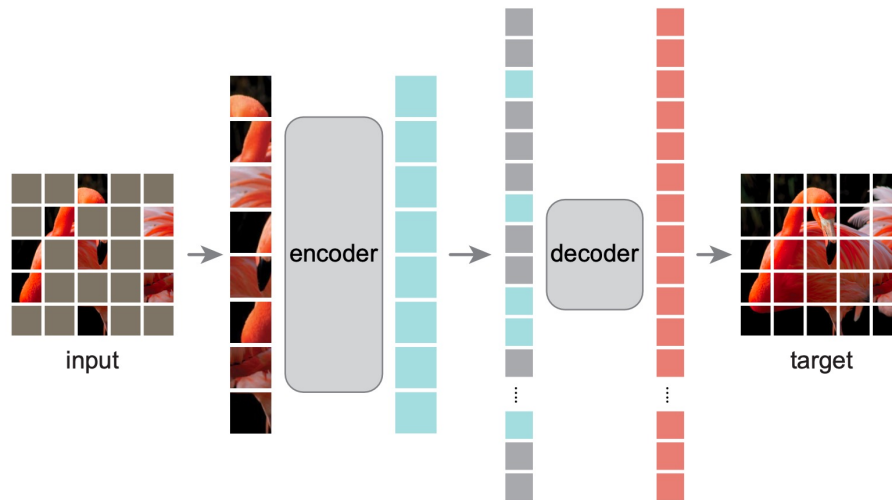
2. Masked Image Modeling



- 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)

Masked Image Modeling

- **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

Masked Image Modeling

- **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

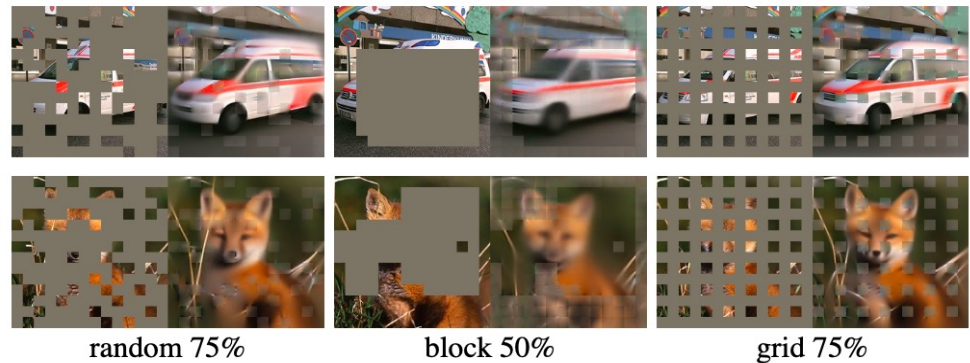
- MAE behaves decently even if using no data augmentation

Masked Image Modeling

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

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

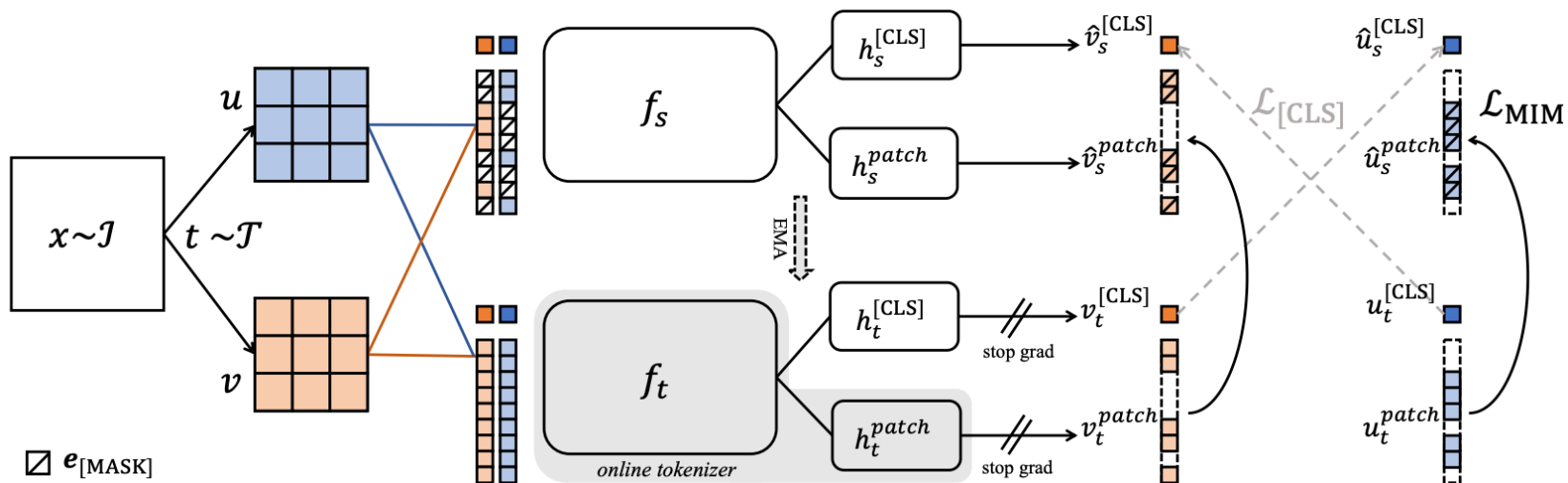
(f) **Mask sampling.** Random sampling works the best. See Figure 6 for visualizations.



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

- **Image-BERT Pretraining with online tokenizer (IBOT)** [Zhou et al., 2022]

- Perform patch-level self-distillation on masked patch tokens (while DINO is done with image-level objective)
- Use data augmentation for invariance learning
- Unlike BEiT, image tokenizer is jointly learned (i.e., online tokenizer)



- **Image-BERT Pretraining with online tokenizer (IBOT)** [Zhou et al., 2022]
 - IBOT shows strong performance on linear probing as well as fine-tuning

Table 1: k -NN and linear probing on ImageNet-1K. [†] denotes using selective kernel. [‡] denotes pre-training on ImageNet-22K.

Method	Arch.	Par.	im/s	Epo. ¹	k -NN	Lin.
<i>SSL big ResNets</i>						
MoCov3	RN50	23	1237	1600	-	74.6
SwAV	RN50	23	1237	2400	65.7	75.3
DINO	RN50	23	1237	3200	67.5	75.3
BYOL	RN200w2	250	123	2000	73.9	79.6
SCLRv2	RN152w3 [†]	794	46	2000	73.1	79.8
<i>SSL Transformers</i>						
MoCov3	ViT-S/16	21	1007	1200	-	73.4
MoCov3	ViT-B/16	85	312	1200	-	76.7
SwAV	ViT-S/16	21	1007	2400	66.3	73.5
DINO	ViT-S/16	21	1007	3200	74.5	77.0
DINO	ViT-B/16	85	312	1600	76.1	78.2
EsViT	Swin-T/7	28	726	1200	75.7	78.1
EsViT	Swin-T/14	28	593	1200	77.0	78.7
iBOT	ViT-S/16	21	1007	3200	75.2	77.9
iBOT	Swin-T/7	28	726	1200	75.3	78.6
iBOT	Swin-T/14	28	593	1200	76.2	79.3
iBOT	ViT-B/16	85	312	1600	77.1	79.5
iBOT	ViT-L/16	307	102	1200	78.0	81.0
iBOT [‡]	ViT-L/16	307	102	200	72.9	82.3

Table 2: Fine-tuning on ImageNet-1K.

Method	Arch.	Epo. ¹	Acc.
Rand.	ViT-S/16	-	79.9
MoCov3	ViT-S/16	600	81.4
DINO	ViT-S/16	3200	82.0
iBOT	ViT-S/16	3200	82.3
<hr/>			
Rand.	ViT-B/16	-	81.8
MoCov3	ViT-B/16	600	83.2
BEiT	ViT-B/16	800	83.4
DINO	ViT-B/16	1600	83.6
iBOT	ViT-B/16	1600	84.0
<hr/>			
MoCov3	ViT-L/16	600	84.1
iBOT	ViT-L/16	1000	84.8
BEiT	ViT-L/16	800	85.2

Table 3: Fine-tuning on ImageNet-1K. Pre-training on ImageNet-22K.

Method	Arch.	Epo. ¹	Acc.
BEiT	ViT-B/16	150	83.7
iBOT	ViT-B/16	320	84.4
<hr/>			
BEiT	ViT-L/16	150	86.0
iBOT	ViT-L/16	200	86.6
iBOT	ViT ₅₁₂ -L/16	200	87.8

- **Image-BERT Pretraining with online tokenizer (IBOT)** [Zhou et al., 2022]

- IBOT shows strong performance on linear probing as well as fine-tuning
- IBOT demonstrates high transferability on various downstream tasks such as semi-supervised learning, unsupervised learning, object detection, and segmentation

Table 4: **Semi-supervised learning on ImageNet-1K.** 1% and 10% denotes label fraction. SD denotes self-distillation.

Method	Arch.	1%	10%
SimCLRv2	RN50	57.9	68.1
BYOL	RN50	53.2	68.8
SwAV	RN50	53.9	70.2
SimCLRv2+SD	RN50	60.0	70.5
DINO	ViT-S/16	60.3	74.3
iBOT	ViT-S/16	61.9	75.1

Table 5: **Unsupervised learning on ImageNet-1K.** [†] denotes k -means clustering on frozen features.

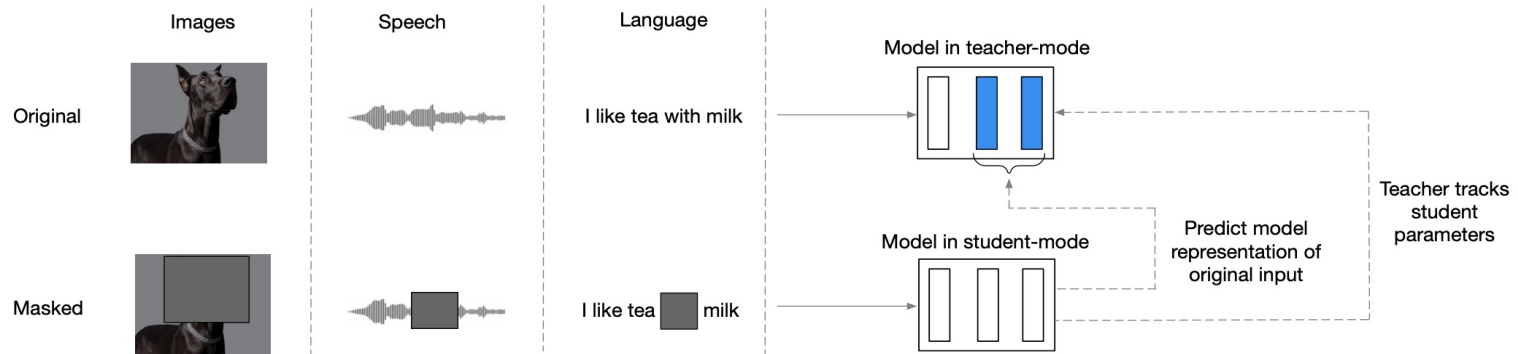
Method	Arch.	ACC	ARI	NMI	FMI
Self-label [†]	RN50	30.5	16.2	75.4	-
InfoMin [†]	RN50	33.2	14.7	68.8	-
SCAN	RN50	39.9	27.5	72.0	-
DINO	ViT-S/16	41.4	29.8	76.8	32.8
iBOT	ViT-S/16	43.4	32.8	78.6	35.6

Table 6: **Object detection (Det.) & instance segmentation (ISeg.) on COCO and Semantic segmentation (Seg.) on ADE20K.** We report the results of ViT-S/16 (left) and ViT-B/16 (right). Seg.[†] denotes using a linear head for semantic segmentation.

Method	Arch.	Param.	Det.	ISeg.	Seg.	Method	Det.	ISeg.	Seg. [†]	Seg.
			AP ^b	AP ^m	mIoU		AP ^b	AP ^m	mIoU	mIoU
Sup.	Swin-T	29	48.1	41.7	44.5	Sup.	49.8	43.2	35.4	46.6
MoBY	Swin-T	29	48.1	41.5	44.1	BEiT	50.1	43.5	27.4	45.8
Sup.	ViT-S/16	21	46.2	40.1	44.5	DINO	50.1	43.4	34.5	46.8
iBOT	ViT-S/16	21	49.4	42.6	45.4	iBOT	51.2	44.2	38.3	50.0

Masked Image Modeling

- **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**

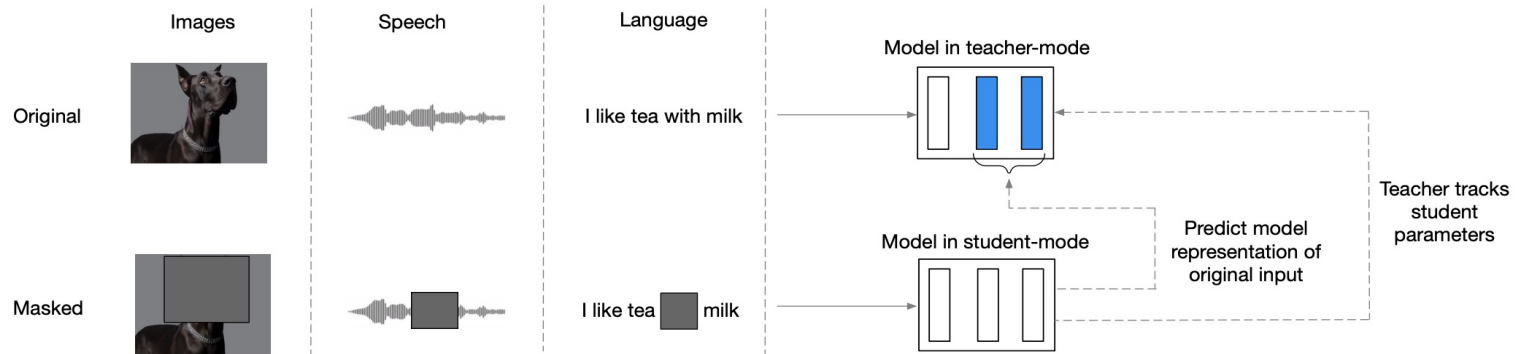


- **Modality-unified algorithm:**
 - 1) Build representations of the full input data with the teacher model
 - The teacher is an exponentially decaying average of the student
 - 2) Encode the masked version of the input sample with the student model and predict the representations of original input
-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

Masked Image Modeling

- **data2vec** [Baevski et al., 2022]

- data2vec is a framework for **general self-supervised learning** for images, speech, and text where the **learning objective is identical in each modality**



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

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

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

- **data2vec** [Baevski et al., 2022]

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

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

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

Vision

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

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

Speech

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

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

NLP

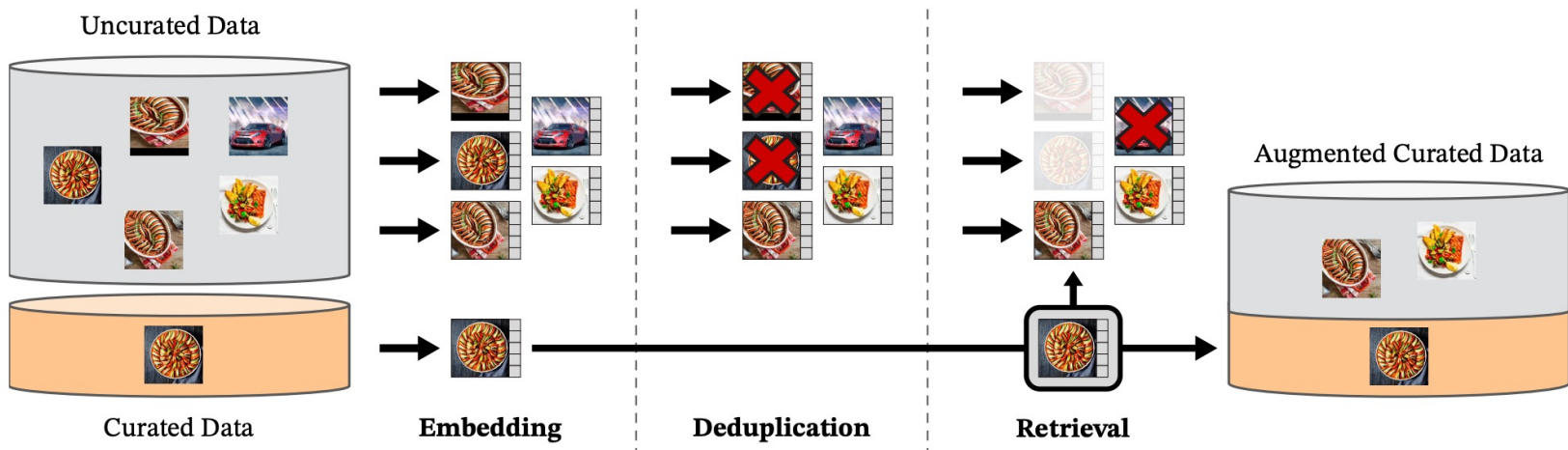
DINO v2: Learning Robust Visual Features without Supervision

- **DINO v2** [Oquab et al., 2023]
 - While there are recent breakthroughs in SSL, CLIP showed better scalability
 - DINO v2 aim to scale the image-only discriminative SSL by
 - Scaling data size by curating data
 - Scaling model size with computational efficient engineering techniques

Method	Arch.	Data	Text sup.	kNN	linear		
				val	val	ReaL	V2
Weakly supervised							
CLIP	ViT-L/14	WIT-400M	✓	79.8	84.3	88.1	75.3
CLIP	ViT-L/14 ₃₃₆	WIT-400M	✓	80.5	85.3	88.8	75.8
SWAG	ViT-H/14	IG3.6B	✓	82.6	85.7	88.7	77.6
OpenCLIP	ViT-H/14	LAION	✓	81.7	84.4	88.4	75.5
OpenCLIP	ViT-G/14	LAION	✓	83.2	86.2	89.4	77.2
EVA-CLIP	ViT-g/14	custom*	✓	83.5	86.4	89.3	77.4
Self-supervised							
MAE	ViT-H/14	INet-1k	✗	49.4	76.6	83.3	64.8
DINO	ViT-S/8	INet-1k	✗	78.6	79.2	85.5	68.2
SEERv2	RG10B	IG2B	✗	–	79.8	–	–
MSN	ViT-L/7	INet-1k	✗	79.2	80.7	86.0	69.7
EsViT	Swin-B/W=14	INet-1k	✗	79.4	81.3	87.0	70.4
Mugs	ViT-L/16	INet-1k	✗	80.2	82.1	86.9	70.8
iBOT	ViT-L/16	INet-22k	✗	72.9	82.3	87.5	72.4
DINOv2	ViT-S/14	LVD-142M	✗	79.0	81.1	86.6	70.9
	ViT-B/14	LVD-142M	✗	82.1	84.5	88.3	75.1
	ViT-L/14	LVD-142M	✗	83.5	86.3	89.5	78.0
	ViT-g/14	LVD-142M	✗	83.5	86.5	89.6	78.4

DINO v2: Learning Robust Visual Features without Supervision

- **DINO v2** [Oquab et al., 2023]
 - Data preprocessing (LVD-142M dataset)
 - Curated dataset from ImageNet and fine-grained dataset
 - Uncurated dataset sourced from crawled web data
 - **Deduplication**: remove near-duplicate images to increase diversity
 - **Self-supervised image retrieval**: using ImageNet-22k pretrained ViT-H/16, retrieve relevant data from uncurated source using K-means clustering



- **DINO v2** [Oquab et al., 2023]
 - Data preprocessing (LVD-142M dataset)
 - Curated dataset from ImageNet and fine-grained dataset
 - Uncurated dataset sourced from crawled web data
 - **Deduplication**: remove near-duplicate images to increase diversity
 - **Self-supervised image retrieval**: using ImageNet-22k pretrained ViT-H/16, retrieve relevant data from uncurated source using K-means clustering
 - LVD-142M maintains ImageNet-1K performance while improving in other domains

Training Data	INet-1k	Im-A	ADE-20k	Oxford-M
INet-22k	85.9	73.5	46.6	62.5
INet-22k \ INet-1k	85.3	70.3	46.2	58.7
Uncurated data	83.3	59.4	48.5	54.3
LVD-142M	85.8	73.9	47.7	64.6

- **DINO v2** [Oquab et al., 2023]

- Training method

- Use both image-level objective in DINO and MIM objective in iBOT
- KoLeo regularizer: minimize the differential entropy of features
 - Encourage features to be uniformly distributed

$$\mathcal{L}_{\text{koleo}} = -\frac{1}{n} \sum_{i=1}^n \log(d_{n,i}), \text{ where } d_{n,i} = \min_{j \neq i} \|x_i - x_j\|$$

- Effect of KoLeo loss term and Masked Image Modeling from iBOT

KoLeo	INet-1k	Im-A	ADE-20k	Oxford-M	MIM	INet-1k	Im-A	ADE-20k	Oxford-M
✗	85.3	70.6	47.2	55.6	✗	85.3	72.0	44.2	64.3
✓	85.8	72.8	47.1	63.9	✓	85.8	72.8	47.1	63.9

(a) Koleo loss

(b) MIM objective in iBOT

DINO v2: Learning Robust Visual Features without Supervision

- **DINO v2** [Oquab et al., 2023]
 - DINO v2 matches domain generalization performance of CLIP
 - Linear probing experiments on ImageNet-A/R/C/Sketch

Method	Arch	Data	Im-A	Im-R	Im-C↓	Sketch
OpenCLIP	ViT-G/14	LAION	63.8	87.8	45.3	66.4
MAE	ViT-H/14	INet-1k	10.2	34.4	61.4	21.9
DINO	ViT-B/8	INet-1k	23.9	37.0	56.6	25.5
iBOT	ViT-L/16	INet-22k	41.5	51.0	43.9	38.5
DINOv2	ViT-S/14	LVD-142M	33.5	53.7	54.4	41.2
	ViT-B/14	LVD-142M	55.1	63.3	42.7	50.6
	ViT-L/14	LVD-142M	71.3	74.4	31.5	59.3
	ViT-g/14	LVD-142M	75.9	78.8	28.2	62.5

- **DINO v2** [Oquab et al., 2023]
 - DINO v2 is better at transferring to vision tasks
 - Semantic segmentation on ADE20K, Cityscapes, Pascal VOC with frozen feature

Method	Arch.	ADE20k (62.9)		CityScapes (86.9)		Pascal VOC (89.0)	
		lin.	+ms	lin.	+ms	lin.	+ms
OpenCLIP	ViT-G/14	39.3	46.0	60.3	70.3	71.4	79.2
MAE	ViT-H/14	33.3	30.7	58.4	61.0	67.6	63.3
DINO	ViT-B/8	31.8	35.2	56.9	66.2	66.4	75.6
iBOT	ViT-L/16	44.6	47.5	64.8	74.5	82.3	84.3
DINOv2	ViT-S/14	44.3	47.2	66.6	77.1	81.1	82.6
	ViT-B/14	47.3	51.3	69.4	80.0	82.5	84.9
	ViT-L/14	47.7	53.1	70.3	80.9	82.1	86.0
	ViT-g/14	49.0	53.0	71.3	81.0	83.0	86.2

DINO v2: Learning Robust Visual Features without Supervision

- **DINO v2** [Oquab et al., 2023]
 - DINO v2 is better at transferring to vision tasks
 - Semantic segmentation on ADE20K, Cityscapes, Pascal VOC with frozen feature
 - Depth estimation on NYUd, KITTI, NYUd -> SUN RGB-D with frozen feature

Method	Arch.	NYUd (0.330)			KITTI (2.10)			NYUd → SUN RGB-D (0.421)		
		lin. 1	lin. 4	DPT	lin. 1	lin. 4	DPT	lin. 1	lin. 4	DPT
OpenCLIP	ViT-G/14	0.541	0.510	0.414	3.57	3.21	2.56	0.537	0.476	0.408
MAE	ViT-H/14	0.517	0.483	0.415	3.66	3.26	2.59	0.545	0.523	0.506
DINO	ViT-B/8	0.555	0.539	0.492	3.81	3.56	2.74	0.553	0.541	0.520
iBOT	ViT-L/16	0.417	0.387	0.358	3.31	3.07	2.55	0.447	0.435	0.426
DINOv2	ViT-S/14	0.449	0.417	0.356	3.10	2.86	2.34	0.477	0.431	0.409
	ViT-B/14	0.399	0.362	0.317	2.90	2.59	2.23	0.448	0.400	0.377
	ViT-L/14	0.384	0.333	0.293	2.78	2.50	2.14	0.429	0.396	0.360
	ViT-g/14	0.344	0.298	0.279	2.62	2.35	2.11	0.402	0.362	0.338

