NeurIPS 2022?

MILAN: Masked Image Pretraining on Language Assisted Representation

Zejiang Hou[1] **Fei Sun**[2] **Yen-Kuang Chen**[2] **Yuan Xie**[2] **Sun-Yuan Kung**[1]

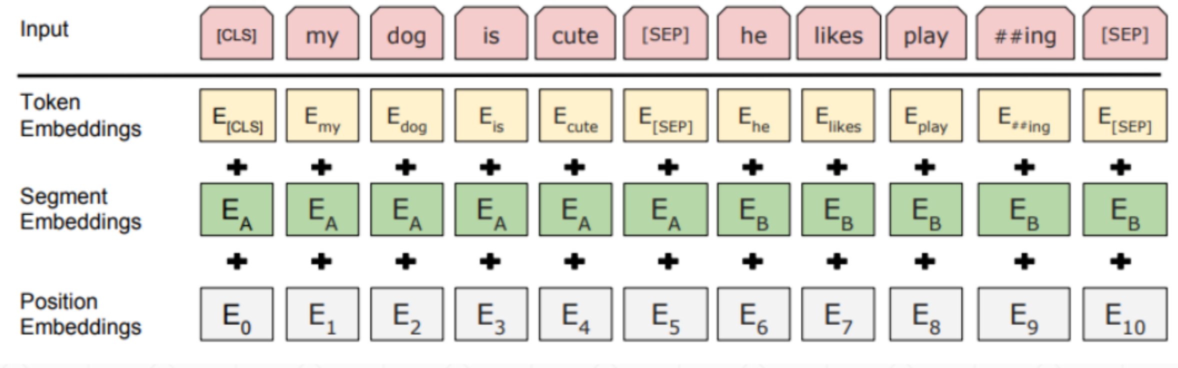
[1]Princeton University [2]DAMO Academy, Alibaba Group

Mengxue

Preview

- **Fully-supervised → Self-supervised**
- **Reconstruction based self-supervised pretraining**
- **Masked Data Modeling**
  - NLP → BERT
    - *MLM (Masked Language Modeling)*
  - V-L → VL-BERT/ViLBERT
    - *MLM*
    - *MIM (Masked Image Modeling)*
    - *… …*
  - CV → MAE
    - *MIM*

BERT



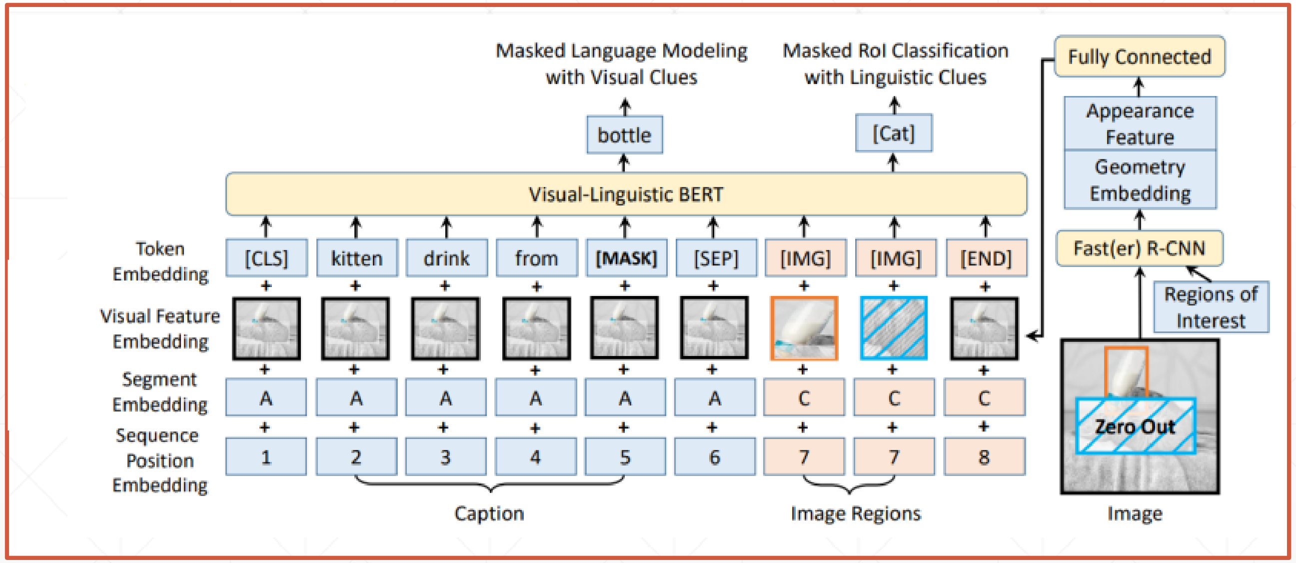
→ 上游任务：Mask Language Modelling & Next Sentence Prediction

→ 下游任务：句子对分类任务、单句子分类任务、问答任务、单句标注任务

Preview

- **Fully-supervised → Self-supervised**
- **Reconstruction based self-supervised pretraining**
- **Masked Data Modeling**
  - NLP → BERT
    - *MLM (Masked Language Modeling)*
  - V-L → VL-BERT/ViLBERT
    - *MLM*
    - *MIM (Masked Image Modeling)*
    - … …
  - CV → MAE
    - *MIM*

