Cross-Lingual (Visual) Language Models on Understanding Physical Concepts

Yihong Liu, Shengqiang Zhang

Center for Information & Language Processing (CIS), LMU Munich

October 8, 2024

Outline

1. Why do we need Vision-Language Models?
2. Vision Transformer
3. An Introduction to Vision-Language Models
   - Contrastive-based VLMs
   - VLMs with masking objectives
   - Generative-based VLMs
   - VLMs from pretrained backbones
4. How to choose from these four types of VLMs?

From Large Language Models to Vision Language Models

Why do we need VLMs?

VLMs are helpful even if in scenarios where vision is not necessary.

- LLMs have difficulties in understanding physical concepts and embodied concepts, such as size, temperature. [1]
- Prior study shows that visual grounding helps learn word meanings in low-data regimes. [2]
- ...

In scenarios where vision is necessary, VLMs are even more important.

- Transforming vision into language would cause information loss.
- ...

[1] Li, Lei, et al. "Can Language Models Understand Physical Concepts?." EMNLP 2023.

[2] Zhuang et al. "Visual Grounding Helps Learn Word Meanings in Low-Data Regimes." NAACL 2024.

World Scopes

"Experience Grounds Language" [3] defines five levels of World Scope:

1. Corpora
2. Internet
3. Perception
4. Embodiment
5. Social

[3] Bisk, Yonatan, et al. "Experience Grounds Language." EMNLP 2020.

World Scopes

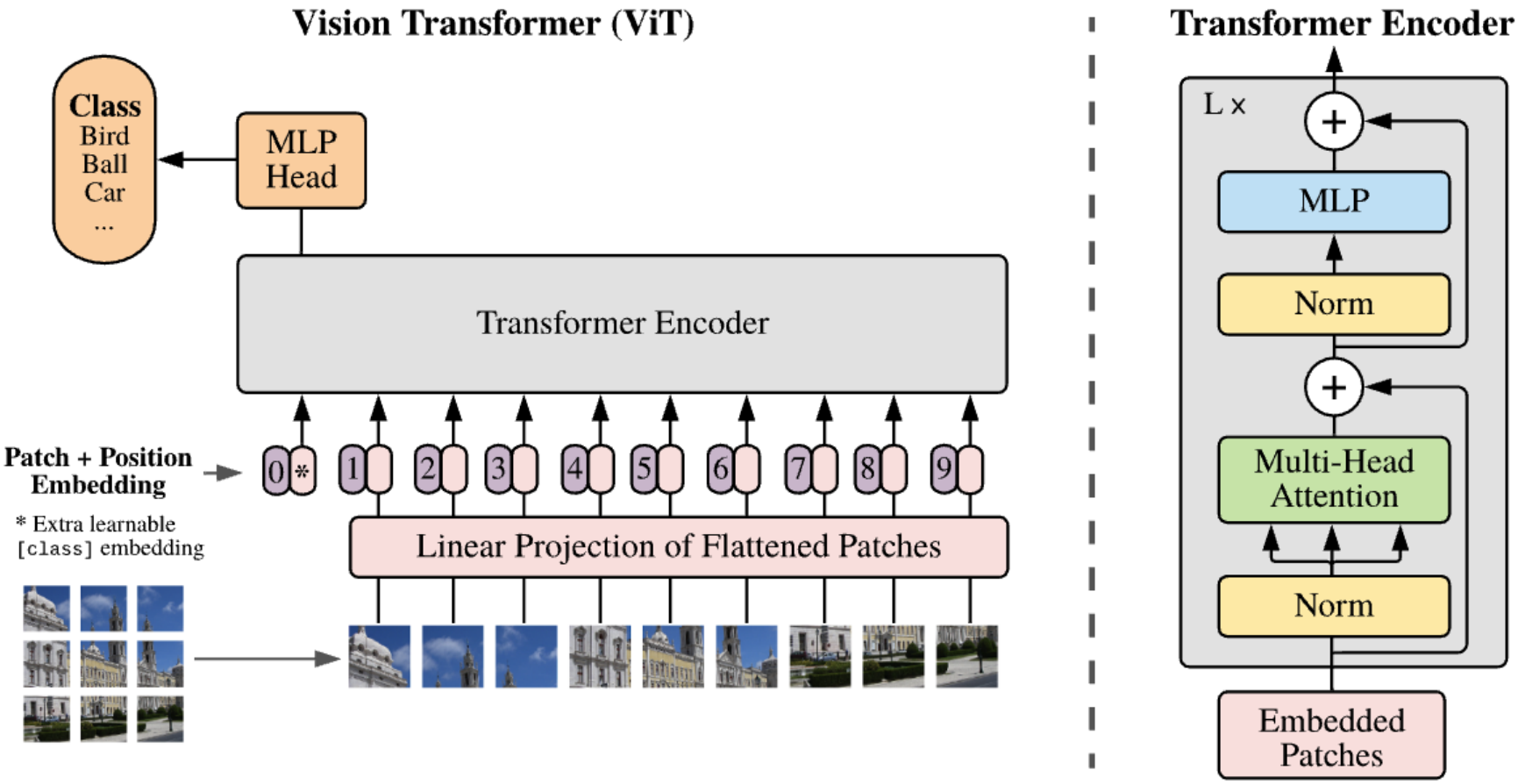
"Experience Grounds Language" [4] defines five levels of World Scope:

1. Corpora (our past)
2. Internet (most of current NLP)
3. Perception (multimodal NLP)
4. Embodiment
5. Social

[4] Bisk, Yonatan, et al. "Experience Grounds Language." EMNLP 2020.

Vision Transformer

Vision Transformer (ViT) Architecture



An image $x \in \mathbb{R}^{H \times W \times C} \rightarrow x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where $N = H \times W / P^2$.

$\mathbf{z}_0 = [\mathbf{x}_{\text{cls}}; \mathbf{x}_p^1 \mathbf{E}; \ldots; \mathbf{x}_p^N \mathbf{E};] + \mathbf{E}_{\text{pos}}$, where $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times d}$, $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times d}$.

ViT Variants

Supervised pre-training on 14M-300M images.

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Notation

- ViT-L/16: ViT "Large" variant with 16x16 input patch size.

An Introduction to Vision-Language Models

VLM Usage Examples

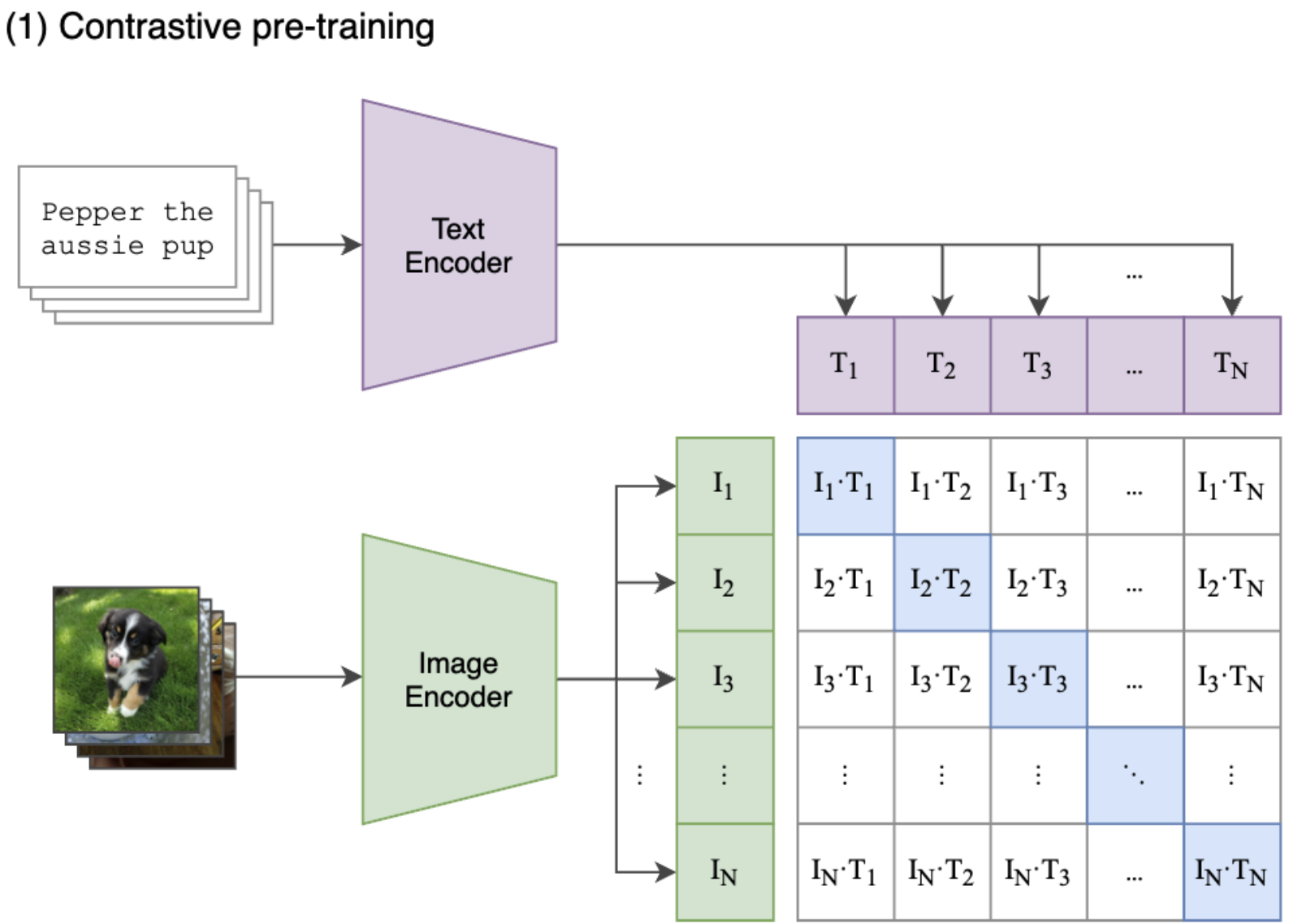
VLM Categories

VLMs can be classified into four categories according to the training methods:

- Contrastive-based VLMs
- VLMs with masking objectives
- Generative-based VLMs
- VLMs from pretrained backbones