1. Introduction

- Foundation models in vision tasks

2. Discriminative Visual Foundation Models

- Self-supervised Learning
- Image-text Contrastive Learning
- Multimodal LLM

3. Generative visual foundation models

- Text-to-Image Diffusion models
- Applications

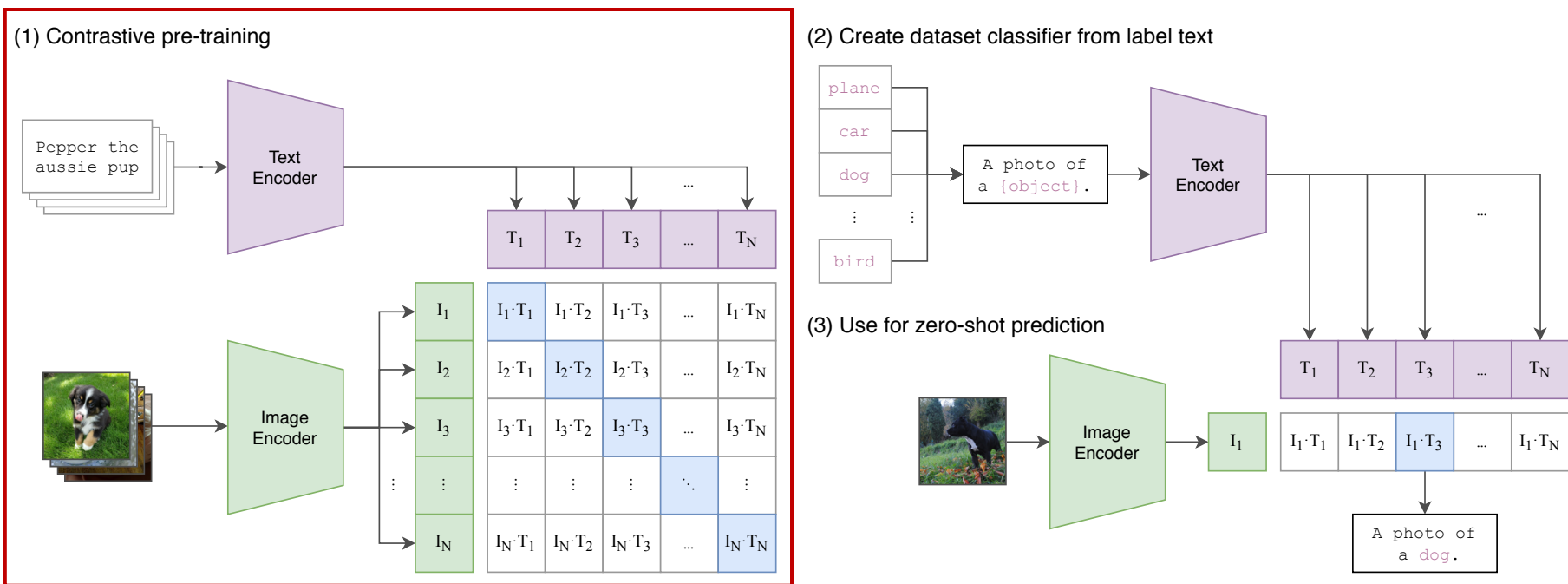
4. Segment Anything

CLIP: Contrastive Language-Image Pre-training

CLIP [Radford et al., 2020]

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

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

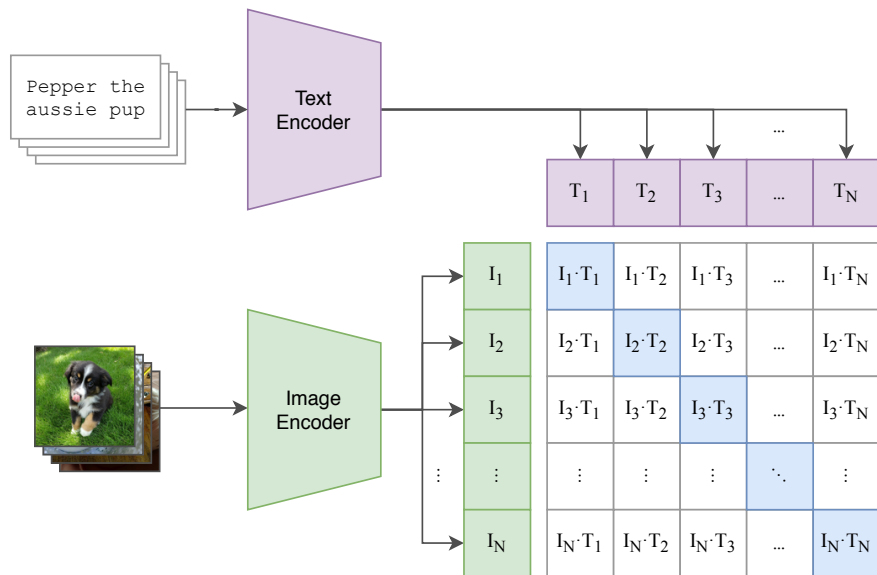


CLIP: Contrastive Language-Image Pre-training

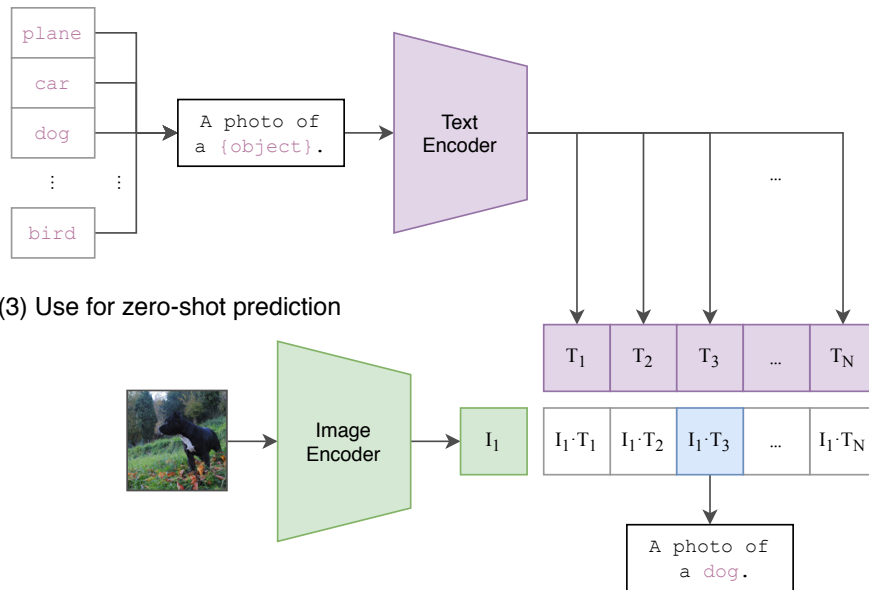
CLIP [Radford et al., 2020]

- 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

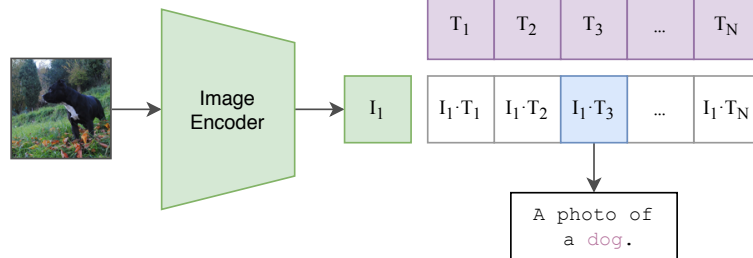
(1) Contrastive pre-training



(2) Create dataset classifier from label text



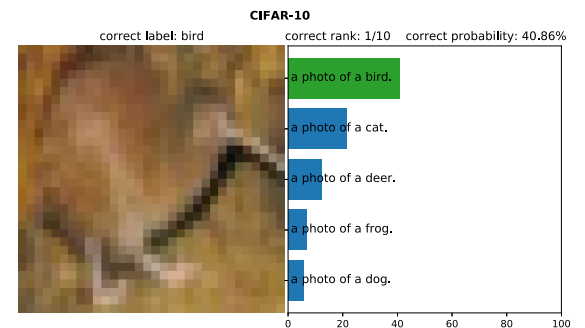
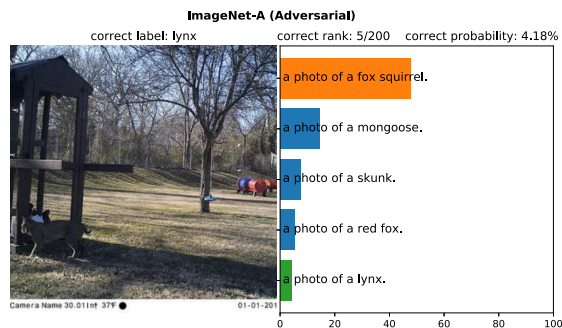
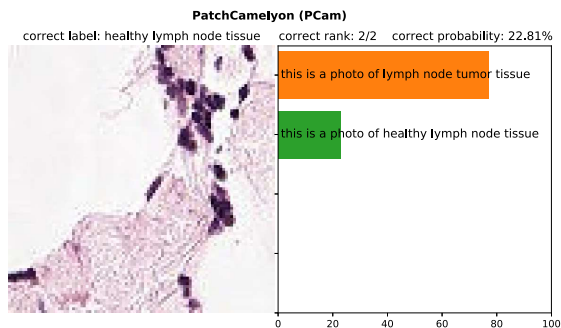
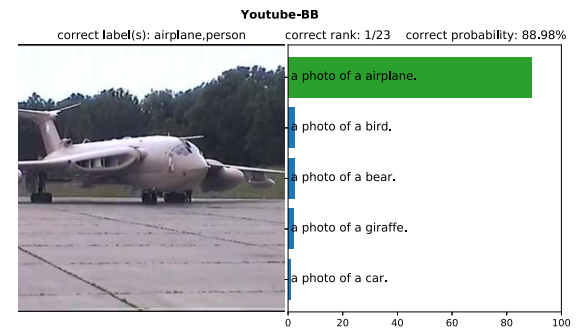
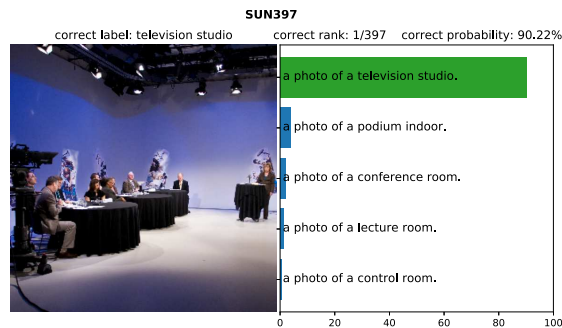
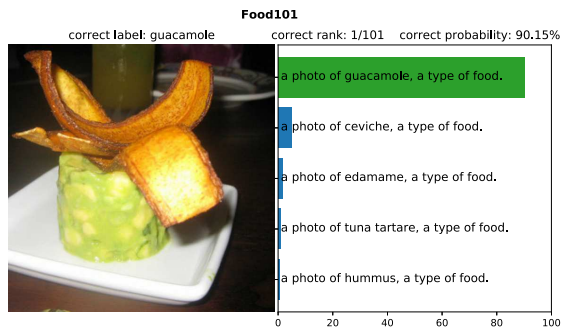
(3) Use for zero-shot prediction



CLIP: Contrastive Language-Image Pre-training

CLIP [Radford et al., 2020]

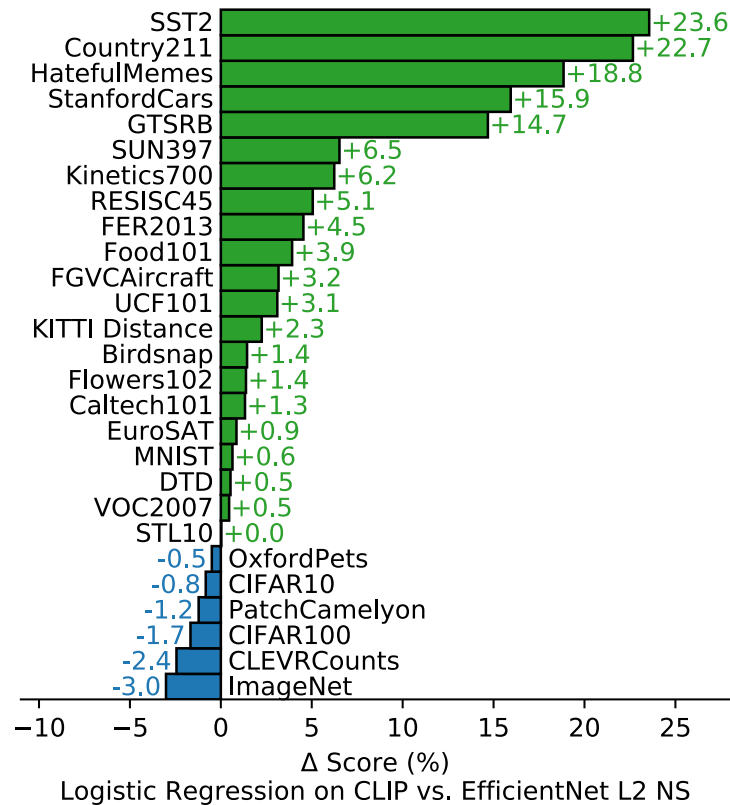
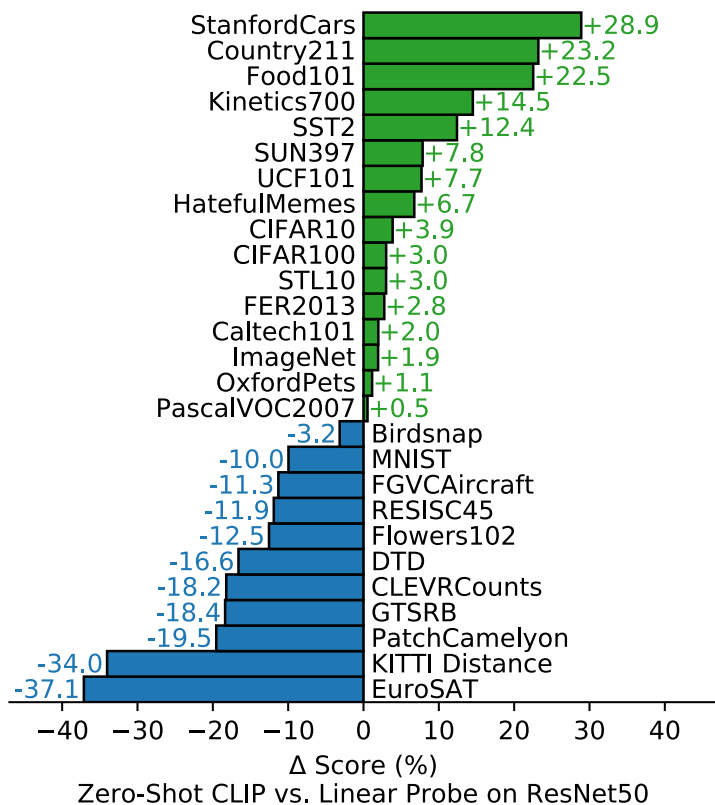
- Zero-shot transfer
 - Transfer learning without seeing the images or labels
 - **Prompt Engineering:** "A photo of a [MASK]"
 - Choose class that maximizes similarity with respect to image



CLIP: Contrastive Language-Image Pre-training

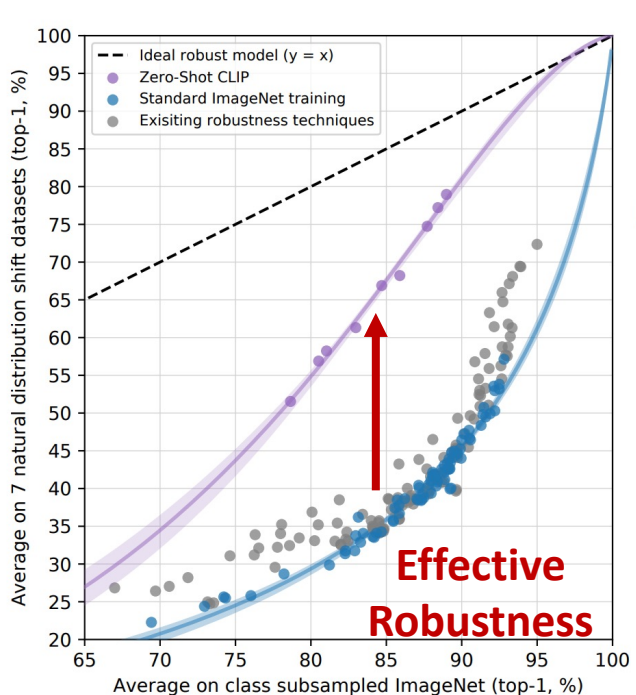
CLIP [Radford et al., 2020]

- 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

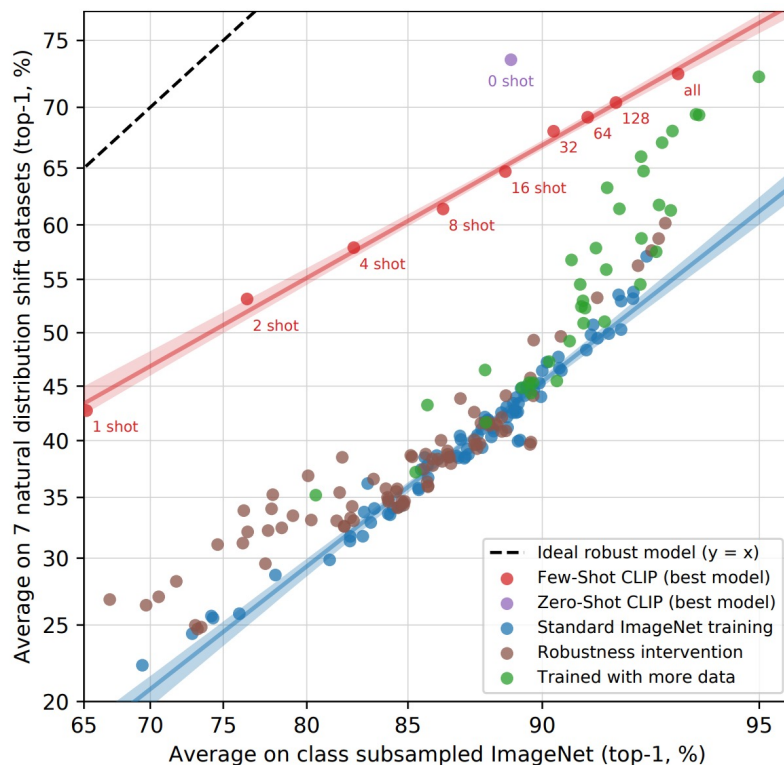


	Dataset Examples	ImageNet ResNet101	Zero-Shot CLIP	Δ Score
ImageNet		76.2	76.2	0%
ImageNetV2		64.3	70.1	+5.8%
ImageNet-R		37.7	88.9	+51.2%
ObjectNet		32.6	72.3	+39.7%
ImageNet Sketch		25.2	60.2	+35.0%
ImageNet-A		2.7	77.1	+74.4%

CLIP: Contrastive Language-Image Pre-training

CLIP [Radford et al., 2020]

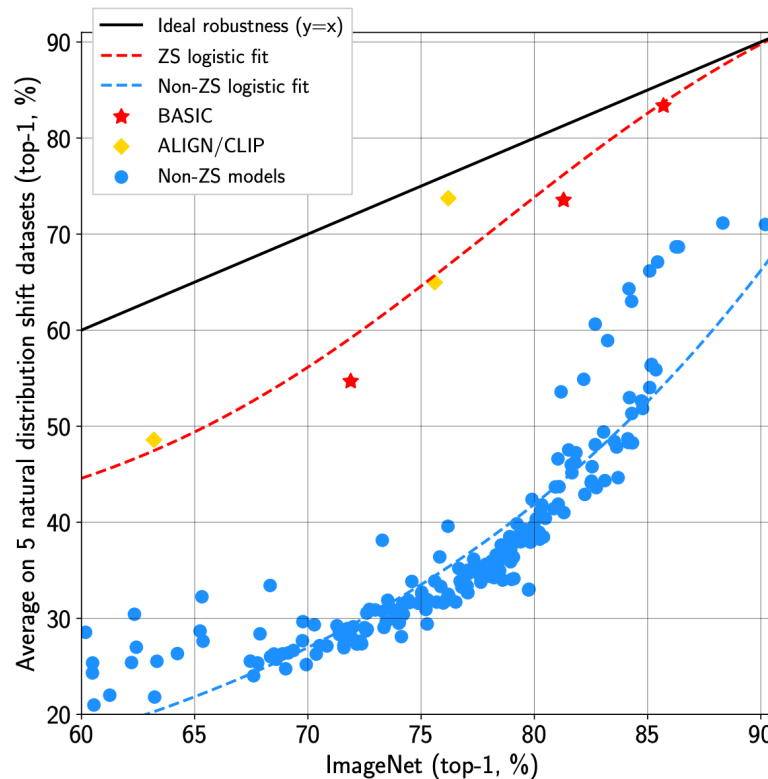
- Zero-shot CLIP classifier is more robust to natural **distributional shift**
 - Interestingly, [Ilharco et al., 2021] show that CLIP have high **effective robustness** even at small scale
- Few-shot CLIP classifier also shows high effective robustness, but less than zero-shot CLIP classifier



Scaling Up dataset size for improved CLIP

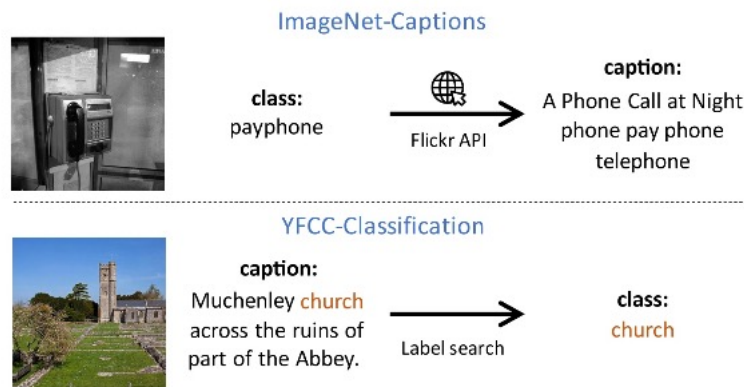
Follow-up studies showed scaling dataset size improves performance

- 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



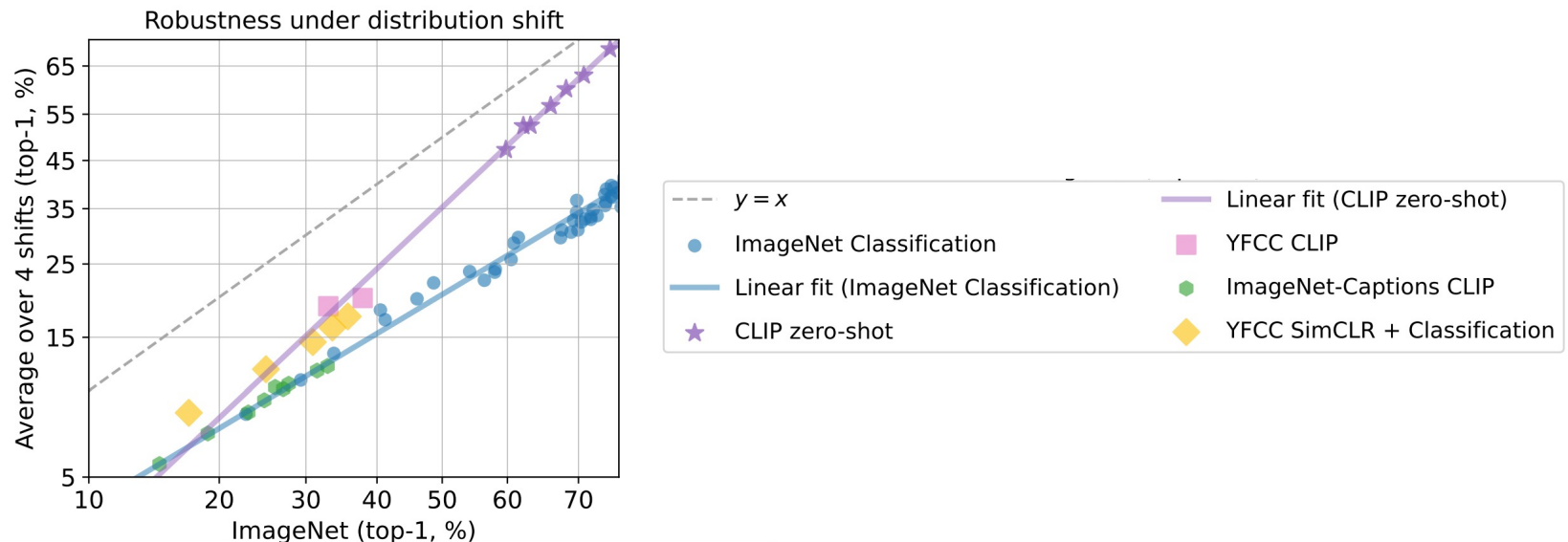
Motivation: What causes CLIP's unprecedented robustness?

- [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!