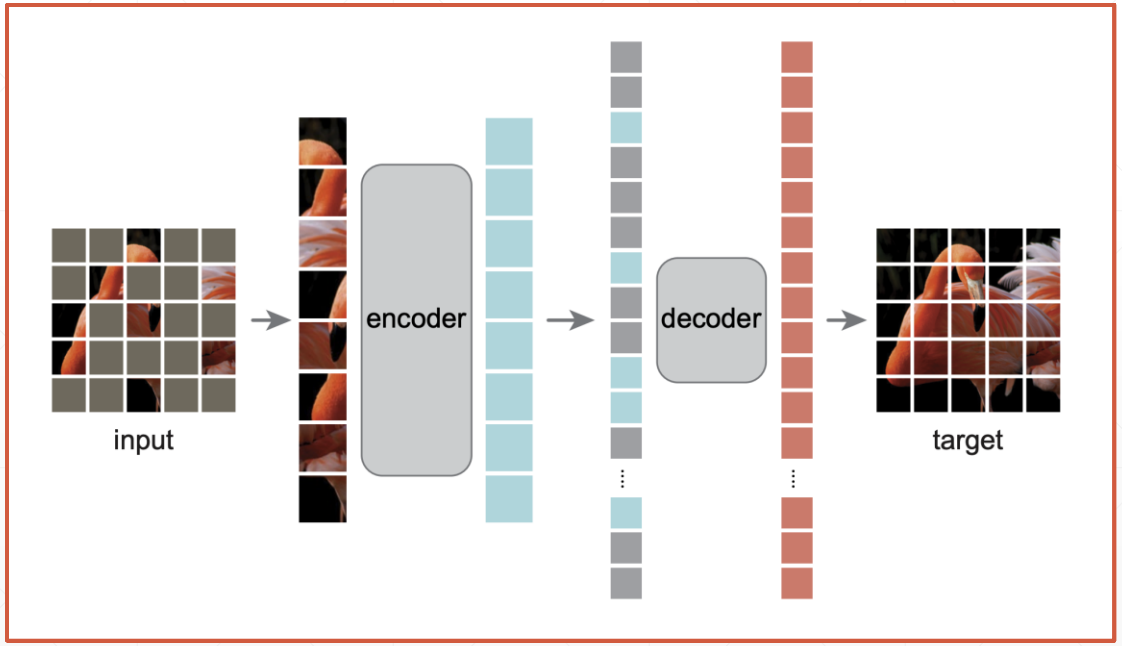
VL-BERT



Preview

- **Fully-supervised → Self-supervised**
- **Reconstruction based self-supervised pretraining**
- **Masked Data Modeling**
  - NLP → BERT
    - *MLM (Masked Language Modeling)*
  - V-L → VL-BERT/ViLBERT
    - *MLM*
    - *MIM (Masked Image Modeling)*
    - *… …*
  - CV → MAE
    - *MIM*

MAE



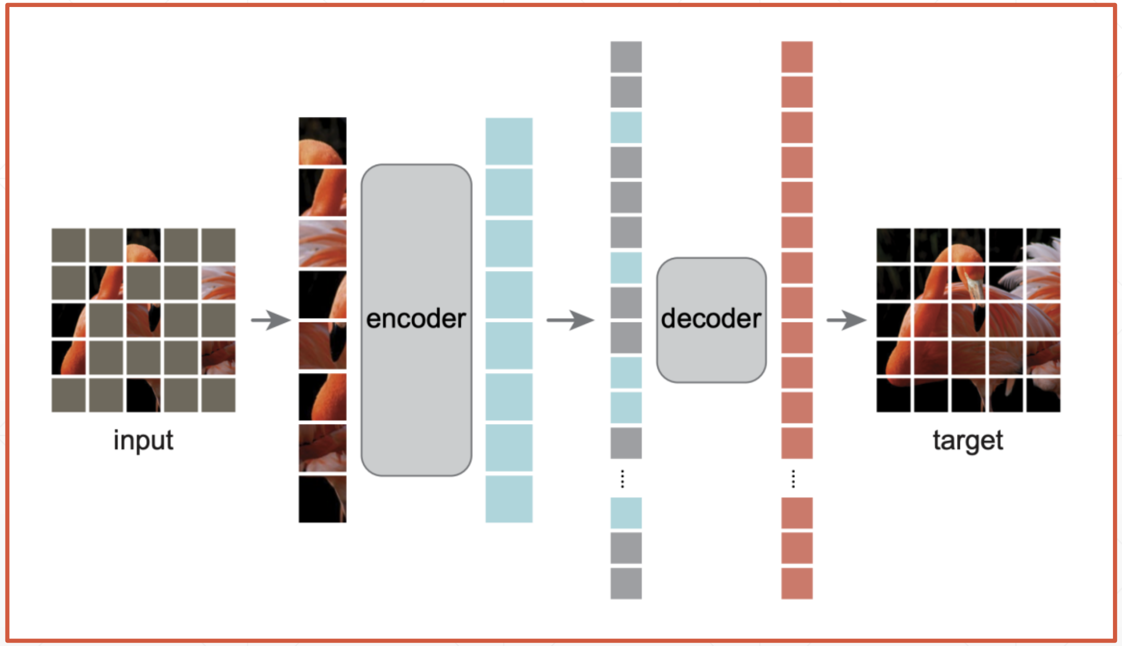
Introduction

- **In this work, we analyze three highly correlated aspects in MAE(masked autoencoders)**:
  - the reconstruction target
  - the decoder design
  - the mask sampling strategy
- We propose a new approach called **MILAN**, which performs **masked image pretraining** on **language assisted** representations.

