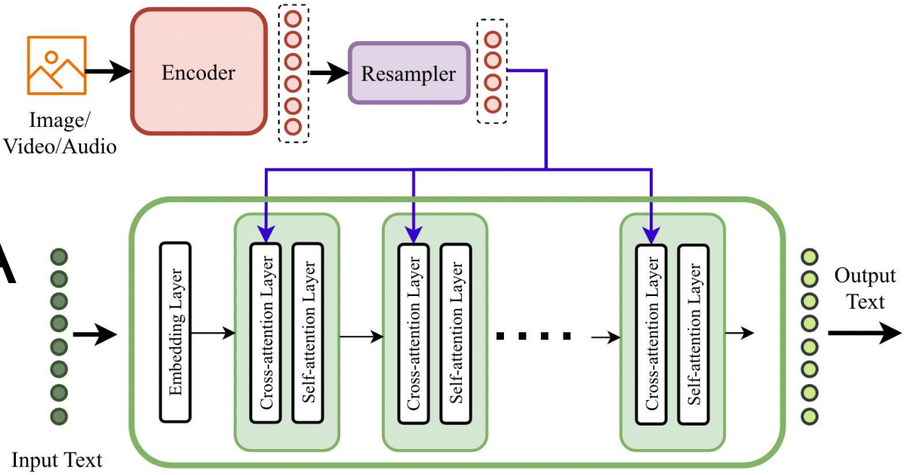
中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

经典多模态模型分类及细粒度CLIP变体

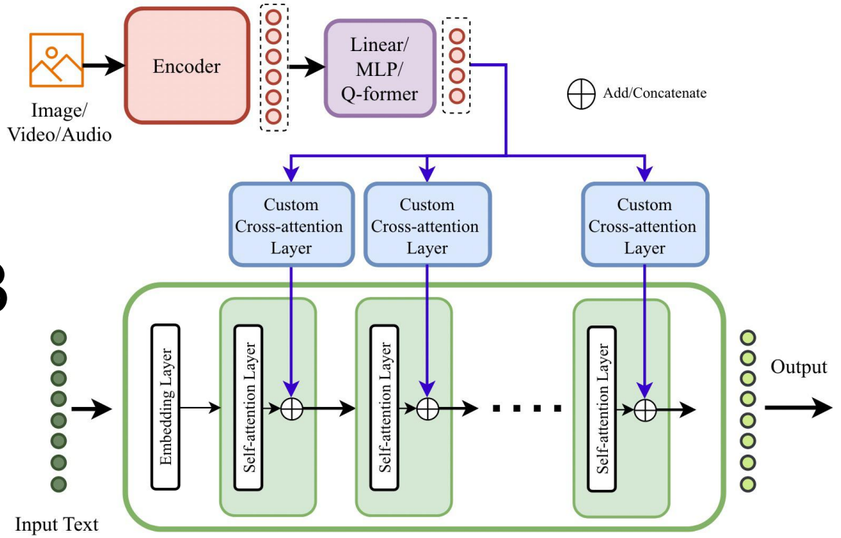
陈佳含 2025.05.28

Overview

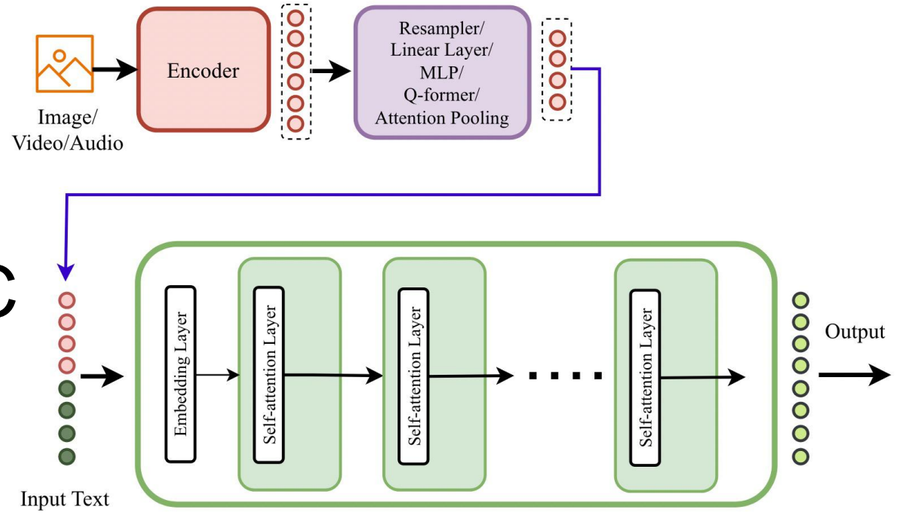
Type A



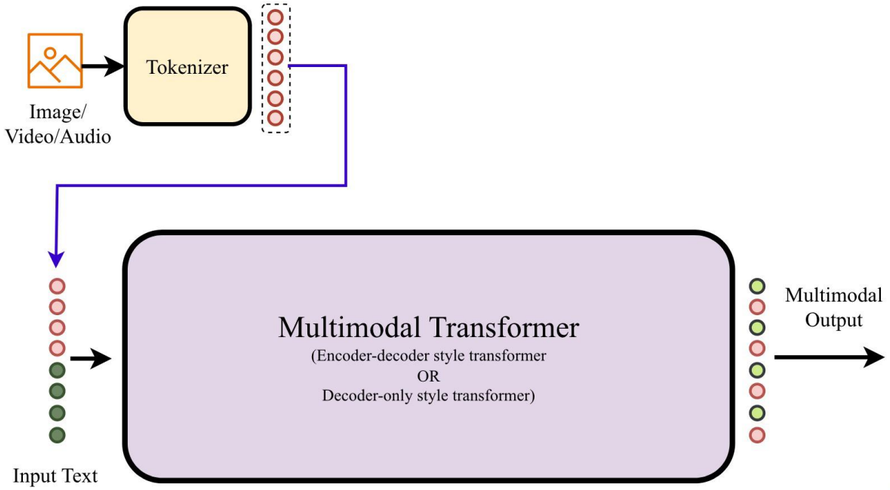
Type B



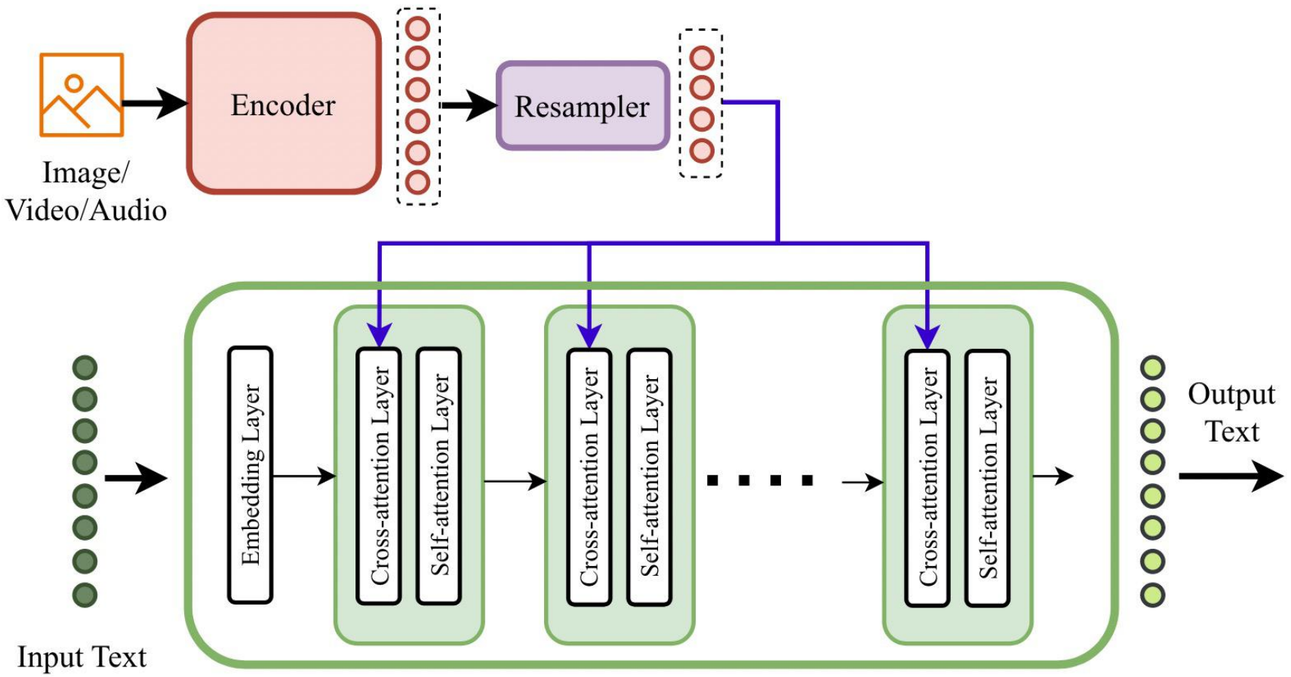
Type C



Type D



Standard Cross-Attention based Deep Fusion (SCDF) - Type A

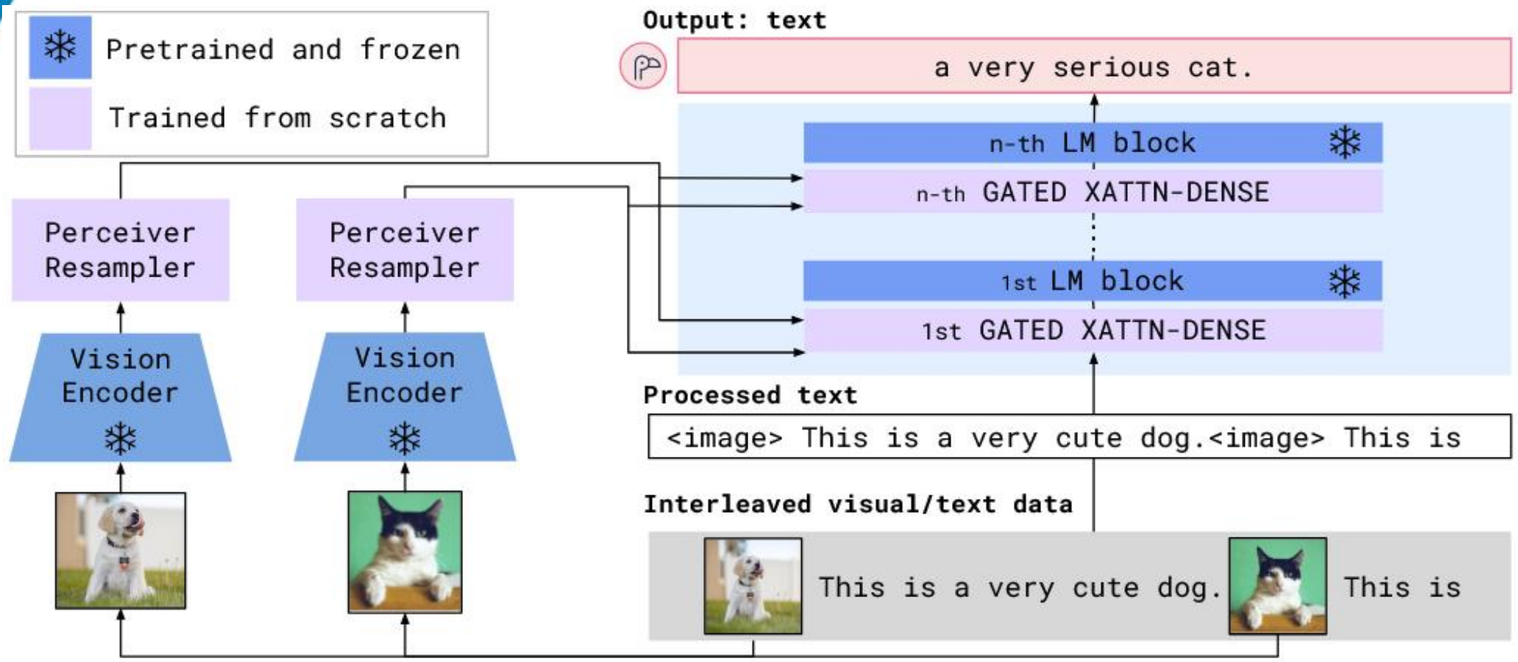


It typically involves **a pre-trained LLM** and integrating **standard cross-attention layers** into its internal architecture to achieve **deep fusion** of input modalities.

A resampler is used to generate a fixed number of tokens that aligns with the requirements of the decoder layer. These resampler outputs are then directed to the internal layers of the LLM using cross-attention layers.

The cross-attention layer can be added **before or after** the self-attention layer in the model's architecture.

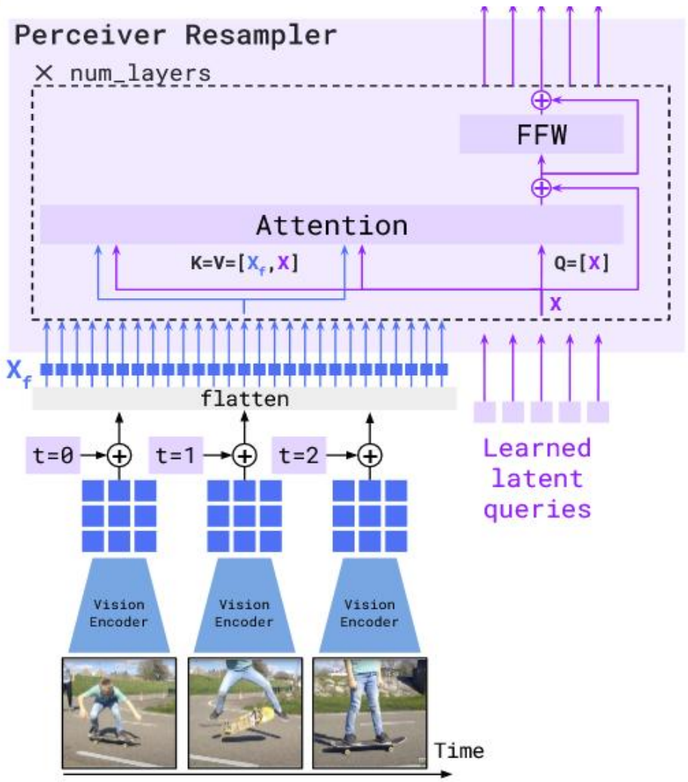
Flamingo（2022.04）



Vision Encoder: A pre-trained and frozen vision model.
Perceiver Resampler: Receives variable-length features from the visual encoder.
Frozen Language Model: A large pre-trained language model.(Chinchilla models)
GATED XATTN-DENSE layers.