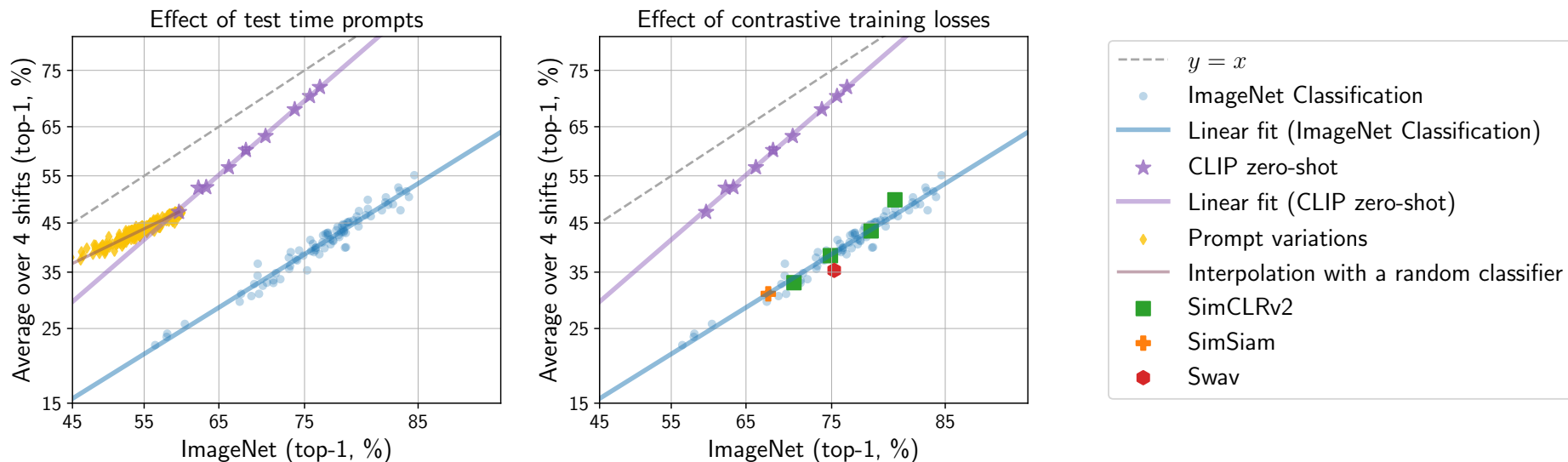


Motivation: What causes CLIP's unprecedented robustness?

- [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

Dataset Design and Distributional Robustness

- **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

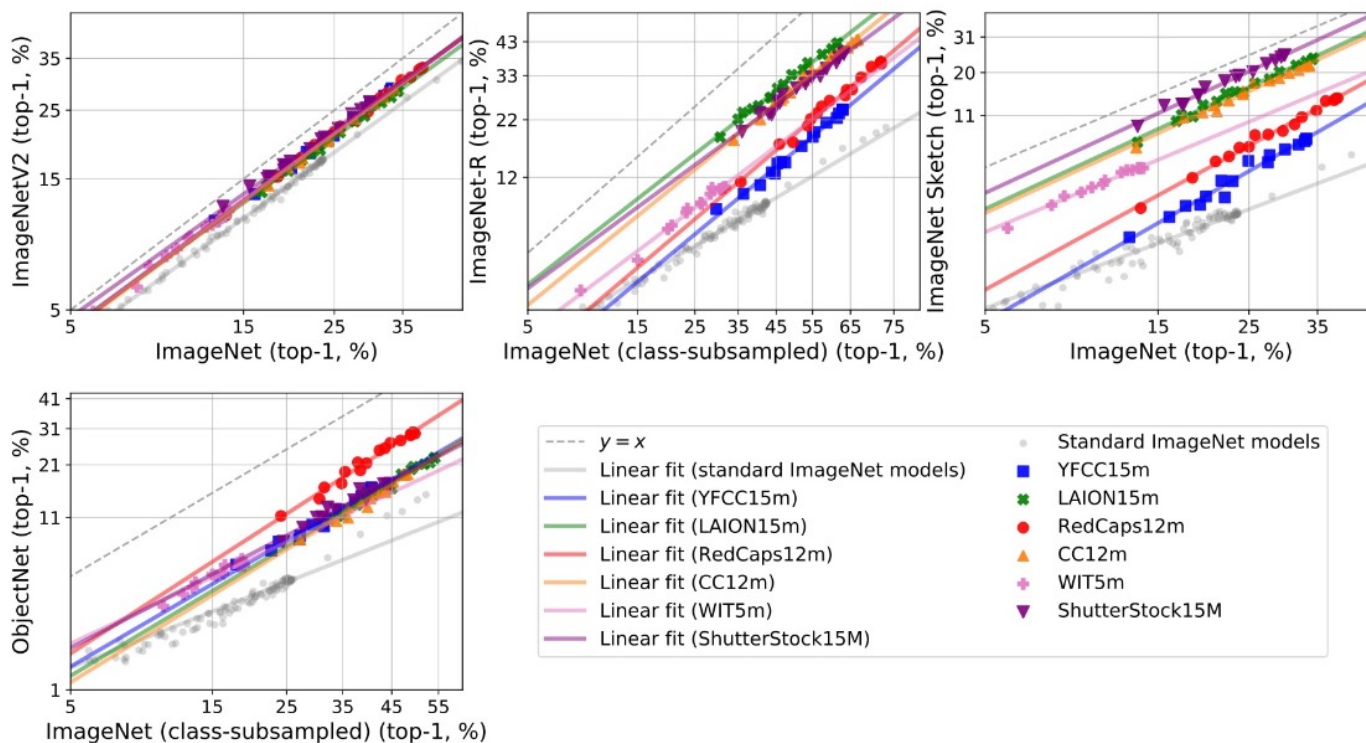


Motivation: What causes CLIP's unprecedented robustness?

- [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 at 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?**

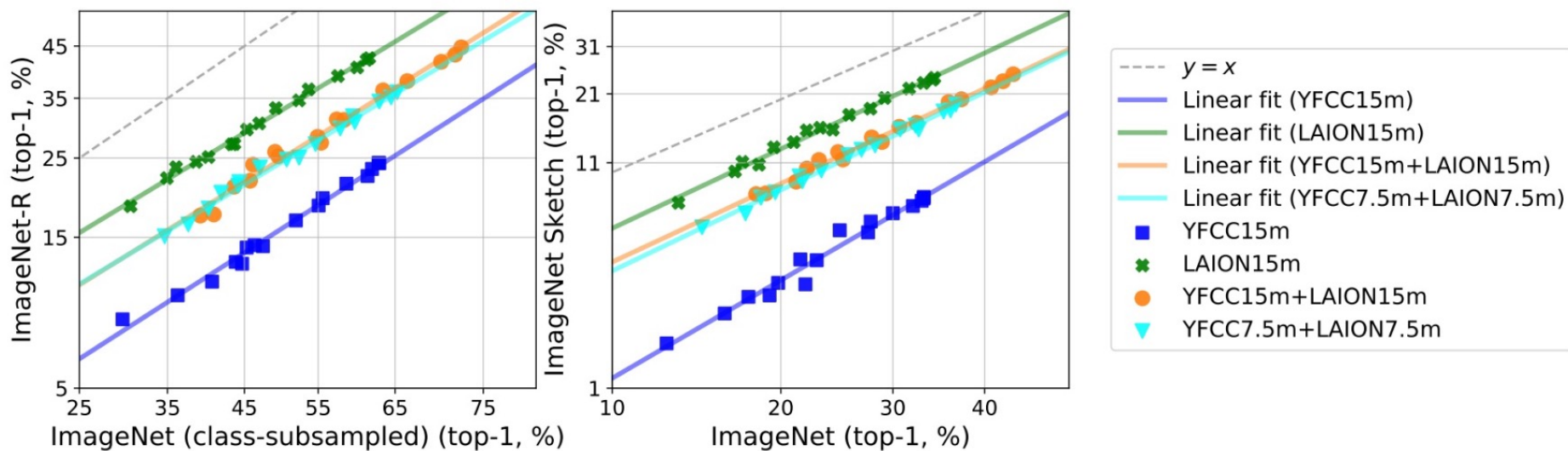
Motivation: Why don't we simply gather all image-text pairs for training data?

- [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
 - $\text{Robustness}(\text{YFCC}) < \text{Robustness}(\text{YFCC}+\text{LAION}) < \text{Robustness}(\text{LAION})$



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
 - $\text{Robustness}(\text{YFCC}) < \text{Robustness}(\text{YFCC}+\text{LAION}) < \text{Robustness}(\text{LAION})$
- 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

Reproducible Scaling law for CLIP

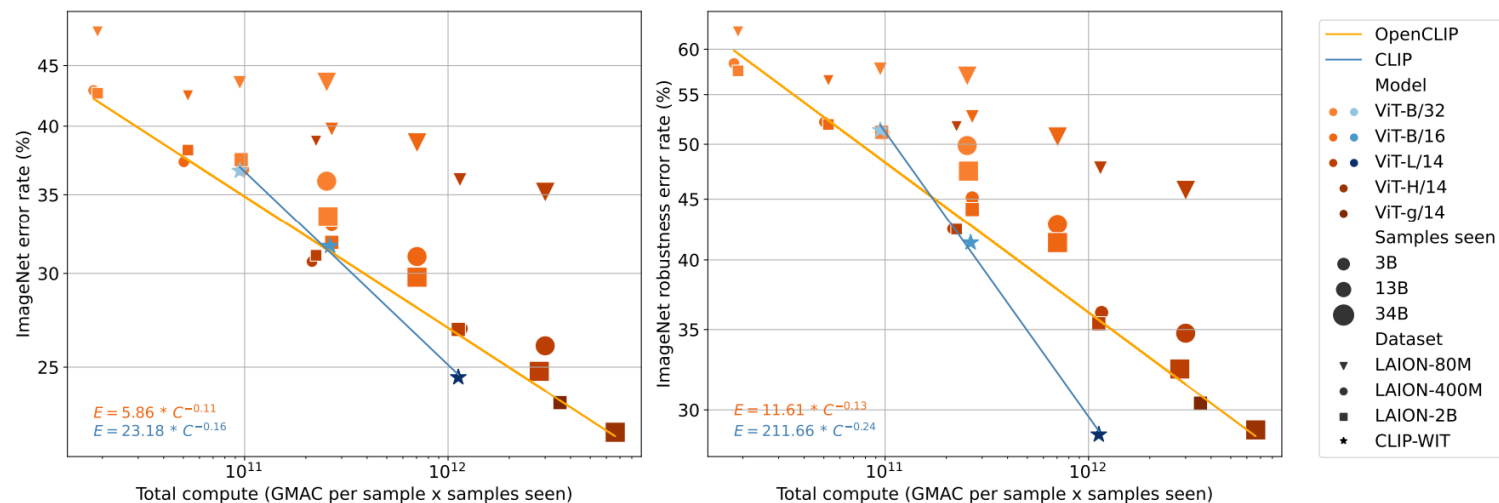
Since OpenAI do not release dataset, many have tried to reproduce its performance

Some open-source approaches in reproducing CLIP:

- OpenCLIP [Ilharco et al., 2021] is a open-source re-implementation of CLIP
- LAION [Schuhmann et al., 2022] is a public large-scale image-text pair

Then, they together performed a study on the scaling behavior of CLIP

- OpenAI's WIT dataset show better scaling than LAION

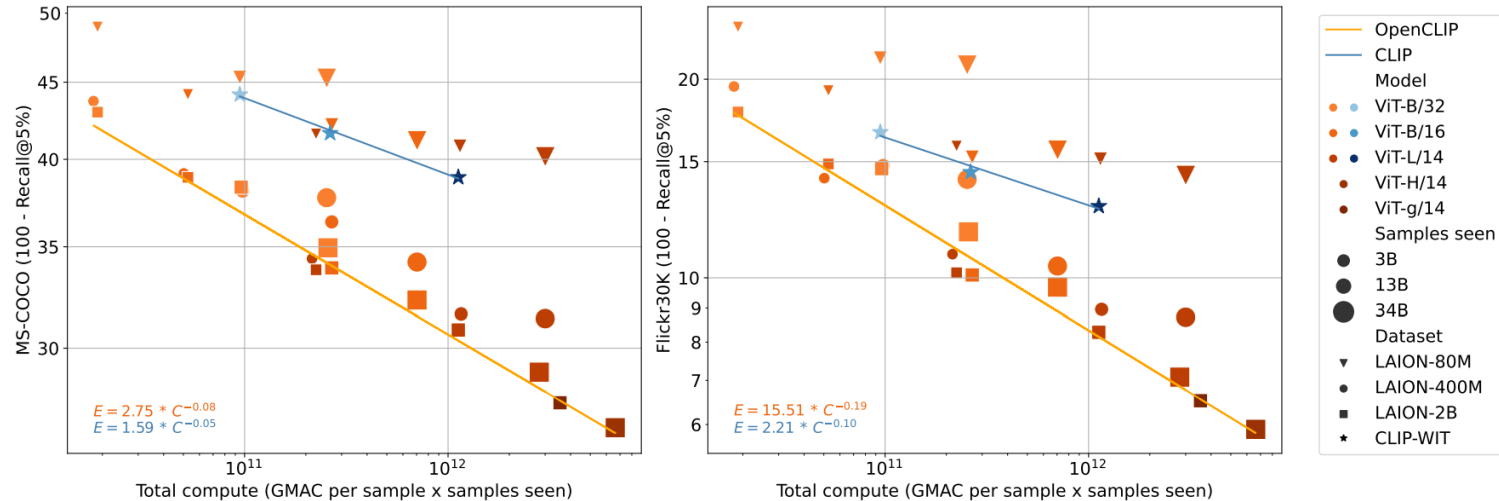


Reproducible Scaling law for CLIP

Then, they together performed a study on the scaling behavior of CLIP

- OpenAI's WIT dataset show better scaling than LAION on ImageNet accuracy
- LAION dataset show better scaling than OpenAI's WIT on COCO image-text retrieval

=> **Scaling leads to better performance, but scaling behavior depends on task type and dataset**



Sigmoid Loss for Language Image Pre-training

We have seen that training data is crucial in CLIP, then how about training loss?

SigLIP [Zhai et al., 2023] propose Sigmoid loss for image-text pretraining which

- Has more efficient implementation
- And better scaling performance compared to CLIP's softmax loss

In specific, recall the CLIP's softmax normalization for image-text contrastive loss:

- t is a learnable temperature parameter
- The normalization should be performed twice: across images and texts

$$-\frac{1}{2|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \left(\overbrace{\log \frac{e^{t\mathbf{x}_i \cdot \mathbf{y}_i}}{\sum_{j=1}^{|\mathcal{B}|} e^{t\mathbf{x}_i \cdot \mathbf{y}_j}}}_{\text{image} \rightarrow \text{text softmax}} + \overbrace{\log \frac{e^{t\mathbf{x}_i \cdot \mathbf{y}_i}}{\sum_{j=1}^{|\mathcal{B}|} e^{t\mathbf{x}_j \cdot \mathbf{y}_i}}}_{\text{text} \rightarrow \text{image softmax}} \right)$$

Sigmoid Loss for Language Image Pre-training

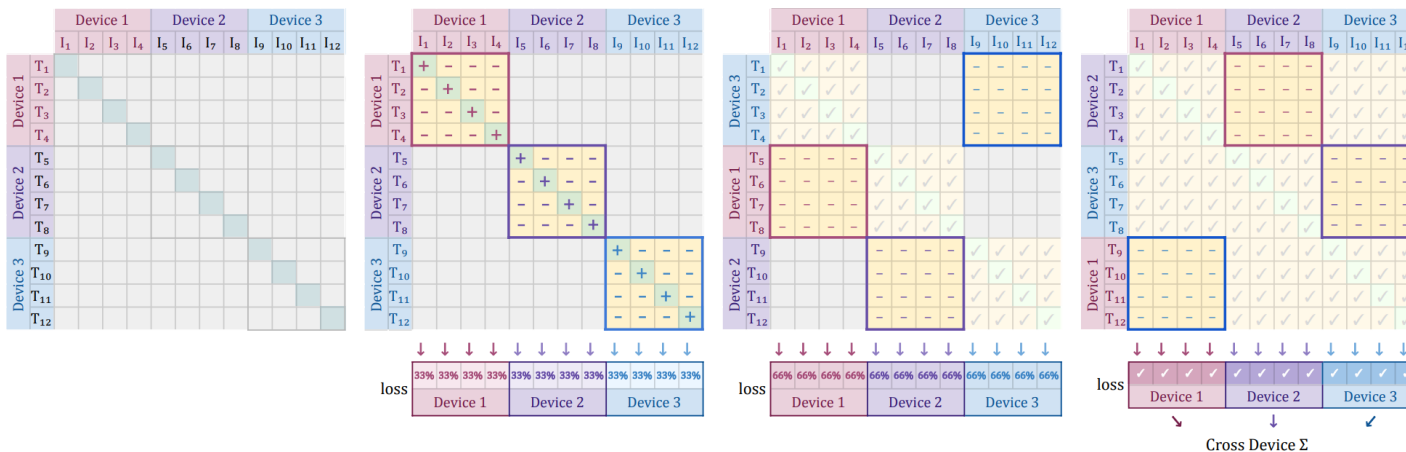
Instead, Sigmoid loss compute every image-text pair independently:

- z_{ij} : 1 if paired -1 otherwise
- t : learnable temperature parameter
- b : learnable bias term

$$-\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \sum_{j=1}^{|\mathcal{B}|} \underbrace{\log \frac{1}{1 + e^{z_{ij}(-t\mathbf{x}_i \cdot \mathbf{y}_j + b)}}}_{\mathcal{L}_{ij}}$$

Efficient implementation

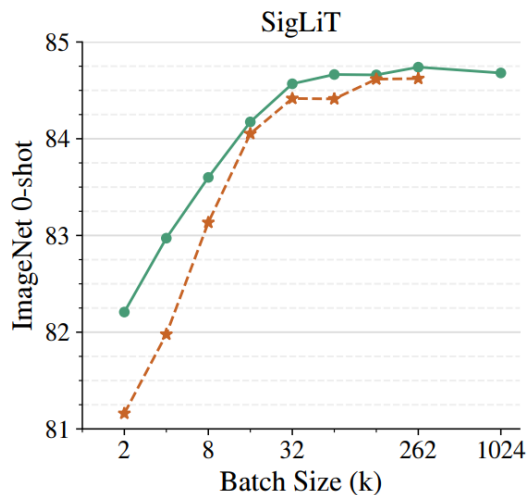
- Conventional contrastive loss requires expensive ‘all-gather’ of embeddings that results in memory-intensive $B \times B$ matrix
- On the other hand, Sigmoid loss is memory efficient, fast, and stable by summation of the loss by swapping negatives across device:



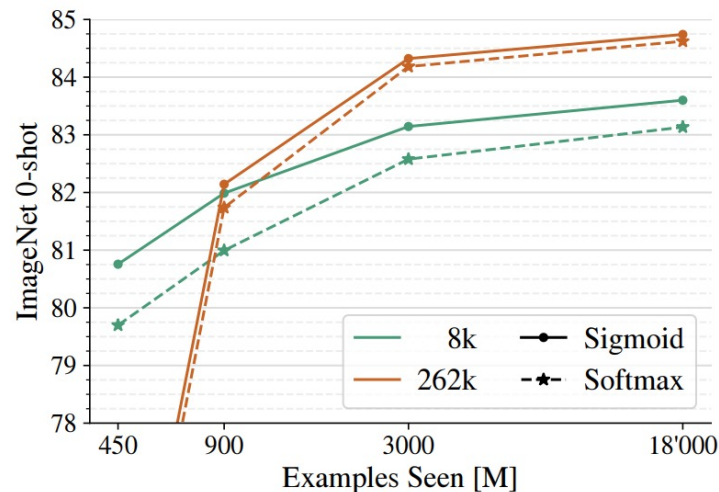
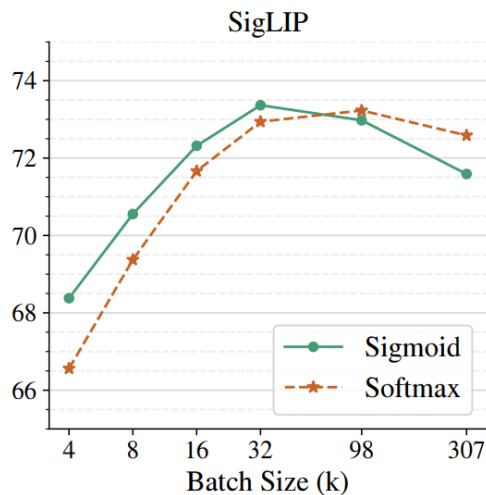
Sigmoid Loss for Language Image Pre-training

As a result, SigLIP (i.e., image-text pretraining with Sigmoid loss) can afford larger batch size with stable training loss, thus results in better scalability

- Sigmoid is better than Softmax at small batch size, but similar at large batch size
- Sigmoid show better scaling behavior than Softmax



Effect of batch size



Effect of data scaling

*SigLiT is Sigmoid loss with Locked-Image Tuning which use pretrained ViT from ImageNet-22K and only fine-tune text encoder using image-caption pairs

Sigmoid Loss for Language Image Pre-training

As a result, SigLIP (i.e., image-text pretraining with Sigmoid loss) can afford larger batch size with stable training loss, thus results in better scalability

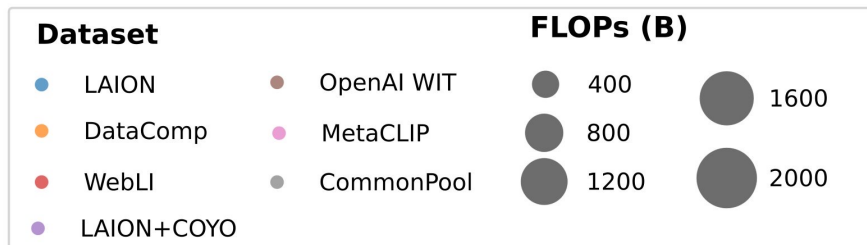
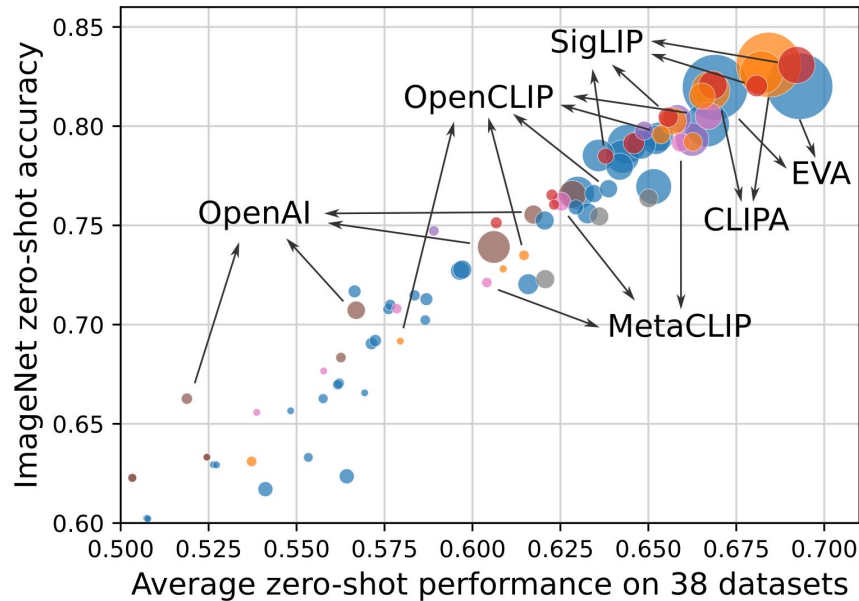
- Comparison to various models

Method	Image Encoder		ImageNet-1k				COCO R@1	
	ViT size	# Patches	Validation	v2	ReaL	ObjectNet	I → T	T → I
CLIP	B	196	68.3	61.9	-	55.3	52.4	33.1
OpenCLIP	B	196	70.2	62.3	-	56.0	59.4	42.3
EVA-CLIP	B	196	74.7	67.0	-	62.3	58.7	42.2
SigLIP	B	196	76.2	69.6	82.8	70.7	64.4	47.2
SigLIP	B	256	76.7	70.0	83.1	71.3	65.1	47.4
SigLIP	B	576	78.6	72.1	84.5	73.8	67.5	49.7
SigLIP	B	1024	79.2	73.0	84.9	74.7	67.6	50.4
CLIP	L	256	75.5	69.0	-	69.9	56.3	36.5
OpenCLIP	L	256	74.0	61.1	-	66.4	62.1	46.1
CLIPA-v2	L	256	79.7	72.8	-	71.1	64.1	46.3
EVA-CLIP	L	256	79.8	72.9	-	75.3	63.7	47.5
SigLIP	L	256	80.5	74.2	85.9	77.9	69.5	51.1
CLIP	L	576	76.6	72.0	-	70.9	57.9	37.1
CLIPA-v2	L	576	80.3	73.5	-	73.1	65.5	47.2
EVA-CLIP	L	576	80.4	73.8	-	78.4	64.1	47.9
SigLIP	L	576	82.1	75.9	87.0	81.0	70.6	52.7
OpenCLIP	G (2B)	256	80.1	73.6	-	73.0	67.3	51.4
CLIPA-v2	H (630M)	576	81.8	75.6	-	77.4	67.2	49.2
EVA-CLIP	E (5B)	256	82.0	75.7	-	79.6	68.8	51.1
SigLIP	SO (400M)	729	83.2	77.2	87.5	82.9	70.2	52.0

