Overview of Image-Text Pre-training

Jianfeng Wang

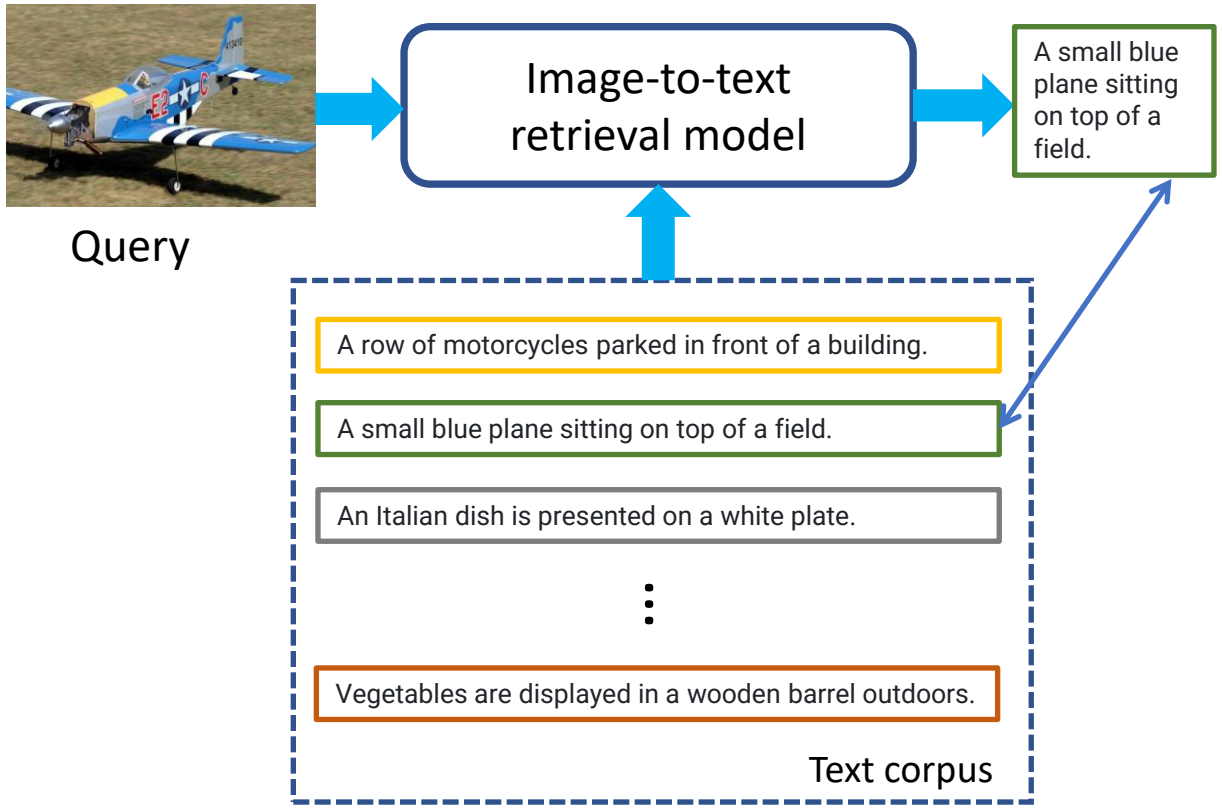
Microsoft Azure AI

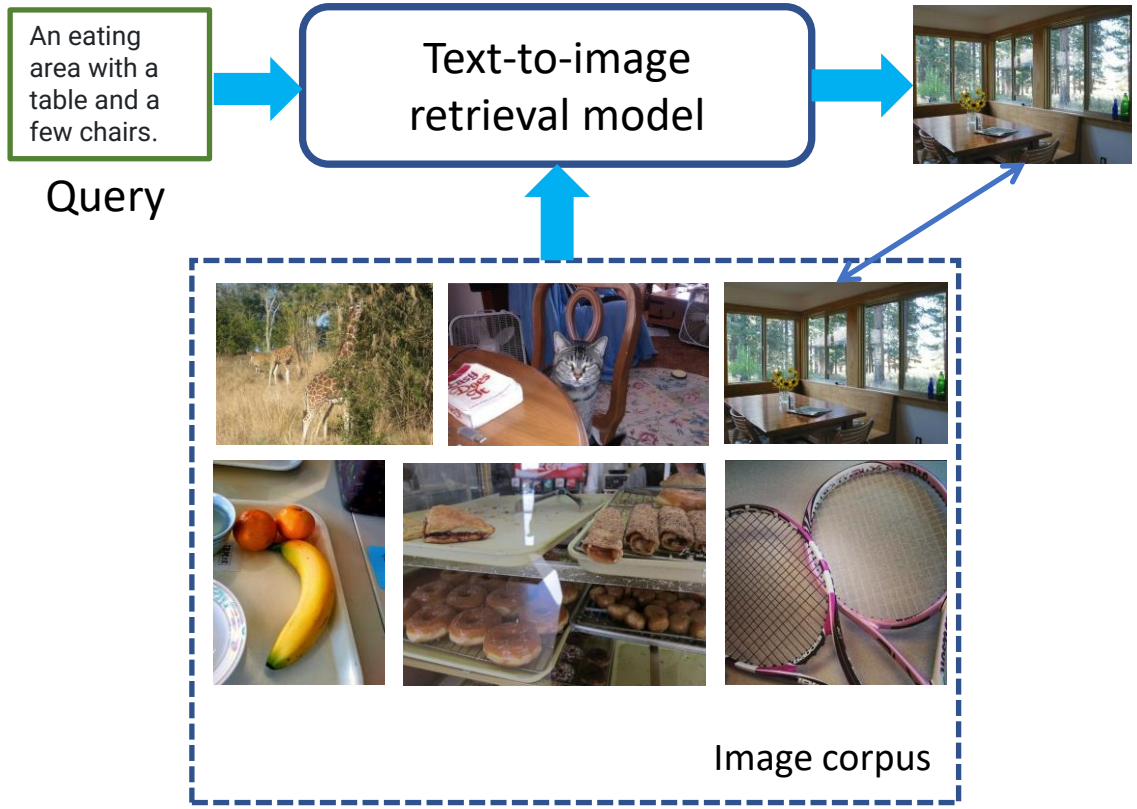
Outline

- Application
  - Retrieval, captioning, question answering
- Network architecture
  - Image encoder, text encoder, multi-model fusion
- Pre-training tasks
  - ITC, ITM, MLM
- Adaptation to downstream tasks

Application

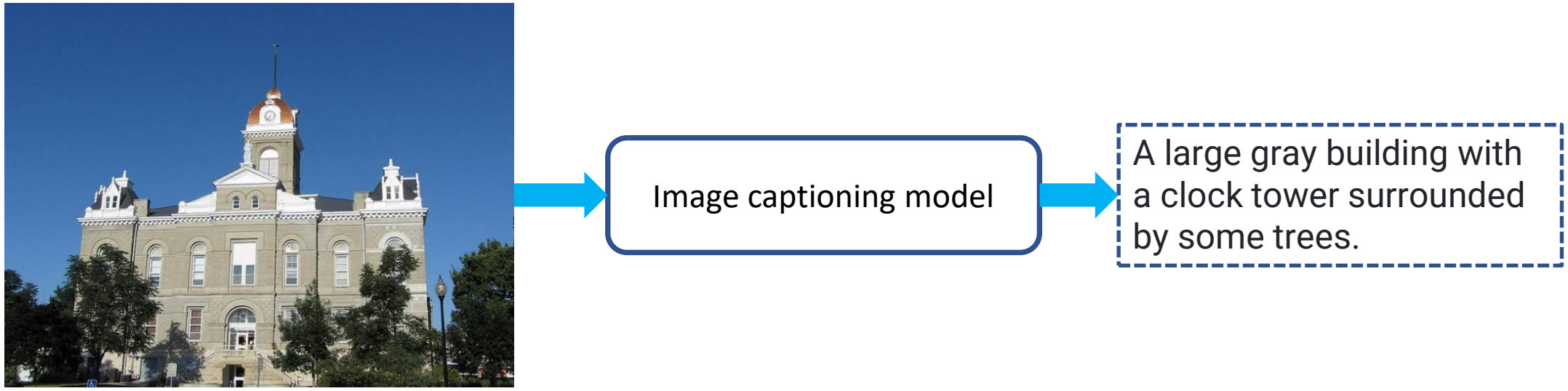
- Multi-modal retrieval





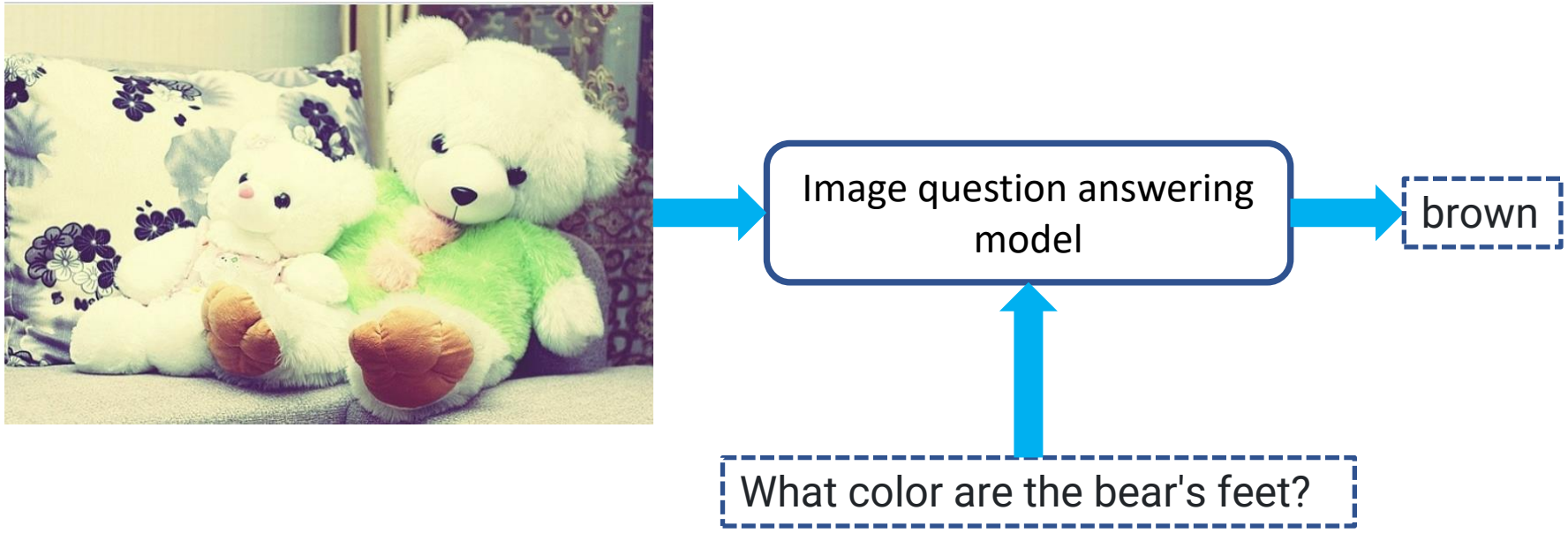
Application

- Image captioning



Application

- Image question answering



- *Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering*, 2016

Application

- Key problem
  - Understand the image, the text, and the **relation**
- How?
  - Image-text pre-training
    - Large-scale dataset of image-text pairs