Flamingo is trained using a mixture of three types of datasets:
M3W
ALIGN: Composed of 1.8 billion images paired with alt-text.
LTIP: Contains 312 million pairs of high-quality, long-text descriptions of images and text.
VTP: It contains 27 million short videos and their corresponding text descriptions.

Flamingo（2022.04）



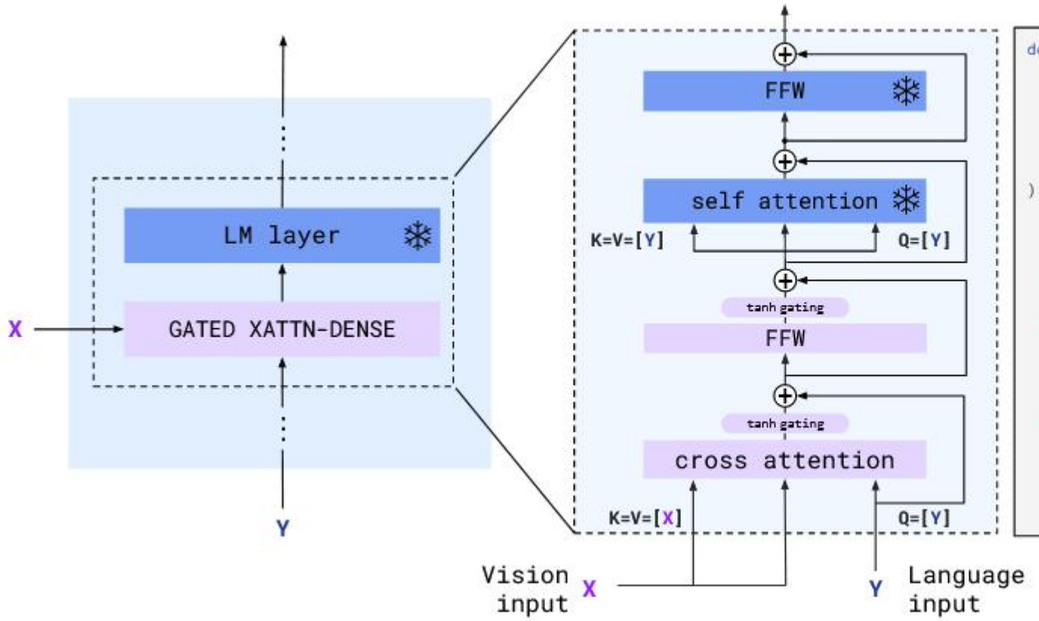

```
def perceiver_resampler(
    x_f,  # The [T, S, d] visual features (T=time, S=space)
    time_embeddings,  # The [T, 1, d] time pos embeddings.
    x,  # R learned latents of shape [R, d]
    num_layers,  # Number of layers
):
  """The Perceiver Resampler model."""

  # Add the time position embeddings and flatten.
  x_f = x_f + time_embeddings
  x_f = flatten(x_f)  # [T, S, d] -> [T * S, d]
  # Apply the Perceiver Resampler layers.
  for i in range(num_layers):
    # Attention.
    x = x + attention_i(q=x, kv=concat([x_f, x]))
    # Feed forward.
    x = x + ffw_i(x)
  return x
```

The Perceiver Resampler is a Transformer Encoder whose input is a feature map (image) ora sequence of feature maps with temporal embedding (video). And a set of learnable latent queries, with the output being a set of fixed-size vision tokens.

Therefore, the Perceiver Resampler uses a set of learnable queries to apply attention to the visual tokens encoded by the Vision Encoder before the LM, extracts a fixed number of denser visual Tokens, and then inputs them into the LM.

Flamingo（2022.04）



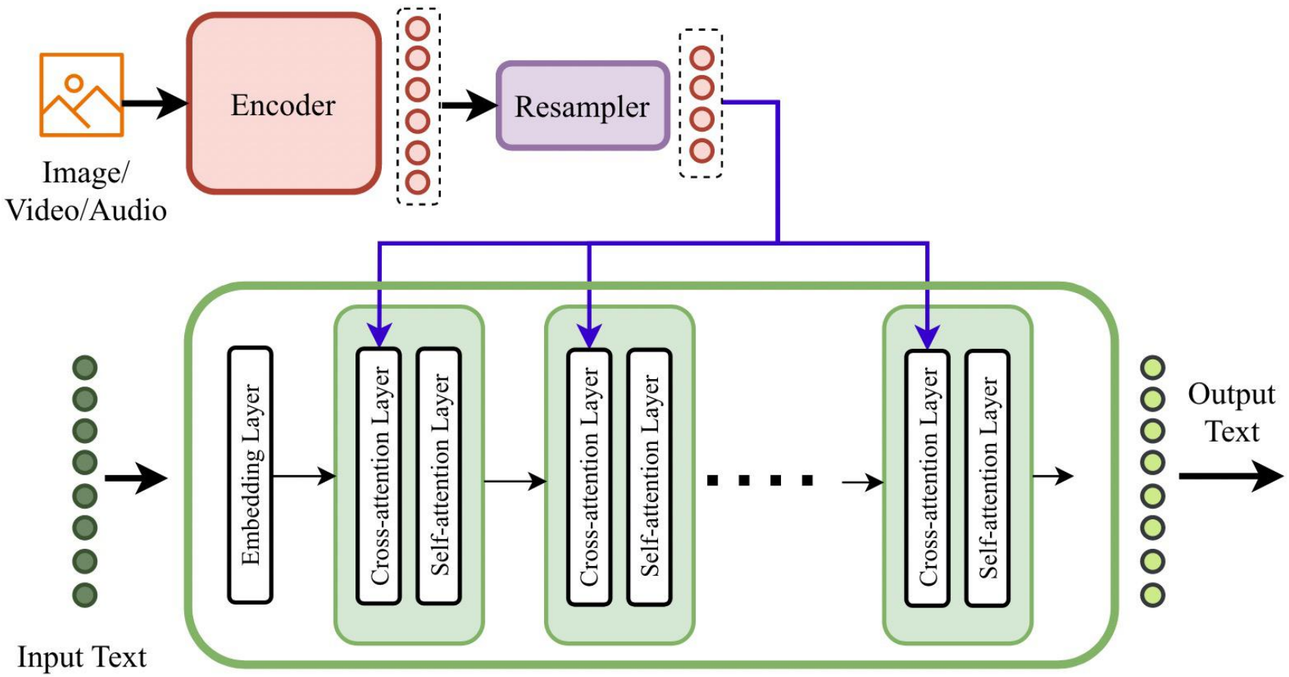

```
def gated_xattn_dense(
    y,  # input language features
    x,  # input visual features
    alpha_xattn, # xattn gating parameter – init at 0.
    alpha_dense, # ffw gating parameter – init at 0.
):
  """Applies a GATED XATTN-DENSE layer."""

  # 1. Gated Cross Attention
  y = y + tanh(alpha_xattn) * attention(q=y, kv=x)
  # 2. Gated Feed Forward (dense) Layer
  y = y + tanh(alpha_dense) * ffw(y)

  # Regular self-attention + FFW on language
  y = y + frozen_attention(q=y, kv=y)
  y = y + frozen_ffw(y)
  return y  # output visually informed language features
```

Visual features and text features are combined through the gated cross-attention dense module. The gated cross-attention dense module uses the tanh-gating mechanism and the output after multiplying tanh(α)by text and visual modal cross-attention. α is initialized to 0. The tanh-gating mechanism ensures that during initialization, the model is not affected by visual features, and the output is the output of the language model.

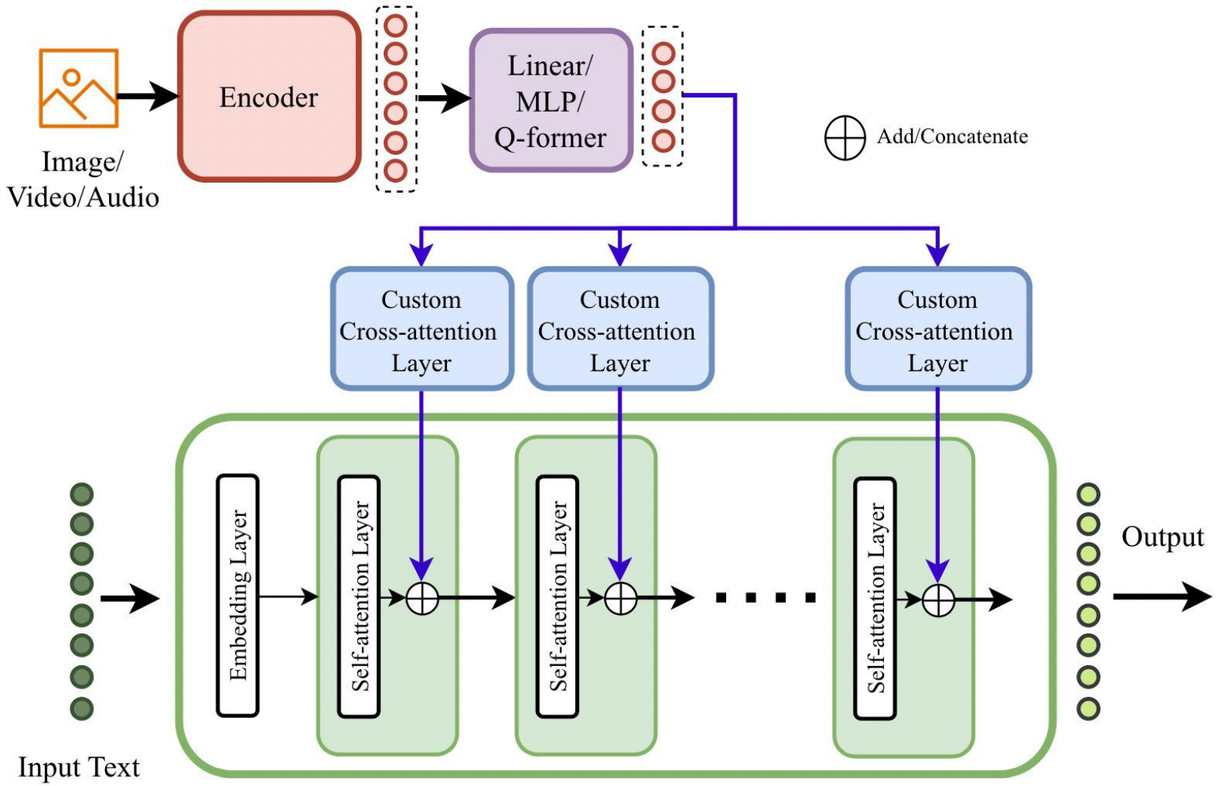
Standard Cross-Attention based Deep Fusion (SCDF) - Type A



Type-A multimodal model architecture enables **fine-grained control** of how modality information **flows** in the model. It is end-to-end trainable and omits design of custom layers by using standard learnable layers of transformers.

Architecture is difficult to scale, especially if pretraining step is involved, because of the large number of training parameters and computational requirements. Adding more modalities is challenging, because in Type-A, after adding image modality cross-attention layer to the LLM layer, adding other modalities to each LLM layer is difficult.

Custom Layer based Deep Fusion (CLDF) – Type B

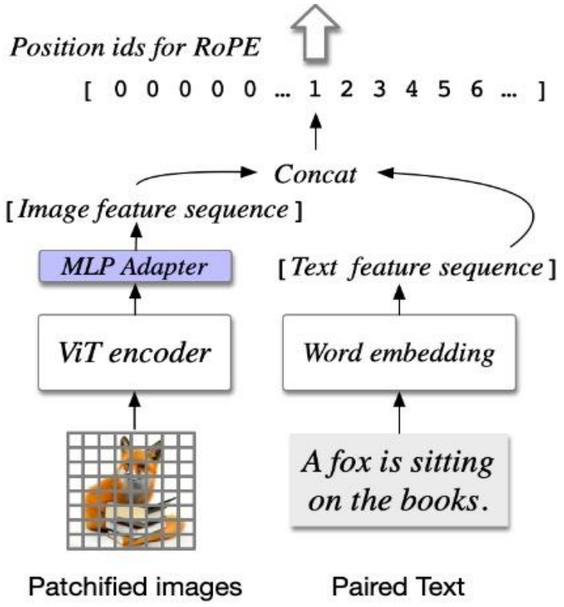


In Type-A, a standard cross-attention layer is utilized, while in Type-B, **a custom-designed layer** is or can be used. Similar to Type-A, in Type-B the input modalities are deeply fused into the **internal layers** of the model.