Sigmoid Loss for Language Image Pre-training

As a result, SigLIP (i.e., image-text pretraining with Sigmoid loss) can afford larger batch size with stable training loss, thus results in better scalability

- Comparison to various models
- Trends of CLIP models



1. Introduction

- Foundation models in vision tasks

2. Discriminative Visual Foundation Models

- Self-supervised Learning
- Image-text Contrastive Learning
- Multimodal LLM

3. Generative visual foundation models

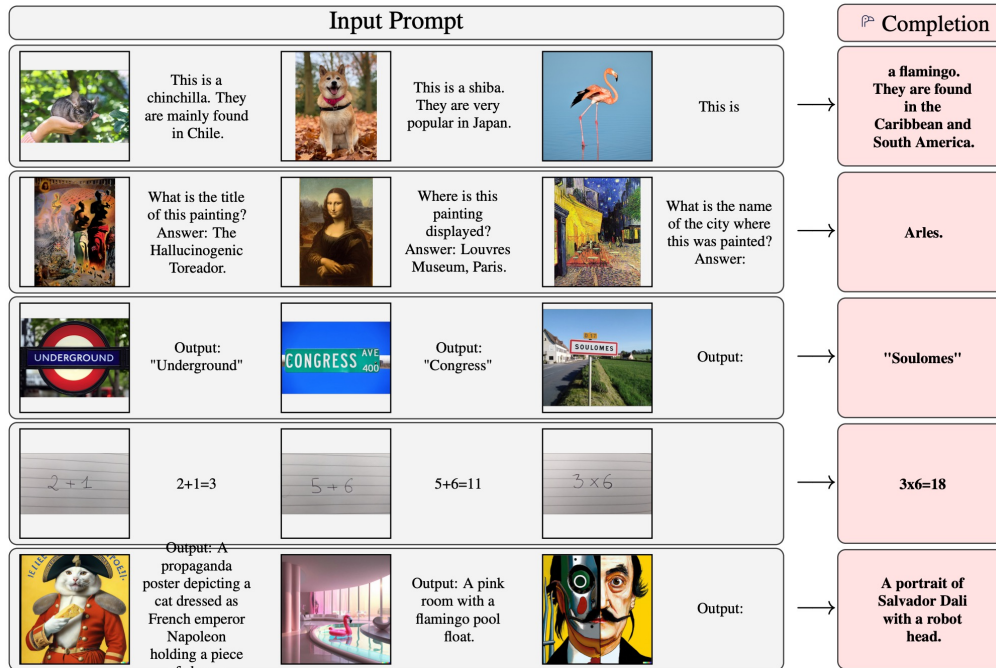
- Text-to-Image Diffusion models
- Applications

4. Segment Anything

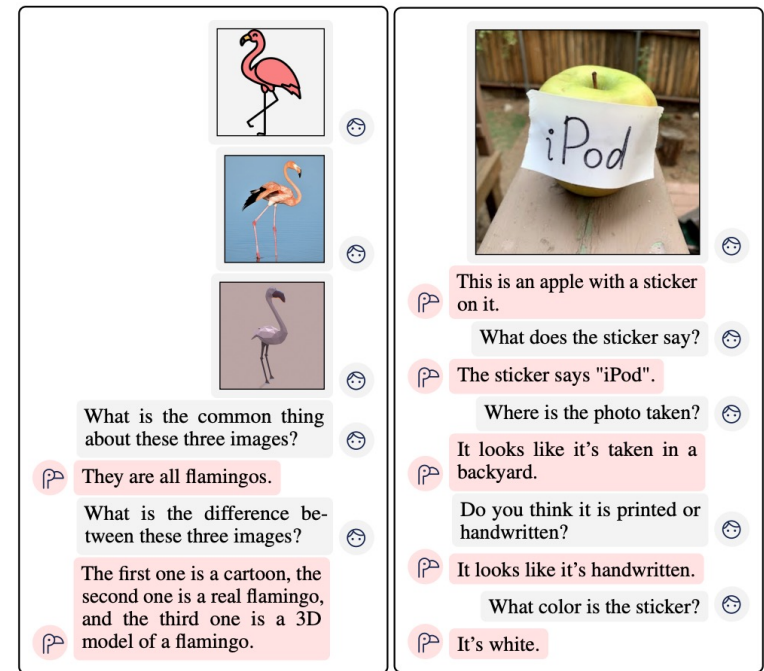
Flamingo: a Visual Language Model for Few-Shot Learning

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

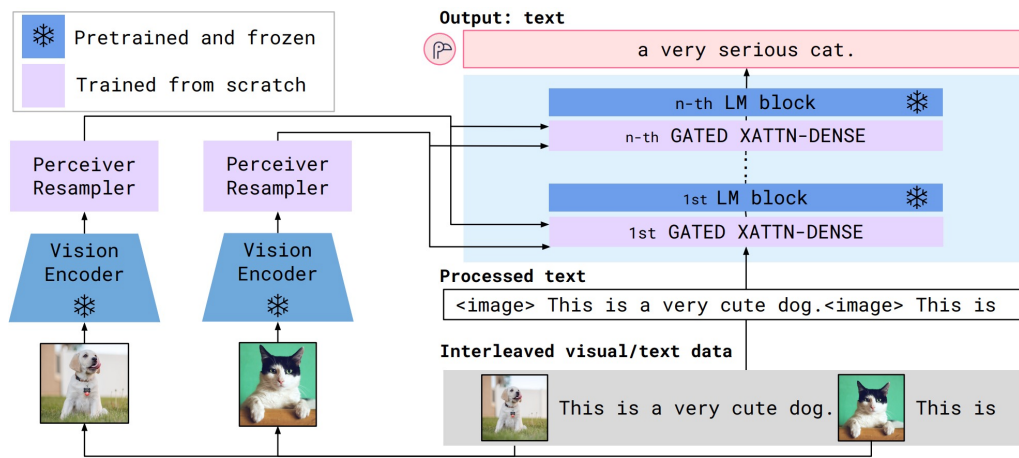


Multimodal visual dialogue

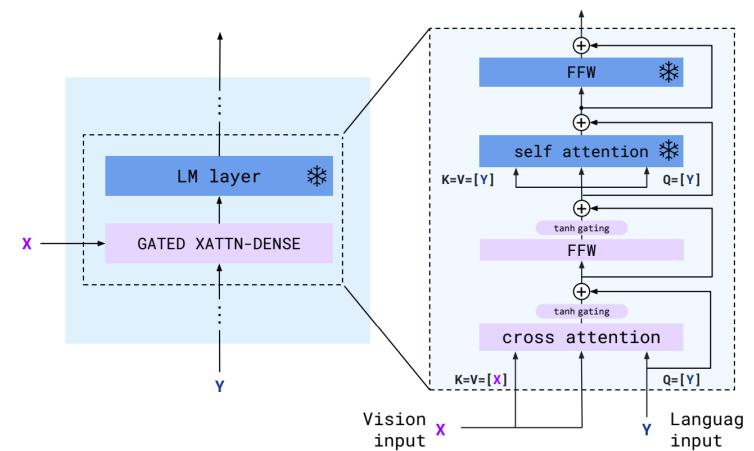
Flamingo: a Visual Language Model for Few-Shot Learning

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

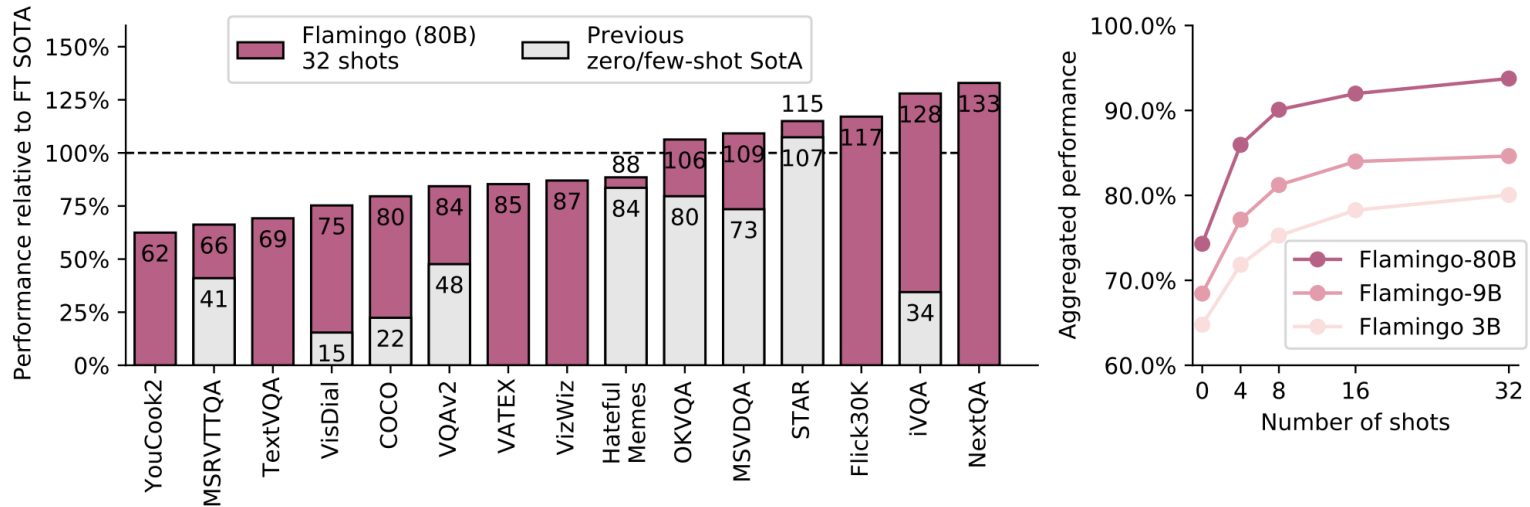
Flamingo [Alayrac et al., 2022]

- 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



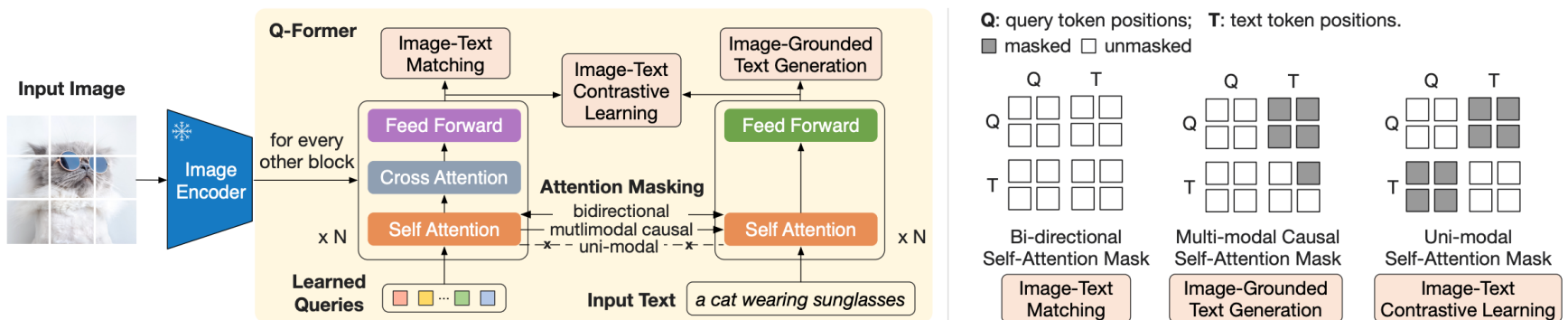
- When fine-tuned, it achieves SOTA various tasks

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 [†]	81.3 [†]	149.6[†]	81.4 [†]	57.2 [†]	60.6 [†]	46.8	75.2	75.4[†]	138.7	54.7	73.7	84.6 [†]
	[133]	[133]	[119]	[153]	[65]	[65]	[51]	[79]	[123]	[132]	[137]	[84]	[152]

BLIP-2: BLIP with Frozen Image Encoders and LLM

BLIP-2 [Li et al., 2023]

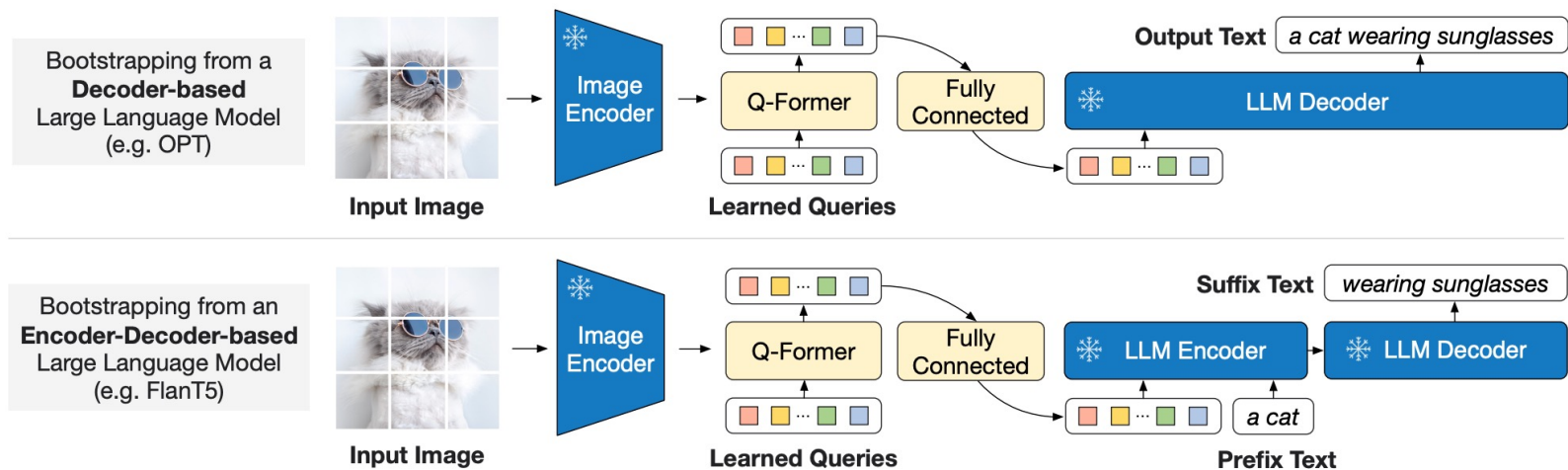
- Lighter approach for aligning pretrained vision encoder and LLM for VL tasks
- Propose two-stage alignment using **Q-former**
 - **Stage 1:** Representation learning with Q-former
 - Q-former: BERT initialized transformer that encodes visual information given query
 - Various learning objectives used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



BLIP-2: BLIP with Frozen Image Encoders and LLM

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: BLIP with Frozen Image Encoders and LLM

BLIP-2 [Li et al., 2023]

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

Models	#Trainable Params	Open-sourced?	Visual Question Answering	Image Captioning		Image-Text Retrieval	
			VQAv2 (test-dev) VQA acc.	NoCaps (val) CIDEr	SPICE	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		OK-VQA	GQA
			val	test-dev	test	test-dev
VL-T5 _{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	<u>63.1</u>	<u>63.0</u>	40.7	44.2
BLIP-2 ViT-G FlanT5 _{XXL}	108M	12.1B	65.2	65.0	<u>45.9</u>	44.7

BLIP-2: BLIP with Frozen Image Encoders and LLM

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
<i>Dual-encoder models</i>													
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	<u>84.8</u>	<u>96.5</u>	<u>98.3</u>	<u>67.2</u>	87.7	92.8
<i>Fusion-encoder models</i>													
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
<i>Dual encoder + Fusion encoder reranking</i>													
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	<u>92.6</u>

Visual Instruction Tuning

LLaVA [Liu et al., 2023]

- Using pre-trained vision encoder and pre-trained LLM (LLaMA) for visual understanding
- Given pre-trained LLM, map an image with vision encoder (CLIP ViT-L/14) into grid features and map to LLM word embedding space using learnable projector
 - Stage 1. feature alignment: pretrain projector on small image-text pairs to map vision encoders into LLM word embedding space
 - Stage 2. Instruction tuning: keep the visual encoder frozen, and fine-tune projector and LLM
- For Stage 2., they collected multimodal instruction-following dataset using GPT-4 and ChatGPT for conversation, detailed description, and complex reasoning

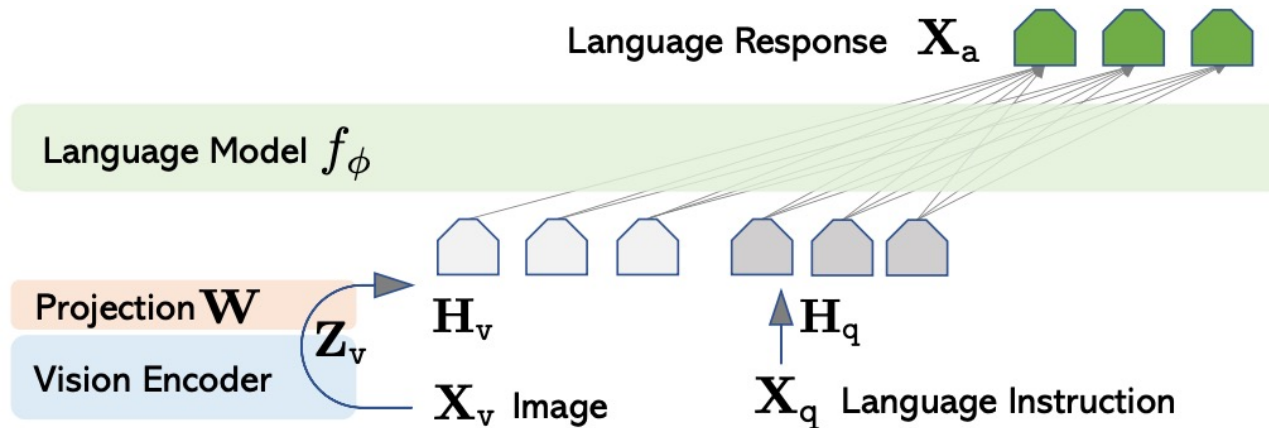


Figure 1: LLaVA network architecture.

LLaVA [Liu et al., 2023]

- Using pre-trained vision encoder and pre-trained LLM (LLaMA) for visual understanding
- LLaVA show comparable performance to GPT-4 in multimodal understanding and conversational capability

Visual input example, Extreme Ironing:



Source: <https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg>

User	What is unusual about this image?
LLaVA	The unusual aspect of this image is a man ironing clothes on the back of a minivan or van. This is not a typical place to perform this activity, as one would usually iron clothes in a more stationary and safe location, such as a home, using a regular ironing board. The scene depicted in the image is peculiar as it involves a makeshift ironing setup on a vehicle, which can be both unsafe and unconventional. Additionally, it is not clear how the man is able to maintain balance and stability while ironing clothes in such an unstable environment.
User	[Start a new conversation, and clear the history]
User	What's happening in the scene?
LLaVA	The image depicts a man standing on top of a yellow SUV in a busy city street. He is holding a portable ladder, seemingly ironing clothes while standing on the vehicle. Around the scene, there are other cars, a traffic light, and a fire hydrant. Additionally, there are two other people in the vicinity, one close to the man on the car and another further away. The man on the ladder appears to be attracting attention due to his unconventional choice of ironing his clothes on top of a moving car . The city street around him is bustling with activity, adding to the unique nature of the scene.
User	What is unusual about this image?
GPT-4 [32]	The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.
User	Can you explain this meme in detail?
BLIP-2	a man is sitting on the back of a yellow cab
User	Can you explain this meme in detail?
OpenFlamingo	The man is drying his clothes on the hood of his car.

Visual input example, Chicken Nugget Map:

Sometimes I just look at pictures of the earth from space and I marvel at how beautiful it all is.



User	Can you explain this meme in detail?
LLaVA	The meme in the image is a creative and humorous take on food, with a focus on chicken nuggets, as the center of the universe. The meme begins with the phrase "Sometimes I just look at pictures of the Earth from space and I marvel at how beautiful it all is..." The main image features a tray of cooked chicken nuggets, with the chicken pieces positioned to resemble the continents and islands on a world map. The meme playfully suggests that the chicken nuggets represent the Earth, and the various locations depicted in the photo are actually chicken nugget versions of different places. The punchline of the meme appears in smaller text at the bottom, reading "I mean, it's not the real Earth, but how beautiful it is all is." This lighthearted and entertaining meme combines food and humor, using the image of chicken nuggets to create a funny and imaginative representation of the world.
User	Can you explain this meme?
GPT-4 [32]	This meme is a joke that combines two unrelated things: pictures of the earth from space and chicken nuggets. The text of the meme suggests that the image below is a beautiful picture of the earth from space. However, the image is actually of chicken nuggets arranged to vaguely resemble a map of the world. The humor in this meme comes from the unexpected juxtaposition of the text and the image. The text sets up an expectation of a majestic image of the earth, but the image is actually something mundane and silly.
User	Can you explain this meme in detail?
BLIP-2	Can you explain this meme in detail? sometimes i just look at pictures of the earth from space and marvel how beautiful it is
User	Can you explain this meme in detail?
OpenFlamingo	It's a picture of a chicken nugget on the International Space Station.

1. Introduction

- Foundation models in vision tasks

2. Discriminative Visual Foundation Models

- Self-supervised Learning
- Image-text Contrastive Learning
- Multimodal LLM

3. **Generative visual foundation models**

- Text-to-Image Diffusion models
- Applications

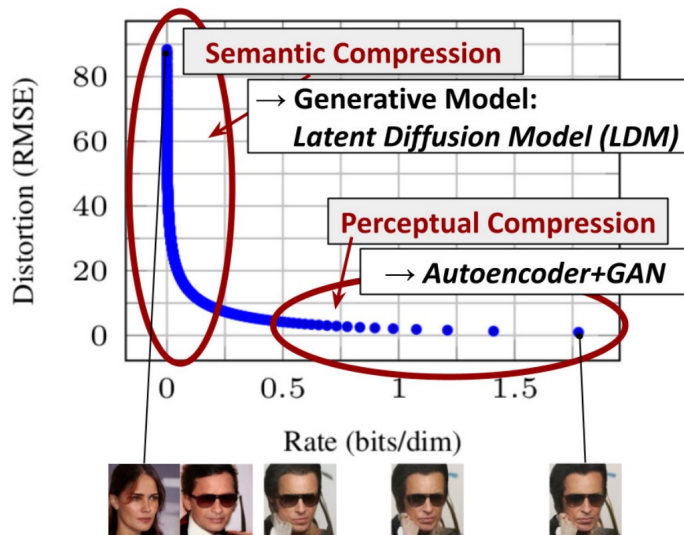
4. Segment Anything

Development of Text-to-Image (T2I) Diffusion Models

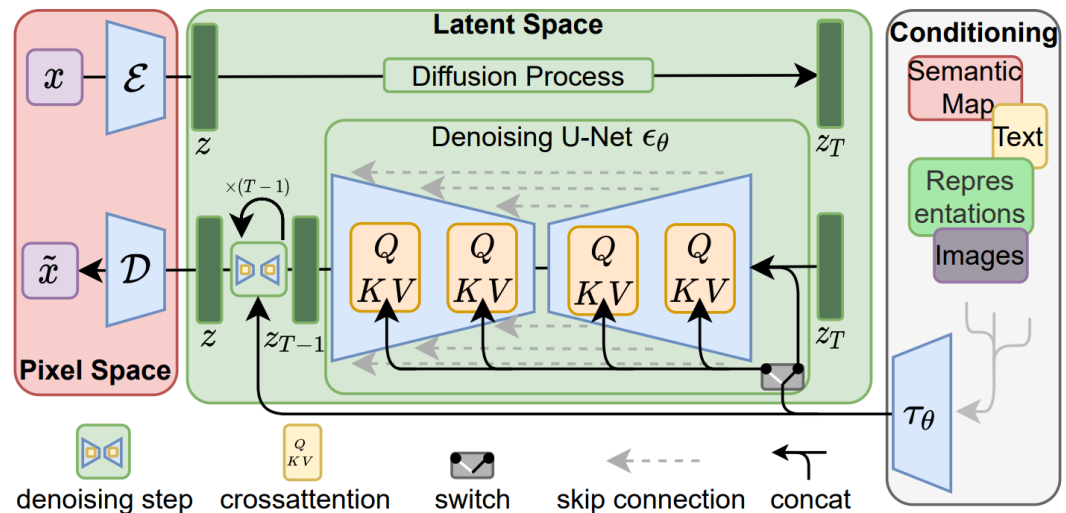
- Due to the scalability of diffusion models and the presence of numerous image-caption pairs, various T2I Diffusion models have been proposed
 - Latent Diffusion Models (i.e., Stable Diffusion)
 - Cascaded Diffusion Models (e.g., Imagen, DeepFloyd-IF)
- In this lecture, we will explore various text-to-image diffusion models and their applications to various tasks, expanding the capabilities
 - Image editing
 - Controllable generation and personalization
 - Extending to other modalities (e.g., Text-to-3D, Text-to-Video)

Latent Diffusion Models (a.k.a Stable Diffusion) [Rombach et al., 2022]

- Training a diffusion model on the pixel space is too memory expensive
- Latent Diffusion Models (LDMs) handle this problem by compressing an image into lower dimensional latent, and train diffusion model on the latent space
- LDM first use condition text embeddings on cross-attention layer



Perceptual & Semantic Compression

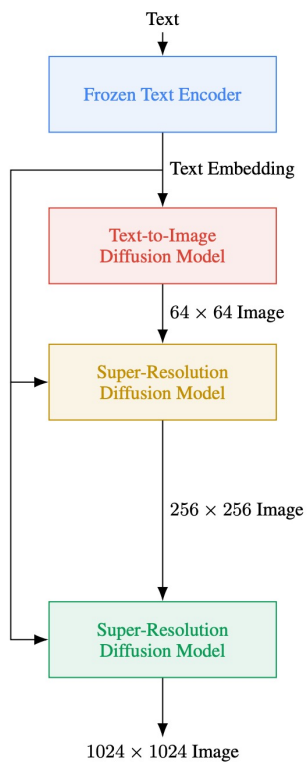


LDM architecture

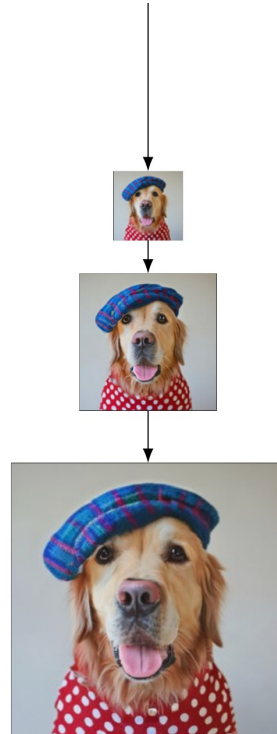
Text-to-Image Diffusion Models

Imagen [Saharia et al., 2022]

- Imagen first used large language models (i.e., T5) as text encoder, and train by conditioning on cascaded U-Nets of size 64 -> 256 -> 1024
- Imagen use Classifier-Free Guidance [Ho et al., 2022] to control sample quality and diversity



“A Golden Retriever dog wearing a blue checkered beret and red dotted turtleneck.”