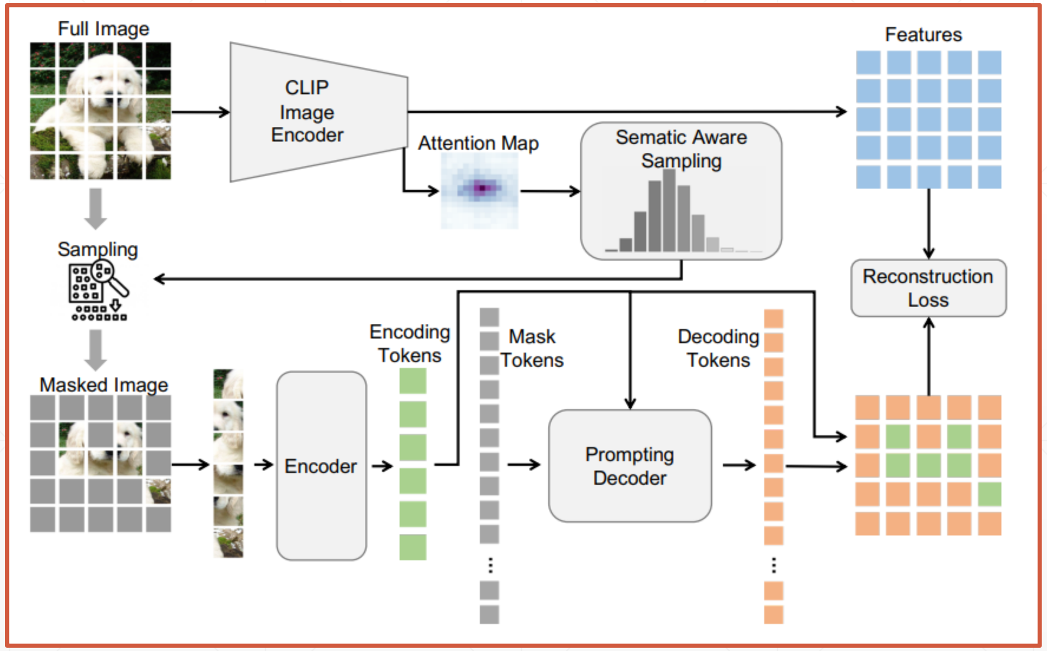
Methodology—Difference in MAE & MILAN

Predict Target / Sampling strategy / Decoder Design

MAE



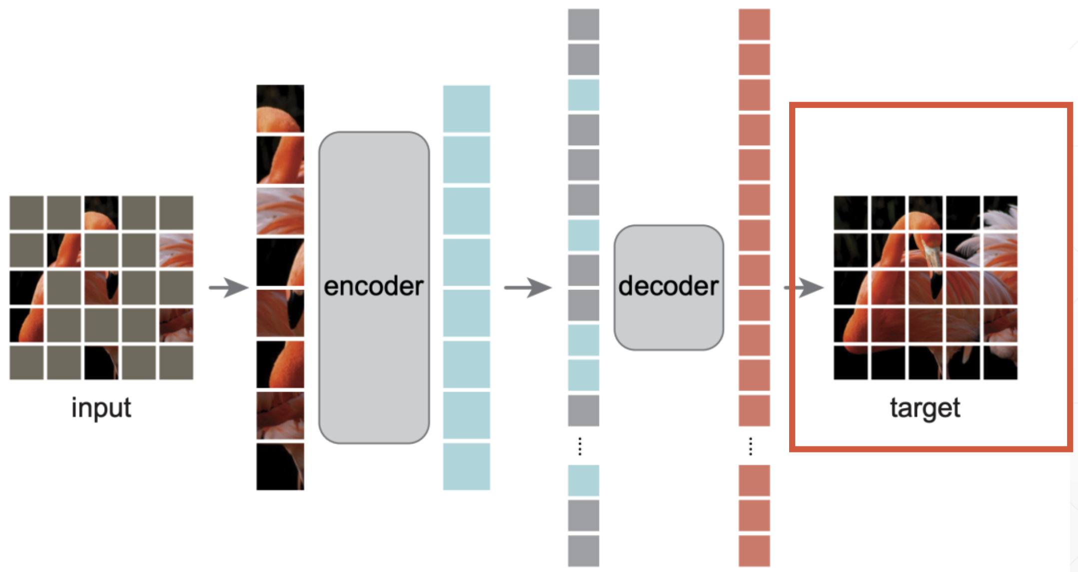
MILAN

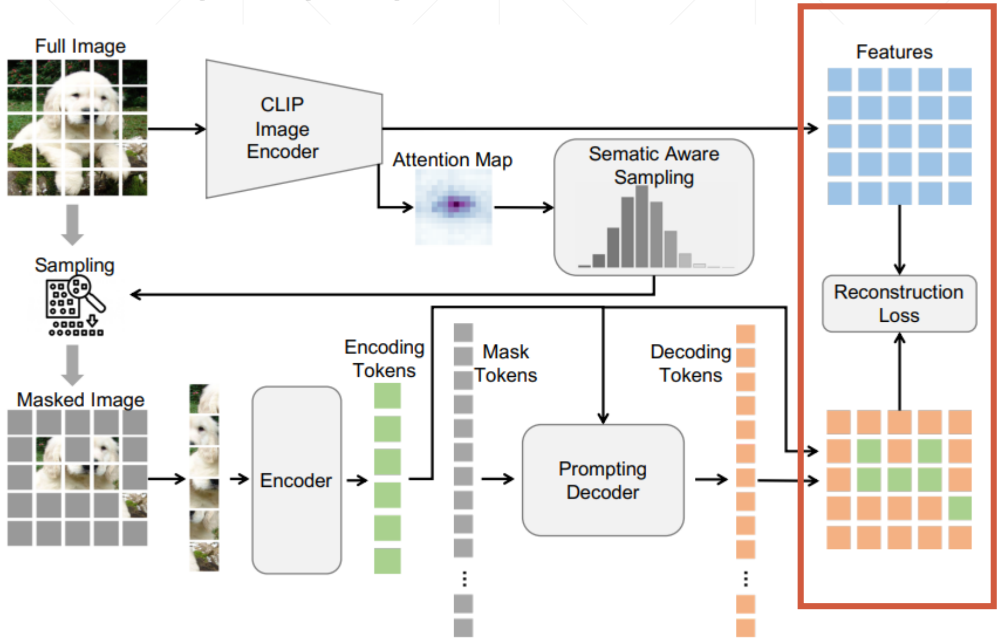


Methodology—Difference in MAE & MILAN

Predict Target / Sampling strategy / Decoder Design

- **Predict Target**
  - MAE: raw pixels
  - MILAN: latent representations obtained with language guidance $\mathcal{L}_{\xi,\nu} = (1/N) \cdot \sum_{j=1}^{N} \|\bar{\boldsymbol{p}}_j - \bar{\boldsymbol{t}}_j\|_2^2.$



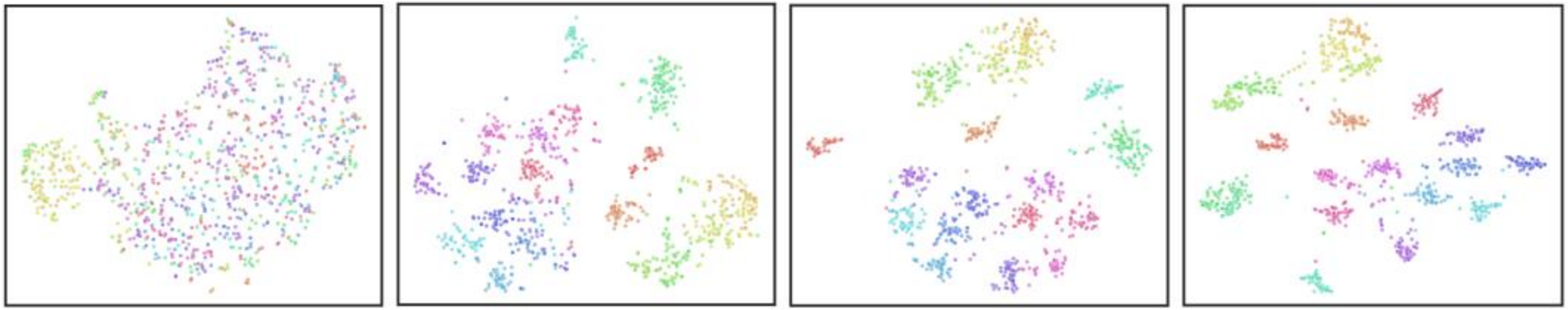


Methodology

Reconstruction target: language assisted representation

- **Why use CLIP feature as targets?**

  → The learned representations are better clustered for different categories



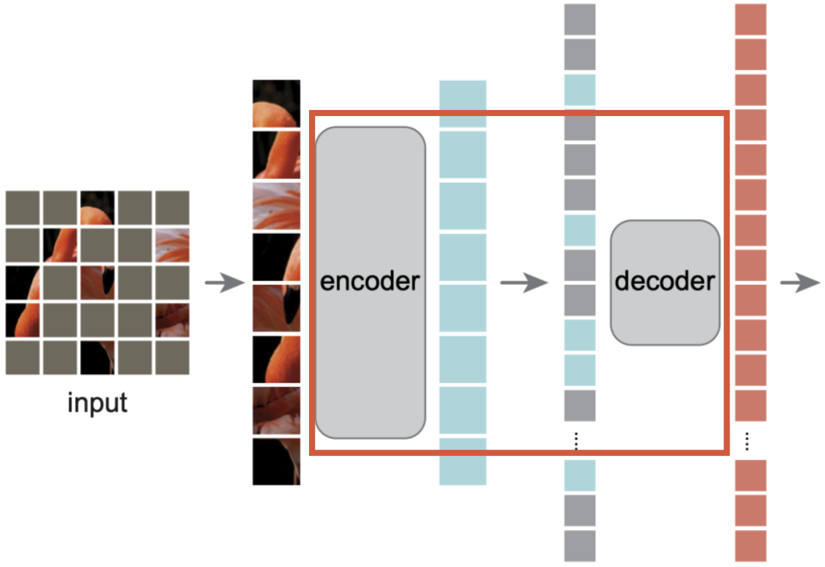
(a) MAE pretrained (b) CLIP image encoder (c) MILAN pretrained (d) MILAN finetuned