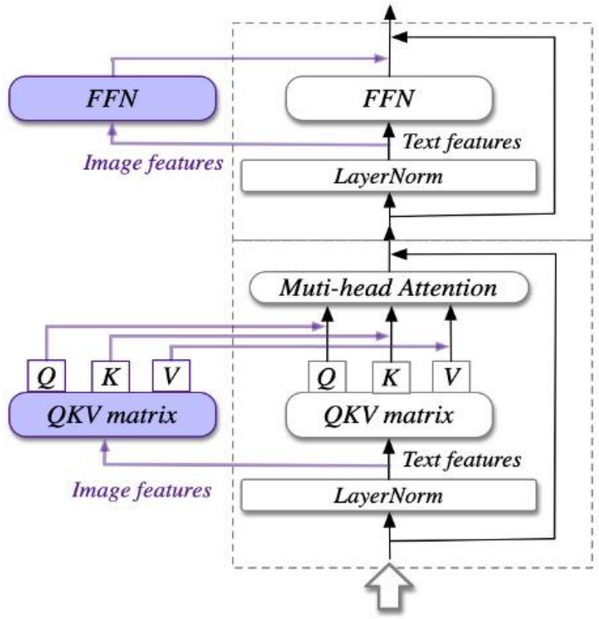
Additionally, a learnable gating factor is included to control the contribution of the cross-attention layer on the output of self-attention layer.

- Uses custom learnable layer other than cross-attention layer.
- Adds custom-cross-attention layer to the internal layers of the LLM

CogVLM（2023.11）



(a) The input of visual language model

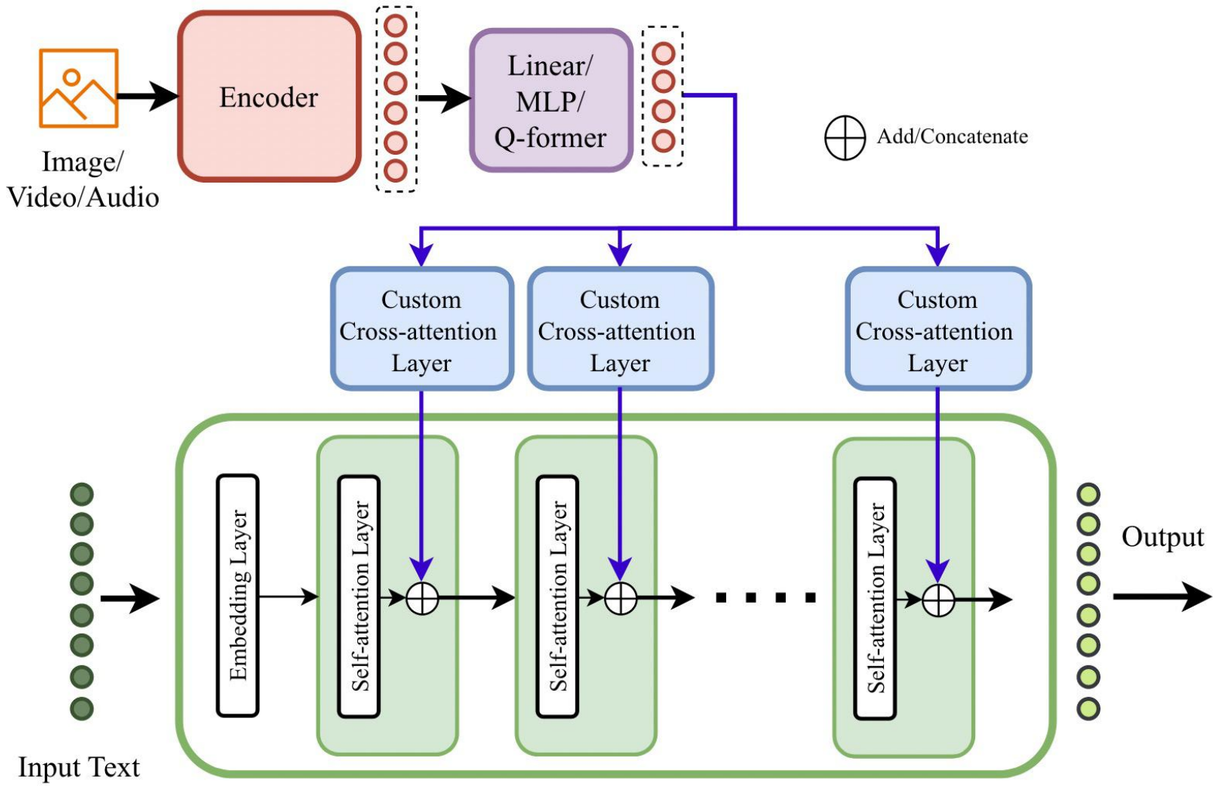


(b) The visual expert built on the language model

CogVLM model comprises four fundamental components:
a vision transformer (ViT) encoder
an MLP adapter（map the output of ViT into the same space as the text features from word embedding）
a pretrained large language model (Vicuna1.5-7B)
a visual expert module（add a visual expert module to each layer）

The first stage of pretraining is for image captioning loss, i.e. next token prediction in the text part.（LAION-2B and COYO-700M）
The second stage of pretraining is a mixture of image captioning and Referring Expression Comprehension (REC, predict the bounding box in the image given the text description of an object).

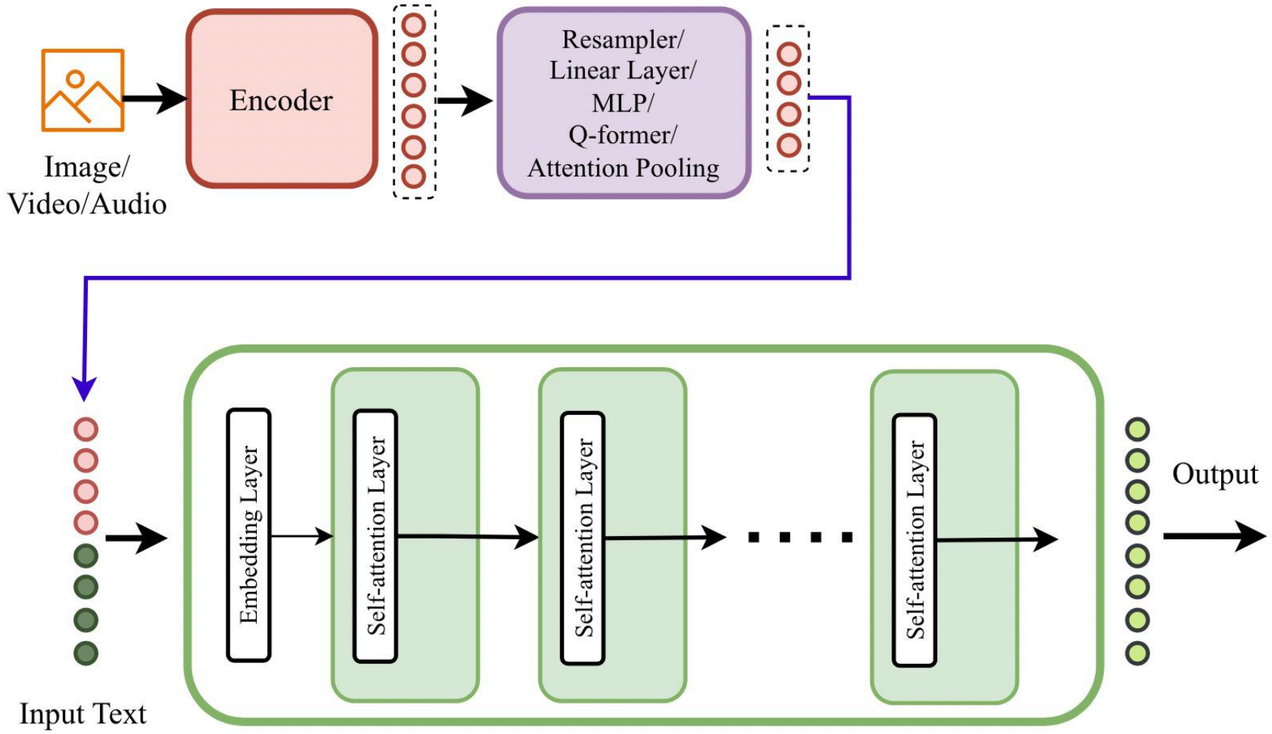
Custom Layer based Deep Fusion (CLDF) – Type B



Type-B also benefits from fine-grained control of how modality information flows in the model. It is end-to-end trainable.

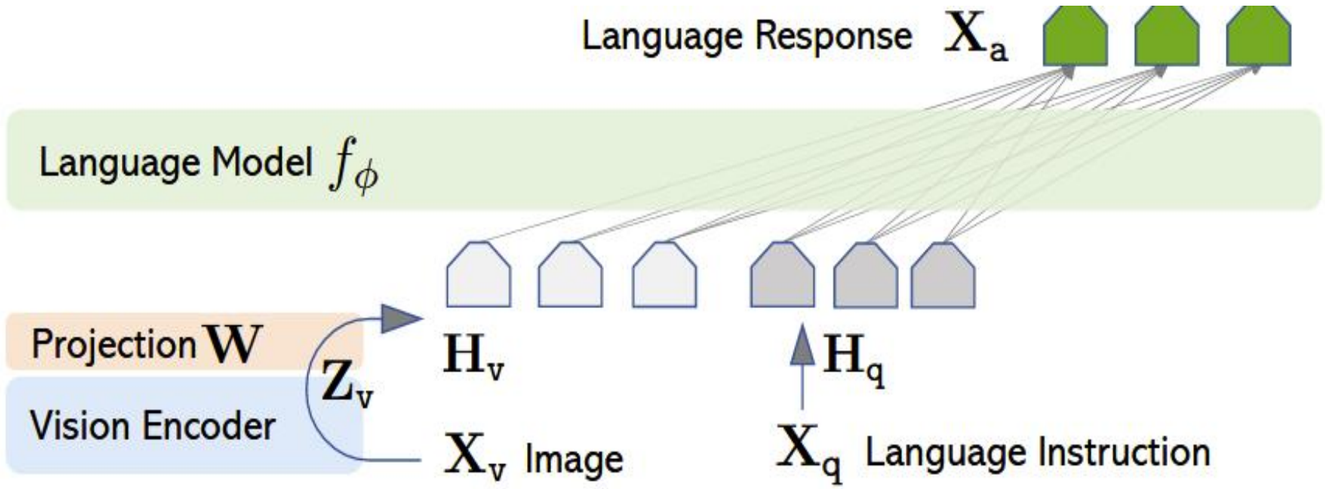
The **custom design** adds to more fine-grained control of modality fusion. Type-B architecture is **more scalable**, due to customizable nature and computational efficiency of the custom learnable connector layers. The Type-B, uniquely provides an alternative by introducing a gating mechanism which can be utilized to add other modalities. The gating mechanism enables direct addition of input modalities to the output LLM layers.

Non-Tokenized Early Fusion (NTEF) - Type C



The modality encoder output is solely directed and fused at the input of the model, without involvement in the internal layers of the model. Pretrained LLM as decoder is used without any major architectural changes to its internal layers. Pretrained image encoder or other modality encoder are used. The encoder/s and the decoder are combined together with **learnable module** like single Linear-Layer, MLP, Q-former, attention-pooling layer, convolutional layer, perceiver resampler or variants of Q-former.

Linear Layer/MLP - LLaVA （2023.04）

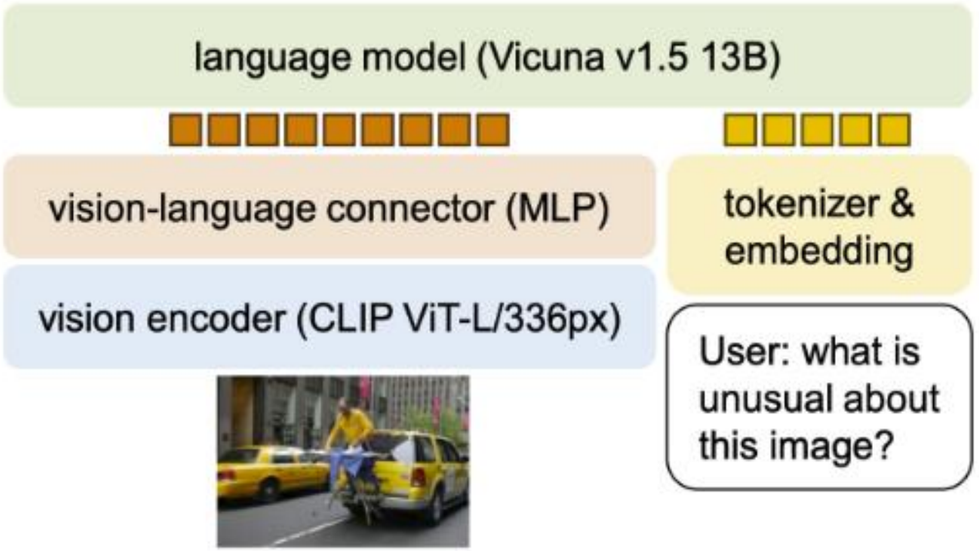


Pre-trained image encoder (Pre-trained CLIP visual encoder ViT-L/14)
Pre-trained LLM (Vicuna)
Projection W: A trainable projection matrix W (simple linear layer), mapping the image embedding to the same dimension as the text embedding

Phase 1: Pre-training Projection W, froze LLM and vision encoder

Phase 2: Fine-tune LLM+Projection W end-to-end and continue with froze vision encoder

Linear Layer/MLP - LLaVA-1.5 (2023.10)



Improve LLaVA.