Sprouts in the shape of text 'Imagen' coming out of a fairytale book.



A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.



Teddy bears swimming at the Olympics 400m Butterfly event.



A cute corgi lives in a house made out of sushi.

Stable Diffusion XL (SDXL) [Podell et al., 2022]

- After the introduction of Latent Diffusion Models, various organizations have open-sourced the scaled version of LDMs
 - Stable Diffusion (v1.5 & v2.1): LDM trained on LAION image-text pairs
- SDXL is the updated version of Stable Diffusion with better autoencoder and larger model size and scale
 - The model size is increased from 860M (SD 1.5) to 2.6B
 - The model is conditioned with the size of image and cropping parameters to generate more centered images



Table of Contents

1. Introduction

- Foundation models in vision tasks

2. Discriminative Visual Foundation Models

- Self-supervised Learning
- Image-text Contrastive Learning
- Multimodal LLM

3. **Generative visual foundation models**

- Text-to-Image Diffusion models
- Applications

4. Segment Anything

Due to the existence of large-scale pretrained T2I models, many following works focused on extending the capability beyond image generation

From now on, we explore recent topics in leveraging T2I models for

- Image editing (or image-to-image translation) using text
- Controllable generation
- Personalization
- Text-to-3D generation

Prompt-to-Prompt Image Editing with Cross-Attention Control [Hertz et al., 2023]

Motivation: **Image editing** is challenging in text-driven synthesis diffusion models

- Small modification in text prompt leads to **different outcome**
- Prior works require a **spatial mask** for localized image editing

Contribution: Textual editing method via **Prompt-to-Prompt manipulations**

- Text-only editing (w/o spatial mask) based on cross-attention maps



Cross-attention maps: High-dim tensors binding **pixels and tokens** from the prompt

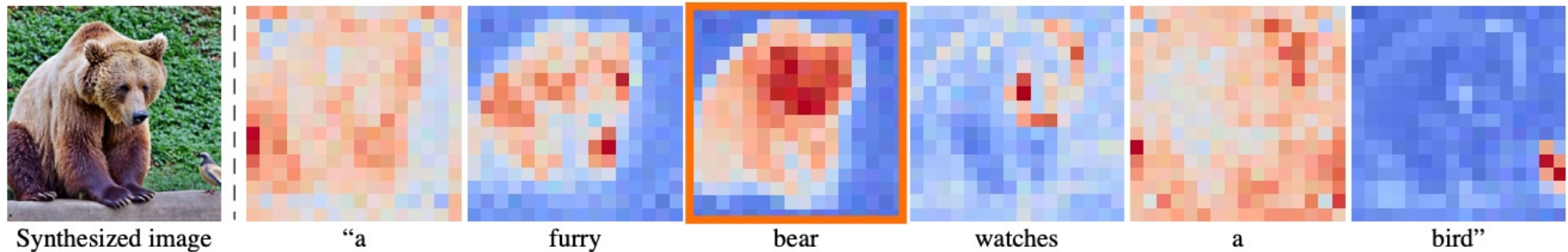
- Contain semantic relations which affects the generated images

Observation: **Spatial layout** and **geometry** depend on the cross-attention maps

- Pixels are more attracted to the words describing them (e.g., bear)

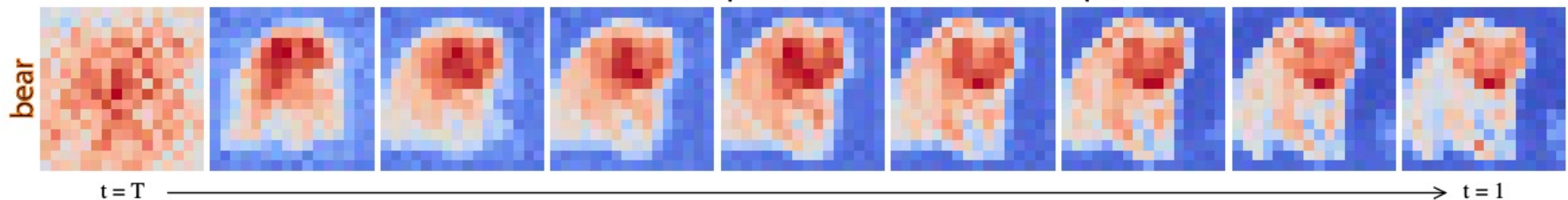
🤔 How to utilize **cross-attention maps** for image editing?

💡 **Inject the attention maps** of original prompt to the modified prompt



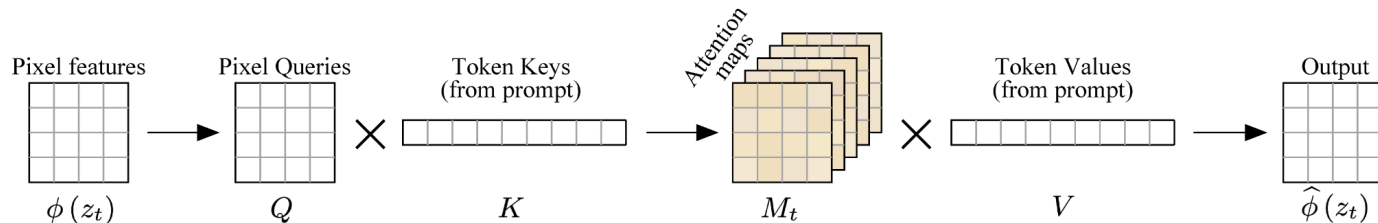
Average cross-attention maps across all timestamps

Cross-attention maps for individual timestamps



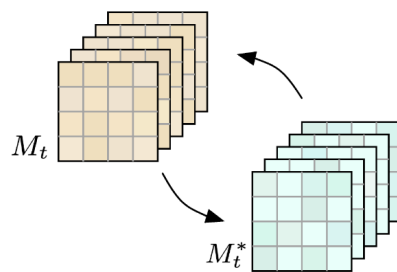
Main Idea: Injecting **cross-attention maps** during the diffusion process

- **Word swap:** attention injection of the source image
 - E.g., “a big **bicycle**” → “a big **car**”
- **Prompt refinement:** attention injection over the common tokens
 - E.g., “a castle” → “**children drawing of** a castle”
- **Attention Re-weight:** increase / decrease the attention weights of specified tokens
 - E.g., more or less “**fluffy**” on “a fluffy ball”

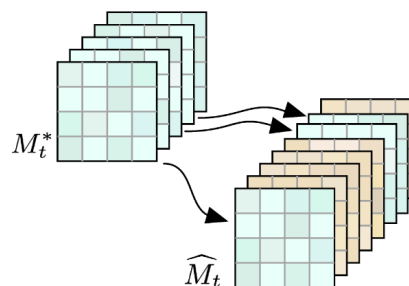


Text to Image Cross Attention

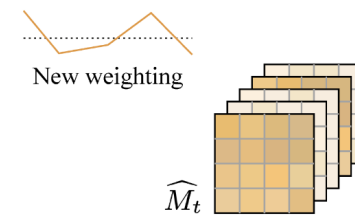
Cross Attention Control



Word Swap



Prompt Refinement



Attention Re-weighting

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

Word Swap



Prompt Refinement



Attention Re-weighting

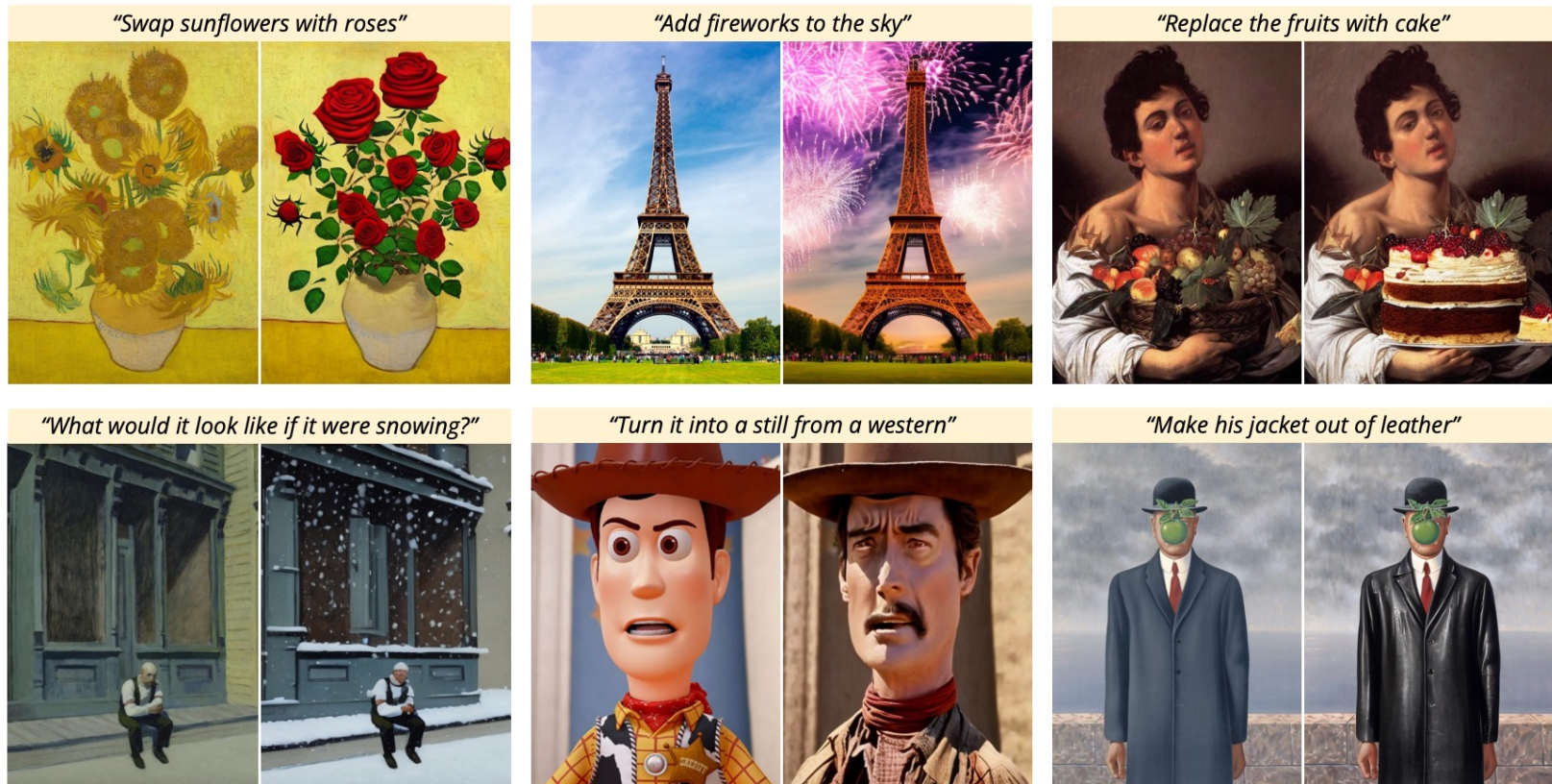


InstructPix2Pix: Learning to Follow Image Editing Instructions [Brooks et al., 2023]

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

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

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



Main Idea: Treat instruction-based image editing as a **supervised problem**

- **Dataset generation:** Text editing instructions and images before/after the edit
 - Two large-scale models on **different modalities**: GPT-3 and Stable Diffusion
 - **GPT-3:** Fine-tuned to produce the instructions and the edited caption
 - **Stable Diffusion:** Transform a pair of captions into a pair of images (w/ p2p)

Training Data Generation

(a) Generate text edits:

Input Caption: *"photograph of a girl riding a horse"* →

GPT-3

Instruction: *"have her ride a dragon"*

Edited Caption: *"photograph of a girl riding a dragon"*

(b) Generate paired images:

Input Caption: *"photograph of a girl riding a horse"* →

Edited Caption: *"photograph of a girl riding a dragon"* →

Stable Diffusion
+ Prompt2Prompt



(c) Generated training examples:

"convert to brick"



"Color the cars pink"



"Make it lit by fireworks"



"have her ride a dragon"



...

Main Idea: Treat instruction-based image editing as a **supervised problem**

- **Dataset generation:** Text editing instructions and images before/after the edit
 - Two large-scale models on **different modalities**: GPT-3 and Stable Diffusion
 - **GPT-3:** Fine-tuned to produce the instructions and the edited caption
 - **Stable Diffusion:** Transform a pair of captions into a pair of images (with PtP)
- **Training:** Train Stable diffusion on generated paired dataset

$$L = \mathbb{E}_{\mathcal{E}(x), \mathcal{E}(c_I), c_T, \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_{\theta}(z_t, t, \mathcal{E}(c_I), c_T)\|_2^2 \right]$$

■ : Input image conditioning

■ : Text instruction conditioning

- **Classifier-free guidance for two conditionings**

- Leverage classifier-free guidance w.r.t. **input image c_I** and **text instruction c_T**

$$\begin{aligned} \tilde{e}_{\theta}(z_t, c_I, c_T) &= e_{\theta}(z_t, \emptyset, \emptyset) \\ &\quad + s_I \cdot (e_{\theta}(z_t, c_I, \emptyset) - e_{\theta}(z_t, \emptyset, \emptyset)) \\ &\quad + s_T \cdot (e_{\theta}(z_t, c_I, c_T) - e_{\theta}(z_t, c_I, \emptyset)) \end{aligned}$$

InstructPix2Pix performs many challenging edits

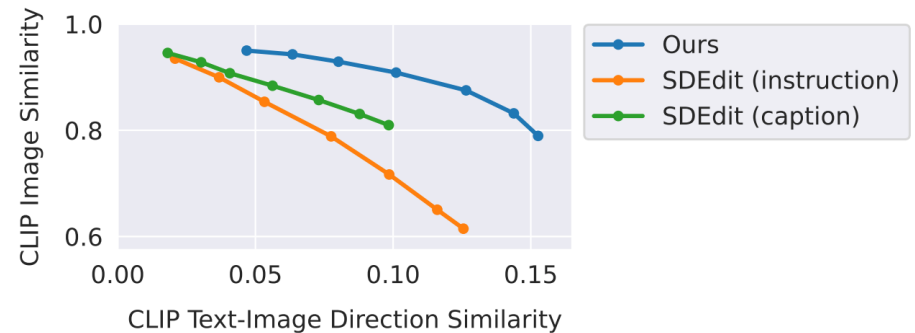
- E.g., replacing object, changing seasons, replacing backgrounds and etc.



Trade-off in consistency

- Consistency with the input images (y-axis)
- Consistency with the edit (x-axis)

→ Higher image consistency



Adding Conditional Control to Text-to-Image Diffusion Models [Zhang et al., 2023]

Motivation: Challenges in **additional control** on the text-to-image diffusion models

- Text prompt is not enough for matching **mental imagery**; need trial-and-error cycles
- Lack of data: Available data for a specific condition is small (e.g., human pose)

Contribution: **End-to-end** way that learns **conditional controls**

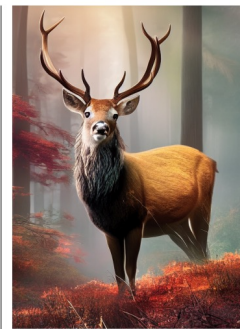
- while preserving the **quality** and **capabilities** of the large model



Input Canny edge



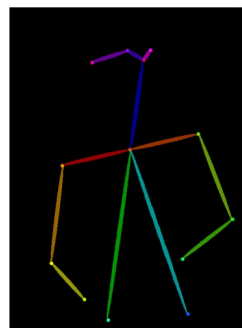
Default



“masterpiece of fairy tale, giant deer, golden antlers”



“..., quaint city Galic”



Input human pose



Default



“chef in kitchen”



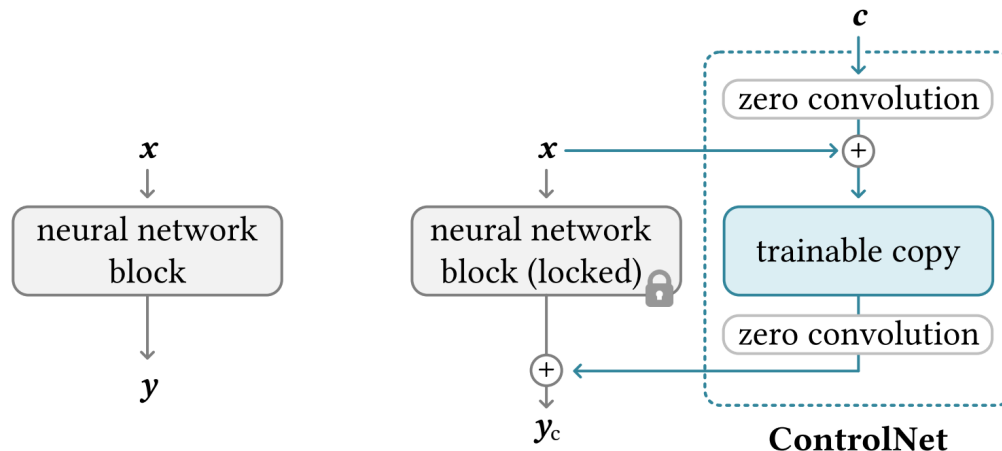
“Lincoln statue”

Main Idea: End-to-end neural network with **trainable copy** and **locked copy**

- **Trainable copy:** Cloning of the neural network block for task-specific dataset
- **Locked copy:** Preserve the capability of large-scale model

Effect of **zero convolution:**

- Reduce number of trainable parameters
- Elimination of harmful noise in training



Zero convolution

1×1 convolution layer
with zero weights and bias

$$y_c = \mathcal{F}(x; \Theta) + \mathcal{Z}(\mathcal{F}(x + \mathcal{Z}(c; \Theta_{z1}); \Theta_c); \Theta_{z2})$$

■ : locked copy

■ : trainable copy

Main Idea: End-to-end neural network with **trainable copy** and **locked copy**

- **Trainable copy:** Cloning of the neural network block for task-specific dataset
- **Locked copy:** Preserve the capability of large-scale model

Effect of **zero convolution**:

- Reduce number of trainable parameters
- Elimination of harmful noise in training

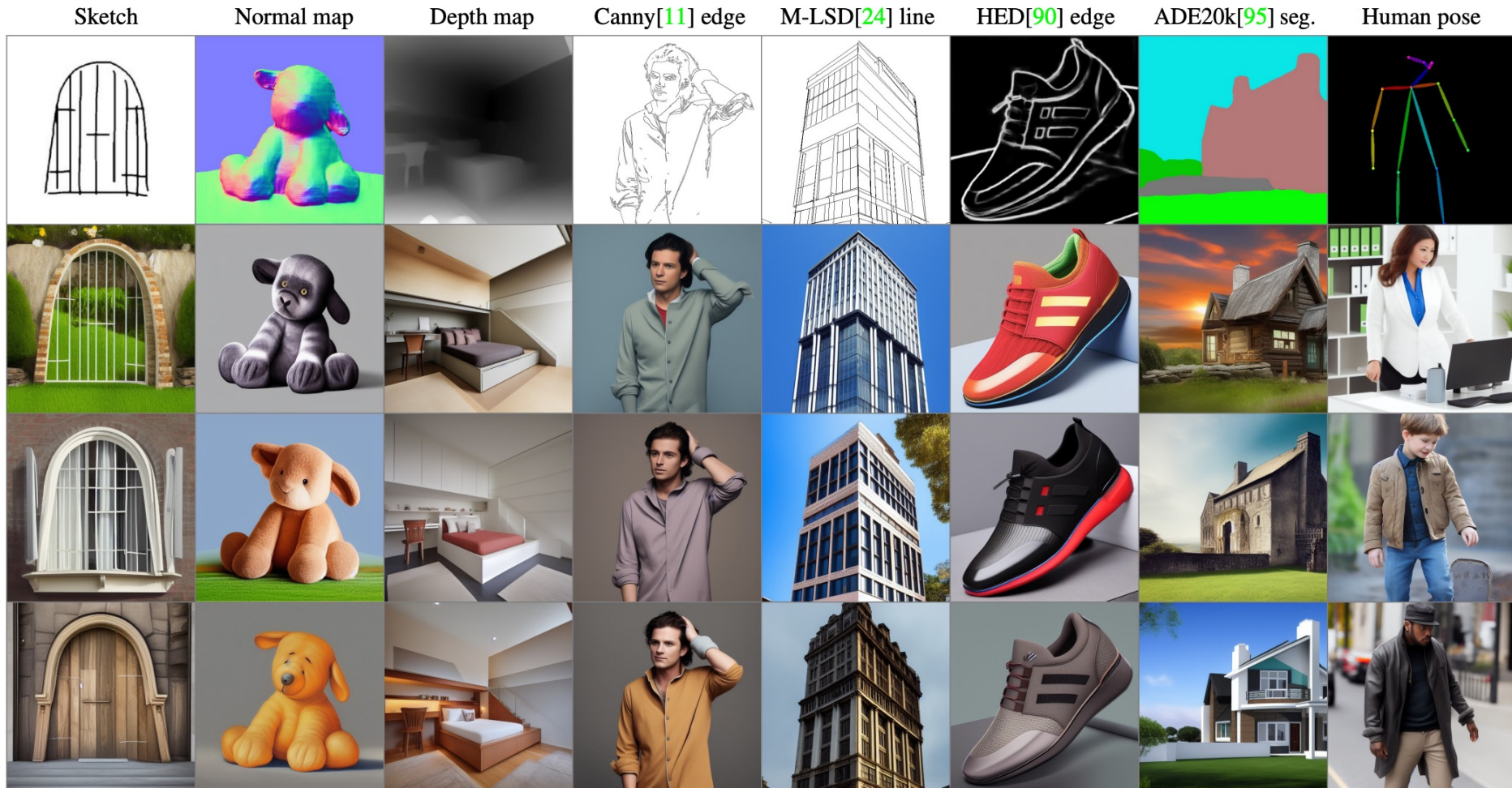
Training: **Fine-tune** the entire diffusion model with **ControlNet**

$$\mathcal{L} = \mathbb{E}_{\mathbf{z}_0, \mathbf{t}, \mathbf{c}_t, \mathbf{c}_f, \epsilon \sim \mathcal{N}(0,1)} \left[\|\epsilon - \epsilon_{\theta}(\mathbf{z}_t, \mathbf{t}, \mathbf{c}_t, \mathbf{c}_f)\|_2^2 \right]$$

 : text prompt

 : task-specific condition

ControlNet robustly interprets content semantics in diverse input conditioning



An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion [Gal et al., 2023]

Motivation: Difficulty in introducing **new concepts** into large scale models

- Re-training requires **huge amount of cost**
- Fine-tuning on few examples leads to **catastrophic forgetting**