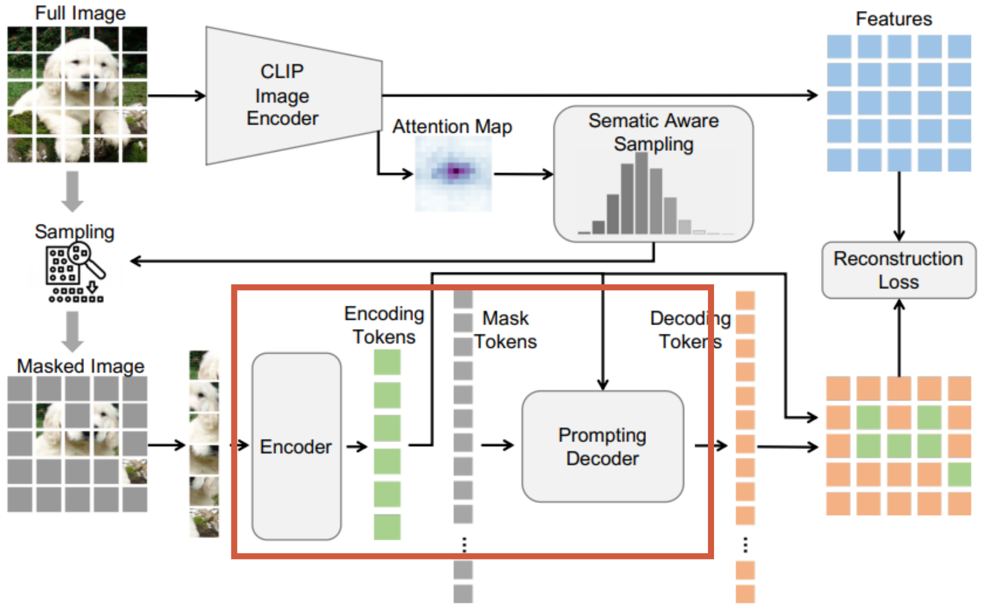
Figure 2: t-SNE visualization of the learned features from ViT-B/16 obtained by different pretraining methods. We plot the features before the final linear head. We use images of randomly sampled 20 classes in ImageNet-1K validation split.

Methodology—Difference in MAE & MILAN

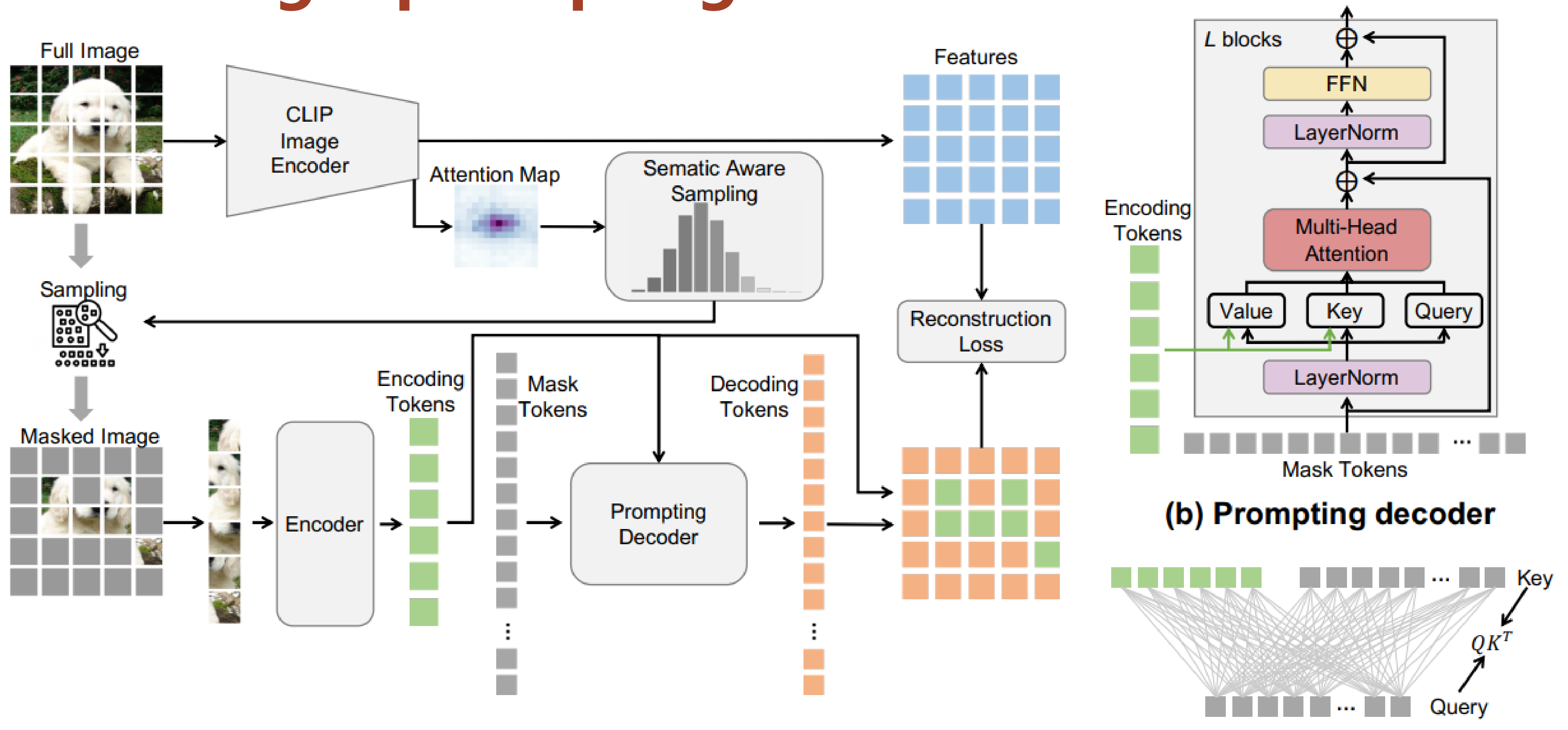
Predict Target / Sampling strategy / **Decoder Design**