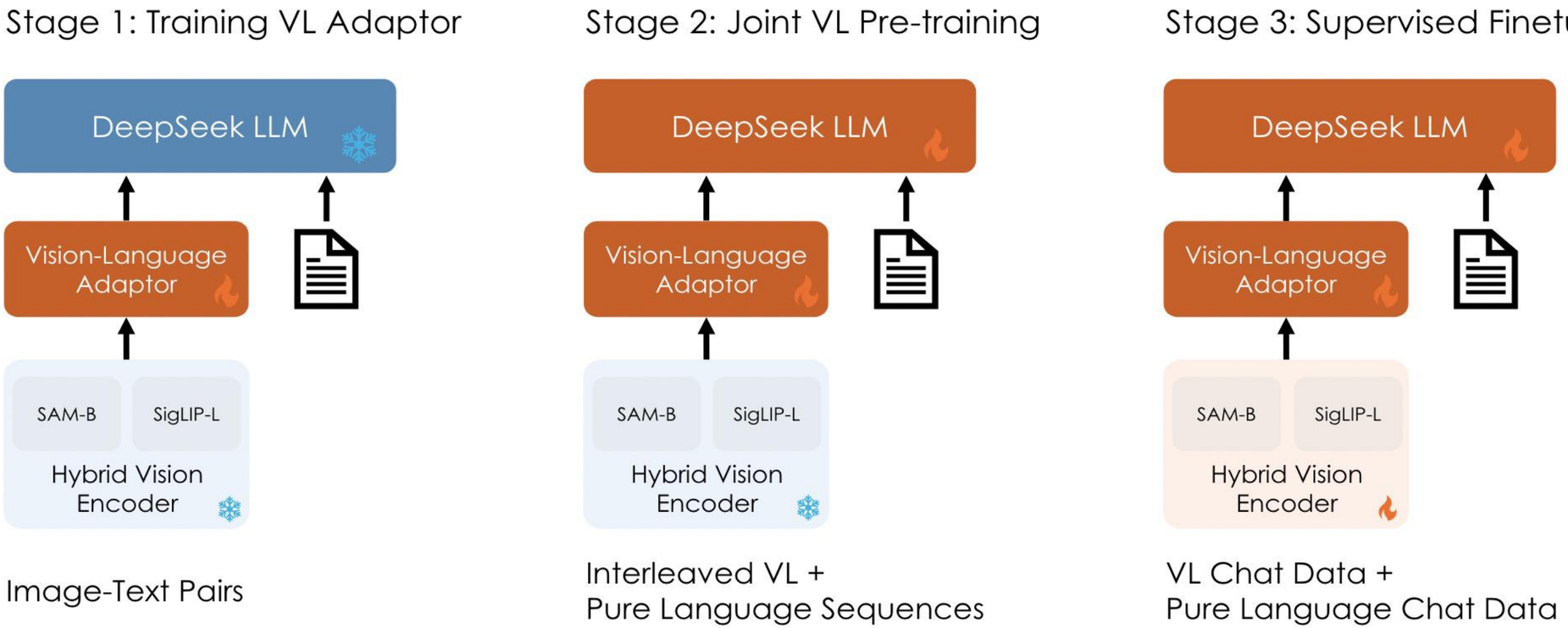
Change the simple linear layer to two layers of MLP, and replace the original simple linear layer with two layers of MLP to improve the ability of the multimodal model;

Add datasets in fields such as VQA/OCR;

prompt with a clearly specified output format (for solving short text question answering);

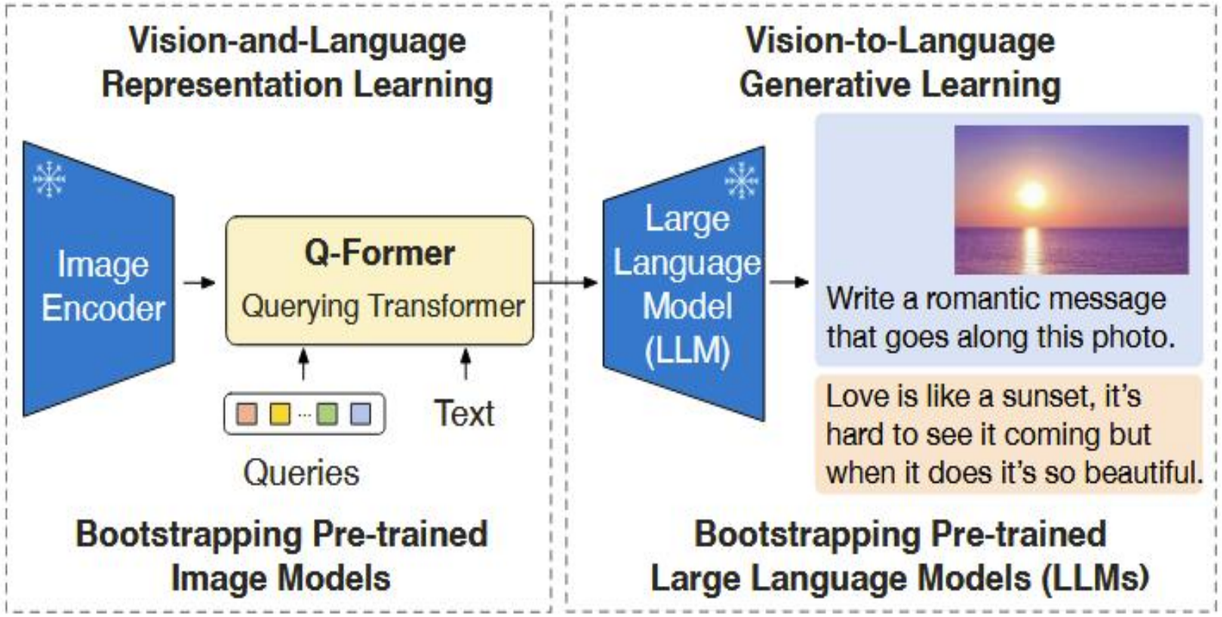
The resolution of the picture has been increased to 336, LLM expanded to 13B.

Linear Layer/MLP - deepseek-VL（2024.03）



The pre-training dataset contains extensive public resources and carefully selected proprietary data;
In the first stage, the pre-trained dataset is used to set up and warm up the model to adapt to the interaction between vision and language.
In the second stage, the visual-language model is further trained through joint pre-training.
In the third stage, the supervised fine-tuning dataset is utilized for the final training of the model, specifically optimized for specific visual-language tasks

Q-former - BLIP-2（2023.01）



The pre-trained image encoder
Pre-trained LLM (decoder-based LLM/encoder-decoder-based LLM)
Q-Former (Bridging the gap between different Modalities)

The first stage bootstraps vision-language representation learning from a frozen image encoder.
The second stage bootstraps vision-to-language generative learning from a frozen LLM, which enables zero-shot instructed image-totext generation.

Q-former - BLIP-2 （2023.01）

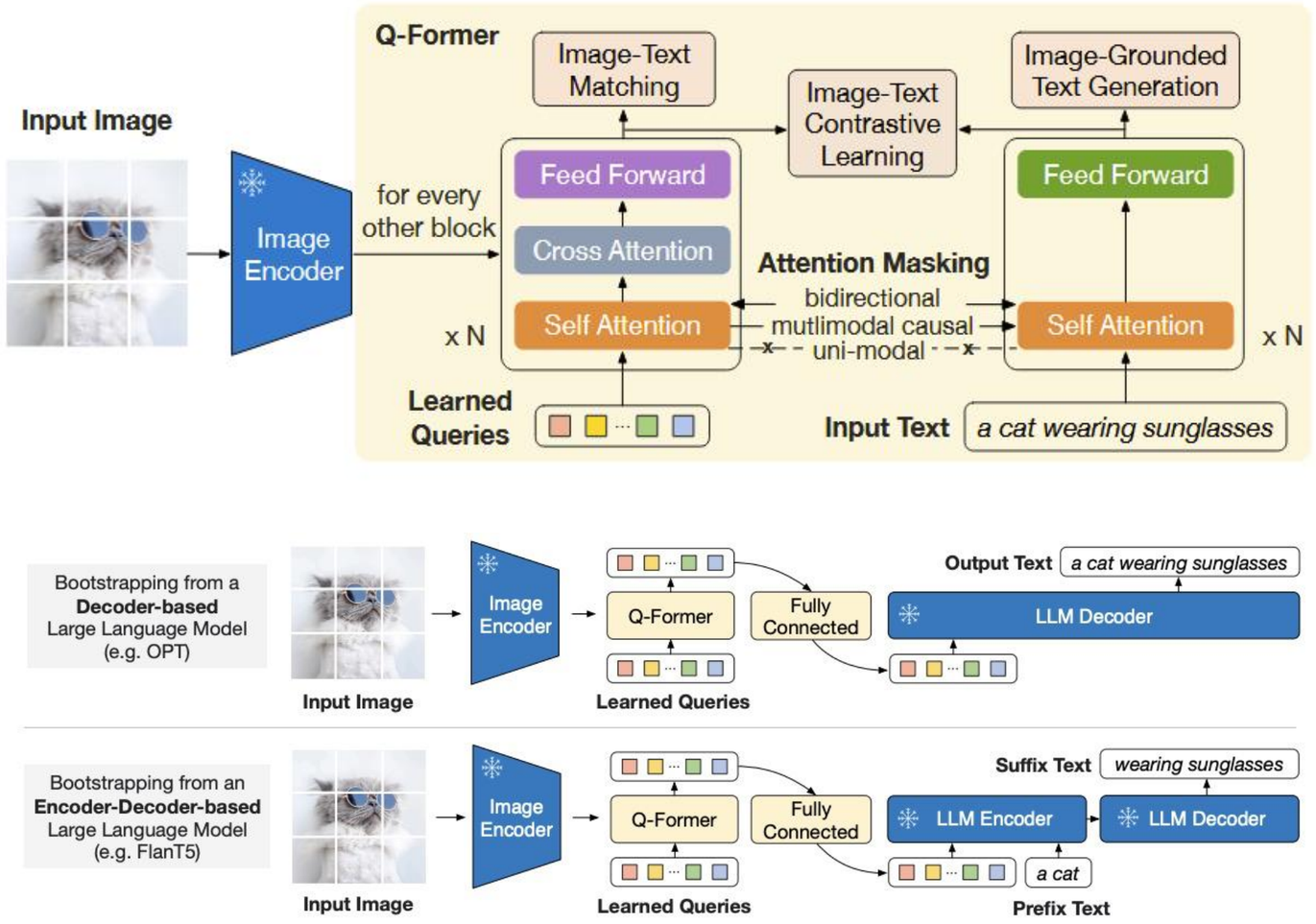
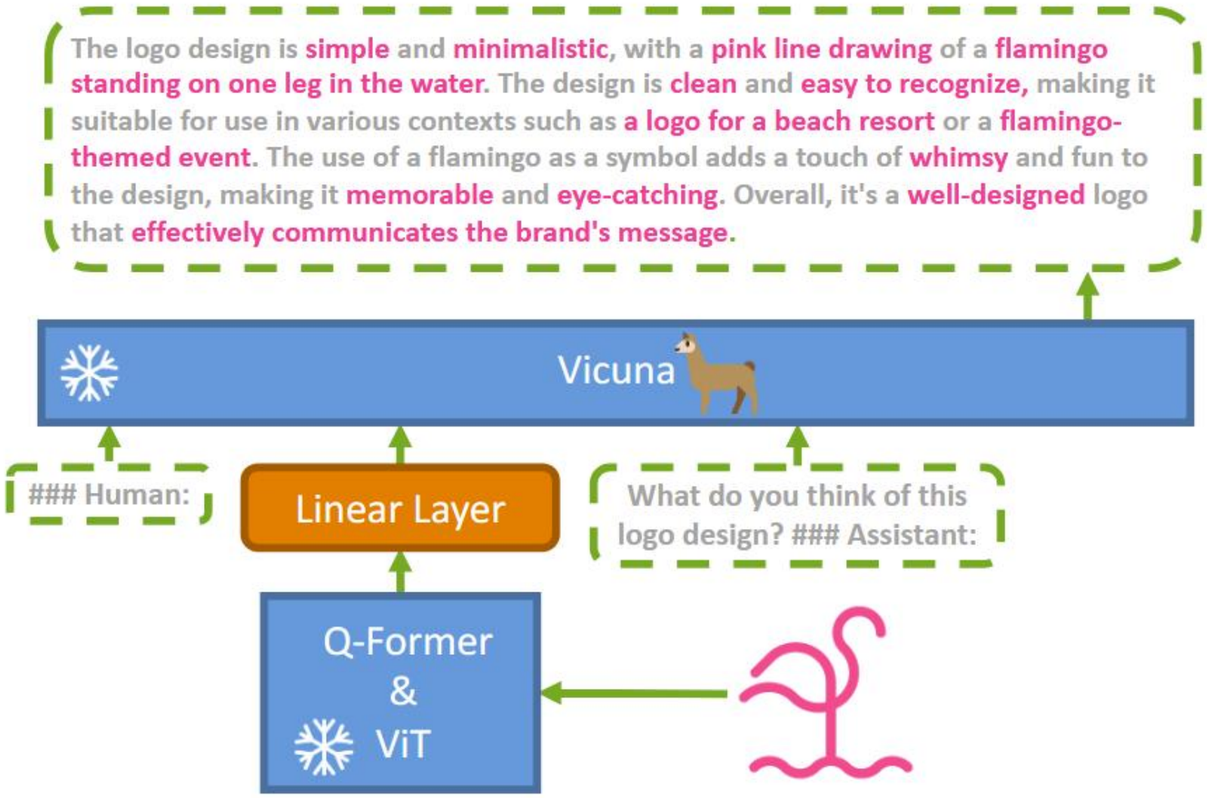


Image transformer: Interact with the frozen image encoder to obtain image features; The input is a set of learnable queries embeddings, which can interact with the input text either through self-attention shared with the text transformer or with the image encoder on the left. They can also interact with each other through self-attention.