a very typical bus station

functions of government : 1 . form a more perfect union

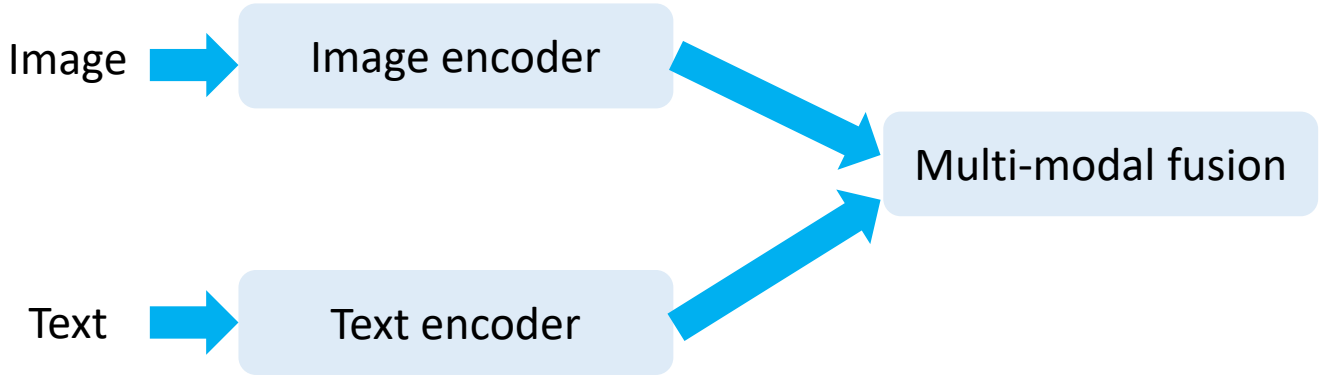
safe deposit with money around it on a white background photo

emergency services were called after a car smashed through a set of traffic lights

- *Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning*, 2018

Network architecture

- Image encoder, Text encoder, Multi-modal fusion



Network architecture

- Image encoder, Text encoder, Multi-modal fusion

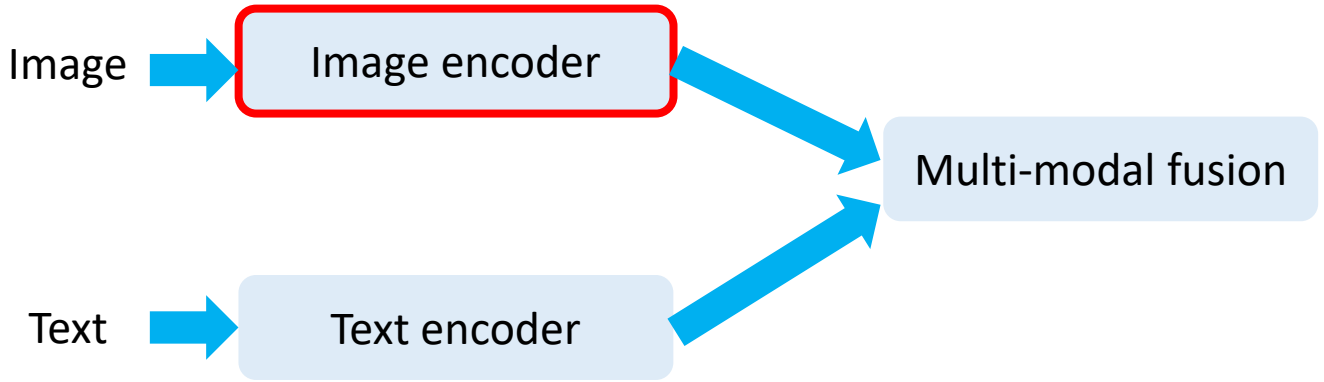
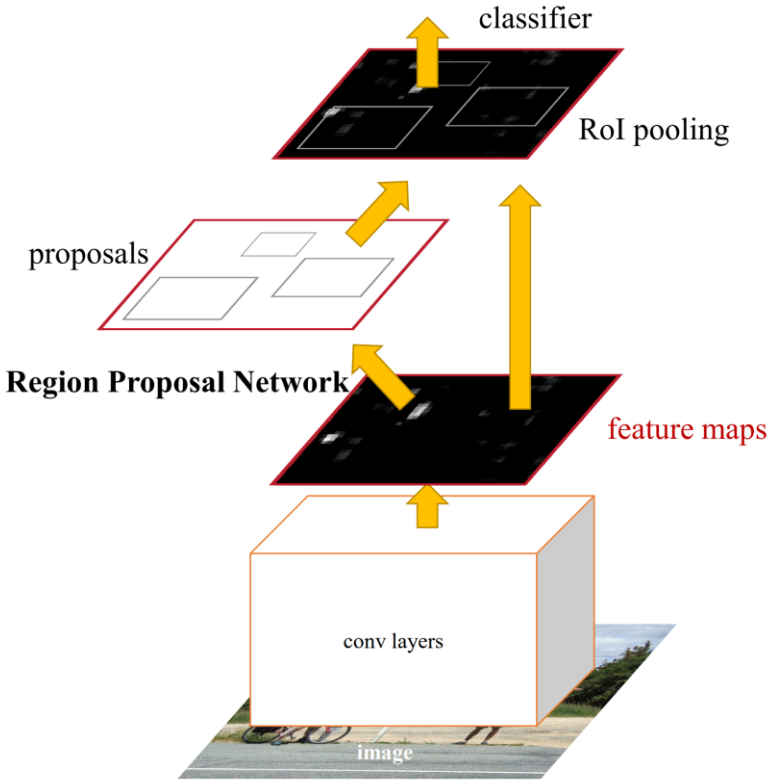


Image encoder

- Sparse feature
  - Object detector
- Dense feature
  - Convolutional neural network (CNN), Vision transformer (ViT), e.t.c.

Sparse feature - Object detector

- Faster RCNN



- Training data
  - Visual Genome
    - 1k+ categories with attributes
- Network
  - Resnet 101 as the backbone
    - (BUTD, 2018)

- *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, 2015
- *Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering*, 2018

Sparse feature - Object detector

- Stronger object detector
  - VinVL
    - Resnet101 -> X152
    - Pretraining with
      - VG + COCO + Objects365 + OpenImages
- Faster object detector
  - MiniVLM
    - Resnet101 -> EfficientNet
      - Pretraining with Objects365

vision \ vl	no VLP	OSCAR$_B$ [21]	OSCAR+$_B$ (ours)
R101-C4 [2]	68.52 ±0.11	72.38	72.46±0.05
VinVL (ours)	71.34 ±0.17	–	74.90±0.05

Table 12: Effects of vision (V) and vision-language (VL) pre-training on VQA.

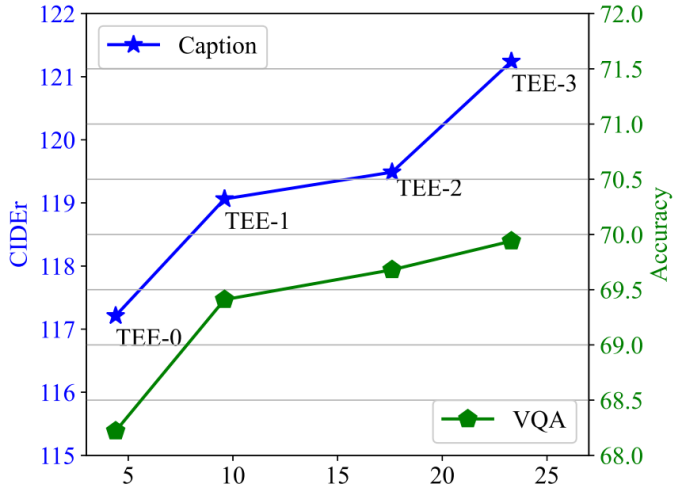


Figure 3: Impact of different backbones in TEE for COCO captioning task and VQA (`test-dev`). Overall, the stronger feature extractor leads to the higher score.