- **Decoder Design**
  - MAE: normal encoder-decoder, both update
  - MILAN: prompting decoder, does not update the encoder (more efficient)





Methodology
Decoder design: prompting decoder



(a) Overall flow of the MILAN framework

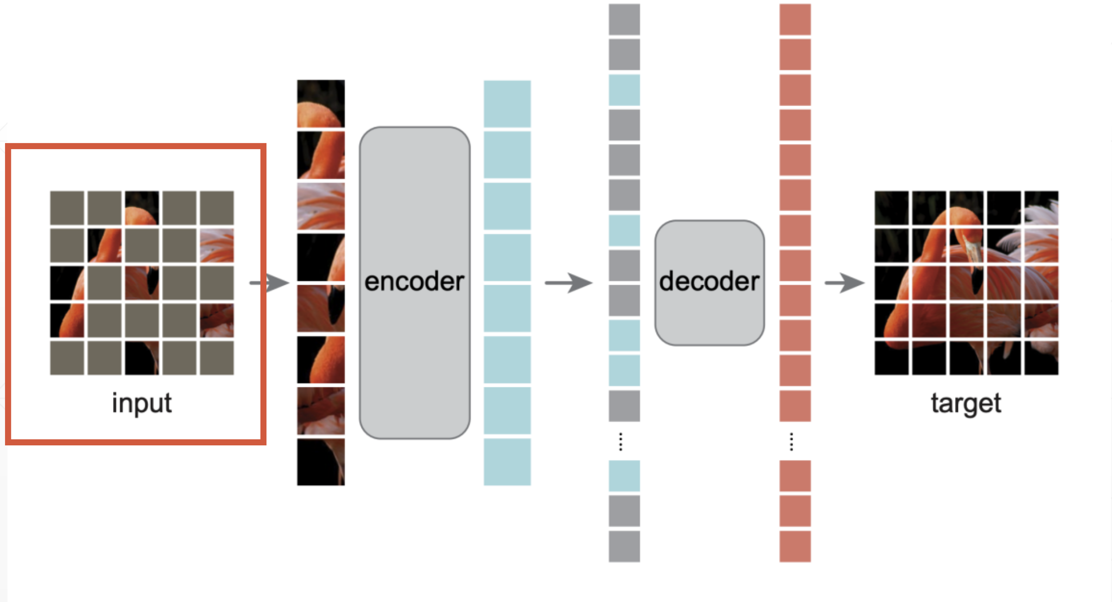
(c) Attention in the prompting decoder

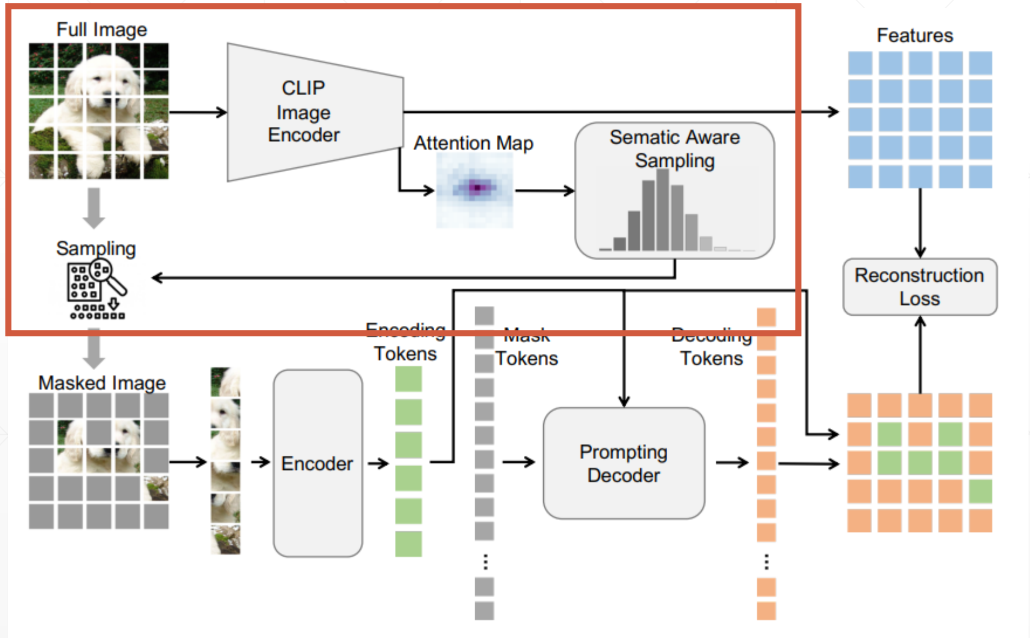
Using the default 75% masking ratio, our prompting decoder reduces the computation cost by **20%** compared to MAE [25].