Contribution: Personalized text-to-image generation (given 3-5 images)

- **Textual inversion:** find new pseudo-words capturing visual semantics and details



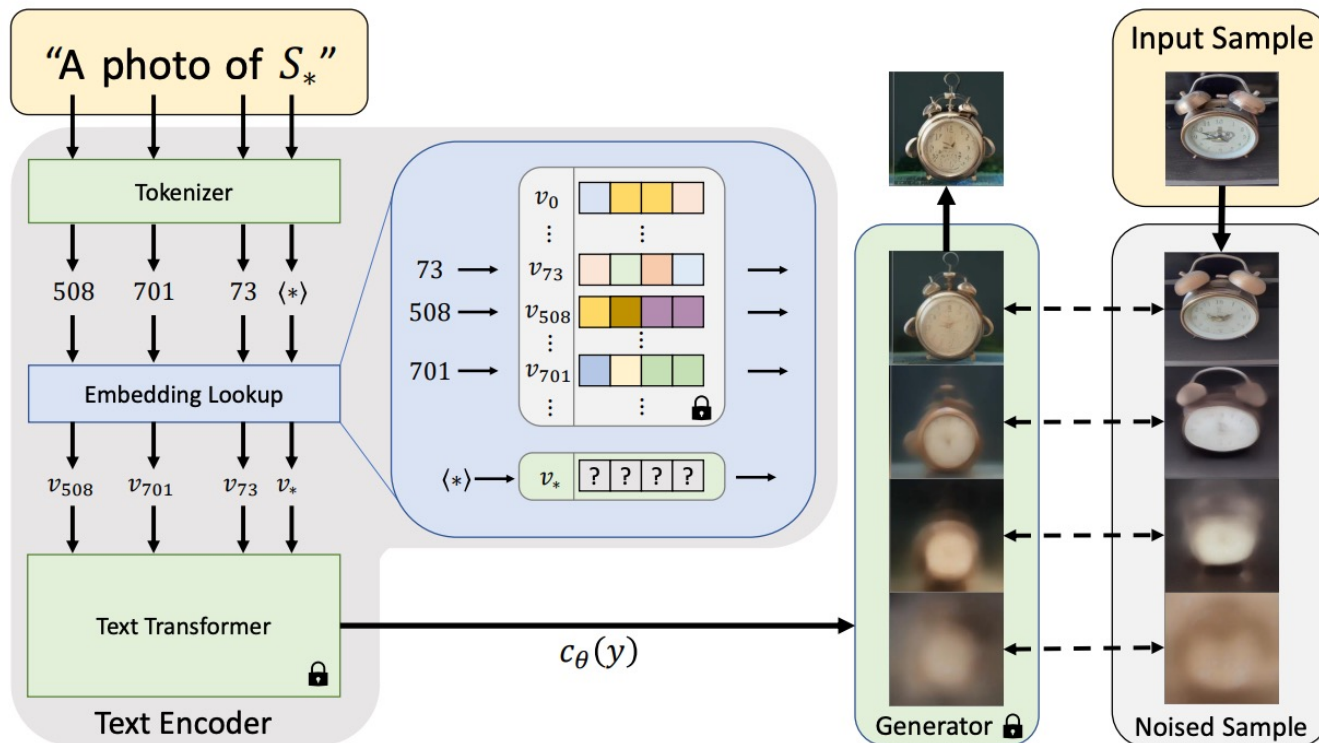
Main Idea: Find new **pseudo-word** in text embedding space (in LDMs)

- For pseudo-word S^* , directly optimize textual embedding v^* of S^*

$$v_* = \arg \min_v \mathbb{E}_{z \sim \mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_\theta(z_t, t, c_\theta(y))\|_2^2 \right]$$

■ : Learnable new token embedding

■ : Frozen LDM model



Textual Inversion enables **capturing** and **recreating** variations of an object

- Image synthesis guided by a caption lacks **fine-grained detail** (e.g., color patterns)
- Capture finer details and compose novel scenes w/ only a **single token embedding**



Input samples



“A mosaic depicting S_* ”



“Death metal album cover featuring S_* ”



“Masterful oil painting of S_* hanging on the wall”



“An artist drawing a S_* ”



Input samples



“A photo of S_* full of cashew nuts”



“A mouse using S_* as a boat”



“A photo of a S_* mask”



“Ramen soup served in S_* ”

DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation [Ruiz et al., 2023]

Motivation: Lack the ability to synthesize **same subjects** in different context

- Output domain is limited; detailed textual description yield different appearances

Contribution: Personalization of text-to-image diffusion models (given 3-5 images)

- **Fine-tuning method** to implant the given subject into the model's output domain



Input images



in the Acropolis



swimming



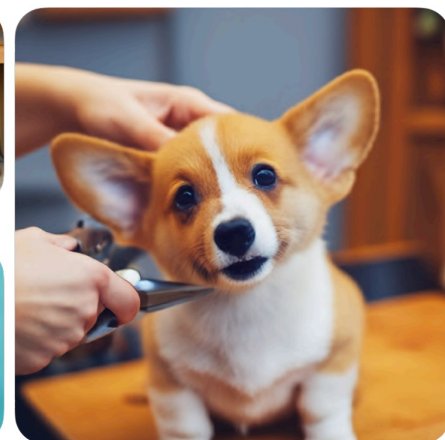
sleeping



in a doghouse



in a bucket



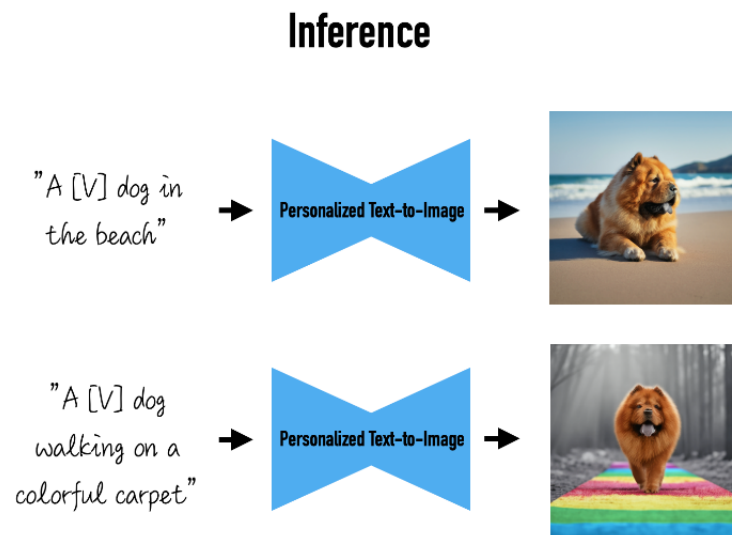
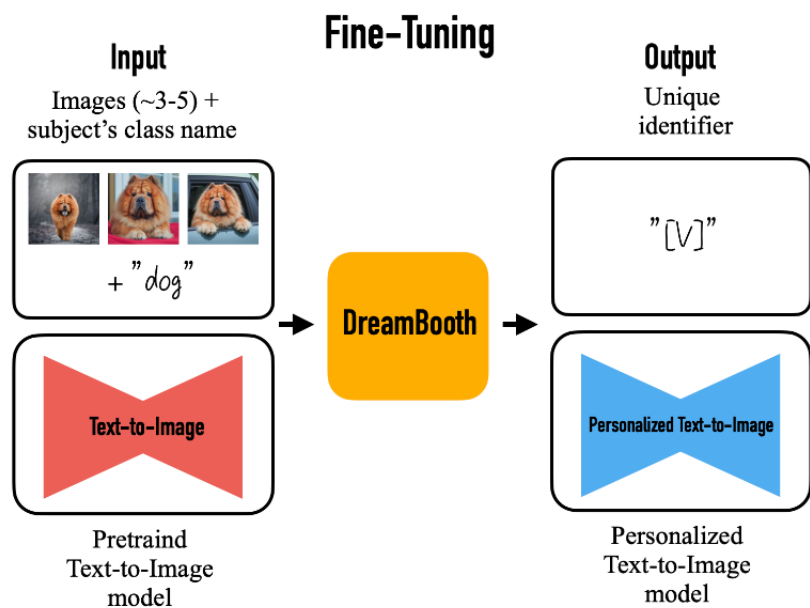
getting a haircut

Main Idea: Fine-tune text-to-image model w/ **few images** of a subject and **class name**

- Text prompt with **unique identifier** and the **class name** (e.g., a [V] dog)
 - **Unique identifier:** class-specific instance
 - **Class name:** prior knowledge on the subject class

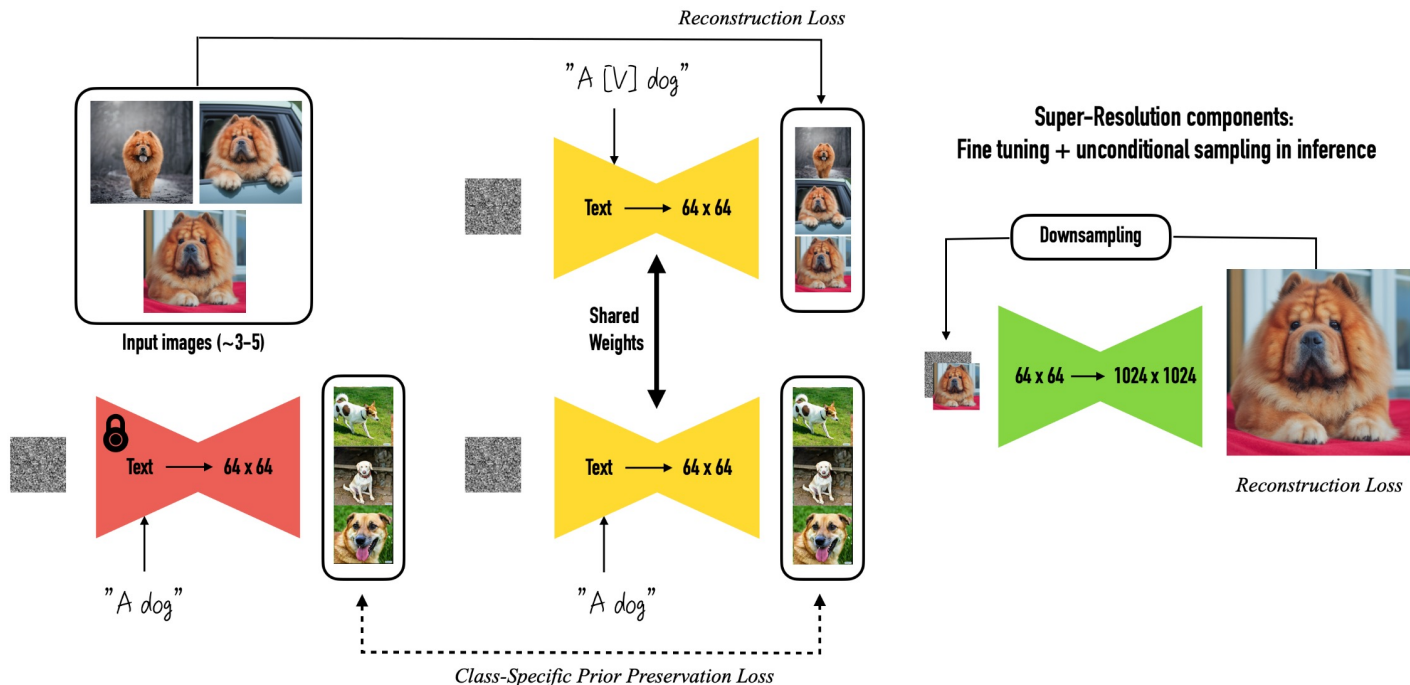
However, fine-tuning text-to-image model with small set may cause:

1. Language drift
2. Reduced output diversity



Main Idea: Fine-tune text-to-image model w/ few images of a subject and class name

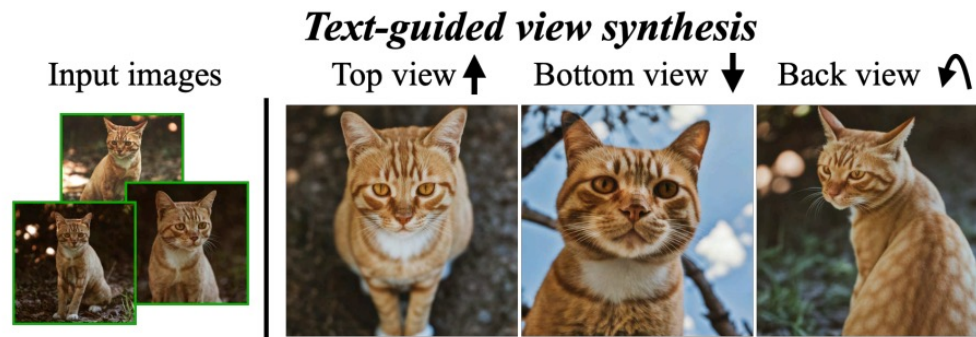
- Text prompt with **unique identifier** and the **class name** (e.g., a [V] dog)
 - **Unique identifier:** class-specific instance
 - **Class name:** prior knowledge on the subject class
- **Class-specific prior preservation loss**
 - Supervise the model w/ own generated samples
 - Leverages the semantic prior that the model has on the class



- Generates image with high preservation of **subject details** in **various context**



- Generate **novel views** with preserving subject identity



DreamBooth fine-tuning with LoRA

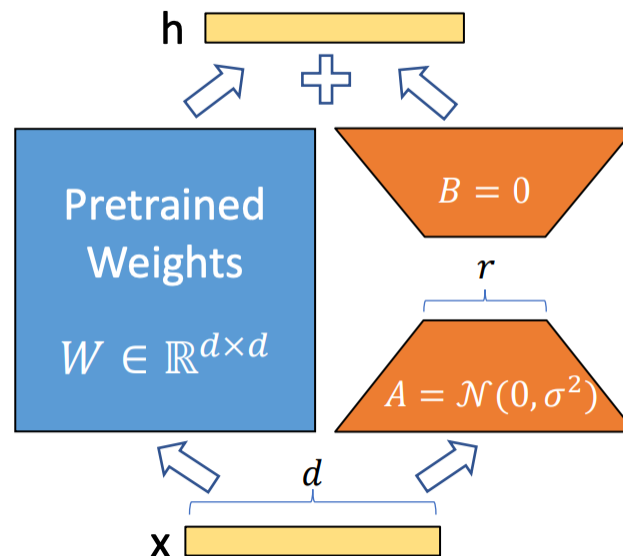
🤔 How to efficiently **fine-tune** large models (e.g., DreamBooth)?

💡 Reduce the **number of trainable parameters**, not fine-tuning all parameters

LoRA: Low-Rank Adaptation of Large Language Models [Hu et al., 2022]

- Freeze the original weights and update only **low-rank decomposed matrices**

$$h = W_0x + \Delta Wx = W_0x + BAx$$



→ LoRA enables **faster** and **memory efficient** DreamBooth fine-tuning

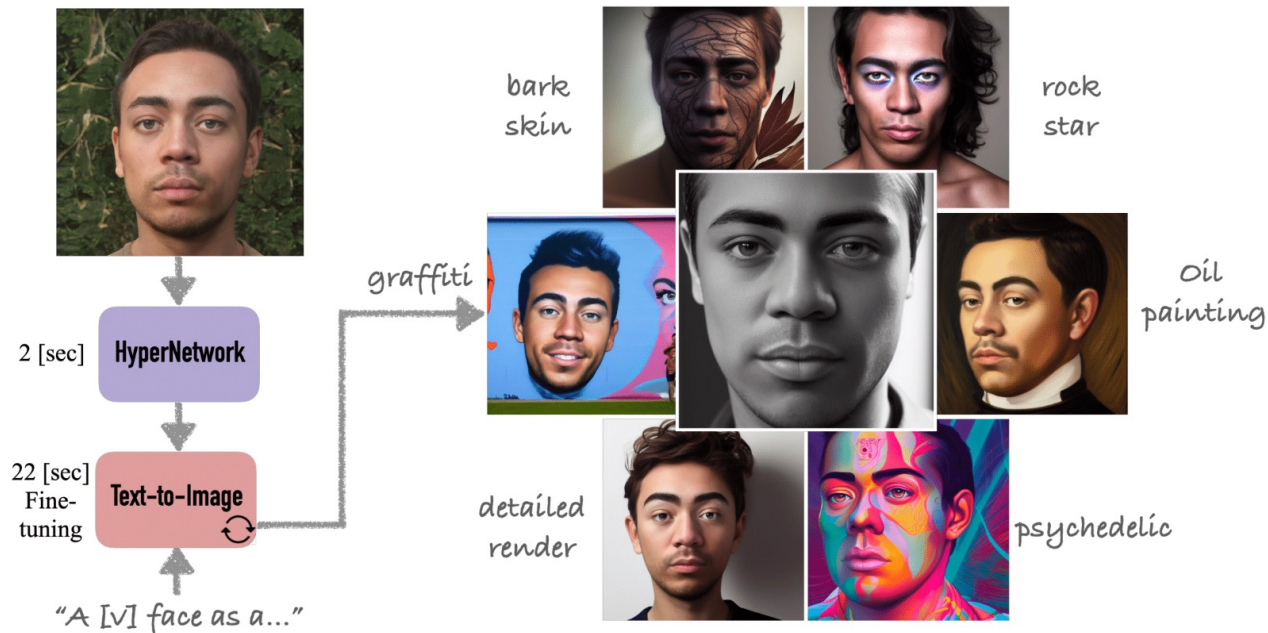
HyperDreamBooth: HyperNetworks for Fast Personalization of Text-to-Image Models [Ruiz et al., 2023]

Motivation: Personalization requires huge amount of **time** and **memory**

- GPU time for fine-tuning the entire model
- Storage for each personalized model

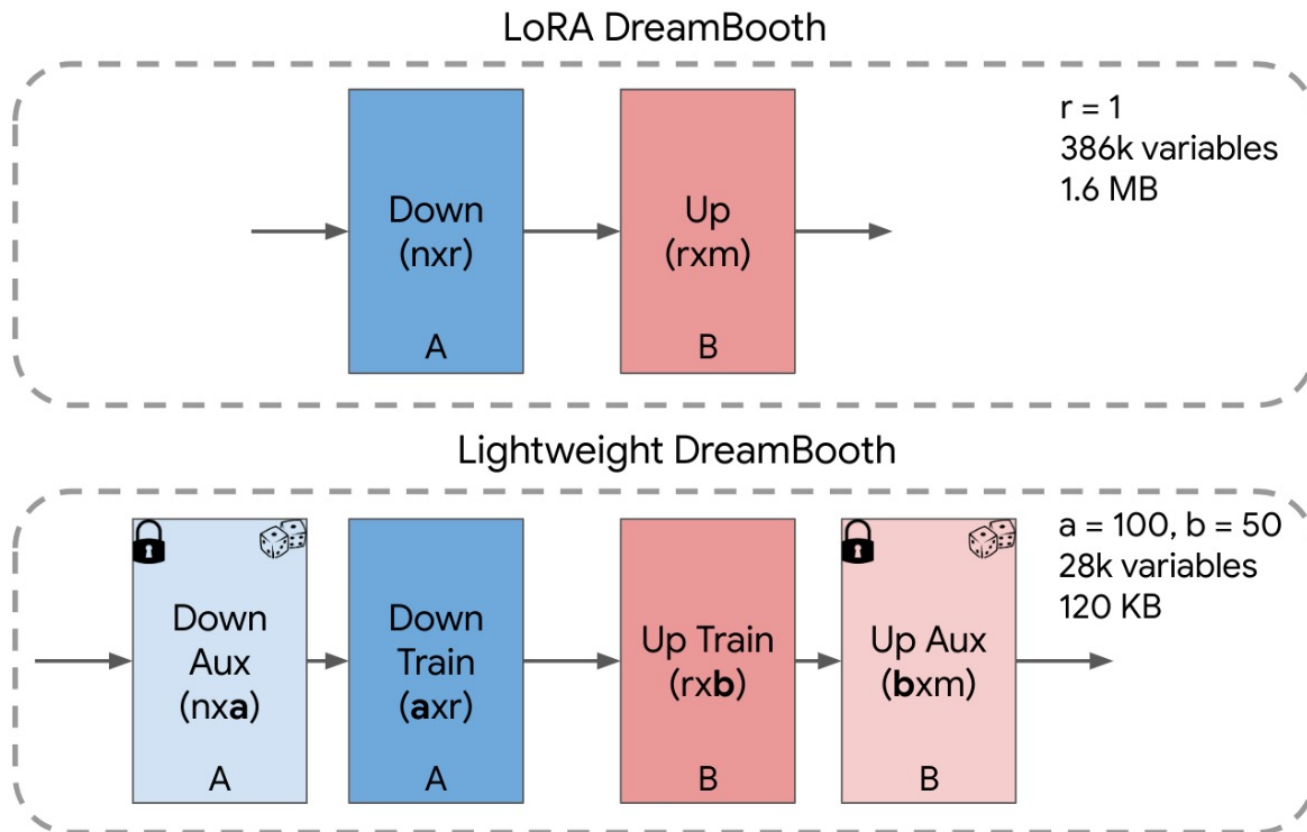
Contribution: Tackles the problem of **speed** and **size** of DreamBooth

- while preserving *model integrity, editability* and *subject fidelity*



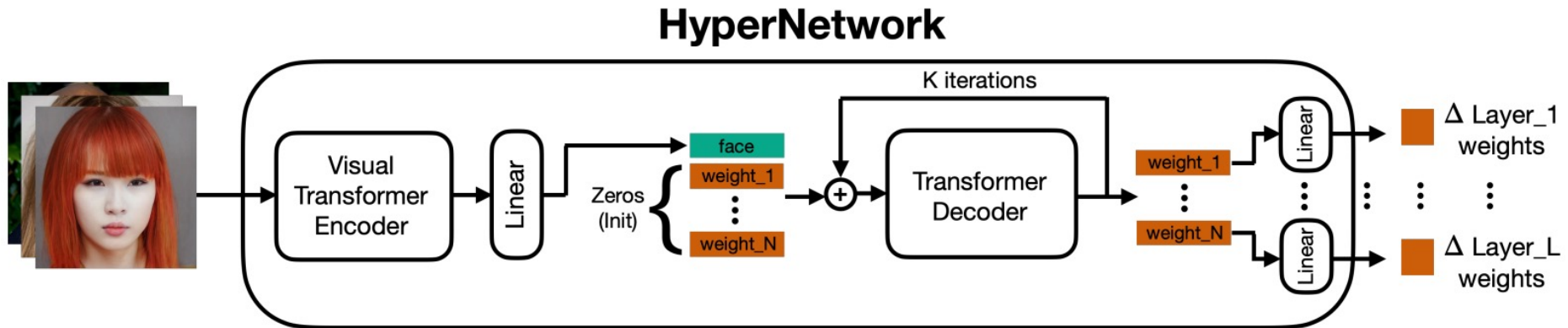
Main Idea: Propose **HyperNetwork** for fast personalization

- **Lightweight DreamBooth (LiDB):** Training a model in a low-dim weight-space
 - Further decompose A and B matrices of LoRA into two matrices



Main Idea: Propose **HyperNetwork** for fast personalization

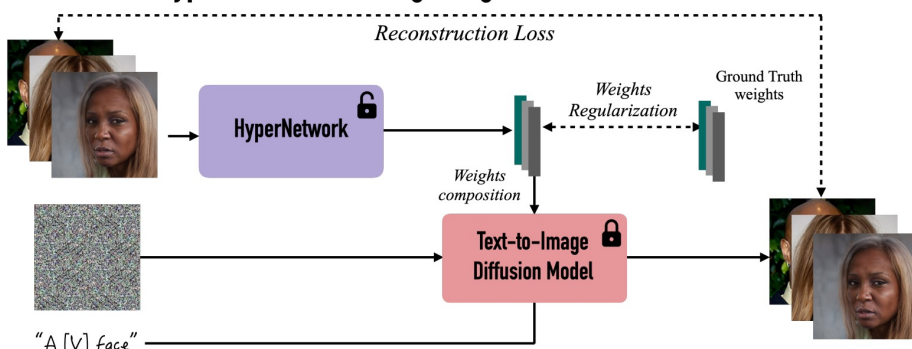
- **Lightweight DreamBooth (LiDB):** Training a model in a low-dim weight-space
- **HyperNetwork:** Generate an initial prediction of LiDB weight
 - **ViT Encoder:** Translates face images into latent face features
 - **Transformer Decoder:** Iteratively predicts the values of weight features



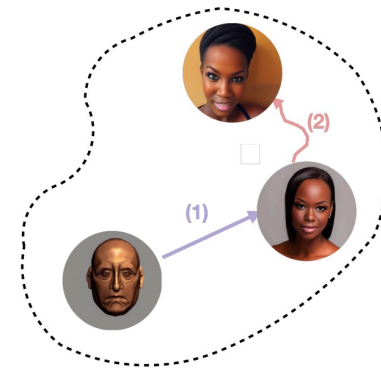
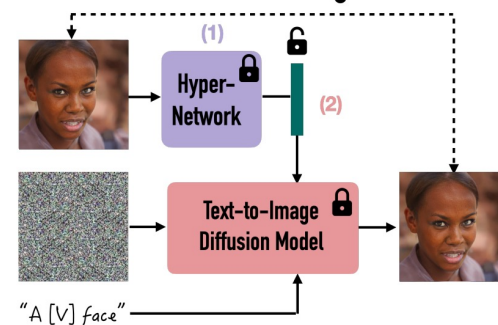
Main Idea: Propose **HyperNetwork** for fast personalization