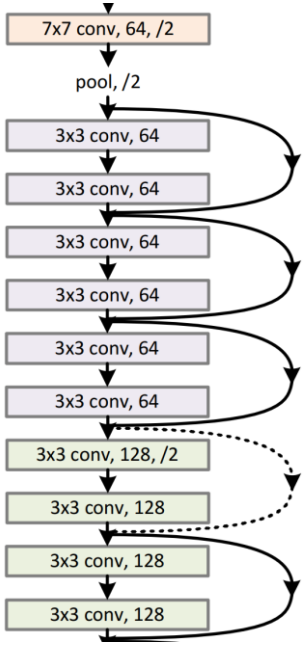
- *VinVL: Revisiting Visual Representations in Vision-Language Models*, 2021
- *MiniVLM: A Smaller and Faster Vision-Language Model*, 2020

Sparse feature -> dense feature

- Sparse feature
  - Object detector
    - Box labels
      - Expensive to annotate
    - Unclear of how to train in an end-to-end way with Faster-RCNN
- Dense feature
  - No need of the box labels

Sparse feature – Dense feature - CNN

- Convolutional neural network
  - Resnet50, Resnet101
    - Pre-trained on ImageNet



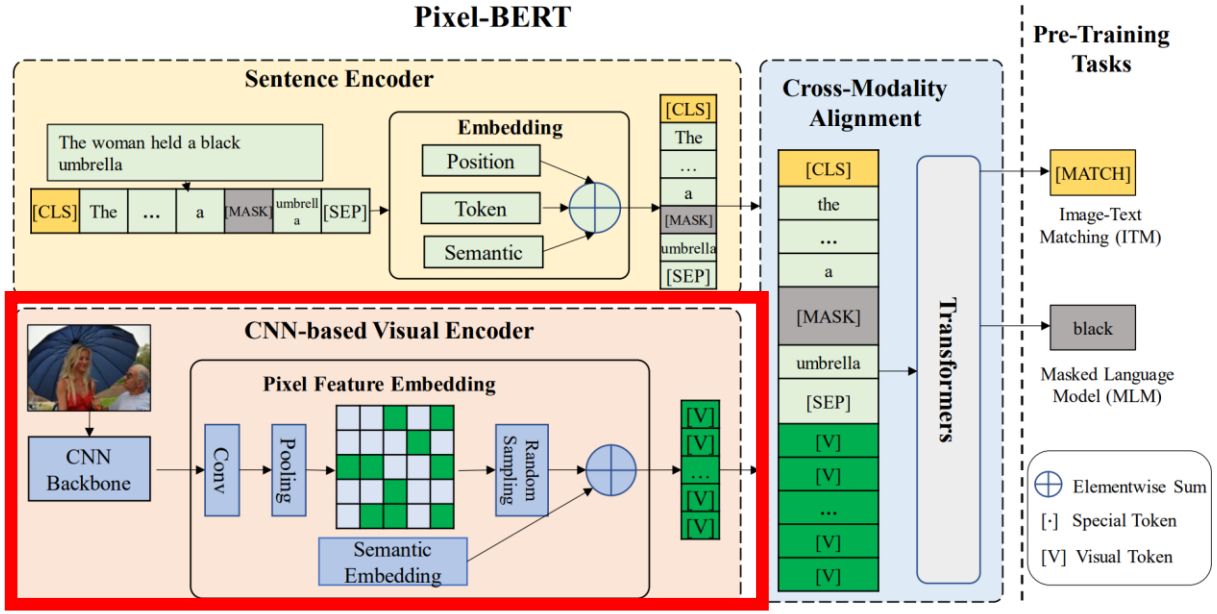


Fig. 2. Pixel-BERT: The model contains a visual feature embedding module, a sentence feature embedding module, and a cross-modality alignment module. Pixel-BERT takes image-sentence pairs as input, and outputs the attention features of each input element. Images are passed into a pixel feature embedding module pixel by pixel and sentences are fed into a sentence feature embedding module token by token. The model can be pre-trained by MLM and ITM tasks, and can be flexibly applied to downstream tasks (e.g. VQA, retrieval, etc).

- *Deep Residual Learning for Image Recognition*, 2015
- *Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers*, 2020

Sparse feature – Dense feature - CNN

- Convolutional neural network
  - Resnet101 and Resnet152
    - Random initialization

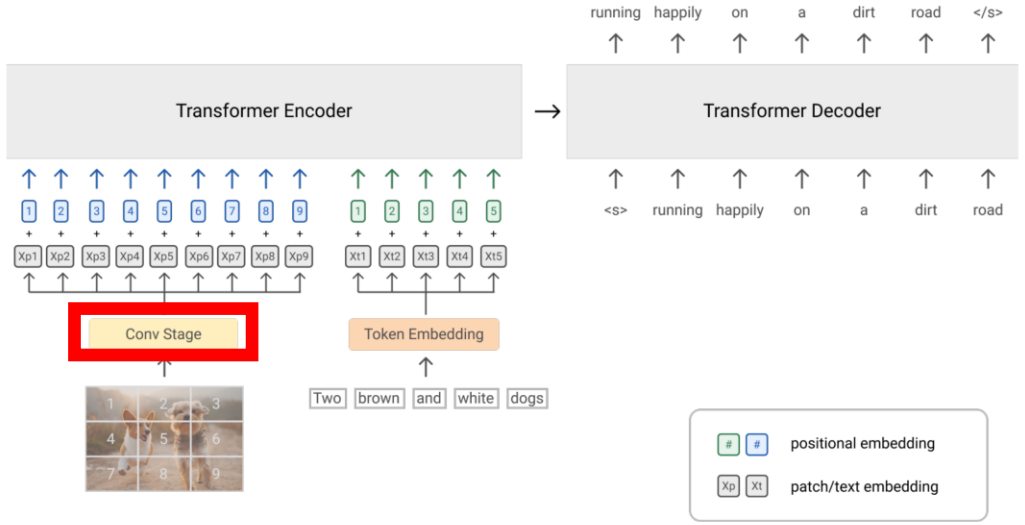


Figure 1: Illustration of the SimVLM model. This shows an example of training with PrefixLM of an image-text pair. For text-only corpora, it is straightforward to remove the image patches and utilize textual tokens only.

- *SimVLM: Simple Visual Language Model Pretraining with Weak Supervision*, 2021

Sparse feature – Dense feature - ViT

- Vision transformer
  - ViT
    - Pretrained with Imagenet cls. or CLIP
  - BEiT
    - Pretrained with self-supervised learning
  - Swin
    - Pretrained with ImageNet cls.

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

Table 3. Comparisons of different vision encoders without VLP. RoBERTa is used as the default text encoder. IR/TR: Flickr30k image/text retrieval; B: Base. The results of ImageNet classification are copied from their corresponding papers. All the results on VL tasks are from their test-dev/val sets. N and M in ViT-N/M denote the image resolution and patch size, respectively.

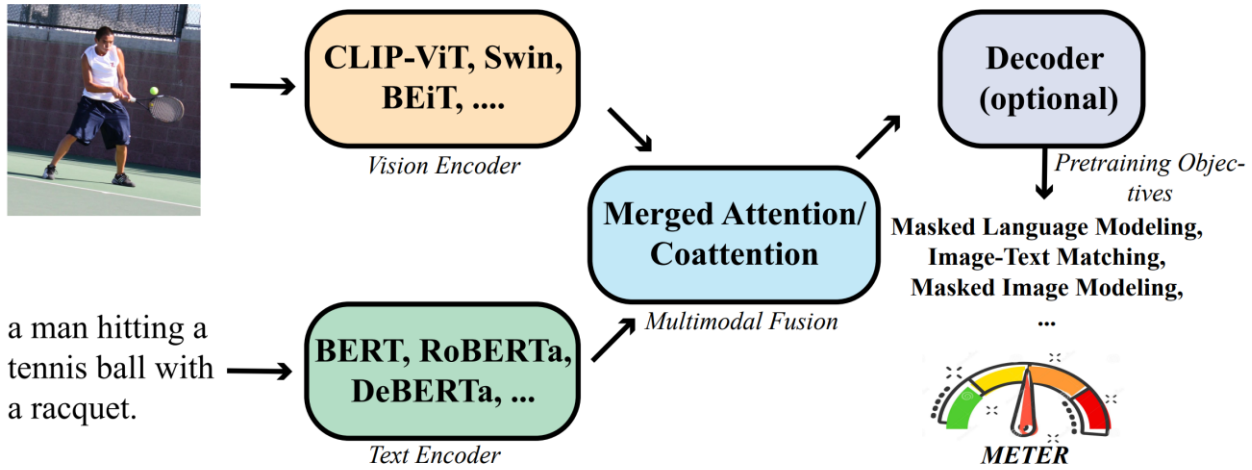


Figure 1. An overview of the proposed METER framework. We systematically investigate how to train a performant vision-and-language transformer, and dissect the model designs along multiple dimensions: vision encoder, text encoder, multimodal fusion module, architectural design (encoder-only vs. encoder-decoder), and pre-training objectives.

- *An Empirical Study of Training End-to-End Vision-and-Language Transformers*, 2021