Contrastive-based VLMs: OpenAI's CLIP

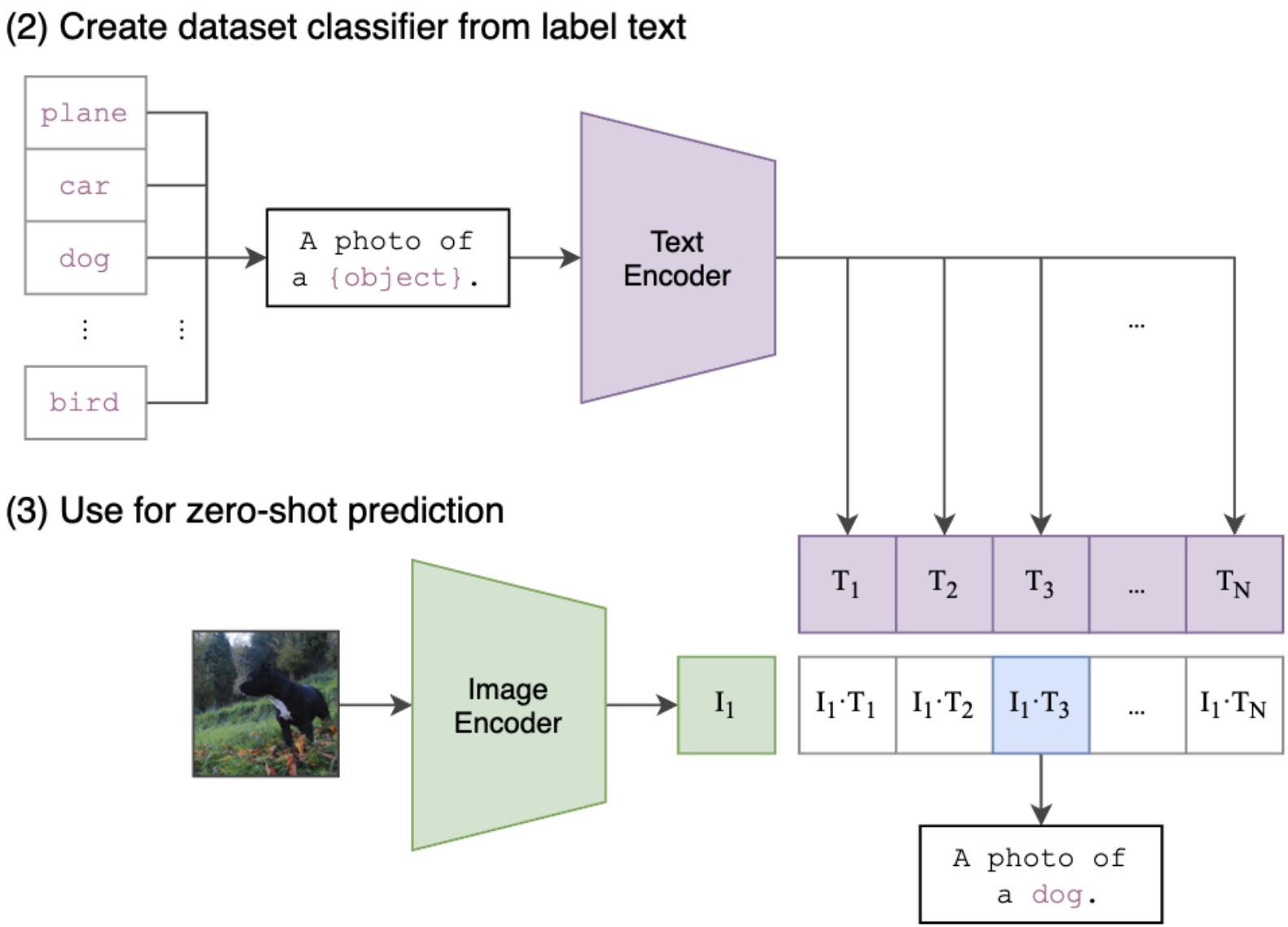
CLIP Pre-training



CLIP Pre-training Implementation Pseudocode

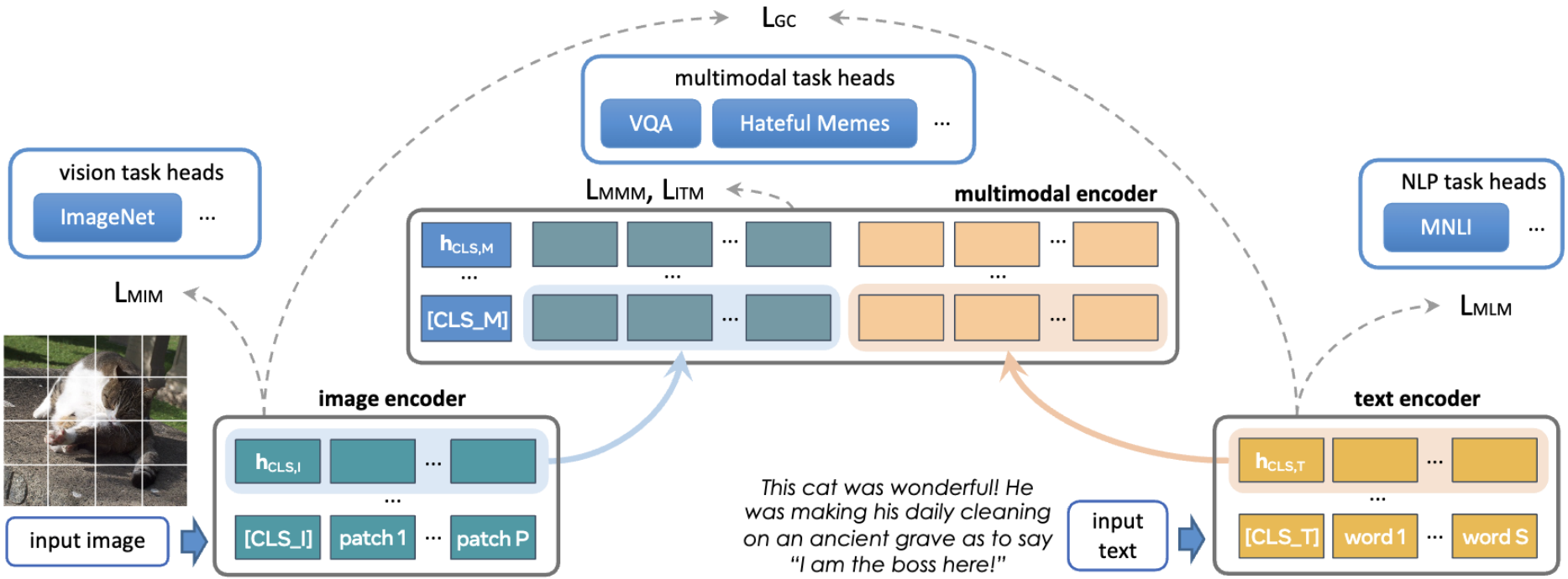
```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

CLIP Inference



VLMs with Masking Objectives: FLAVA

FLAVA Architecture



Pre-training

Unimodal pre-training:

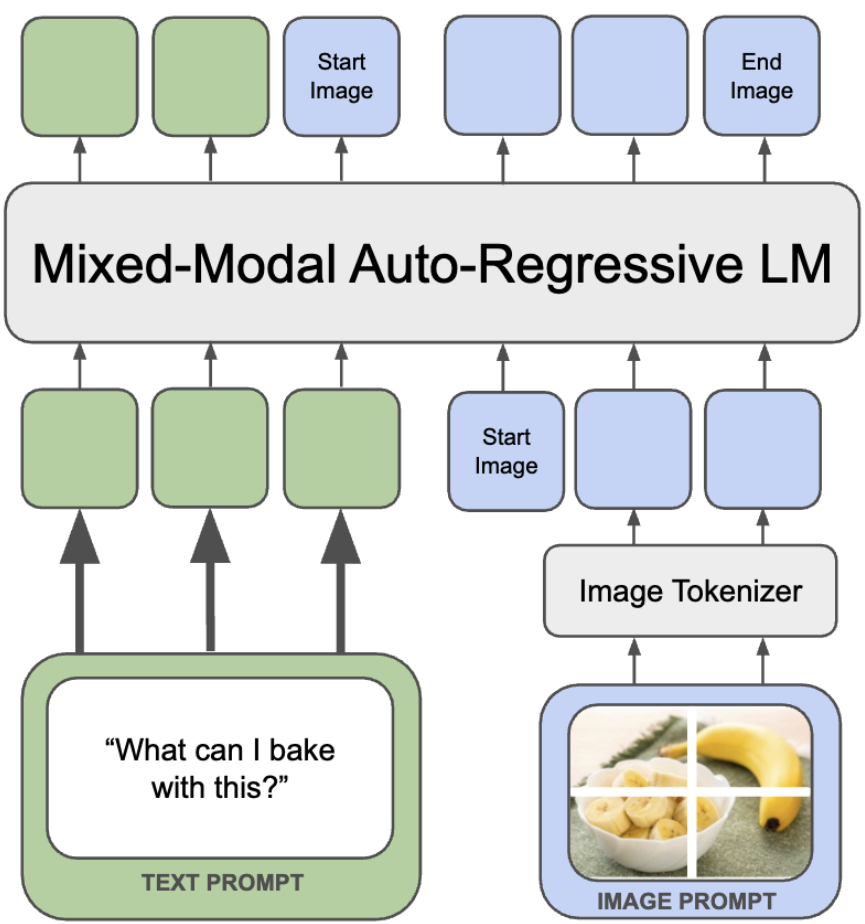
- Masked Image Modeling (MIM)
- Masked Language Modeling (MLM)

Multimodal pre-training:

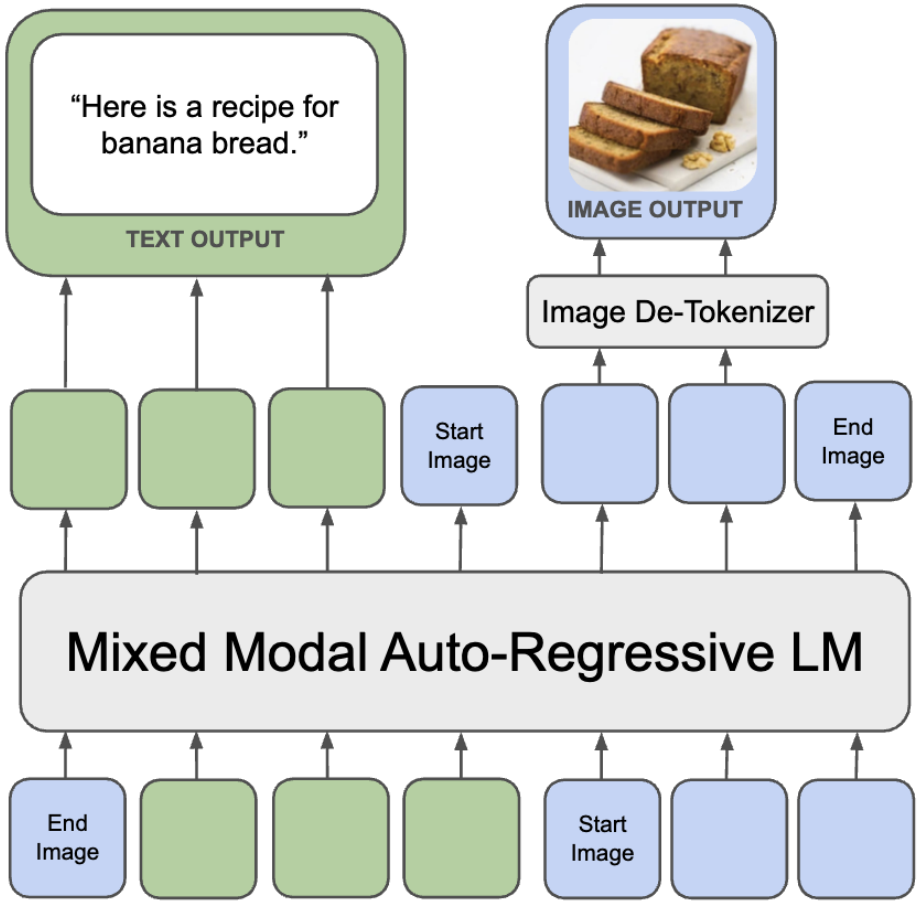
- Global contrastive loss
- Masked Multimodal Modeling (MMM)
- Image Text Matching (ITM)