- **Lightweight DreamBooth (LiDB):** Training a model in a low-dim weight-space
- **HyperNetwork:** Generate an initial prediction of LiDB weight
- **Rank-relaxed fine-tuning**
 - **HyperNetwork training:** Training a hypernetwork to predict network weights
 - **Fast fine-tuning:** Fine-tuned using reconstruction loss with given image

Phase 1 - HyperNetwork Training (Large Scale)



Phase 2 - Fast Fine-Tuning

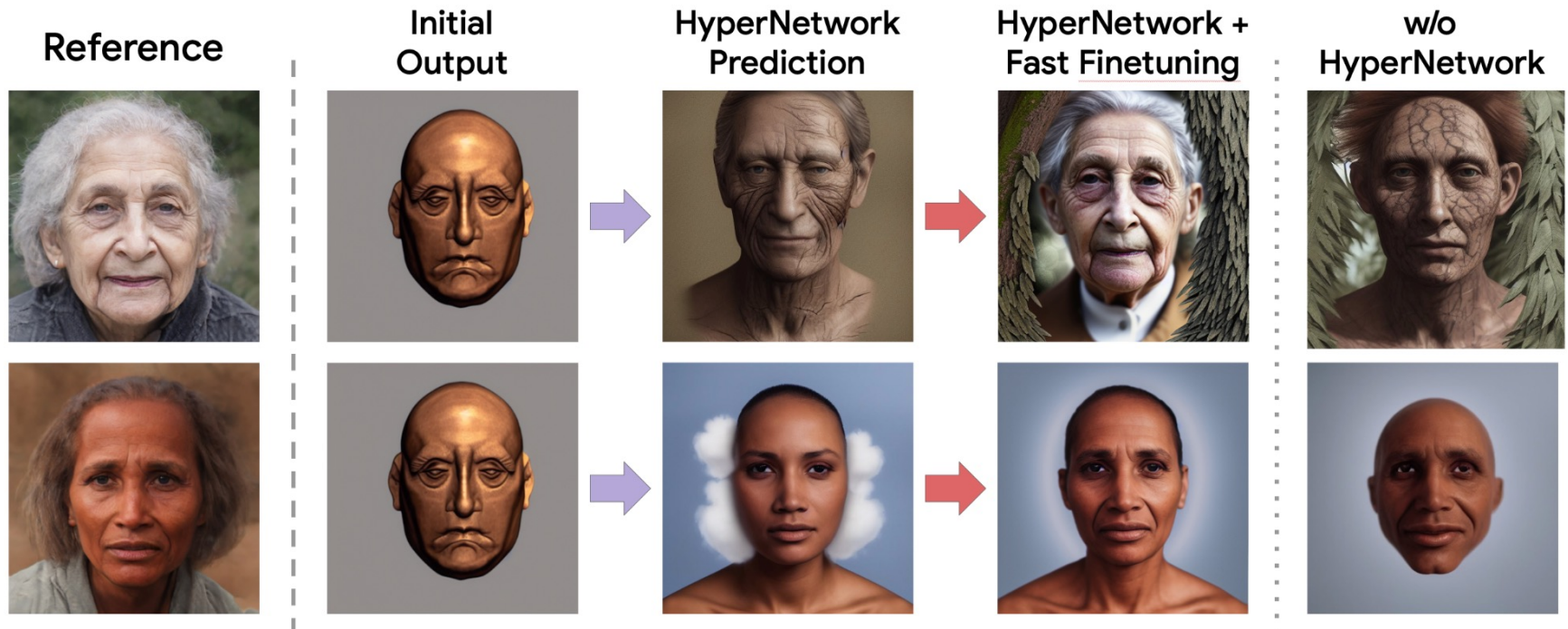


HyperNetwork + Fast Finetuning achieves strong results

- Preserves **identity** and **subject fidelity** more closely compared to prior works\

Method	Face Rec. \uparrow	DINO \uparrow	CLIP-I \uparrow	CLIP-T \uparrow
Ours	0.655	0.473	0.577	0.286
DreamBooth	0.618	0.441	0.546	0.282
Textual Inversion	0.623	0.289	0.472	0.277

- Strong personalization results** for diverse faces



- Recent, Text-to-image (T2I) diffusion models have shown impressive capabilities
 - Synthesizing high-quality, realistic, diverse images with the text given as input
- How can we **utilize T2I diffusion models to 3D synthesis** without 3D training data?
- How can we **use DMs as a critic to optimize** the underlying 3D representation?
- Poole et al. (2023): **Score Distillation Sampling (SDS)**
 - Probabilistic density distillation enabling the use of a 2D diffusion models for priors
- **DreamFusion**: Optimize NeRF using T2I diffusion models with SDS
 - Optimize NeRF $g(\theta)$, that look like images x when rendered from random angles
 - The optimized NeRF yields good images appropriate for given text prompt
 - Does not require 3D training data and no modification to the image diffusion models

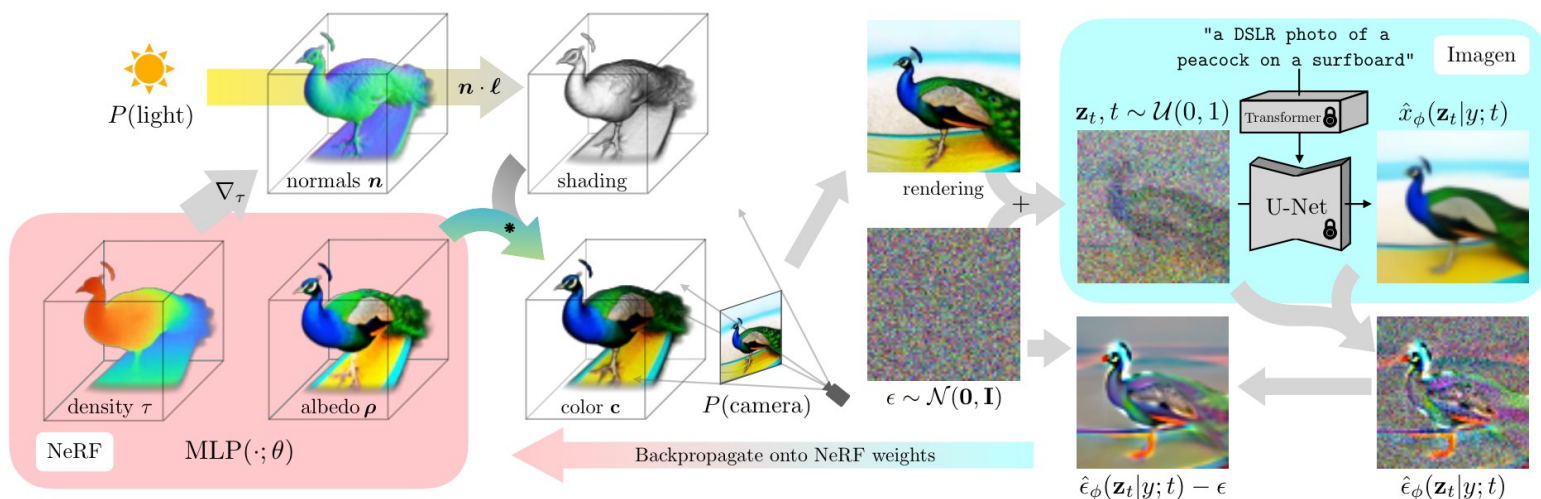
• How does DreamFusion create 3D assets from text descriptions?

1. Initialization:
NeRF is randomly initialized and trained from scratch for each caption
2. NeRF parameter updates:
DreamFusion diffuses the rendering and reconstructs it with a (frozen) Imagen

$$\hat{\epsilon}_{\phi}(\mathbf{z}_t | y; t) - \epsilon$$

prediction of injected noise
injected noise

- Subtracting the injected noise produces a low variance update direction
- Backpropagated through the rendering process to update the NeRF MLP parameters



- **Score distillation sampling enables sampling in parameter space, not pixel space**

- create 3D models that look like good images when rendered from random angles

1. Training objective of diffusion models is as follows:

$$\mathcal{L}_{\text{Diff}}(\phi, \mathbf{x}) = \mathbb{E}_{t \sim \mathcal{U}(0,1), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [w(t) \|\epsilon_\phi(\alpha_t \mathbf{x} + \sigma_t \epsilon; t) - \epsilon\|_2^2] :$$

2. Minimize the diffusion model training loss w.r.t a generated data point $\mathbf{x} = g(\theta)$

$$\theta^* = \operatorname{argmin}_\theta \mathcal{L}_{\text{Diff}}(\phi, \mathbf{x} = g(\theta))$$

3. Gradient of the training objective becomes:

$$\nabla_\theta \mathcal{L}_{\text{Diff}}(\phi, \mathbf{x} = g(\theta)) = \mathbb{E}_{t, \epsilon} \left[w(t) \underbrace{(\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon)}_{\text{Noise Residual}} \underbrace{\frac{\partial \hat{\epsilon}_\phi(\mathbf{z}_t; y, t)}{\partial \mathbf{z}_t}}_{\text{U-Net Jacobian}} \underbrace{\frac{\partial \mathbf{x}}{\partial \theta}}_{\text{Generator Jacobian}} \right]$$

4. Score Distillation Sampling

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, \mathbf{x} = g(\theta)) \triangleq \mathbb{E}_{t, \epsilon} \left[w(t) (\hat{\epsilon}_\phi(\mathbf{z}_t; y, t) - \epsilon) \frac{\partial \mathbf{x}}{\partial \theta} \right]$$

DreamFusion: Text-to-3D using 2D diffusion [Poole et al., 2023]

- DreamFusion generates coherent 3D scenes from a variety of text prompts

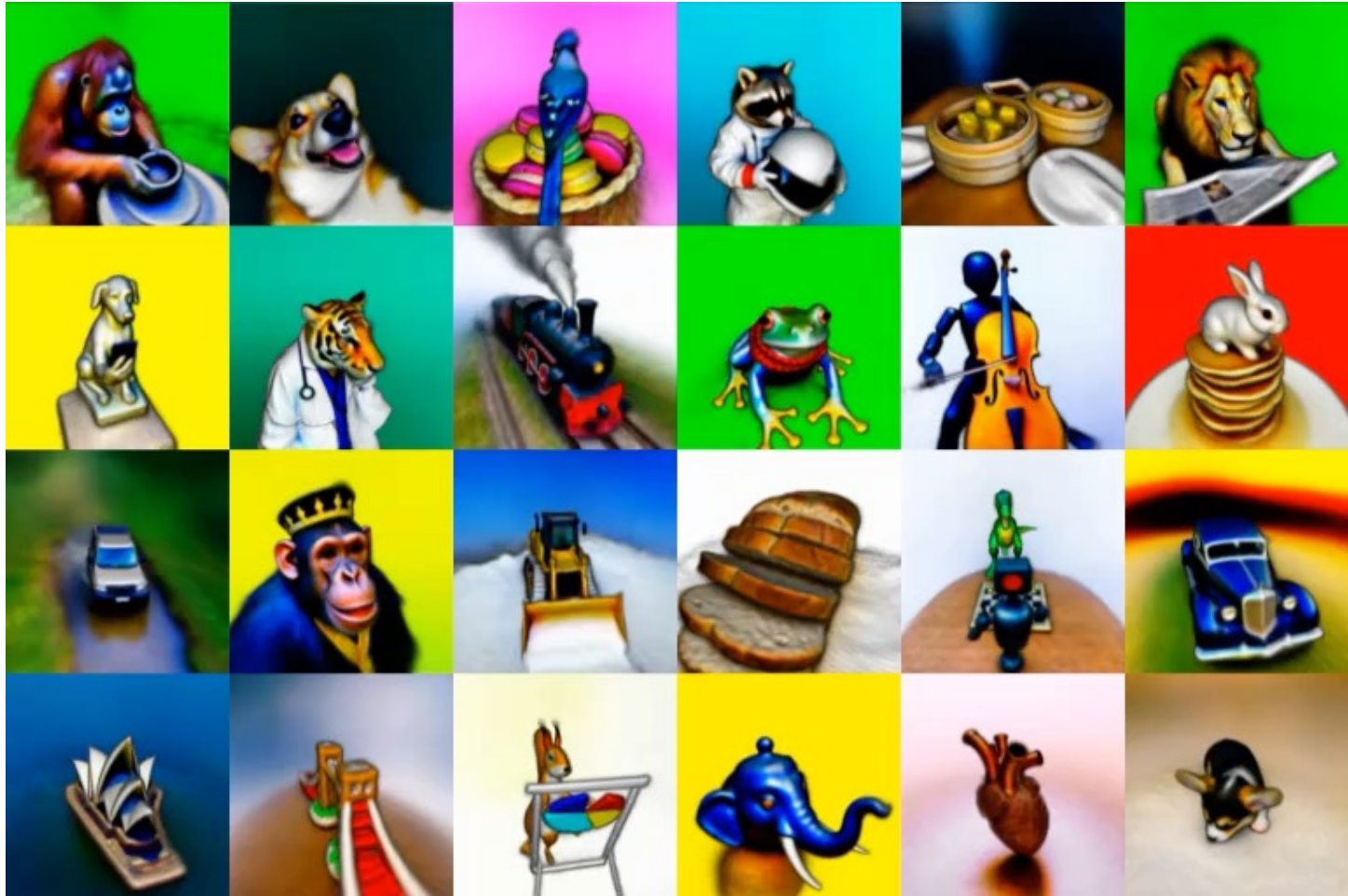


Table of Contents

1. Introduction

- Foundation models in vision tasks

2. Discriminative Visual Foundation Models

- Self-supervised Learning
- Image-text Contrastive Learning
- Multimodal LLM

3. Generative visual foundation models

- Text-to-Image Diffusion models
- Applications

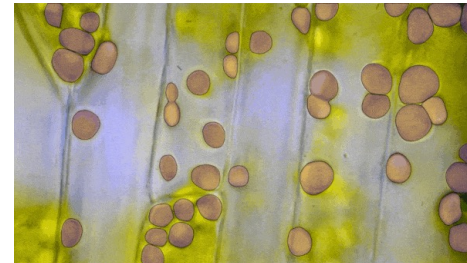
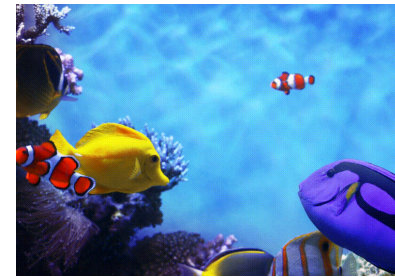
4. Segment Anything

Segment Anything Model (SAM) [Kirillov et al., 2023]

- A foundation model for image segmentation, *i.e.*, predicting object masks
- SA-1B dataset
 - Web-scale 11M photography and 1.1B segmentation masks¹
- Enables strong **zero-shot transfer** on new domains
 - e.g., segmenting underwater scenes, or microscopy



SA-1B examples



Zero-shot transfer with SAM

Segment Anything Model (SAM) [Kirillov et al., 2023]

- Promptable Segmentation via **points** and **boxes**
 - User can steer the image segmentation, like prompting MLs
- For example, user can prompt regions to be **included** & **excluded** by the model
 - Segmenting the whole image can be done by prompting a grid of points

include

exclude



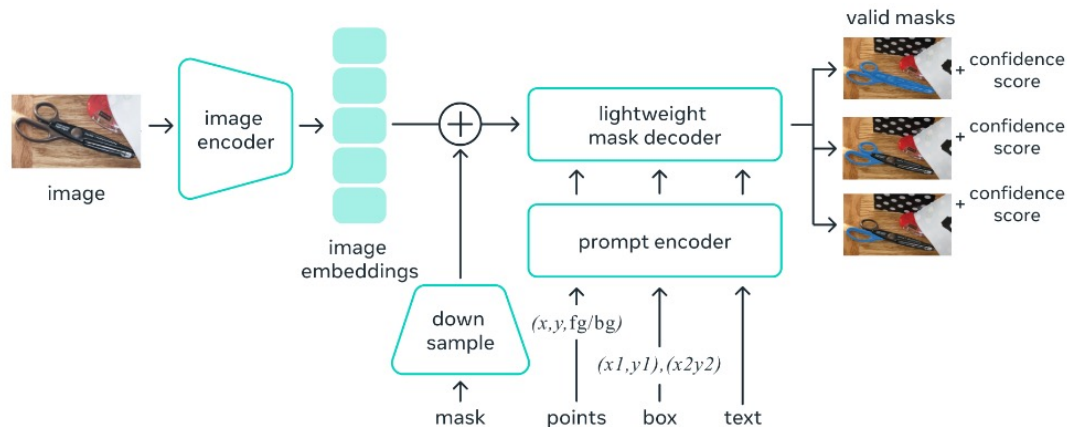
Prompt-based
Image Segmentation by SAM



Segmenting the whole image
by prompting a grid of points

Component of SAM model

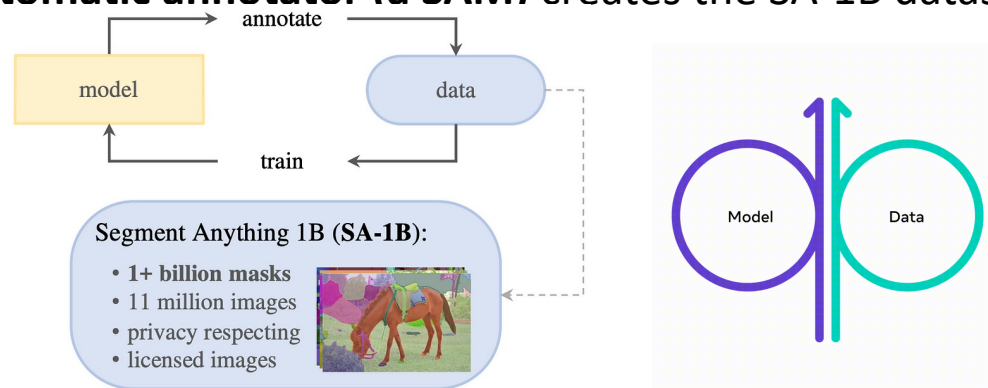
- Image Encoder
 - A ViT model producing a one-time embedding for segmentation
 - The embedding can be shared for different prompts
- Prompt Encoder
 - Encodes point, box, or text¹ prompts into transformer tokens
- Mask Decoder
 - Prompt token and image embedding goes through a transformer decoder
 - Decoder predicts multiple candidates for segmentation mask and the confidence



1. Text encoding function is not published.

SA-1B dataset

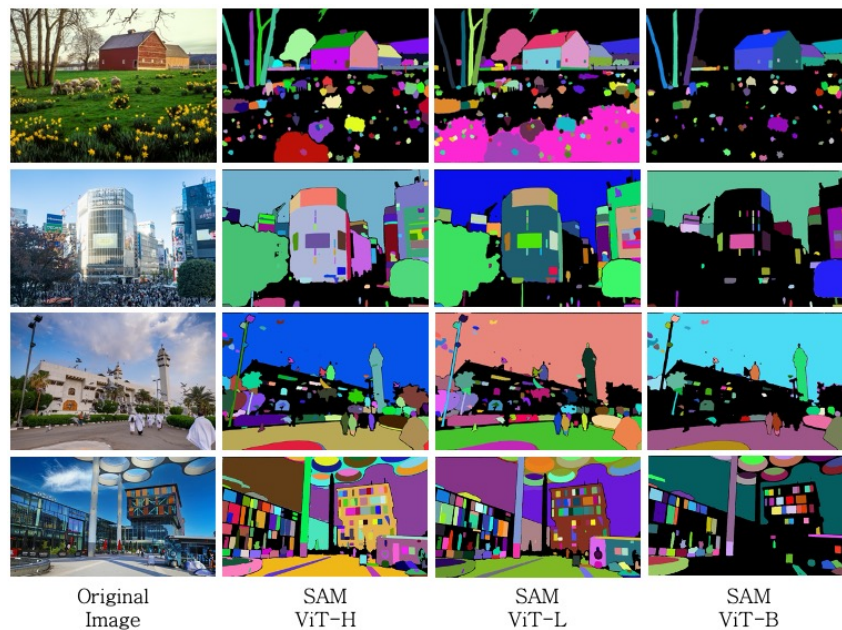
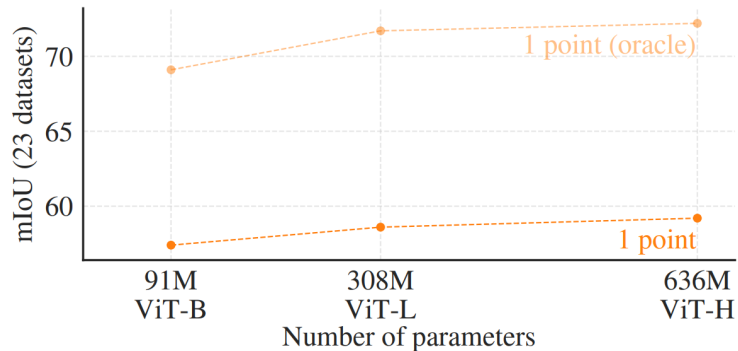
- Web-scale 11M photography and **1.1B segmentation masks**
 - Challenge: manually annotating the images is **too expensive**
- **Model-in-the-loop design**
 1. The data annotators use and fix SAM's outputs to annotate images (semi-auto)
 2. Newly available annotations are then used to re-train SAM
 3. The process is repeated and SAM's performance is bootstrapped
- Finally, the **automatic annotator (a SAM)** creates the SA-1B dataset



Model-in-the-loop process is repeated +10 times to get the final automatic annotation

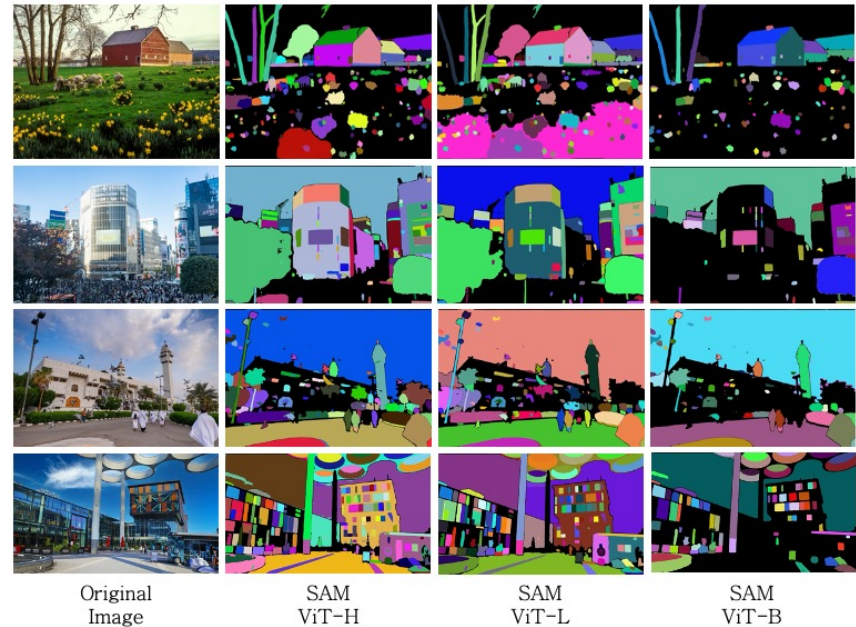
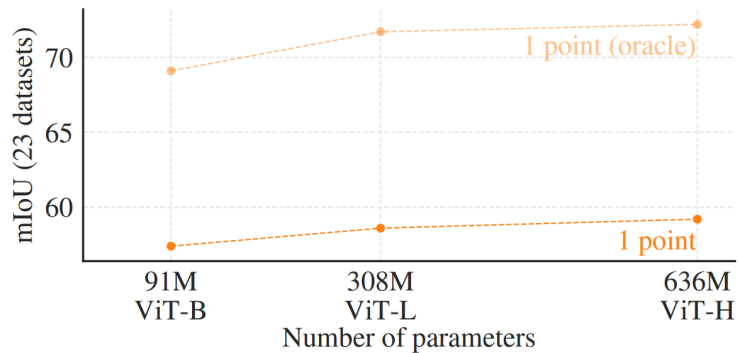
SAM model variants

- Default variants by the original research paper
 - Considers different image encoders: ViT-B, ViT-L, ViT-H
 - A **direct trade-off** on performance vs. computation cost



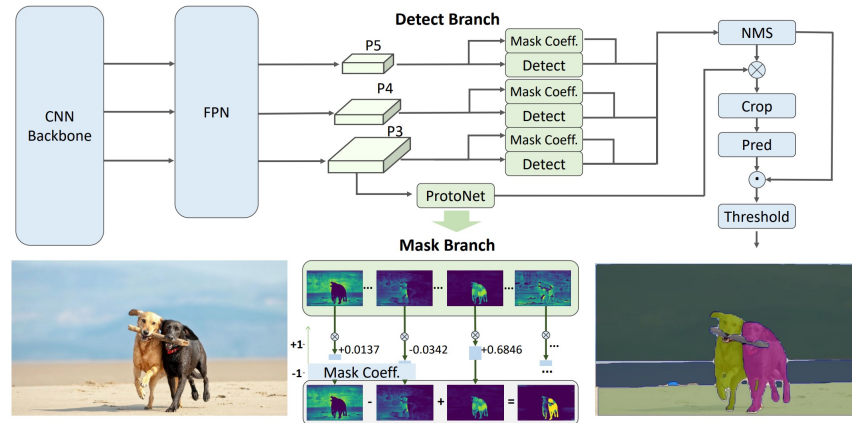
SAM model variants

- Default variants by the original research paper
 - Considers different image encoders: ViT-B, ViT-L, ViT-H
 - A **trivial trade-off** on segmentation accuracy vs. computation cost
- More effective way for the efficiency?