Sparse feature – Dense feature – Patch

- One convolutional layer
  - Kernel size == stride

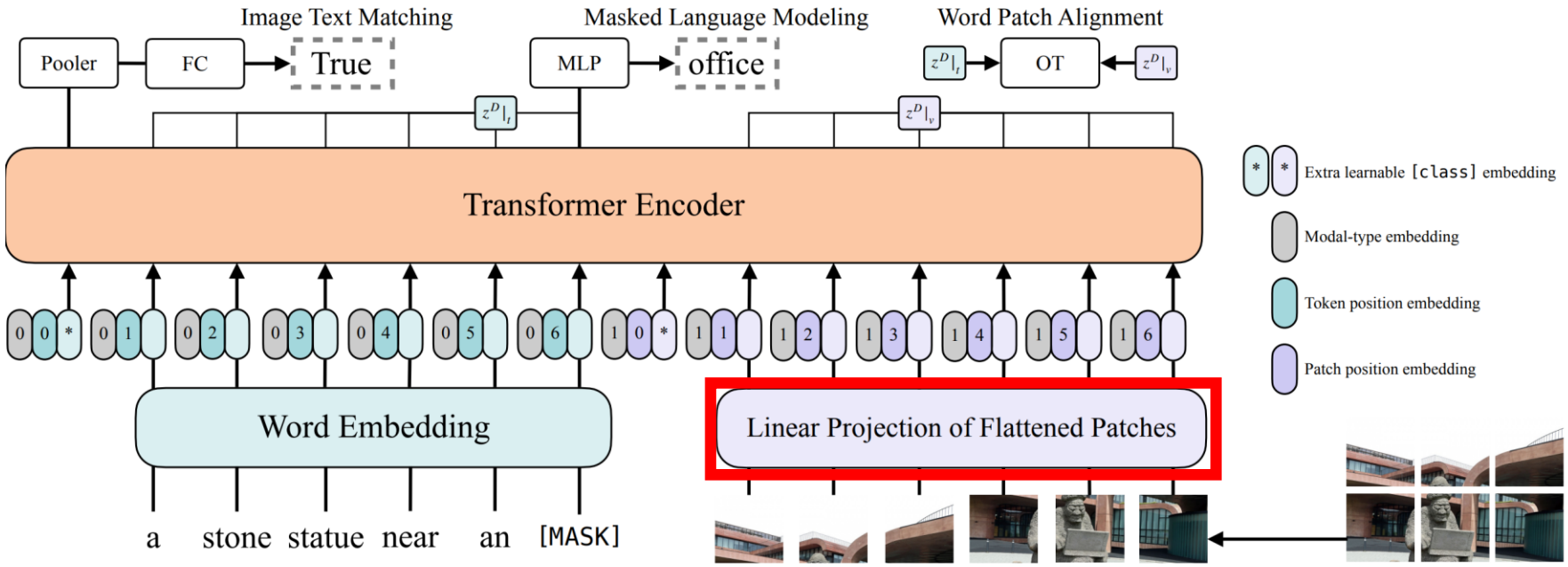


Figure 3. Model overview. Illustration inspired by Dosovitskiy et al. (2020).

- *ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision*, 2021

Sparse feature – Dense feature

- ViT for light fusion
- Patch for deep fusion

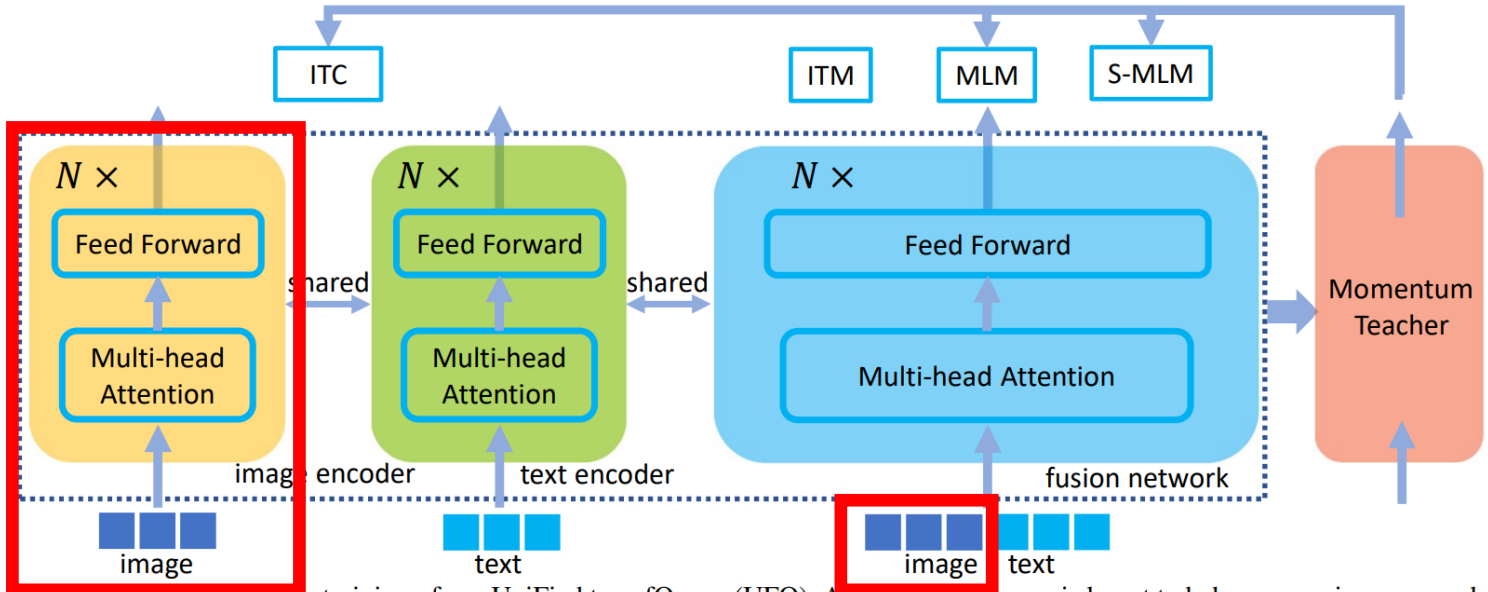
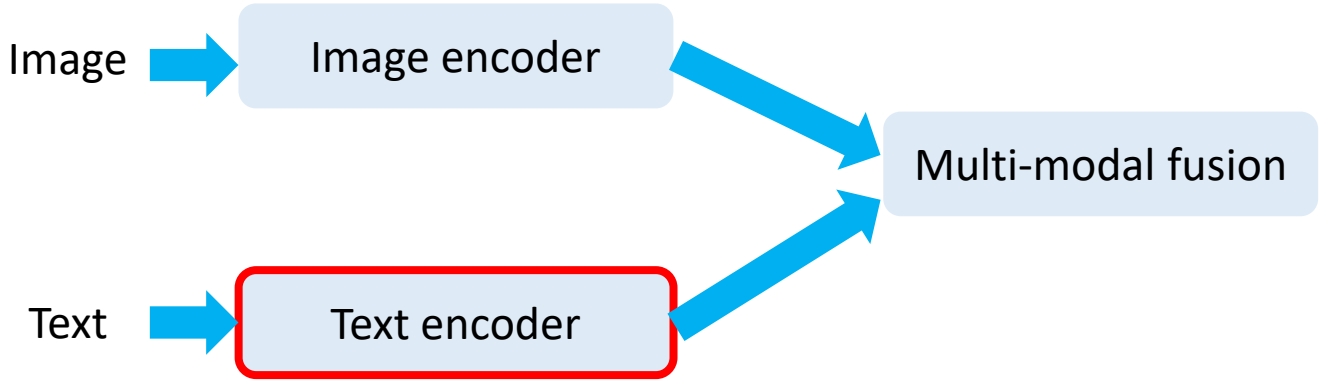


Figure 2. Vision-language pre-training of our UniFied transfOrmer (UFO). A single transformer is learnt to behave as an image encoder, a text encoder and a fusion network. The pre-training losses include the image-text contrastive (ITC) loss, image-text matching (ITM) loss, masked language modeling loss based on the bidirectional (MLM) and `seq2seq` attention mask (S-MLM). ITC empowers the network to understand the unimodal inputs (image or text), while the rest three focus on the joint inputs. In each iteration, one of the losses is randomly selected and is guided by a momentum teacher if the loss is ITC/MLM/S-MLM.

- *UFO: A UniFied TransfOrmer for Vision-Language Representation Learning*, 2021

Network architecture

- Image encoder, Text encoder, Multi-modal fusion



Text encoder - Embedding

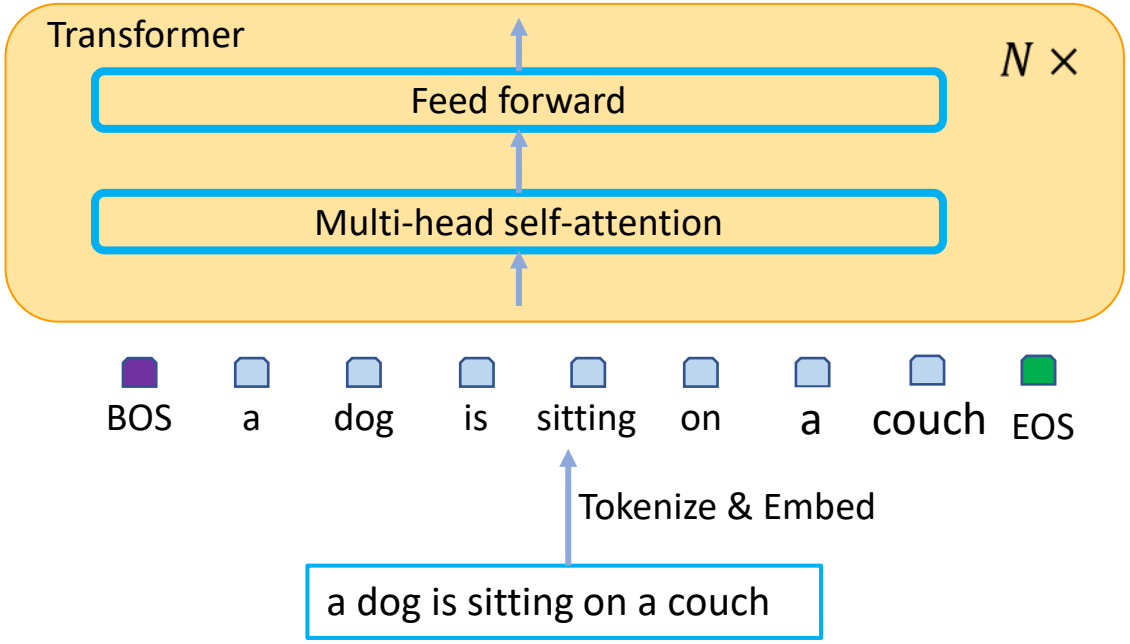
- Tokenize
  - Input: string
  - Output: $x_i \in \{0,1,\dots,T-1\},\ i \in \{0,1,\dots N\}$
    - N: number of tokens
    - T: vocabulary size
- Embedding
  - Input: $x_i \in \{0,1,\dots,T-1\}$
    - Token index
  - Output: $y_i \in \mathrm{R}^{\mathrm{D}}$
    - D: embedding dimension
    - lookup table
- Position embedding

a dog is sitting on a couch → (tokenize) → 1037, 3899, 2003, 3564, 2006, 1037, 6411 → (embed) → $\mathrm{R}^{N\times D}$