Generative-based VLMs: Chameleon

Chameleon Architecture



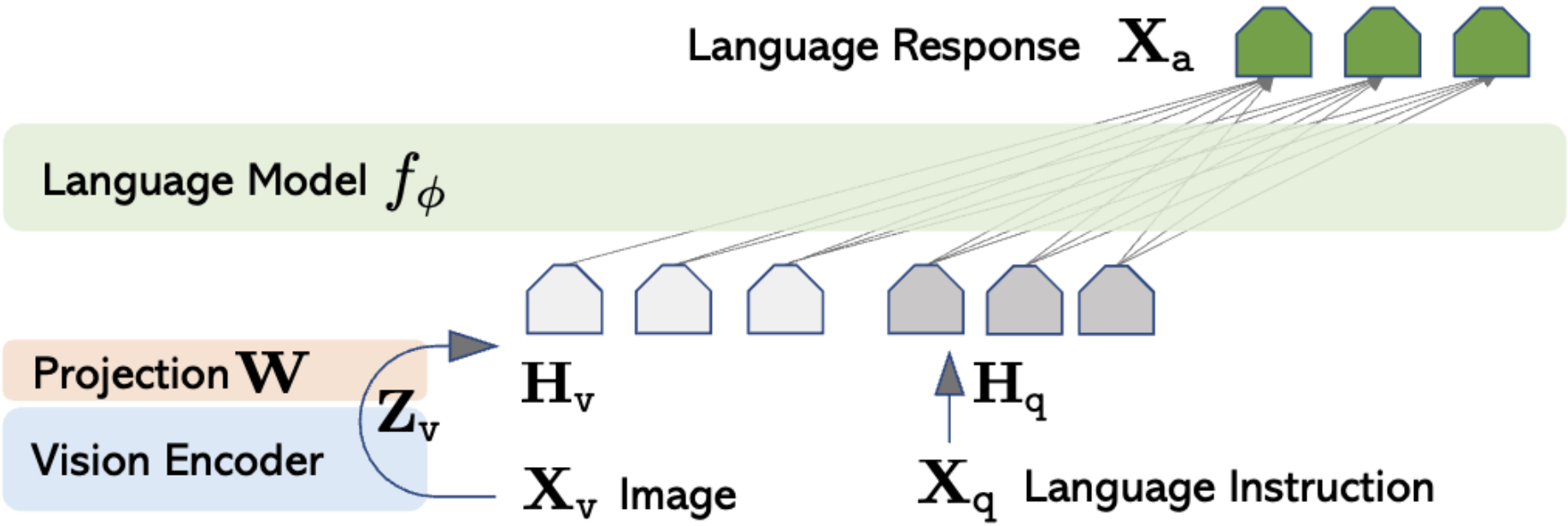
(a) Mixed-Modal Pre-Training



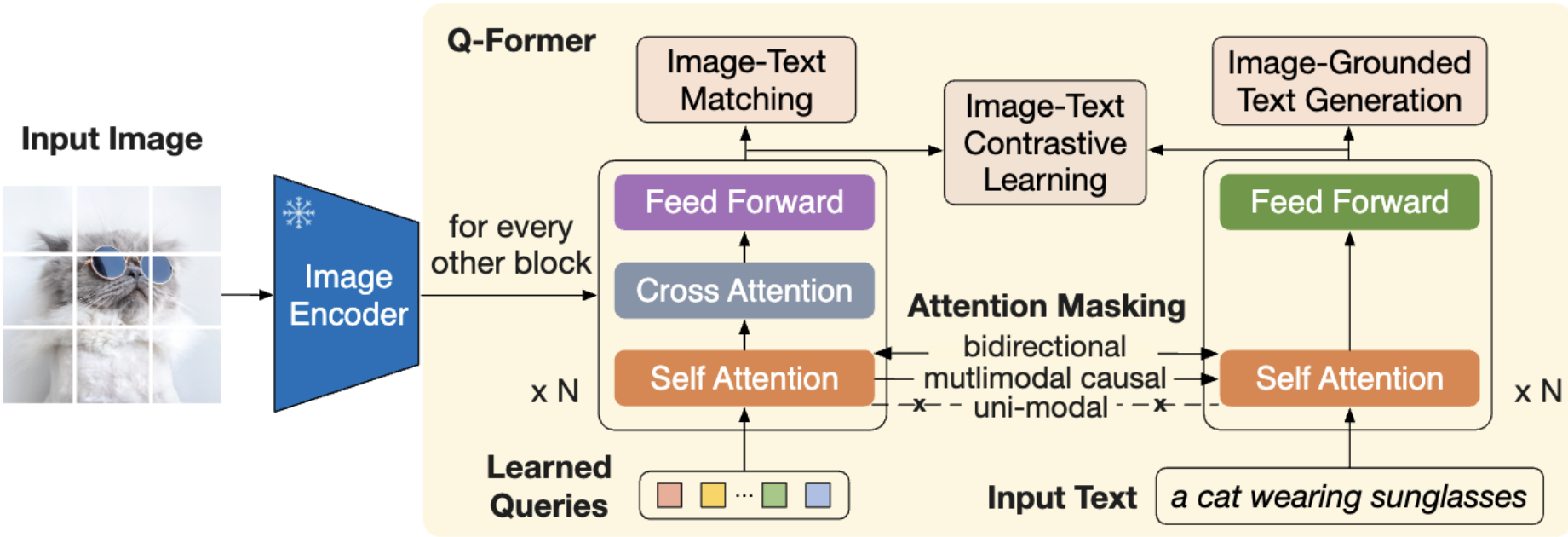
(b) Mixed-Modal Generation

VLMs from Pretrained Backbones: LLaVA, BLIP-2, MolMo

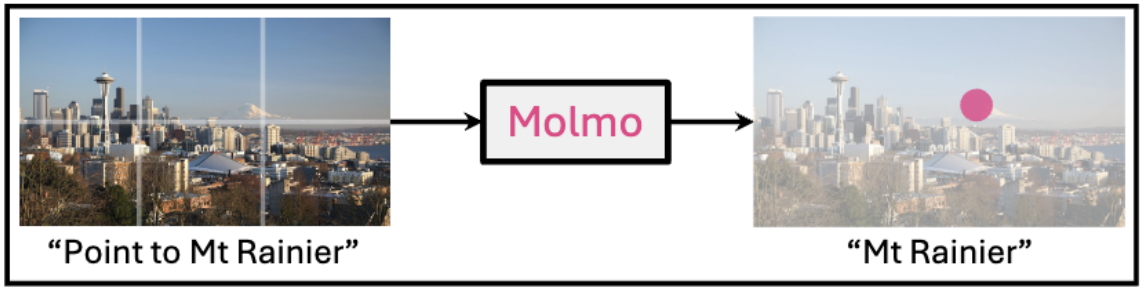
LLaVA

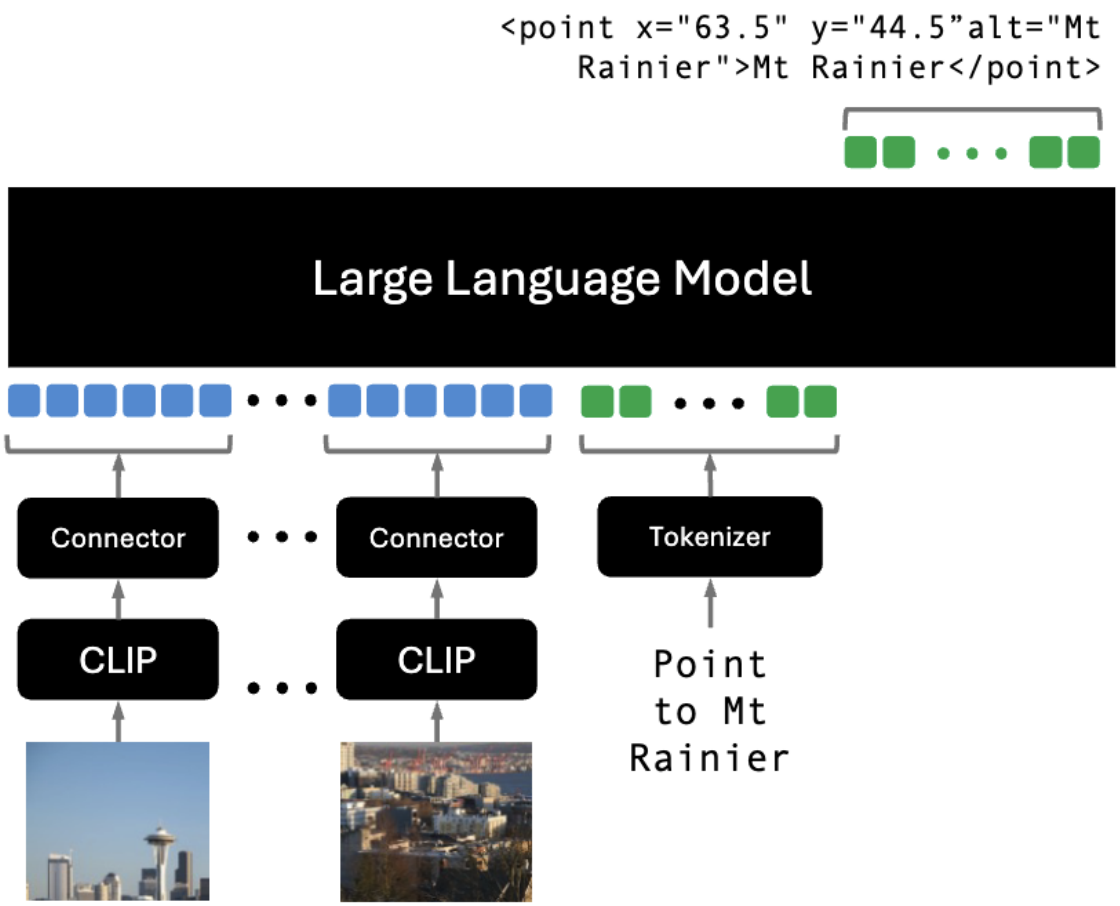

Language Response $\mathbf{X}_a$
Language Model f_ϕ
Projection $\mathbf{W}$
Vision Encoder
$\mathbf{Z}_v$
$\mathbf{H}_v$
$\mathbf{H}_q$
$\mathbf{X}_v$ Image
$\mathbf{X}_q$ Language Instruction

BLIP-2


Input Image
Image Encoder
for every other block
Q-Former
Image-Text Matching