Methodology—Difference in MAE & MILAN

Predict Target / **Sampling strategy** / Decoder Design

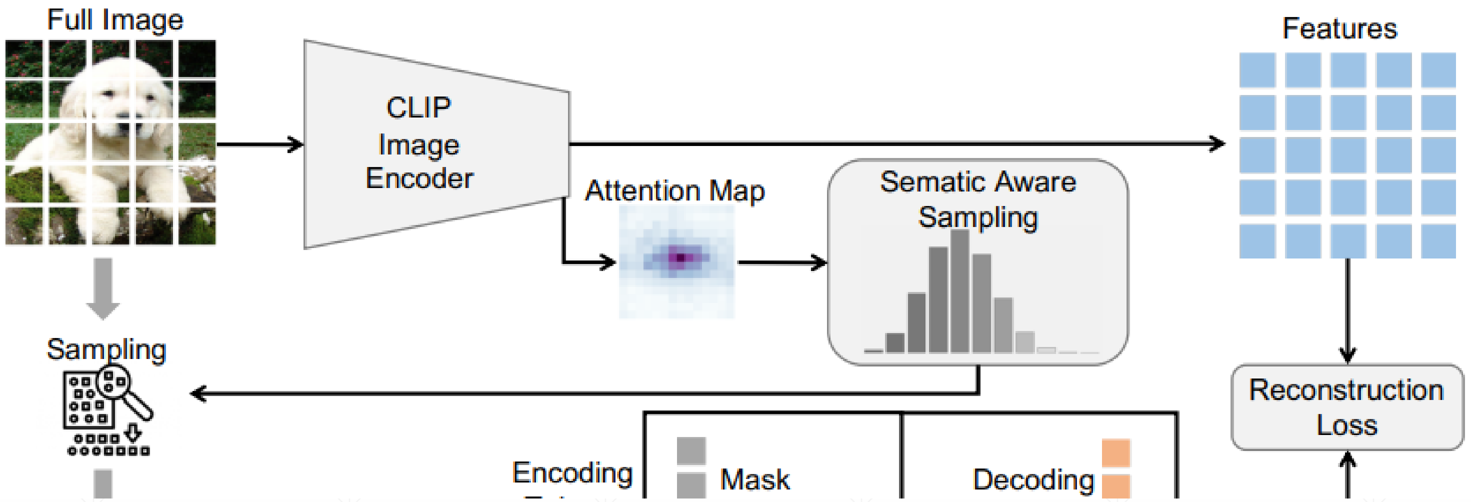
- **Sampling strategy**
  - MAE: uniform sampling
  - MILAN: mask sampling (more adapted to patches' discriminativeness)





Methodology

Masking strategy: semantic aware sampling



CLIP image encoder by $[\mathbf{z}_{\text{class}}; \mathbf{z}_1; \ldots; \mathbf{z}_N] \in \mathbb{R}^{(N+1)\times d}$,

$$\mathbf{s}_{\text{class}} = \text{softmax}(\mathbf{q}_{\text{class}}\mathbf{K}^T/\sqrt{d}),$$

$$\mathbf{q}_{\text{class}} = \mathbf{z}_{\text{class}} W_q$$

$$\mathbf{K} = [\mathbf{z}_{\text{class}}; \mathbf{z}_1; \ldots; \mathbf{z}_N] W_k$$

Because **the class token** from the last layer of the CLIP image encoder is used to **align with the text embedding** from the text encoder

Sclass reflects how much information one image patch contributes to the output features of the CLIP image encoder.

Experiments

- We pretrain the ViT-B/16 and ViT-L/16 models using MILAN method on **ImageNet-1K** dataset for **400 epochs** using PyTorch framework on **A100** machines.
- We use the ViT-B/16 CLIP image encoder obtained from OpenAI’ s paper [43] to produce the reconstruction targets when pretraining both ViT-B/16 and ViT-L/16 models.

Experiments
Classification on ImageNet-1K

Method	Training data	Resolution	ViT-B/16 Epochs	ViT-B/16 Top-1 (%)	ViT-L/16 Epochs	ViT-L/16 Top-1 (%)
Supervised [50]	IN1K	224	-	83.8 (+1.6)	-	84.9 (+1.8)
contrastive or clustering based						
MoCov3 [11]	IN1K	224	300	83.2 (+2.2)	300	84.1 (+2.6)
DINO [6]	IN1K	224	400	82.8 (+2.6)	-	-
iBOT [69]	IN22K+IN1K	224	320	84.4 (+1.0)	200	86.3 (+0.4)
reconstruction based						
BEiT [3]	DALLE250M+IN22K+IN1K	224	150	83.7 (+1.7)	150	86.0 (+0.7)
mc-BEiT [33]	OpenImages9M+IN1K	224	800	84.1 (+1.3)	800	85.6 (+1.1)
PeCo [18]	IN1K	224	800	84.5 (+0.9)	800	86.5 (+0.2)
SimMIM [61]	IN1K	224	800	83.8 (+1.6)	-	-
MaskFeat [56]	IN1K	224	1600	84.0 (+1.4)	1600	85.7 (+1.0)
data2vec [2]	IN1K	224	800	84.2 (+1.2)	1600	86.6 (+0.1)
CAE [9]	IN1K	224	800	83.6 (+1.8)	-	-
MAE [25]	IN1K	224	1600	83.6 (+1.8)	1600	85.9 (+0.8)
language-image pretraining based						
CLIP [43]	OpenAI400M+IN1K	224	-	82.1 (+3.3)	-	85.3 (+1.4)
MVP [57]	OpenAI400M+IN1K	224	300	84.4 (+1.0)	300	86.3 (+0.4)
MILAN	OpenAI400M+IN1K	224	400	**85.4**	400	**86.7**
Supervised [19]	JFT300M+IN1K	384	90	84.2 (+2.2)	90	87.1 (+0.2)
BEiT [3]	DALLE250M+IN1K	384	800	84.6 (+1.8)	800	86.3 (+1.0)
SWAG [47]	IG3.6B+IN1K	384	2	85.3 (+1.1)	-	-
MILAN	OpenAI400M+IN1K	384	400	**86.4**	400	**87.3**

Table 1: Comparison of the **finetuning** top-1 accuracy on ImageNet-1K dataset. All models are pretrained with 224×224 input resolution. We compare finetuning with both 224×224 and 384×384 resolutions. "Epochs" refer to the pretraining epochs. "-": not reported by the original paper.