- (UNITER, 2019), (OSCAR, 2020), (VinVL, 2021), (MiniVLM, 2021), (ViLT, 2021), (UFO, 2021), (ViTCap, 2021), (LEMON, 2021), (GIT, 2022), (Flamingo, 2022)

Text encoder - Transformer

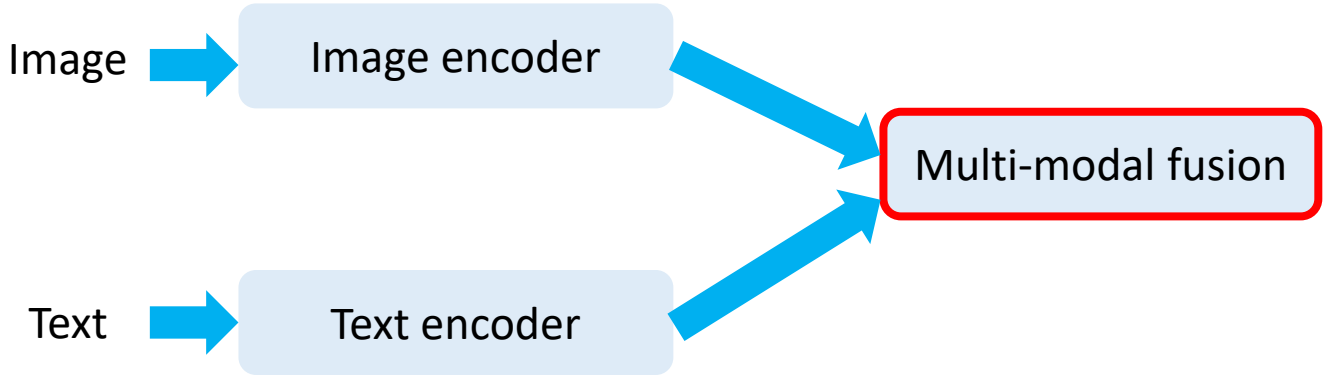
- Transformer
  - self-attention
  - Pretrained
    - BERT, RoBERT



- *LXMERT: Learning Cross-Modality Encoder Representations from Transformers*, 2019
- *An Empirical Study of Training End-to-End Vision-and-Language Transformers*, 2021
- *Align before Fuse: Vision and Language Representation Learning with Momentum Distillation*, 2021

Network architecture

- Image encoder, Text encoder, Multi-modal fusion



Multi-modal fusion – Transformer encoder

- Input concatenation, self-attention, modality-unaware

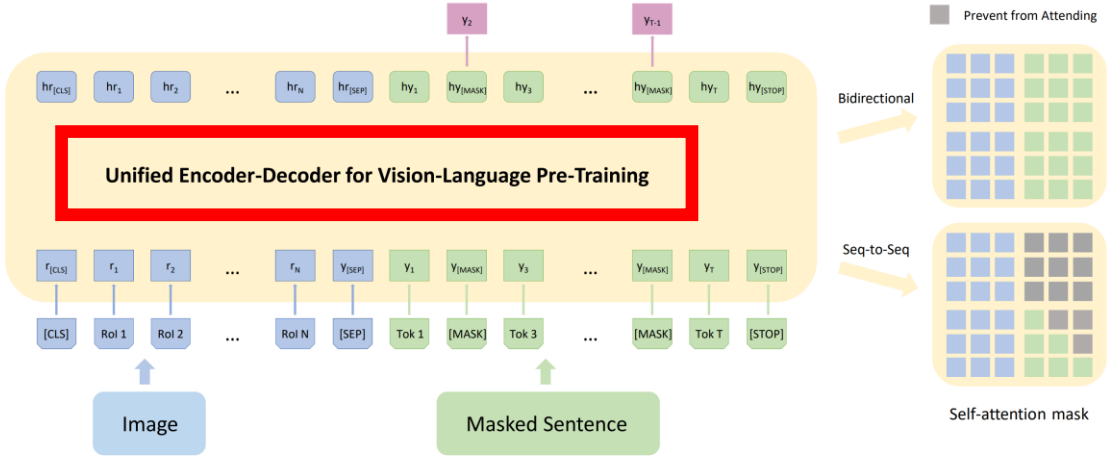


Figure 2: Model architecture for pre-training. The input comprises of image input, sentence input, and three special tokens ([CLS], [SEP], [STOP]). The image is processed as N Region of Interests (RoIs) and region features are extracted according to Eq. 1. The sentence is tokenized and masked with [MASK] tokens for the later masked language modeling task. Our Unified Encoder-Decoder consists of 12 layers of Transformer blocks, each having a masked self-attention layer and feed-forward module, where the self-attention mask controls what input context the prediction conditions on. We implemented two self-attention masks depending on whether the objective is bidirectional or seq2seq. Better viewed in color.

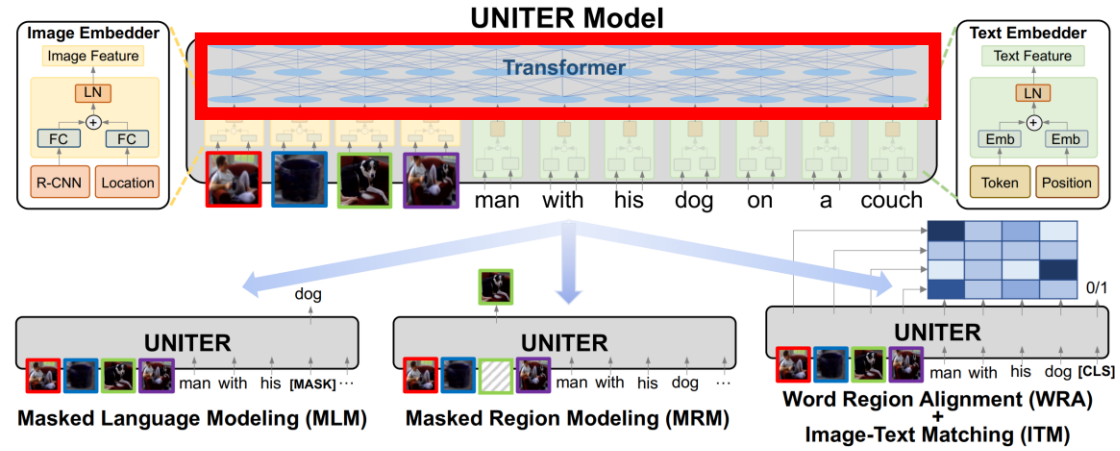


Fig. 1: Overview of the proposed UNITER model (best viewed in color), consisting of an Image Embedder, a Text Embedder and a multi-layer Transformer, learned through four pre-training tasks

- *Unified Vision-Language Pre-Training for Image Captioning and VQA*, 2019
- *UNITER: UNiversal Image-TExt Representation Learning*, 2019

Multi-modal fusion – Transformer encoder

- Self-attention, cross-attention, modality-aware

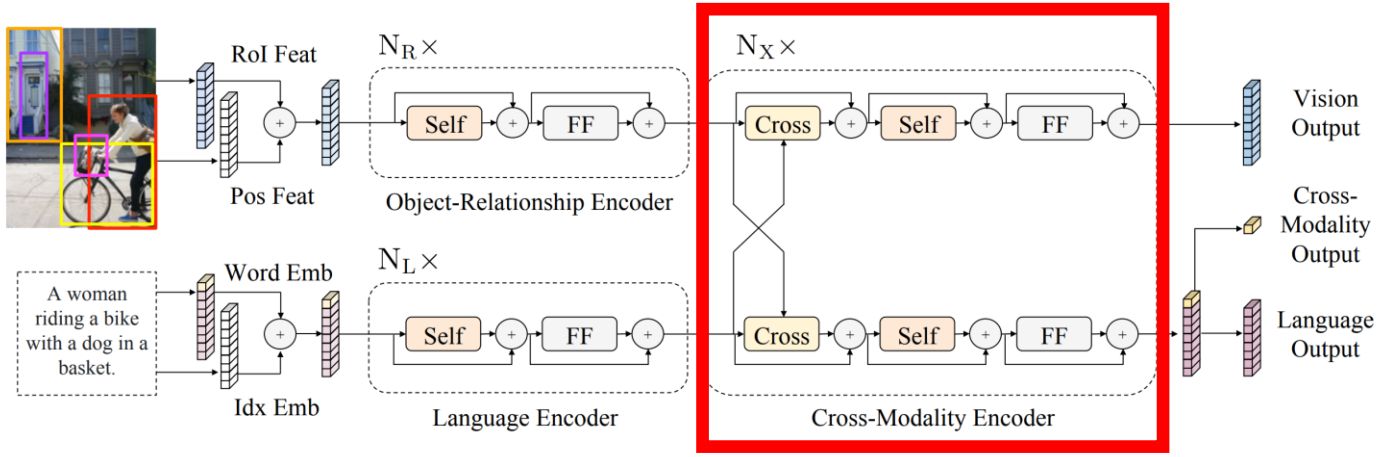


Figure 1: The LXMERT model for learning vision-and-language cross-modality representations. 'Self' and 'Cross' are abbreviations for self-attention sub-layers and cross-attention sub-layers, respectively. 'FF' denotes a feed-forward sub-layer.