Image-Text Contrastive Learning
Image-Grounded Text Generation
Feed Forward
Cross Attention
Self Attention
x N
Attention Masking
bidirectional
mutlimodal causal
uni-modal
Feed Forward
Self Attention
x N
Learned Queries
Input Text
a cat wearing sunglasses

MolMo


Molmo
"Point to Mt Rainier"
"Mt Rainier"


<point x="63.5" y="44.5"alt="Mt Rainier">Mt Rainier</point>
Large Language Model
Connector
Connector
Tokenizer
CLIP
CLIP
Point to Mt Rainier

How To Choose From These Four Types of VLMs?

Which model to use?

VLMs with pre-trained backbones

- For most people who have access to limited resources.
- Only the mapping between vision modality and language modality should be learned.
- VLMs are impacted by the hallucination problem of LLMs.
- It's not clear which design is better, using separate encoders for vision and language is better or learning the distribution of language and vision jointly?

Which model to use?

Contrastive-based VLMs such as CLIP

- CLIP is not a generative model.
- CLIP learns representations that have both meaning in the image and text space, which makes it possible to prompt the CLIP text encoder with words such that we can retrieve the images that map to the corresponding text representations.
- Many data curation pipelines use CLIP.
- CLIP is a good base for building more complex models.

Which model to use?

VLMs with masking objectives

- By learning to reconstruct data from both masked images and text, it is possible to jointly model their distributions.
- Models based on masking might need to leverage a decoder to map back the representation to the input space.
- No batch dependency any more.
- Many VLM methods leverage a mix of masking strategies along with some contrastive loss.

Which model to use?

Generative-based VLMs

- Generative models based on diffusion or autoregressive criteria have demonstrated impressive abilities in generating photorealistic images based on text prompt.
- Most large-scale training efforts on VLM are also starting to integrate image generation components.
- It might be easier to understand and assess what the model has learned when it is able to decode abstract representations in the input data space.
- While models like CLIP would need extensive k-NN evaluations using millions of image data points to show what the images closest to a given word embedding look like, generative models can just output the most probable image directly without such an expensive pipeline.
- They are more computationally expensive to train.