Text transformer: It can be used as both an encoder and a decoder

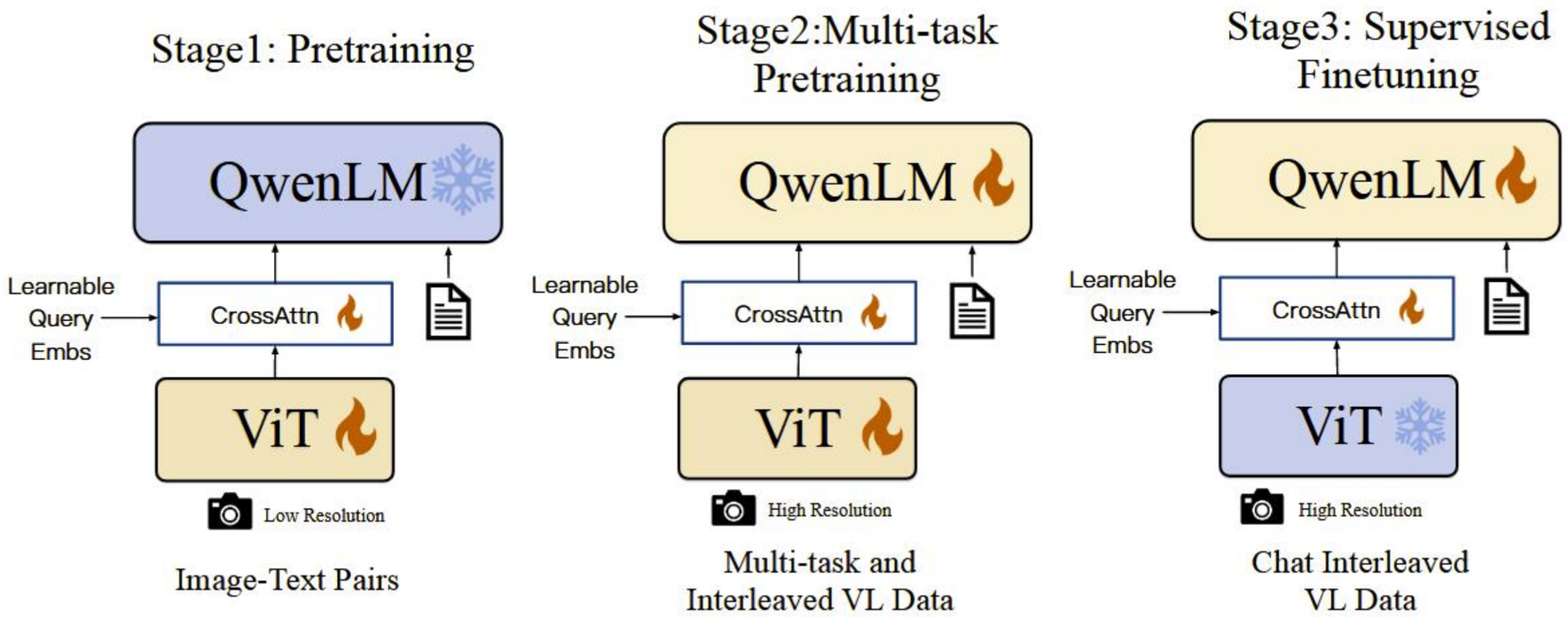
Q-former - MiniGPT-4 （2023.04）



Currently, the update speed of LLM and vision encoder is very fast. Freezing the parameters of ViT&Q-former and LLM and only training the projection layer connecting ViT&Q-former and LLM would be very effective, avoiding the waste of resources from scratch training

The first stage is to pre-train the model on a large number of aligned image-text pairs;
In the second stage, the pre-trained model is fine-tuned using a smaller-scale but higher-quality image-text dataset and dialogue templates.

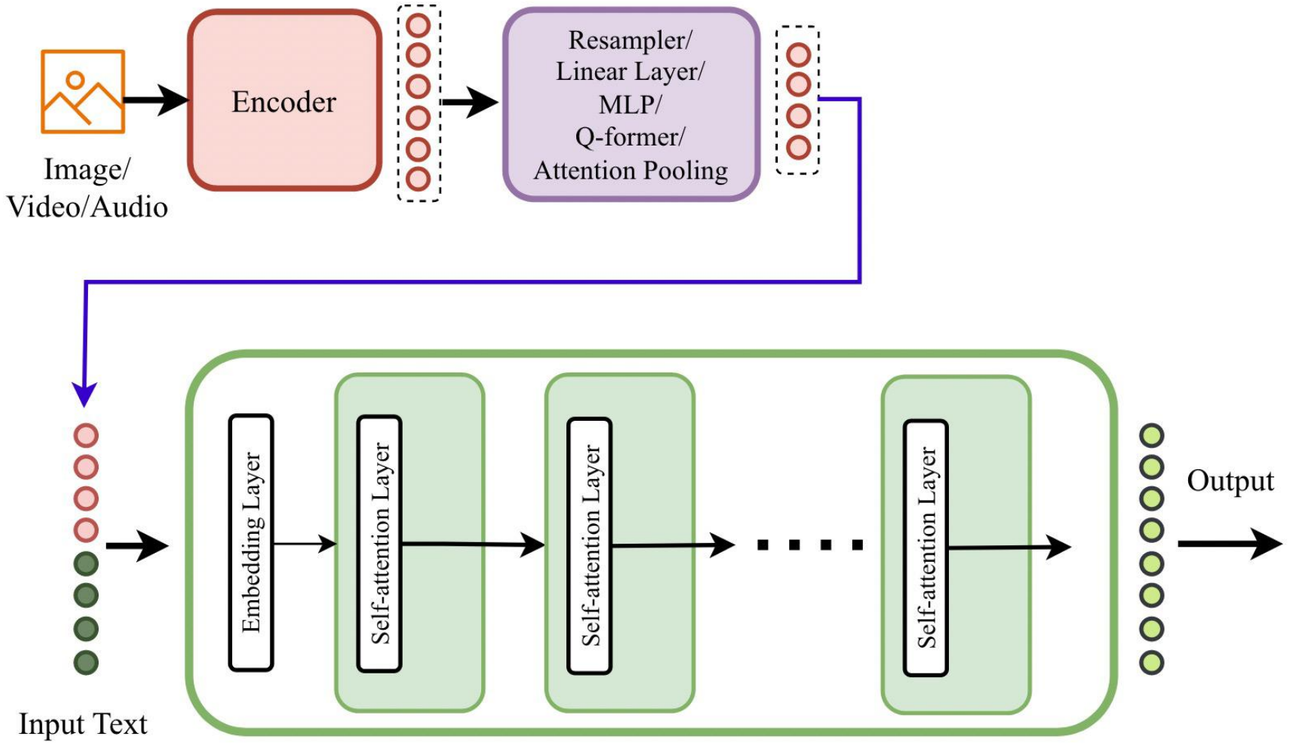
Custom Learnable layer - Qwen-VL （2023.08）



LLM (Qwen-7B)
Visual encoder (ViT-bigG)
Visual-Language Adapter (A randomly initialized single-layer cross-attention module that still feels similar to Q-former, with the input being trainable queries embedding)
Output as the vision representation that is most suitable for ViT

Phase 1: Pre-training. Use image-text pairs obtained from large-scale, weakly labeled, and crawlers, freeze LLM, and only optimize ViT and adapter. The training goal is to minimize the cross-entropy of text tokens
Phase 2: Multi-task pre-training (7 tasks), using high-quality and fine-grained VL annotated data with larger input resolution and interlaced image-text data for pre-training
Phase 3: Instruction fine-tuning, freezing ViT, optimizing LLM and adapter. The multimodal instruction fine-tuning data is derived from the dialogue data of image content understanding + an additional set of dialogue data (manual annotation, model generation, and policy connection construction).

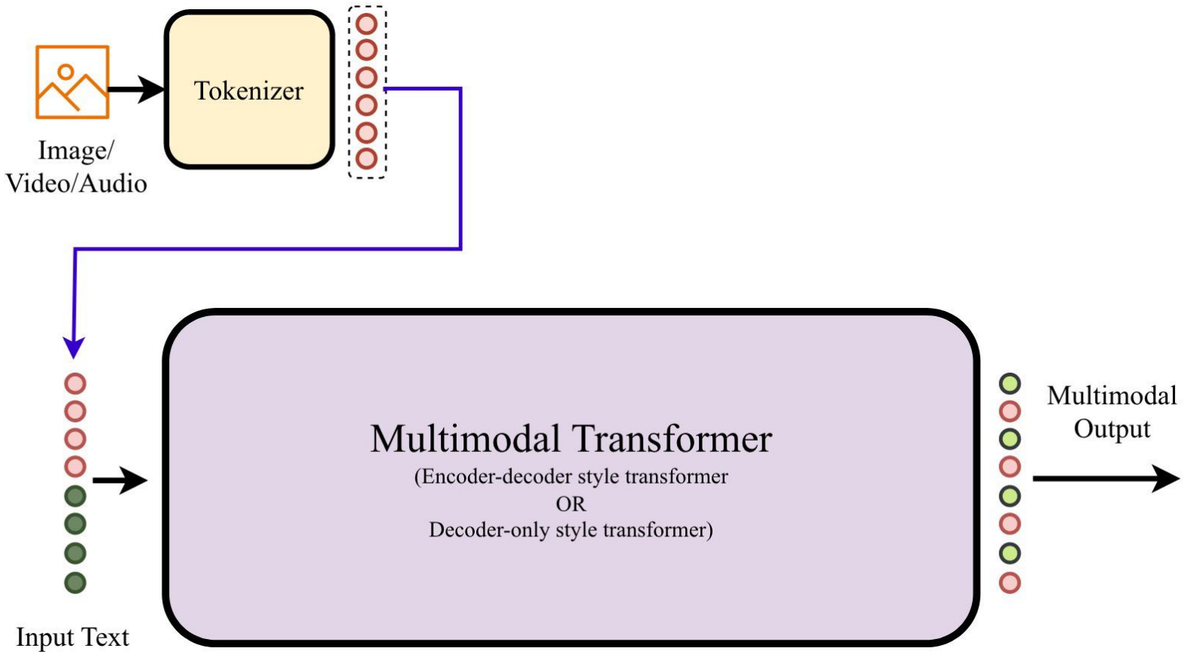
Non-Tokenized Early Fusion (NTEF) - Type C



Type-C does not have fine-grained control of how modality information flows in the model. Different modality inputs are **fused only at the input of decoder** (LLM). It is end-to-end trainable. It is easier to build compared to all other type of multimodal architectures, owing to its modular architecture.

Type-C architecture is scalable due to its modular design, reduced training data requirements, and computational efficiency.

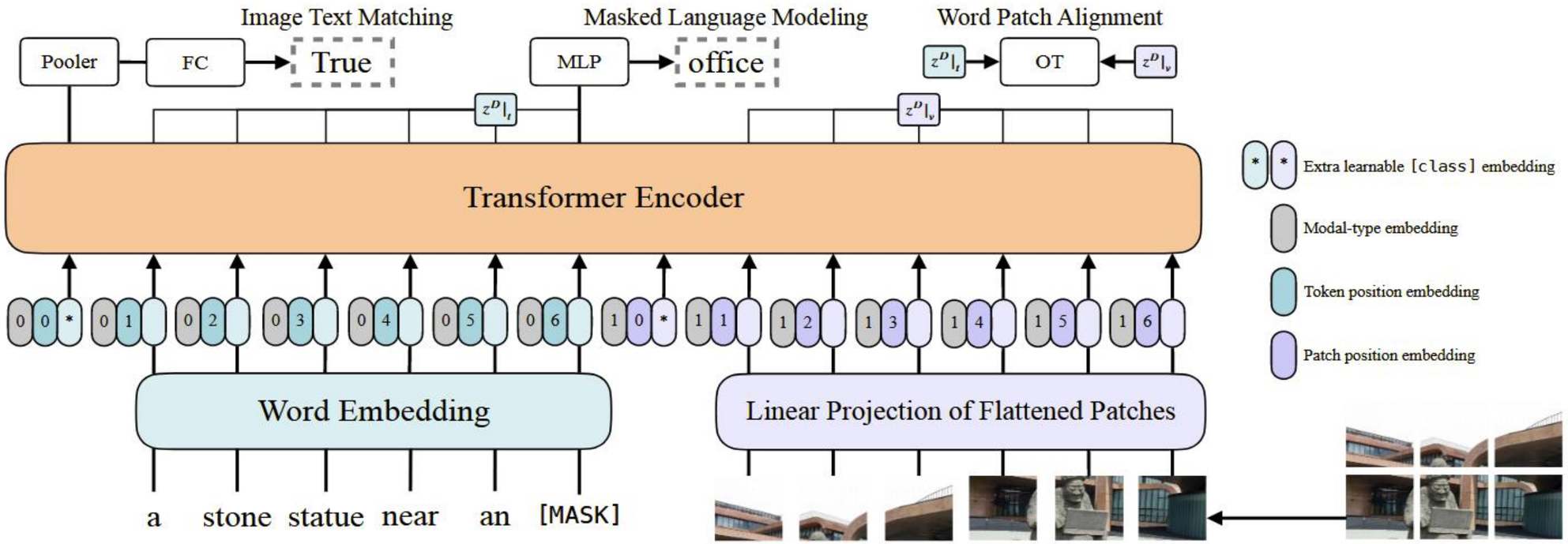
Tokenized Early Fusion (TEF) – Type D



In Type-D, multimodal inputs are **tokenized** using a common tokenizer or modality specific tokenizers. The tokenized inputs are then given to a pretrained LLM or an encoder-decoder transformer model, which generates multimodal outputs.