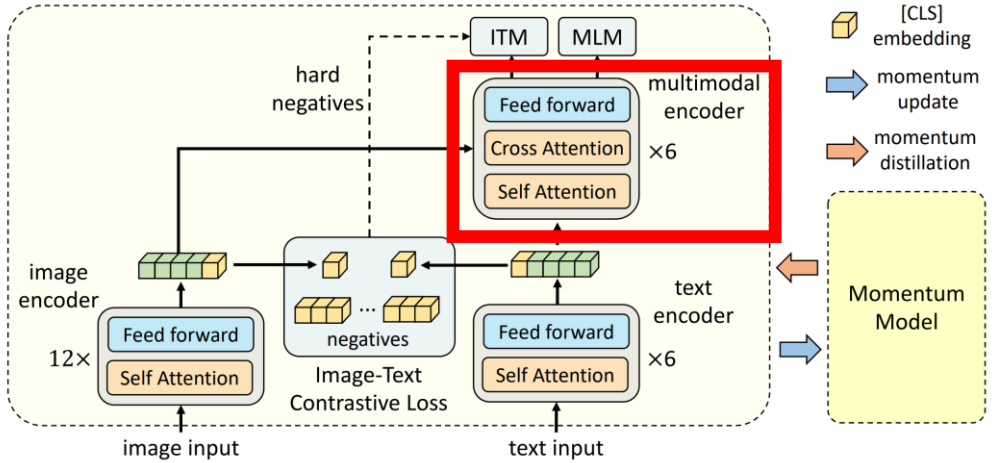


Figure 1: **Illustration of ALBEF.** It consists of an image encoder, a text encoder, and a multimodal encoder. We propose an image-text contrastive loss to align the unimodal representations of an image-text pair before fusion. An image-text matching loss (using in-batch hard negatives mined through contrastive similarity) and a masked-language-modeling loss are applied to learn multimodal interactions between image and text. In order to improve learning with noisy data, we generate pseudo-targets using the momentum model (a moving-average version of the base model) as additional supervision during training.

- *LXMERT: Learning Cross-Modality Encoder Representations from Transformers*, 2019
- *Align before Fuse: Vision and Language Representation Learning with Momentum Distillation*, 2021

Multi-modal fusion – Transformer decoder

- Input concatenation, self-attention, modality-unaware
  - Decoder: text generation
  - Masked language modeling
    - Predict masked token
    - seq2seq

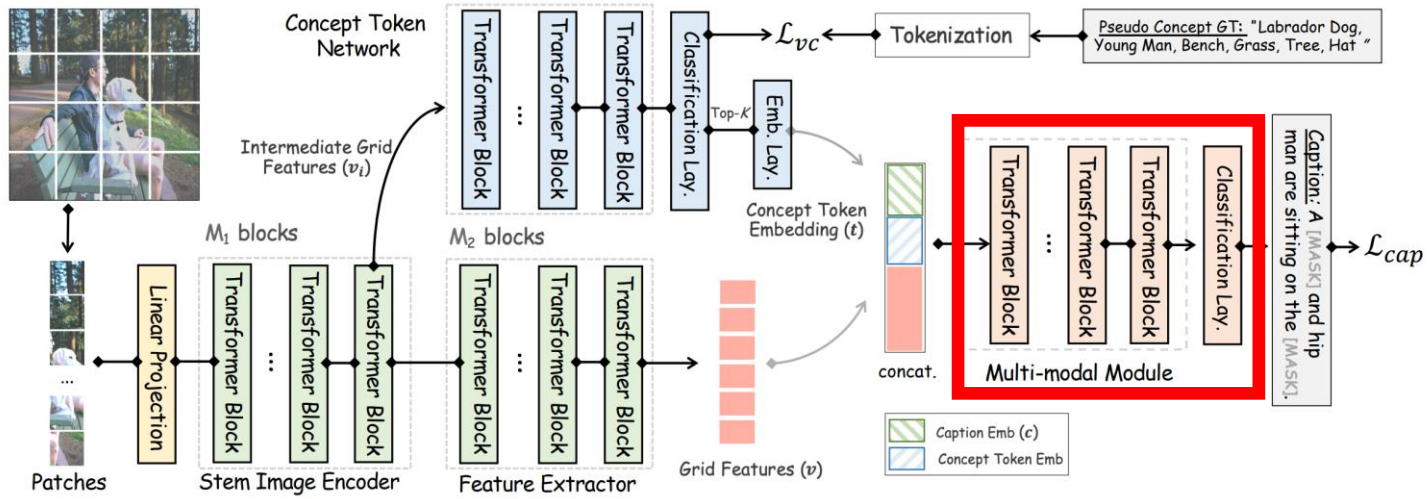


Figure 2. Architecture of our proposed ViTCAP image captioning model. ViTCAP is a detector-free image captioning model based on the vision transformer, where image patches are encoded into continuous embeddings as grid representations. The CTN branch roots from an intermediate block of the image encoder, and is a shallow transformer architecture (*e.g.*, 4 self-attention blocks). The CTN is trained via a classification task using object tags gleaned from the Teacher VLM's detector as pseudo-labels and the keywords parsed from image captions as the semantic concept ground-truth. During captioning, the CTN-produced concept tokens from the semantic concept vocabulary are then concatenated with the grid representations and fed into the multi-modal module for decoding. Best viewed in color.

- *Injecting Semantic Concepts into End-to-End Image Captioning*, 2021

Multi-modal fusion – Transformer decoder

- Input concatenation, self-attention, modality-unaware
  - Language modeling task
    - Predict next token

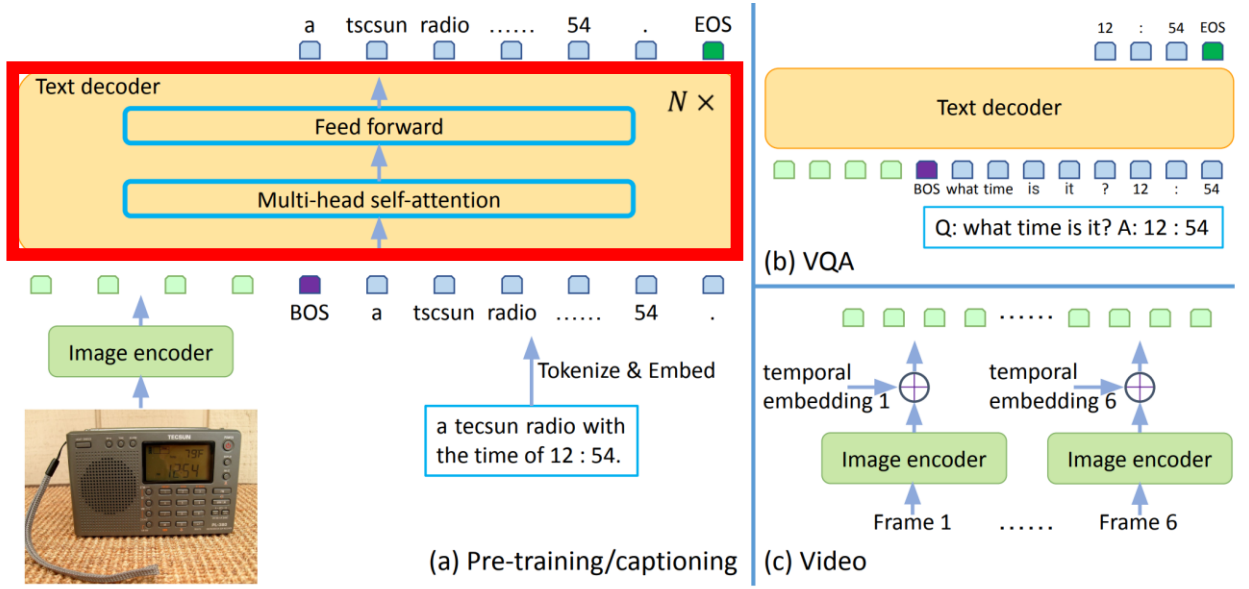


Figure 2: Network architecture of our GIT, composed of one image encoder and one text decoder. (a): The training task in both pre-training and captioning is the language modeling task to predict the associated description. (b): In VQA, the question is placed as the text prefix. (c): For video, multiple frames are sampled and encoded independently. The features are added with an extra learnable temporal embedding (initialized as 0) before concatenation.

- *GIT: A Generative Image-to-text Transformer for Vision and Language*, 2022

Multi-modal fusion – Transformer decoder

- Self-attention, cross-attention, modality-aware
  - Freeze decoder; randomly initialize cross-attention-based modules

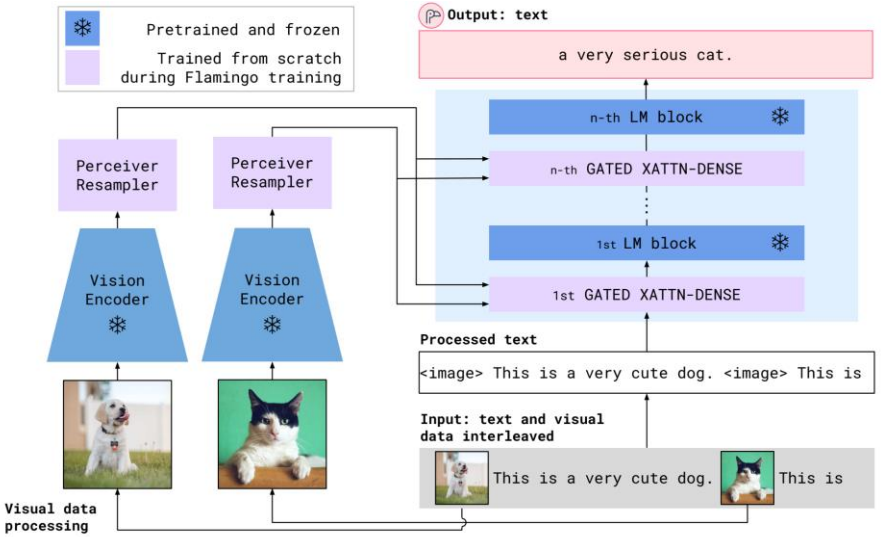
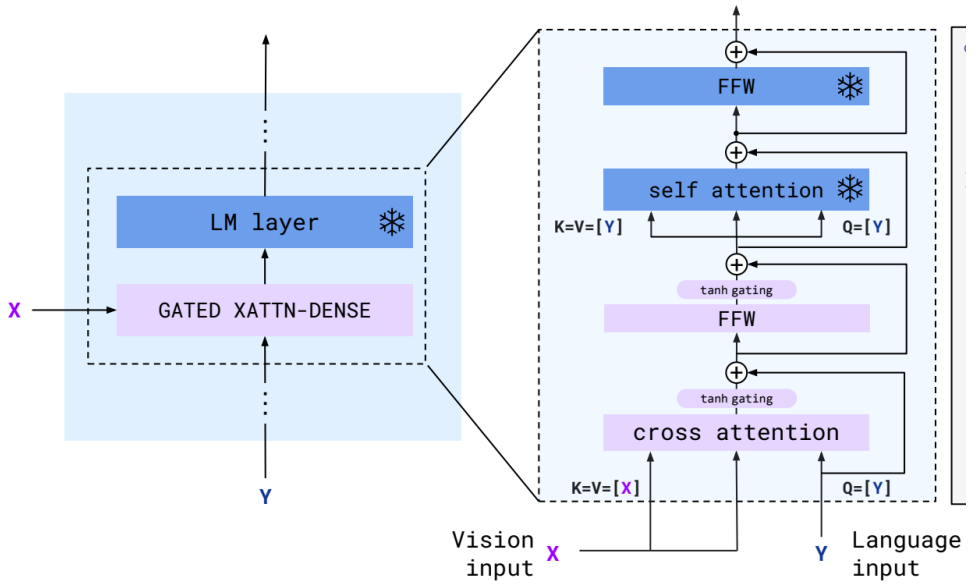