Downstream tasks

- Object detection and instance segmentation on COCO
- Semantic segmentation on ADE20K

Method	Epochs	Detection AP_{box}	Instance Segmentation AP_{mask}	Semantic Segmentation mIoU
Supervised [27, 59]	-	47.9 (+4.7)	42.9 (+2.6)	47.4 (+5.3)
MoCov3 [11]	300	47.9 (+4.7)	42.7 (+2.8)	47.3 (+5.4)
DINO [6]	300	46.8 (+5.8)	41.5 (+4.0)	47.2 (+5.5)
BEiT [3]	300	42.6 (+10.)	38.8 (+6.7)	45.7 (+7.0)
PeCo [18]	300	43.9 (+8.7)	39.8 (+5.7)	46.7 (+6.0)
SplitMask [20]	300	46.8 (+5.8)	42.1 (+3.4)	45.7 (+7.0)
CAE [9]	800	49.2 (+3.4)	43.3 (+2.2)	48.8 (+3.9)
MAE [25]	1600	50.3 (+2.3)	44.9 (+0.6)	48.1 (+4.6)
MILAN	400	**52.6**	**45.5**	**52.7**

Table 3: Results of object detection and instance segmentation are obtained by using Mask R-CNN on COCO dataset with an input resolution of 1024×1024. Semantic segmentation results are obtained by using UperNet on ADE20K with an input resolution of 512×512. All methods use ViT-B/16 pretrained on ImageNet-1K dataset as the backbone. "Epochs" refer to the pretraining epochs.

Experiments

Ablation study

	CLIP target	Prompting decoder	Semantic aware sampling	Epochs	Top-1 (%)
#1	Baseline (MAE)			400 (1600)	83.0 (83.6)
#2	✓			400	83.9
#3		✓		400	83.0
#4			✓	400	83.3
#5		✓	✓	400	83.3
#6	✓		✓	400	84.1
#7	✓	✓		400	85.1
#8	✓	✓	✓	400 (1600)	**85.4** (85.6)
#9	SLIP target	✓	✓	400	84.4

Table 4: Ablation study of different components in MILAN. All results are obtained by pretraining and finetuning ViT-B/16 model on ImageNet-1K dataset at 224×224 resolution.

Experiments
Visualization

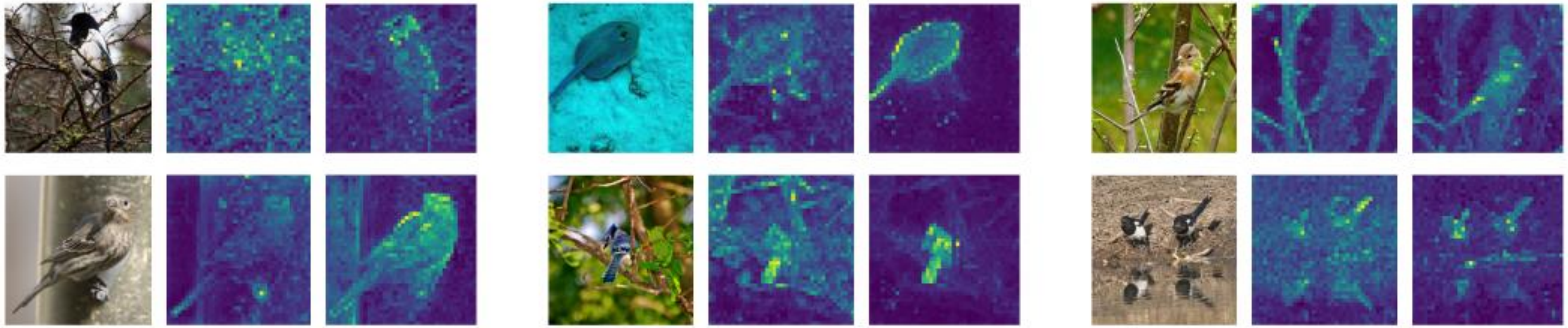


Figure 3: Visualization of original images (left), and the attention features extracted from the last self-attention layer of ViT-B/16 model pretrained by MAE (middle) and MILAN (right).

Figure 4: Visualization of the original images (left), masked images by the semantic aware sampling strategy with 75% masking ratio (middle), and the reconstruction loss patch-by-patch (right). For the plots of reconstruction loss, darker green colors indicate higher loss values. As shown, both unmasked patches and masked foreground patches have lower losses.

Limitation

- Similar to [3, 9, 33] which rely on external datasets to train their image tokenizers, the reconstruction target in MILAN is obtained from the CLIP model which also requires an **extra image-text dataset**. **Training the CLIP model, if it is not amortized for many downstream tasks, is considered an extra training step.**
- Moreover, we recognize that our improvements on ViT-L is not as significant as those on ViT-B. This may be because we employ the ViT-B version of the CLIP image encoder to produce the reconstruction targets for training both ViT-B and ViT-L for the sake of computational efficiency.

Thanks