Either pretrained modality specific tokenizers are used, or a tokenizer training stage is included in the training process.

ViLT (2021.02)

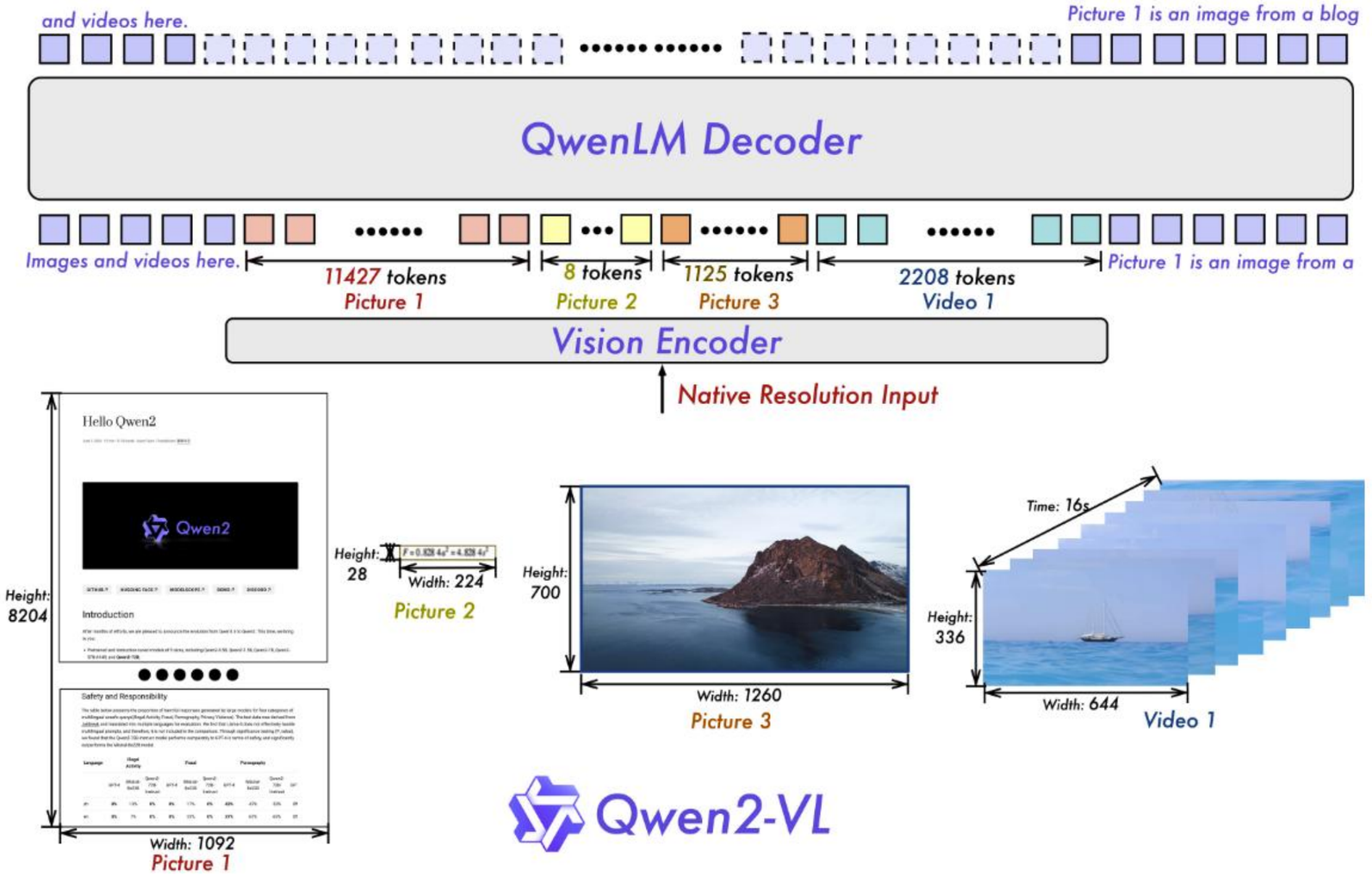


Use transformer to process visual features instead of the original separate visual processing module (it is believed that the original focus was on image feature extraction and encoding, and the interaction between text and images remained at a superficial level).

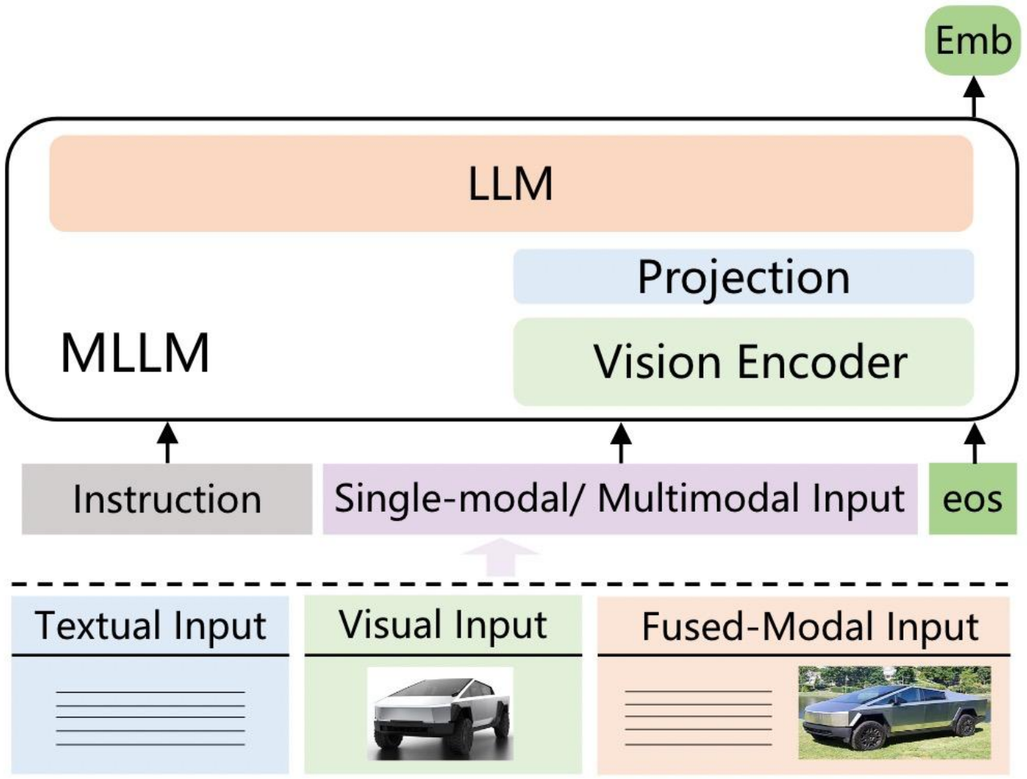
The image preprocessing only segments the patch and linear projection. Then, the concat vectors are input, and the interaction of modal information is achieved through the transformer

Using the conventional pre-training tasks combining text and images, ITM (randomly replacing aligned images with different images with a probability of 0.5) and MLM are employed

Qwen2-VL（2024.09）



GME（2024.12）



MLLM（Qwen2-VL） as the foundation for GME.

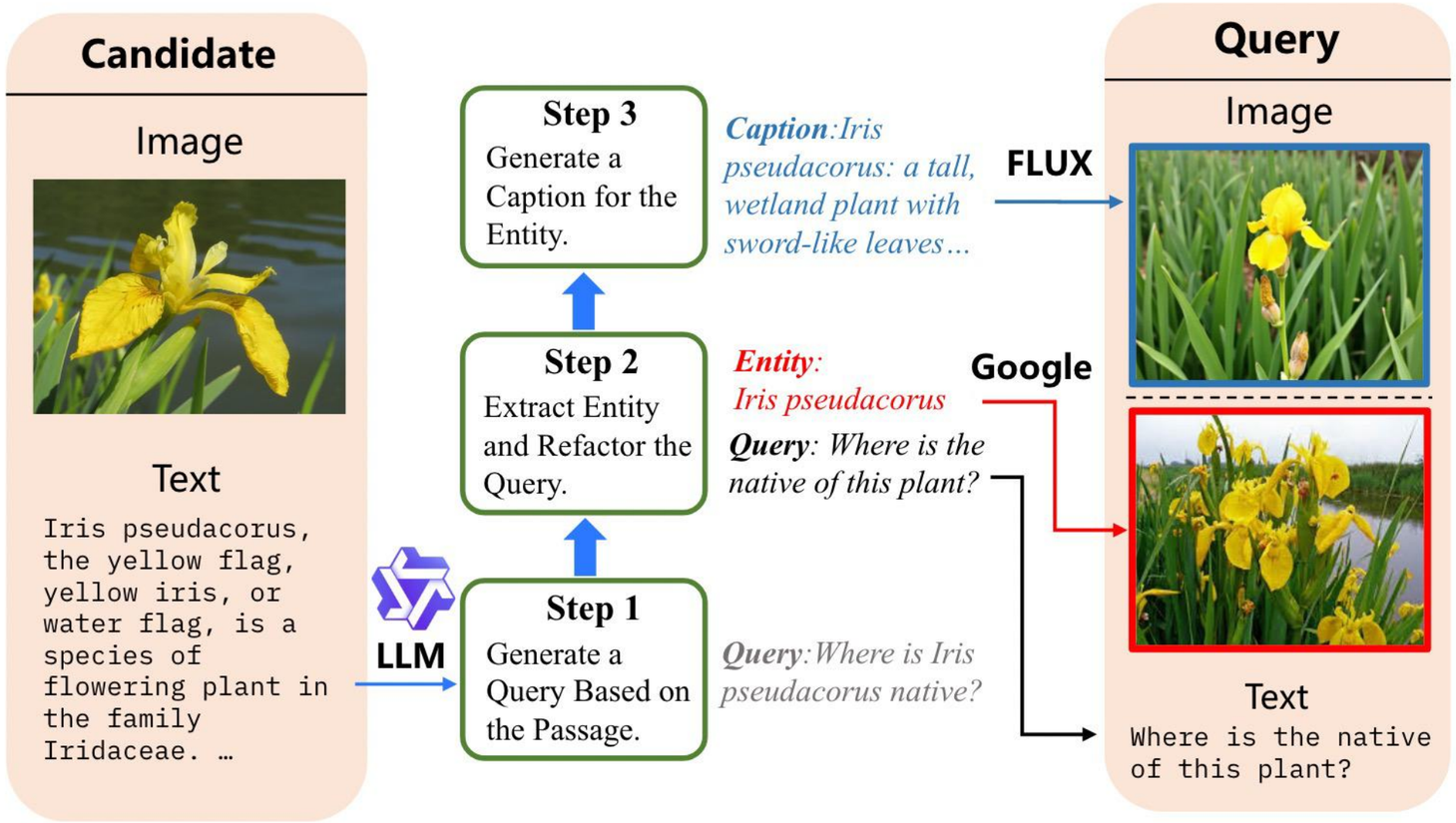
Two-stage training strategy：

(1) Initial Training: We first train the model using randomly selected negative candidates, resulting in Model M1.

(2) Hard Negative Mining and Continue Training: Using M1, we retrieve the top K candidates for each query and select non-relevant candidates from them as hard negatives. We then use these hard negatives to further train M1, refining it into the final model.

single-modal	T->T	MSMARCO、NQ、HotpotQA、TriviaQA、SQuAQ、FEVER
	I->I	ImageNet
cross-modal	T->I	LAION、mscoco
	T->VD	Docmatix
fused-modal	IT->IT	M-BEIR and 1.1 million synthesized fused data

Fused-Modal Data Synthesis



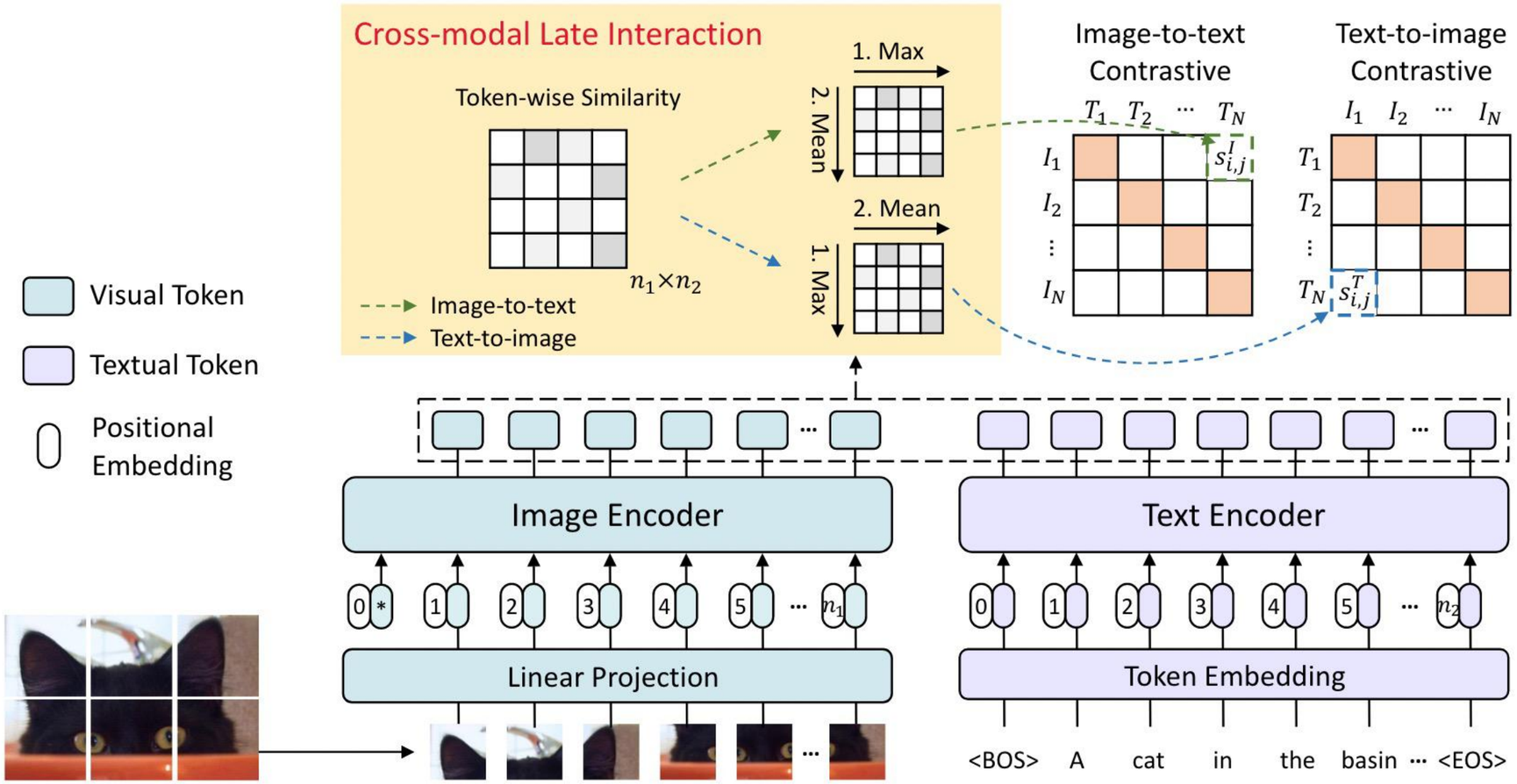
GME (2024.12)