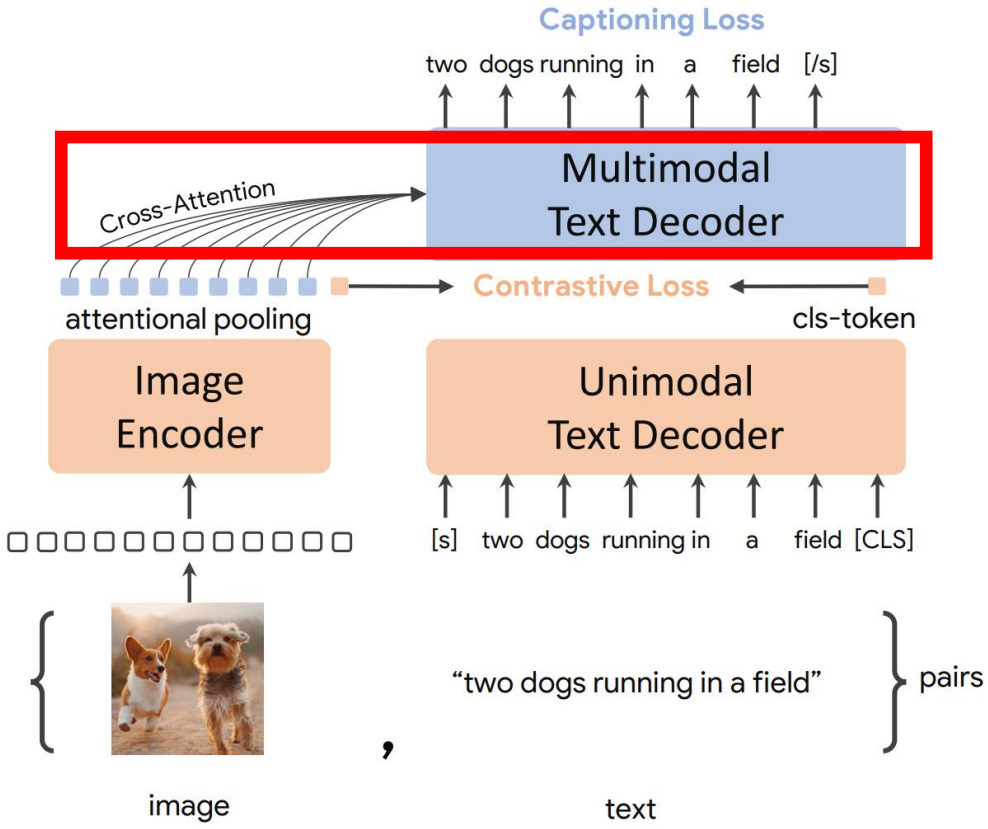
Figure 3 | **Overview of the Flamingo model.** The Flamingo models are a family of visual language model (VLM) that can take as input visual data interleaved with text and can produce free-form text as output. Key to its performance are novel architectural components and pretraining strategies described in Section 3.



- *Flamingo: a Visual Language Model for Few-Shot Learning*, 2022

Multi-modal fusion – Transformer decoder

- Self-attention, cross-attention, modality-aware
  - Random initialization



- *CoCa: Contrastive Captioners are Image-Text Foundation Models*, 2022

Multi-modal fusion – Transformer decoder

- Cross-attention-based vs self-attention-based

Table 23: Comparison between pure self-attention-based decoder and the cross-attention-based decoder under different amounts of pre-training data. No SCST is applied on captioning. No intermediate fine-tuning is applied for VQA.

		Captioning				Visual Question Answering		
data	Cross-Att.	COCO	nocaps	TextCaps	VizWiz	ST-VQA	TextVQA	VizWiz
0.8B	w/o	**144.2**	**120.3**	**143.7**	**107.2**	**65.3**	**58.5**	**59.1**
	w/	143.2	118.2	139.3	103.0	63.1	55.6	58.9
10M	w/o	**139.1**	75.4	92.7	**89.3**	40.9	33.0	51.8
	w/	138.1	**86.2**	**93.9**	88.5	**42.7**	**34.7**	**54.8**

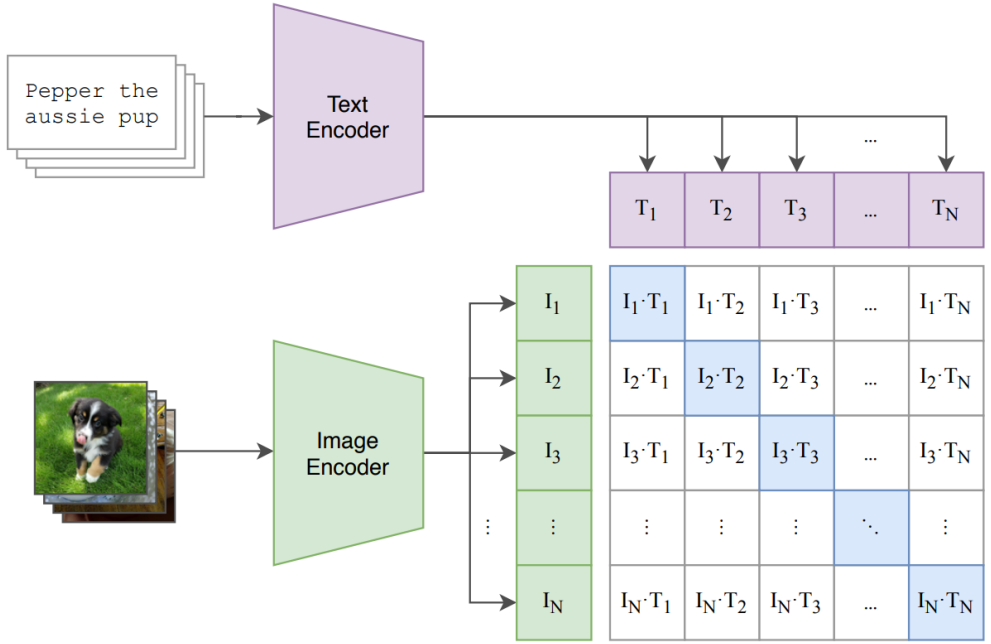
- *GIT: A Generative Image-to-text Transformer for Vision and Language*, 2022

Outline

- Application
  - Retrieval, captioning, question answering
- Network architecture
  - Image encoder, text encoder, multi-model fusion
- **Pre-training tasks**
  - ITC, MLM, ITM
- Adaptation to downstream tasks

Pre-training tasks

- Image-text contrastive (ITC) loss
  - l_2 normalization
    - cosine similarity
  - Temperature
    - Pre-set
    - Learnable
  - Retrieval task



$$l_i = -\log \frac{\exp \frac{I_i T_i}{t}}{\sum_j \exp \frac{I_i T_j}{t}} - \log \frac{\exp \frac{I_i T_i}{t}}{\sum_j \exp \frac{I_j T_i}{t}}$$

- *Learning Transferable Visual Models From Natural Language Supervision*, 2021
- *Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision*, 2021
- *Florence: A New Foundation Model for Computer Vision,* 2021
- *LightningDOT: Pre-training Visual-Semantic Embeddings for Real-Time Image-Text Retrieval*, 2021

Pre-training tasks

- Image-text contrastive (ITC) loss
  - Negative samples
    - In-batch
      - (SimCLR, 2020)
      - (UFO, 2021)
    - Momentum queue
      - (Moco, 2019)
      - (ALBEF, 2021)

Loss	VQA	ZS Flickr TR@1
$l_{\text{m-ITC}}$	70.49	61.6
l_{ITC}	71.39	68.7

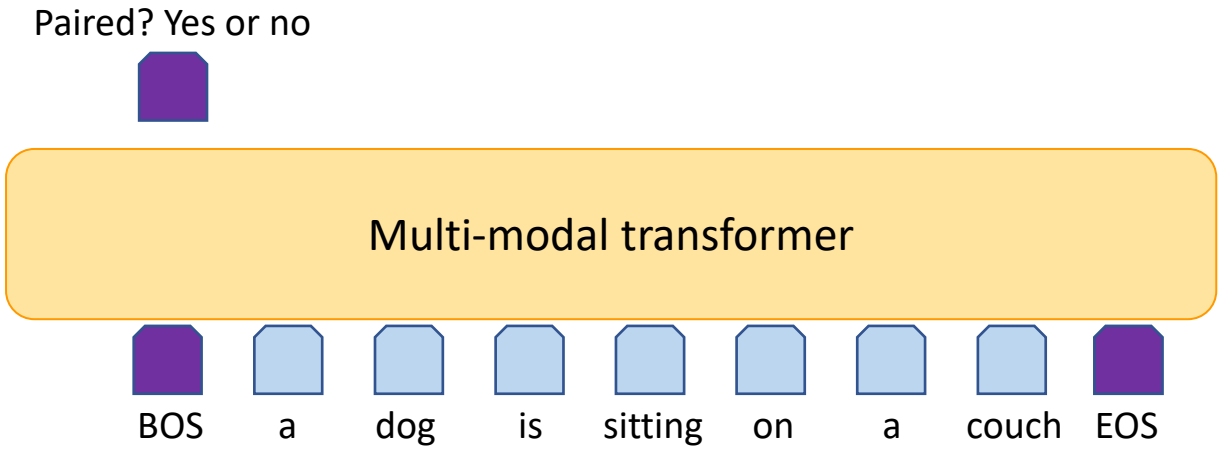
Table 17. Comparison between the in-batch image-text contrastive loss l_{ITC} and the momemtum-based image-text contrastive loss $l_{\text{m-ITC}}$.