FastSAM [Zhao et al., 2023]

- Trains SA-1B on a CNN-based architecture for image segmentation (YOLO v7)
- Predicts all possible masks at once, without conditioning on prompts
 - (+) Better parallelization on the GPUs (Running time is independent to the number of points)
 - (-) Does not learn to utilize user prompts, e.g., points, boxes



YOLO architecture predicts all image segmentations at once

method	params	Running Speed under Different Point Prompt Numbers (ms)					
		1	10	100	E(16×16)	E(32×32*)	E(64×64)
SAM-H [20]	0.6G	446	464	627	852	2099	6972
SAM-B [20]	136M	110	125	230	432	1383	5417
FastSAM (Ours)	68M	40					

FastSAM shows constant running time; independent of the number of masks

Segment Anything Models (SAM)

MobileSAM [Zhang et al., 2023]

- Downsizing the image encoder through **Knowledge Distillation** [Hinton et al., 2015]
- Parameters: 611M (ViT-H) → 5M (tiny transformer)
- Image embedding space tends to be similar after knowledge distillation
 - Can perform well close to the original SAM
 - Realtime inference 452ms (Original SAM) → 8ms (MobileSAM)

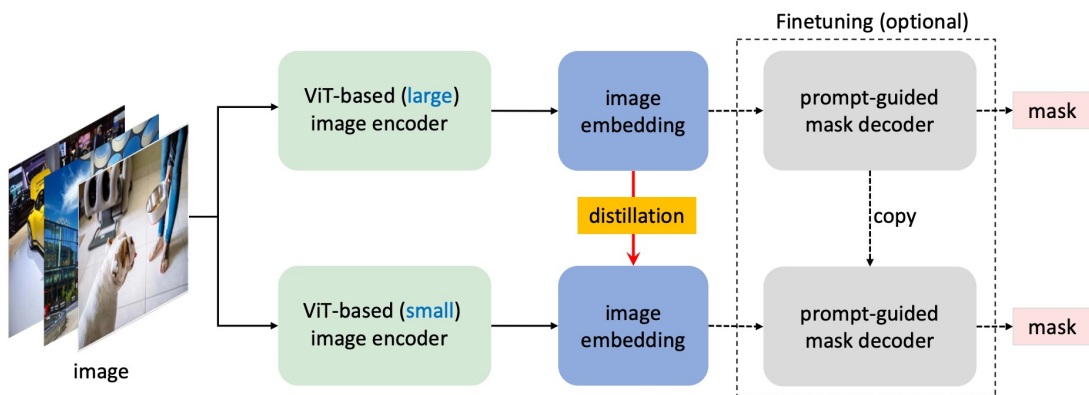


Image encoder is distilled, with a frozen mask decoder



SAM-HQ [Ke et al., 2023]

- Identifies the weakness of SAM and SA-1B dataset
 - Failures on objects with intricate structures (e.g., grate patterns)

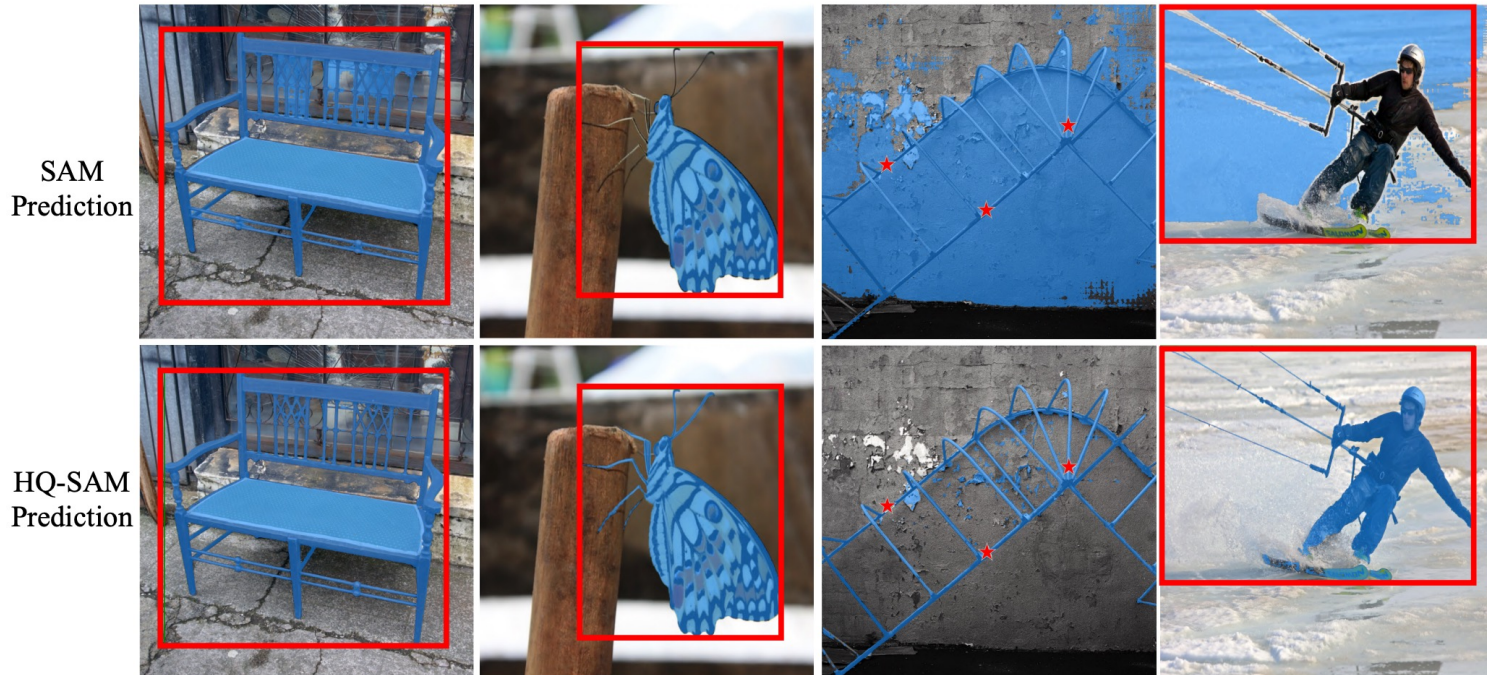


SAM has weakness on intricate structures, which gets fixed by HQ-SAM [Ke et al., 2023]

Segment Anything Models (SAM)

SAM-HQ [Ke et al., 2023]

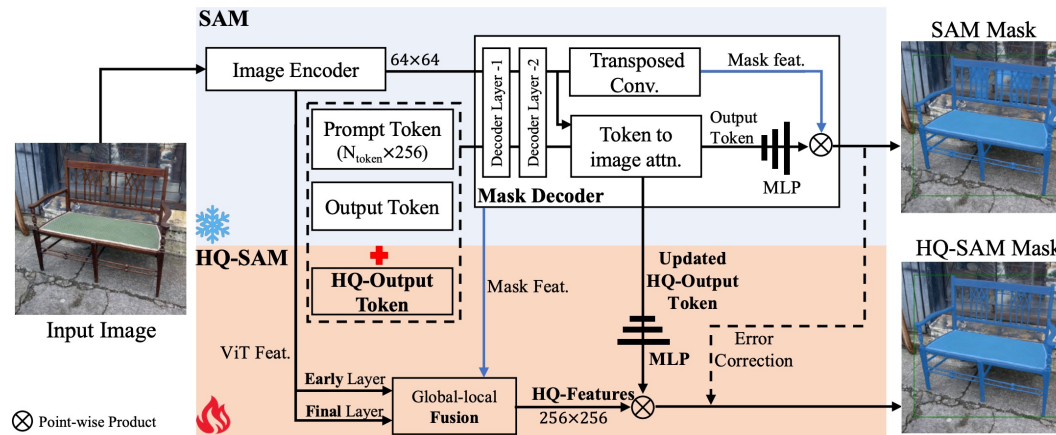
- SAM-HQ introduces fine-tuning to mitigate the failure cases (**HQSeg-44K dataset**)
 - Custom collection of 44K images, with extremely intricate segmentation annotations



SAM vs. HQ-SAM on HQSeq-44k samples

SAM-HQ [Ke et al., 2023]

- The pretrained SAM parameters remain frozen
 - Prevents model overfitting or catastrophic forgetting by a small HQSeg-44K dataset
- SAM-HQ only introduces a tunable prompt token and MLPs for fusion
 - Requires training only 5.1M additional parameters (0.5% of the SAM's parameters)



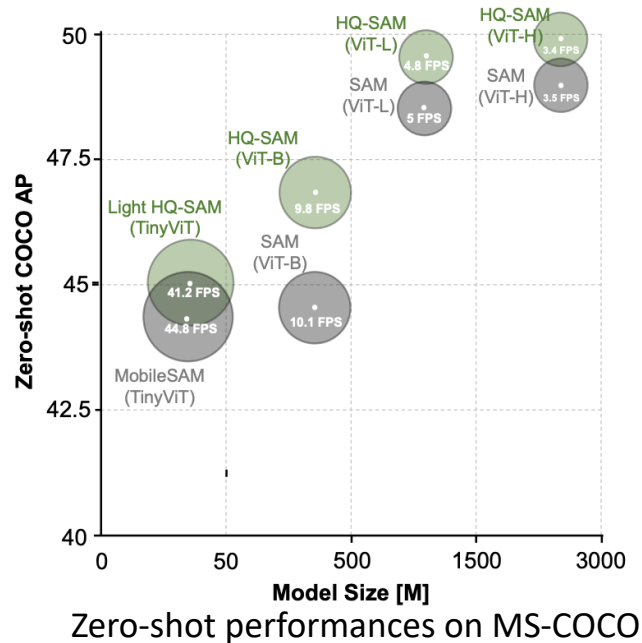
HQ-SAM architecture

Method	Learnable Params (M)	Training			Inference	
		# GPU	Batch Size	Time (h)	FPS	Mem.
SAM [21]	1191	128	128	N/A	5.0	7.6G
HQ-SAM	5.1	8	32	4	4.8	7.6G

Training cost of HQ-SAM

SAM-HQ [Ke et al., 2023]

- The pretrained SAM parameters remain frozen
 - Prevents model overfitting or catastrophic forgetting by a small HQSeg-44K dataset
- SAM-HQ only introduces a tunable prompt token and MLPs for fusion
 - Brings simple and effective performance boosts on all existing SAM variants
 - Including ViT-H, ViT-L, ViT-B and MobileSAM [Zhang et al., 2023]

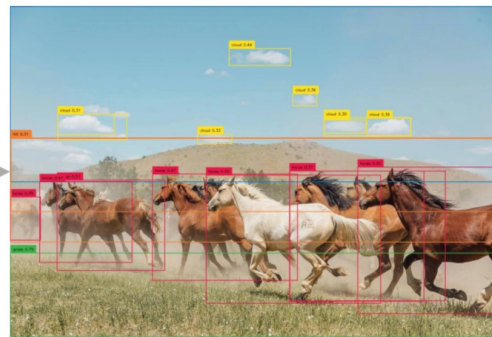


Notable Applications of SAM

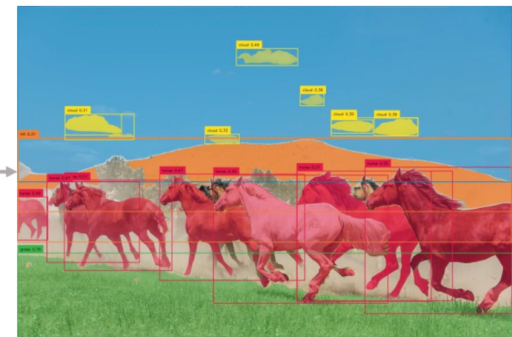
- Open-Vocabulary Semantic Segmentation (*e.g.*, Grounded SAM [Liu et al., 2023])
Basic Idea: **prompting SAM** with boxes, via **text-prompted box predictors**
 - Recent **vision-language models** can make **zero-shot box predictions** at ease
e.g., GroundingDINO [Liu et al., 2023], ViLD [Gu et al., 2022]
 - However, **zero-shot semantic segmentation** has remained **challenging**
- SAM directly **escalates** the semantic box predictions → segmentation masks
 - A break-through in the zero-shot, open vocabulary, semantic segmentation task



Text Prompt:
“Horse. Clouds. Grasses. Sky. Hill.”



Grounding DINO:
Detect Everything



Grounded-SAM:
Detect and Segment Everything

Segment Anything Models (SAM)

Rank	Participant	mean SGinW (↑)	mean PQ (↑)	mean AP (↑)	mean IoU (↑)	Elephants (↑)	Hand- Metal (↑)	Watermelon (↑)	House- Parts (↑)	HouseHold- Items (↑)	Strawberry (↑)	Fruits (↑)	Nutt Squi (↑)
------	-------------	----------------------	-------------------	-------------------	--------------------	------------------	-----------------------	-------------------	------------------------	-------------------------	-------------------	---------------	---------------------

SAM-based models

1	Grounded HQ-SAM (Grounded HQ-SAM- (B+H))	49.6	0.0	0.0	0.0	77.5	81.2	65.6	8.5	60.1	85.6	82.3	77.1
2	Grounded- SAM (Grounded- SAM-(L+H))	46.0	0.0	0.0	0.0	78.6	75.2	61.5	7.2	35.0	82.5	86.9	70.9
3	UNINEXT (UNINEXT)	42.1	0.0	0.0	0.0	72.1	57.0	56.3	0.0	54.0	80.7	81.1	84.1
4	SAN (SAN- CLIP-ViT-L)	41.4	22.1	10.6	43.5	67.4	62.9	43.5	9.0	60.1	81.8	77.4	82.2
5	odise (ODISE-L)	38.7	0.0	0.0	0.0	74.9	51.4	37.5	9.3	60.4	79.9	81.3	71.9
6	OpenSEED (OpenSEED- L)	36.1	19.7	15.0	4.7	72.9	38.7	52.3	1.8	50.0	82.8	76.4	40.0

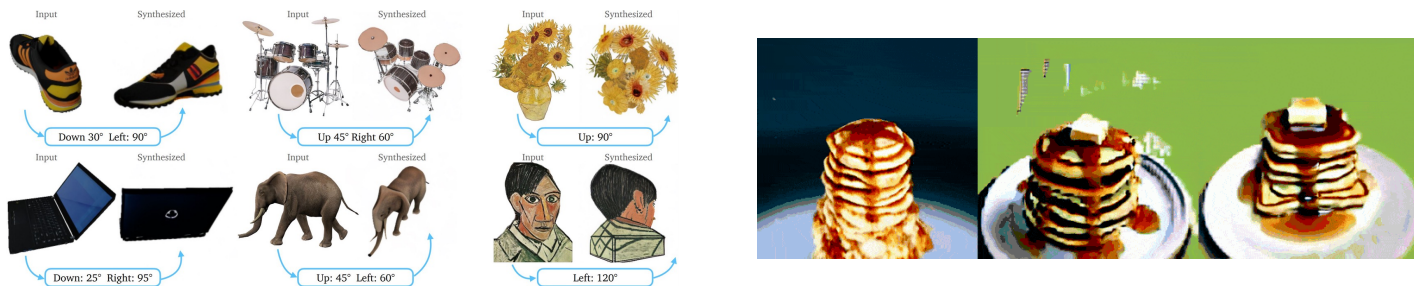
Segmentation in the Wild competition @ CVPR 2023

Notable Applications of SAM

- **Modeling 3D objects for in-the-wild images** (e.g., Anything 3D [NUS, 2023])

Basic Idea: **utilize SAM to segment an object's 2D view, then escalate to 3D.**

- **3D novel-view generation methods** has rapidly emerged recently
e.g., Zero1-to-3 [Liu et al., 2023], 3D Fuse [Seo et al., 2023]



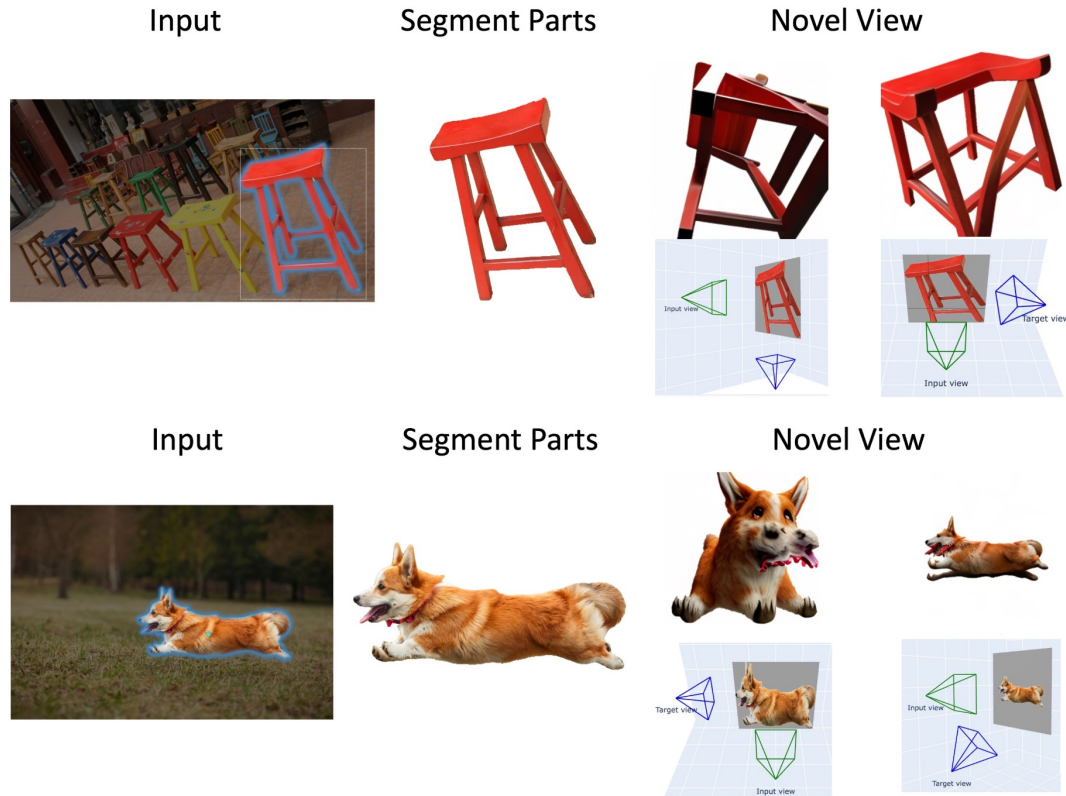
- However, **clean-cut inputs** are required for them to work
 - Constrains in-the-wild usage

Notable Applications of SAM

- **Modeling 3D objects for in-the-wild images (e.g., Anything 3D [NUS team, 2023])**

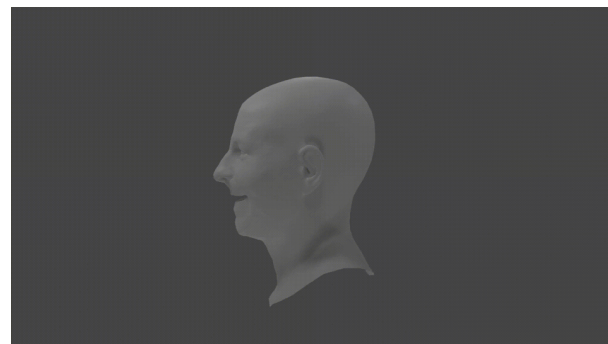
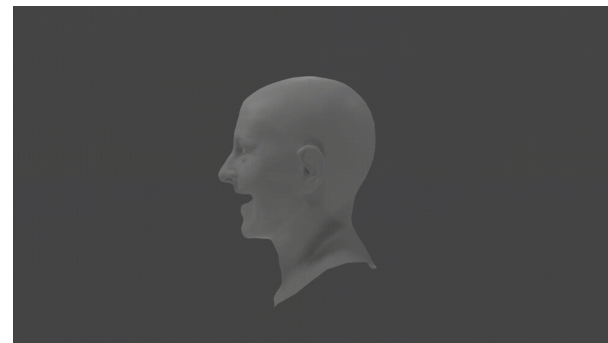
Basic Idea: **utilize SAM to segment an object's 2D view, then escalate to 3D.**

- Given the SAM predictions, clean-cut objects can be readily available



Notable Applications of SAM

- **Modeling 3D objects for in-the-wild images (e.g., Anything 3D [NUS, 2023])**
Basic Idea: **utilize SAM to segment an object's 2D view, then escalate to 3D.**
- Given the SAM predictions, clean-cut objects can be readily available



Anything 3D-Face [NUS team, 2023]

Conclusion

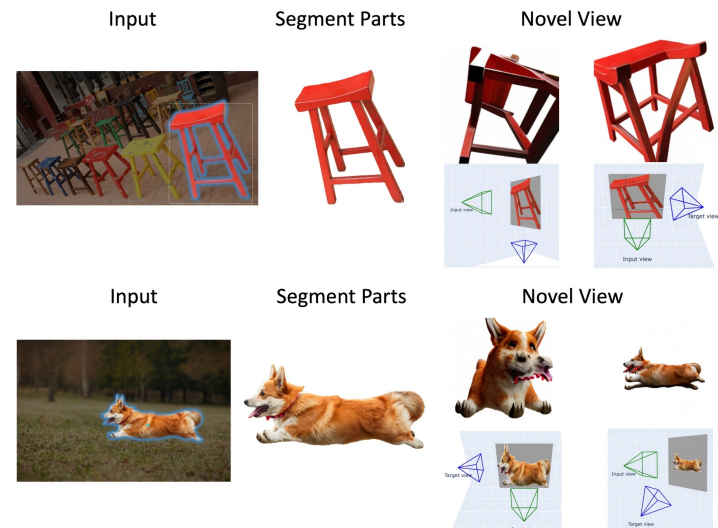
Segment Anything Model, a **foundation model** in Vision AI

- Trained on a **web-scale** dataset of 11M images & 1B+ masks
- **Adaptable** to wide range of image domains & tasks via user prompts

Foundation Model = *scale & flexibility*



SA-1B



Anything 3D [NUS team, 2023]

References

- [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
- [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
- [Zhou et al., 2022] ibot: Image bert pre-training with online tokenizer, ICLR 2022
- [Baeovski et al., 2022] data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language, 2022
- [Oquab et al., 2023] DINOv2: Learning Robust Visual Features without Supervision, 2023
- [Radford et al., 2021] Learning Transferable Visual Models From Natural Language Supervision, ICML 2021
- [Schuhmann et al., 2022] Laion-5b: An open large-scale dataset for training next generation image-text models, NeurIPS 2022
- [Fang et al., 2022] Data Determines Distributional Robustness in Contrastive Language Image Pre-training (CLIP), 2022
- [Zhai et al., 2023] Sigmoid Loss for Language Image Pre-Training, ICCV 2023
- [Mehdi, et al. 2023] Reproducible scaling laws for contrastive language-image learning., CVPR 2023

References

- [Hertz et al., 2022] Prompt-to-Prompt Image Editing with Cross Attention Control, ICLR 2023
- [Brooks et al., 2022] InstructPix2Pix: Learning to Follow Image Editing Instructions, CVPR 2023
- [Zhang et al., 2023] Adding Conditional Control to Text-to-Image Diffusion Models, ICCV 2023
- [Gal et al., 2022] An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion
- [Ruiz et al., 2022] DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation, CVPR 2023
- [Ruiz et al., 2023] HyperDreamBooth: HyperNetworks for Fast Personalization of Text-to-Image Models, 2023
- [Poole et al., 2022] DreamFusion: Text-to-3D using 2D Diffusion, ICLR 2023
- [Kirillov et al., 2023] Segment Anything.
- [Yang et al., 2023] SAM3D: Segment Anything in 3D Scenes.
- [Liu et al., 2023] Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection
- [Gu et al., 2022] Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. ICLR 2022
- [NUS team, 2023] Anything-3D: Towards Single-view Anything Reconstruction in the Wild.