Universal Multimodal Retrieval Benchmark - UMRB

Name	Type	Categ.	Eval Samples	Candidates Nums	Eval Query avg. chars	Eval Candidate avg. chars	In partial
ArguAna	Single-Modal	T→T	10,080	1,406	192.98	166.80	True
Climate-FEVER	Single-Modal	T→T	1,535	5,416,593	20.13	84.76	False
CQADupStack	Single-Modal	T→T	13,145	457,199	8.59	129.09	False
DBPedia	Single-Modal	T→T	400	4,635,922	5.39	49.68	False
FEVER	Single-Modal	T→T	6,666	5,416,568	8.13	84.76	False
FiQA2018	Single-Modal	T→T	648	57,638	10.77	132.32	False
HotpotQA	Single-Modal	T→T	7,405	5,233,329	17.61	46.30	False
MSMARCO	Single-Modal	T→T	6,980	8,841,823	5.96	55.98	False
NFCorpus	Single-Modal	T→T	323	3,633	3.30	232.26	True
NQ	Single-Modal	T→T	3,452	2,681,468	9.16	78.88	False
Quora	Single-Modal	T→T	10,000	522,931	9.53	11.44	True
SCIDOCS	Single-Modal	T→T	1,000	25,657	9.38	176.19	True
SciFact	Single-Modal	T→T	300	5,183	12.37	213.63	False
Touche2020	Single-Modal	T→T	49	382,545	6.55	292.37	False
TRECCOVID	Single-Modal	T→T	50	171,332	10.60	160.77	True
WebQA	Single-Modal	T→T	2,455	544,457	18.58	37.67	False
Nights	Single-Modal	I→I	2,120	40,038	-	-	True
VisualNews	Cross-Modal	T→I	19,995	542,246	18.78	-	False
Fashion200k	Cross-Modal	T→I	1,719	201,824	4.89	-	False
MSCOCO	Cross-Modal	T→I	24,809	5,000	10.43	-	True
Flickr30k	Cross-Modal	T→I	5,000	1,000	12.33	-	True
TAT-DQA	Cross-Modal	T→VD	1,646	277	12.44	-	False
ArxivQA	Cross-Modal	T→VD	500	500	17.12	-	False
DocVQA	Cross-Modal	T→VD	451	500	8.23	-	True
InfoVQA	Cross-Modal	T→VD	494	500	11.29	-	False
Shift Project	Cross-Modal	T→VD	100	1,000	16.01	-	True
Artificial Intelligence	Cross-Modal	T→VD	100	968	12.3	-	False
Government Reports	Cross-Modal	T→VD	100	972	12.62	-	False
Healthcare Industry	Cross-Modal	T→VD	100	965	12.56	-	False
Energy	Cross-Modal	T→VD	100	977	13.49	-	False
TabFQuad	Cross-Modal	T→VD	280	70	16.49	-	False
VisualNews	Cross-Modal	I→T	20,000	537,568	-	18.53	False
Fashion200k	Cross-Modal	I→T	4,889	61,707	-	4.95	False
MSCOCO	Cross-Modal	I→T	5,000	24,809	-	10.43	True
Flickr30k	Cross-Modal	I→T	1,000	5,000	-	12.33	True
WebQA	Fused-Modal	T→IT	2,511	403,196	16.43	12.83	False
EDIS	Fused-Modal	T→IT	3,241	1,047,067	20.07	15.53	False
OVEN	Fused-Modal	IT→T	50,004	676,667	6.52	82.13	False
INFOSEEK	Fused-Modal	IT→T	11,323	611,651	8.76	91.49	False
ReMuQ	Fused-Modal	IT→T	3,609	138,794	13.82	34.26	True
OKVQA	Fused-Modal	IT→T	5,046	114,516	8.09	102.55	True
LLaVA	Fused-Modal	IT→T	5,120	5,994	10.70	90.65	True
FashionIQ	Fused-Modal	IT→I	6,003	74,381	11.70	-	True
CIRR	Fused-Modal	IT→I	4,170	21,551	11.01	-	True
OVEN	Fused-Modal	IT→IT	14,741	335,135	5.91	94.76	True
EVQA	Fused-Modal	IT→IT	3,743	68,313	9.38	211.12	False
INFOSEEK	Fused-Modal	IT→IT	17,593	481,782	7.94	96.00	False

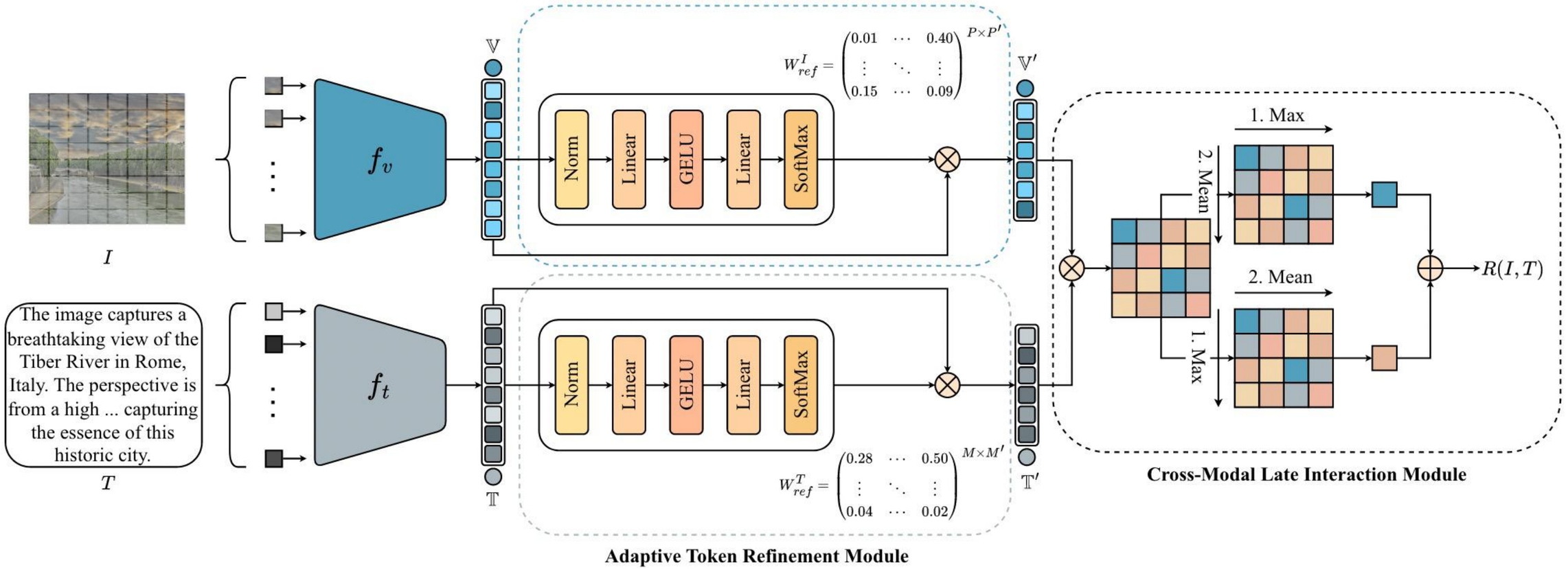
UMRB	Size	Single-Modal		Cross-Modal			Fused-Modal				Avg.
Task (#Datasets)		T→T (16)	I→I (1)	T→I (4)	T→VD (10)	I→T (4)	T→IT (2)	IT→T (5)	IT→I (2)	IT→IT (3)	(47)
VISTA (2024a)	0.2B	55.15	**31.98**	32.88	10.12	31.23	45.81	53.32	8.97	26.26	37.32
CLIP-SF (2024)	0.4B	39.75	31.42	59.05	24.09	62.95	66.41	53.32	34.90	55.65	43.66
One-Peace (2023)	4B	43.54	31.27	61.38	42.9	65.59	42.72	28.29	6.73	23.41	42.01
DSE (2024)	4.2B	48.94	27.92	40.75	78.21	52.54	49.62	35.44	8.36	40.18	50.04
E5-V (2024a)	8.4B	52.41	27.36	46.56	41.22	47.95	54.13	32.90	23.17	7.23	42.52
GME-Qwen2VL-2B	2.2B	55.93	29.86	57.36	87.84	61.93	76.47	64.58	37.02	66.47	64.45
GME-Qwen2VL-7B	8.2B	**58.19**	31.89	**61.35**	**89.92**	**65.83**	**80.94**	**66.18**	**42.56**	**73.62**	**67.44**

FILIP (https://arxiv.org/abs/2111.07783)



Previous methods like CLIP and ALIGN simply encode each image or text separately to a global feature, neglecting finer-grained interactions (e.g., word-patch alignment) between the two modalities. FILIP applies a **cross-modal late interaction** to model the **token-wise** cross-modal interaction.

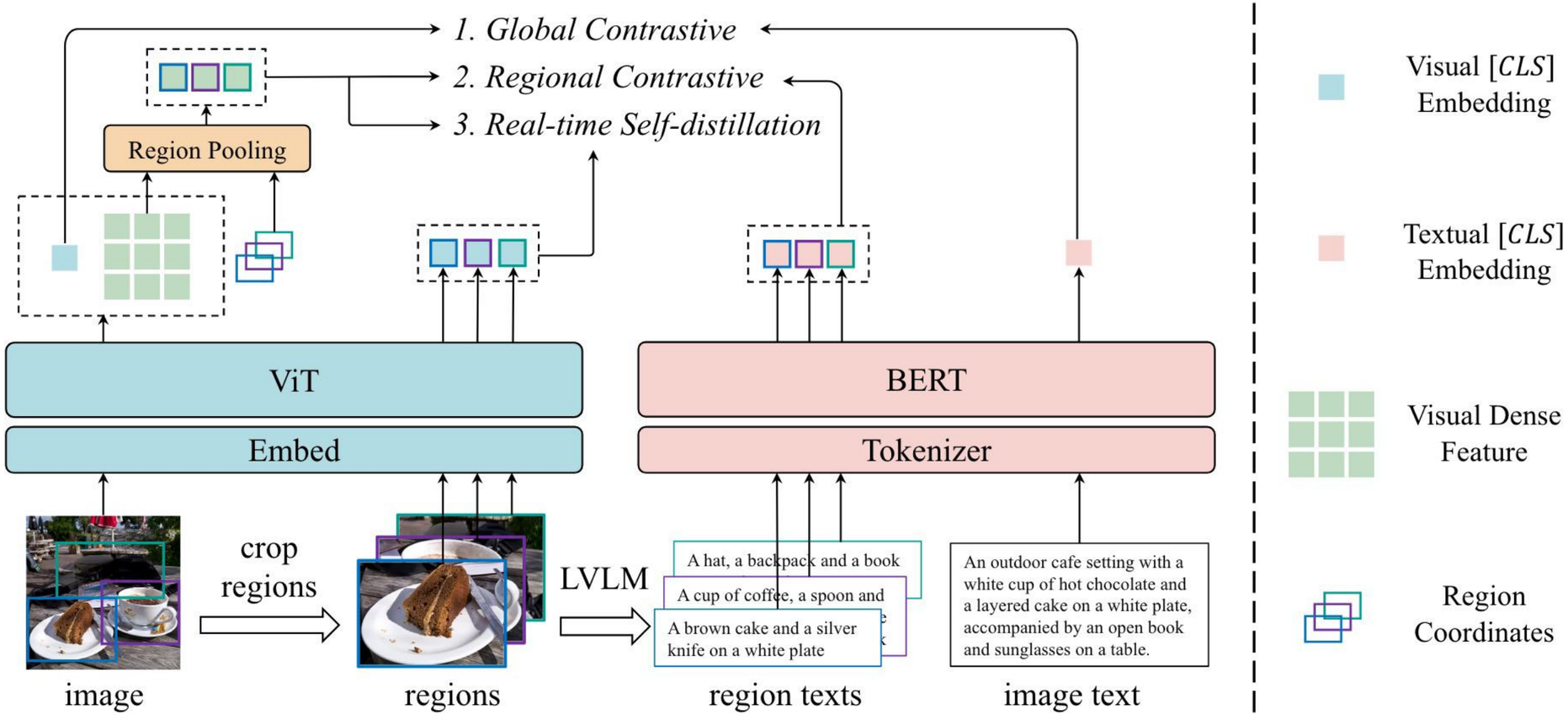
FineLIP (https://arxiv.org/abs/2504.01916 NeurIPS 2024)



$$R(I,T) = \frac{1}{P'}\sum_{i=1}^{P'}\max_j S(v'_i, t'_j) + \frac{1}{M'}\sum_{j=1}^{M'}\max_i S(t'_i, v'_j)$$

$$\mathcal{L}_{i2t} = \max(0, R(I_q, T^-) - R(I_q, T^+) + \alpha)$$
$$\mathcal{L}_{t2i} = \max(0, R(T_q, I^-) - R(T_q, I^+) + \alpha)$$
$$\mathcal{L}_{triplet} = \mathcal{L}_{i2t} + \mathcal{L}_{t2i}$$