- *A Simple Framework for Contrastive Learning of Visual Representations*, 2020
- *Momentum Contrast for Unsupervised Visual Representation Learning*, 2019
- *Align before Fuse: Vision and Language Representation Learning with Momentum Distillation*, 2021
- *UFO: A UniFied TransfOrmer for Vision-Language Representation Learning*, 2021

Pre-training tasks

- Image-text matching (ITM) loss
  - Input
    - (image, paired text) or (image, unpaired text)
  - Output
    - Binary classifier
      - Paired or not
  - VQA (Yes/no question), Retrieval

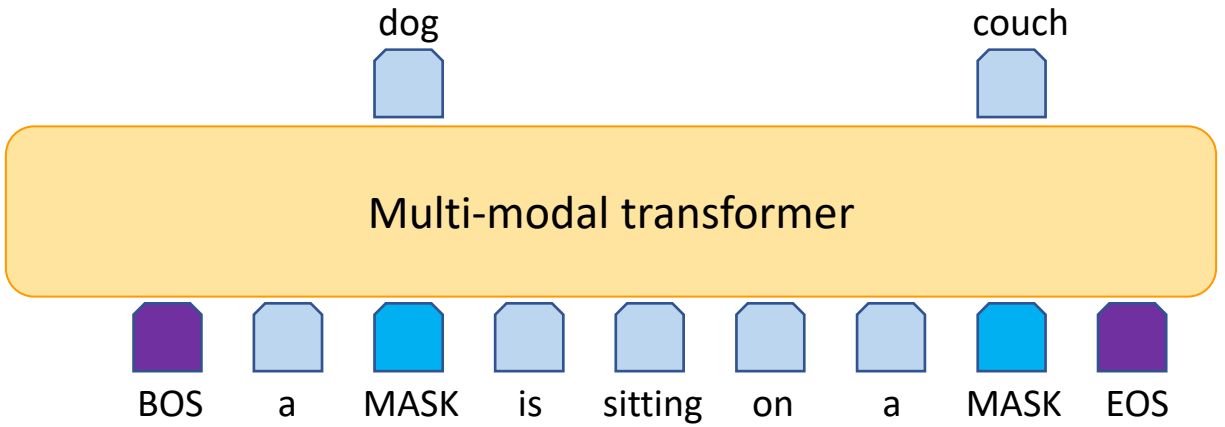
$$-\log p(\text{yes}|\text{paired}) \; -\log p(\text{no}|\text{unpaired})$$



- (LXMERT, 2019), (UNITER, 2019), (OSCAR, 2020), (VinVL, 2021), (MiniVLM, 2021), (ViLT, 2021), (ALBEF, 2021), (UFO, 2021), (VLMO, 2021), (METER, 2021), etc.

Pre-training tasks

- Masked language modeling (MLM) loss
  - Attention mask
    - Bidirectional
    - Unidirectional
  - Captioning/VQA

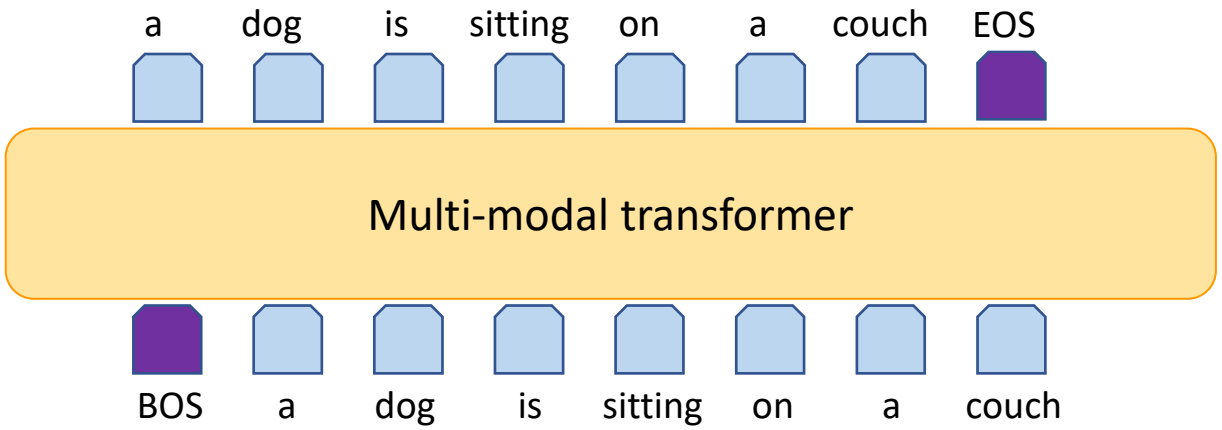


- (LXMETERT, 2019), (UNITER, 2019), (OSCAR, 2020), (VinVL, 2021), (MiniVLM, 2021), (ViLT, 2021), (UFO, 2021), (ViTCap, 2021), (VLMO, 2021), (METER, 2021), (LEMON, 2021), …

Pre-training tasks

- Language modeling (LM)
  - Predict next token
  - Captioning/VQA

$$-\sum_i \log p\left(y_{i+1} \middle| I, y_j, j \leq i\right)$$



- (SimVLM, 2021), (CoCa, 2022), (Flamingo, 2022), (GIT, 2022), (BLIP, 2022), (OFA, 2022)

Pre-training tasks

- MLM vs LM
  - MLM
    - Learn 15% tokens in each iteration
    - Higher performance with enough training cost
  - LM
    - Learn 100% tokens in each iteration
    - Efficient in training
    - Large-scale dataset/model

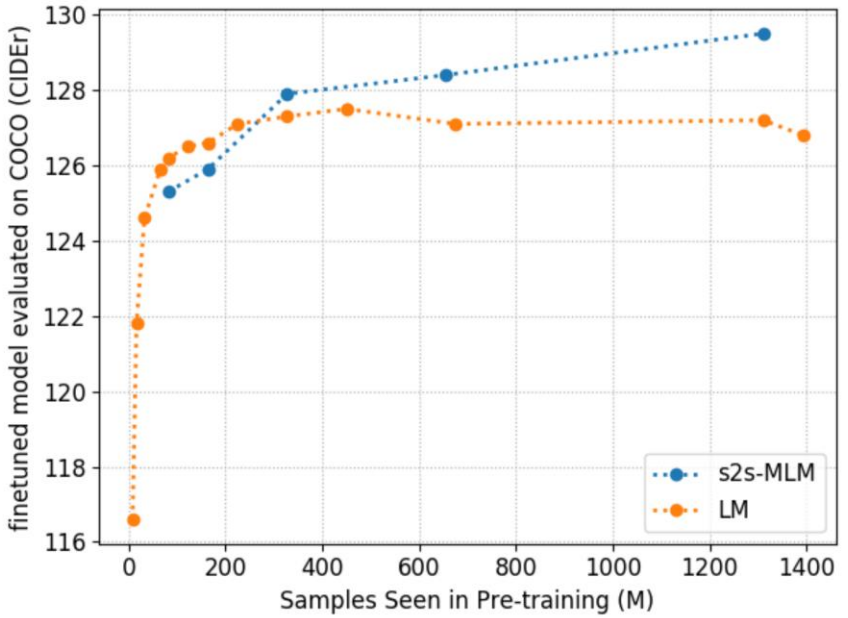


Figure 7. **Comparison of different training objectives** by pre-training on CC12M and finetuning on COCO. The models are fine-tuned from intermediate checkpoints using the same objective as used in pre-training.

- *Scaling Up Vision-Language Pre-training for Image Captioning*, 2021

Adaptation to downstream tasks

- Retrieval
  - Key
    - Evaluate similarity between image and text
  - Fine-tuning consistent with pre-training
    - Pretrained with image-text contrastive loss
      - Inner product to calculate the similarity
    - Pretrained with image-text matching loss
      - Feed forward to calculate the similarity
  - No new parameters/modules
  - Evaluation set
    - COCO and Flickr30K

Adaptation to downstream tasks

- Image captioning
  - Decode tokens autoregressively
  - Pretrained vs fine-tuning
    - Pretrained with bidirectional attention mask in MLM
      - Fine-tuning with seq2seq attention mask
    - Pretrained with seq2seq attention mask in MLM
      - Consistent
    - Pretrained with language modeling task
      - Consistent
  - Evaluation set
    - COCO, nocaps, TextCaps, VizWiz-Captions

Adaptation to downstream tasks

- Visual question answering
  - As classification task over answer candidates
    - Fine-tuning with extra modules, randomly initialized
  - As text generation task
    - Consistent with pre-training
    - Open-vocabulary answer
  - Evaluation set
    - VQAv2, TextVQA, ST-VQA, VizWiz-QA, OCR-VQA, OK-VQA, AVQA, AdVQA

Take-away messages

- A simple approach to study research paper
  - Network
    - Image encoder? Text decoder? Modality fusion?
  - Pre-training task
    - ITC, ITM, MLM, LM?
  - Adaptation to downstream task
    - How?

- Image encoder
  - Sparse features with object detector --> dense feature?
    - Eliminate the bounding box annotation

- Pre-training task
  - Multi-tasks with MLM, ITM, ITC --> Fewer tasks with ITC, LM?
    - Reduce the gap between pre-training and downstream tasks