FineCLIP（https://arxiv.org/abs/2504.01916）



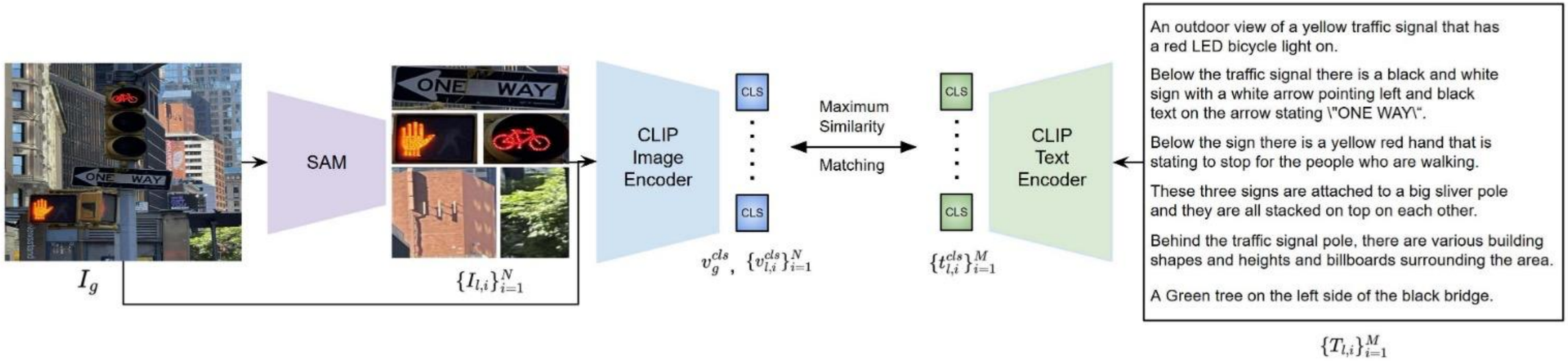
$$L_{GC} = -\frac{1}{2\mathcal{B}}\sum_{i=1}^{\mathcal{B}}(\log\frac{\exp(S(v_i,t_i)/\tau)}{\sum_{j=1}^{\mathcal{B}}\exp(S(v_i,t_j)/\tau)} + \log\frac{\exp(S(t_i,v_i)/\tau)}{\sum_{j=1}^{\mathcal{B}}\exp(S(t_i,v_j)/\tau)}),$$

$$L_{SD} = \frac{1}{\mathcal{M}}\sum_{j=1}^{\mathcal{M}}(1 - S(p_j^r, v_j^r)).$$

$$L_{RC} = -\frac{1}{2\mathcal{M}}\sum_{i=1}^{\mathcal{M}}(\log\frac{\exp(S(p_i^r,t_i^r)/\tau)}{\sum_{j=1}^{\mathcal{M}}\exp(S(p_i^r,t_j^r)/\tau)} + \log\frac{\exp(S(t_i^r,p_i^r)/\tau)}{\sum_{j=1}^{\mathcal{M}}\exp(S(t_i^r,p_j^r)/\tau)}).$$

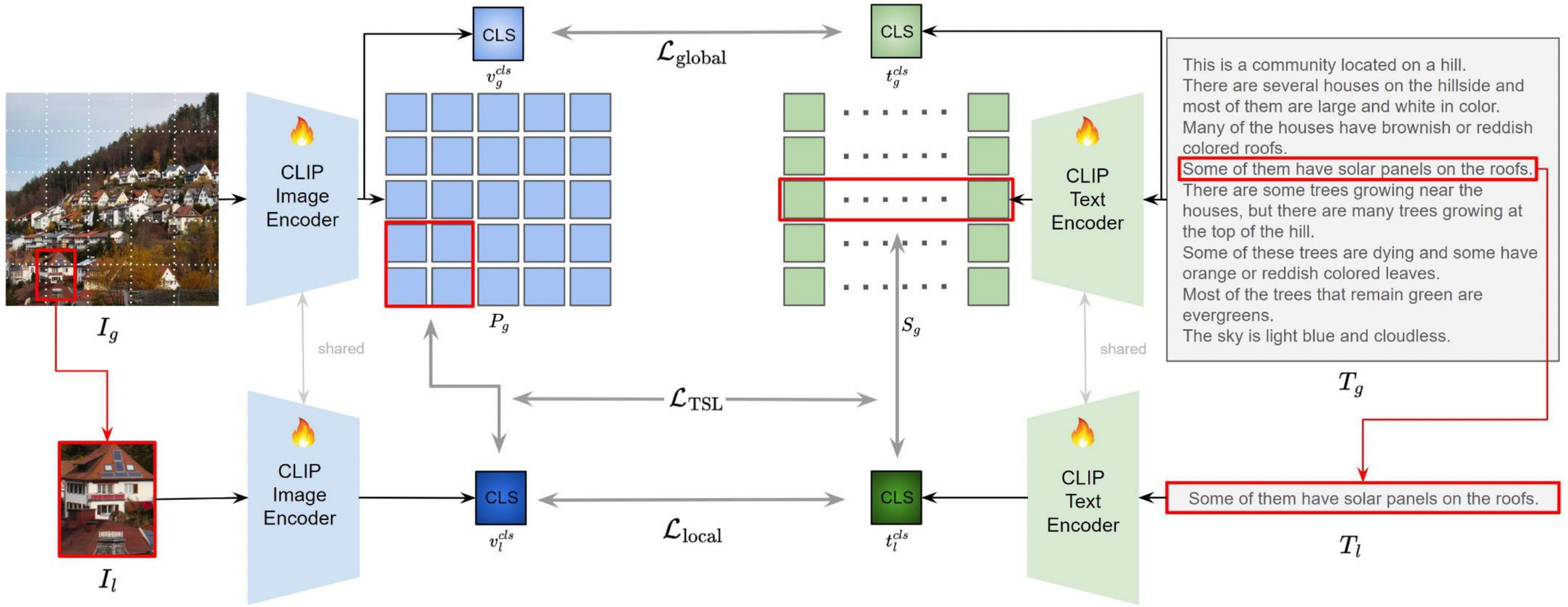
$$L = L_{GC} + \lambda * L_{SD} + \gamma * L_{RC},$$

GOAL（https://arxiv.org/abs/2503.17782）



Given a global image and its detailed caption, GOAL uses SAM to segment the image into local regions and splits the caption into individual sentences. These local pairs are then processed through CLIP encoders to obtain CLS embeddings, which are used for maximum similarity matching to identify the most relevant image-sentence pairs.

GOAL (https://arxiv.org/abs/2503.17782)



$$\mathcal{L}_{\text{global}} = \mathcal{L}_{\text{contrast}}(v_g^{cls}, t_g^{cls}) \quad \mathcal{L}_{\text{local}} = \mathcal{L}_{\text{contrast}}(v_l^{cls}, t_l^{cls})$$

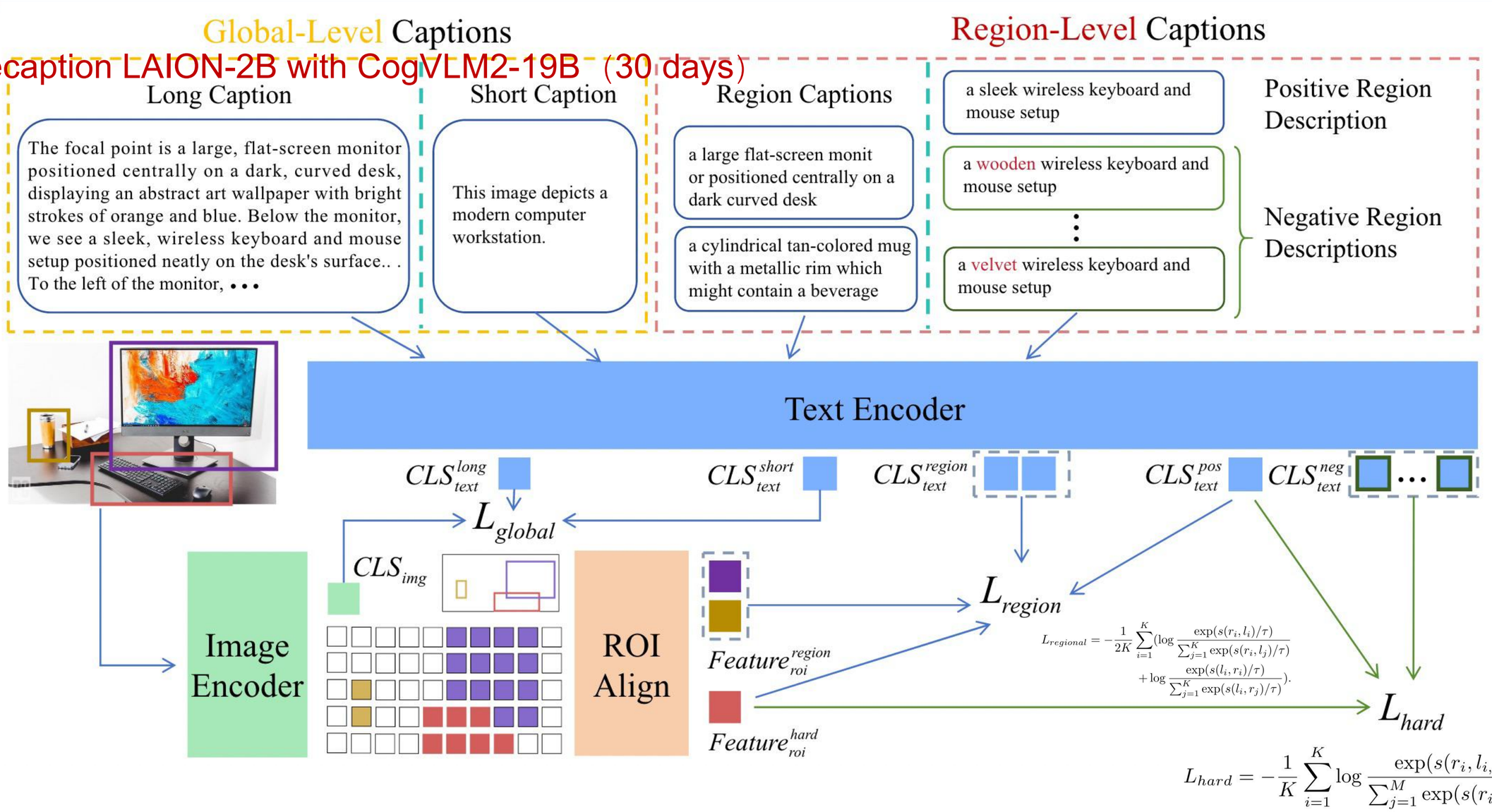
$$\mathcal{L}_{\text{TSL}} = \text{MSE}(sim(\hat{P}_l, v_l^{cls}), \mathbf{1}) + \text{MSE}(sim(\hat{S}_l, t_l^{cls}), \mathbf{1})$$

$$\mathcal{L}_{\text{total}} = \lambda_{global}\mathcal{L}_{\text{global}} + \lambda_{local}\mathcal{L}_{\text{local}} + \lambda_{TSL}\mathcal{L}_{\text{TSL}},$$

FG-CLIP (https://arxiv.org/abs/2505.05071 ICML 2025)

Modify attributes of bounding box descriptions with Llama-3.1-70B (10 neg 7d)

Recaption LAION-2B with CogVLM2-19B (30 days)



FG-CLIP（https://arxiv.org/abs/2505.05071 ICML 2025）

GOAL

Given that each local image region I_l has its bounding box coordinates (x_1, y_1, x_2, y_2) obtained from LISM in the global image I_g, we can leverage this spatial information to identify specific patch tokens from P_g that correspond to the local image region, filtering out patches from other parts of the global image. Let $\mathcal{B}$ denote the set of indices of patch tokens located inside the bounding box. We aggregate these tokens using average pooling to capture comprehensive information from the selected region:

FineCLIP/FG_CLIP：RoIAlign

ative samples. We curate the overall dataset based on GRIT (Peng et al., 2024) images. The process begins with generating detailed image captions using CogVLM2-19B (Hong et al., 2024), ensuring comprehensive and nuanced descriptions that capture the full context of each image. Following (Peng et al., 2024), we then use SpaCy (Honnibal et al., 2020) to parse the captions and extract the referring expressions. Subsequently, the images and referring expressions are fed into the pretrained object detection model, i.e., Yolo-World (Cheng et al., 2024) to obtain the associated bounding boxes. Non-maximum suppression is applied to eliminate overlapping bounding boxes, retaining only those with predicted confidence scores higher than 0.4. This process results in 12 million images and 40 million bounding boxes with fine-grained region captions. We provide examples of the images and their corresponding captions in Appendix A.