A Technical Analysis of Multimodal Large Language Models

Networked Intelligence for Comprehensive Efficiency (NICE) Lab
College of Information Science and Electronic Engineering
Zhejiang University
http://nice.rongpeng.info/

Sep. 8, 2025

Content

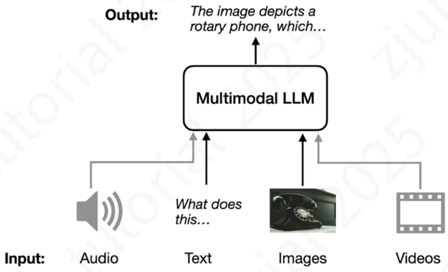
1 Core Architectures and Principles of MLLM
- Background
- The Three Core Components of MLLMs
- Comparison of Mainstream Architectural Paradigms
- Key Technique: Modality Interface Design
- Training Method

2 Classical and Advanced Models
- Feature-level fusion (Flamingo)
- Token-level fusion (BLIP-2)
- Efficient Processing

3 Conclusion

Core Architectures and Principles of MLLM

Background: What is an MLLM?



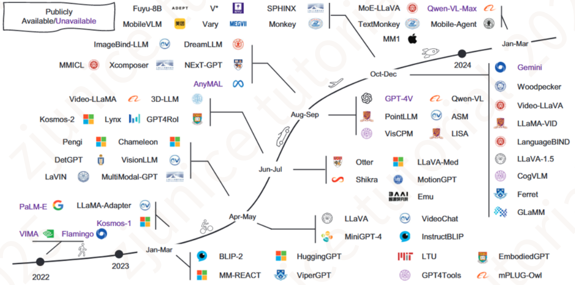
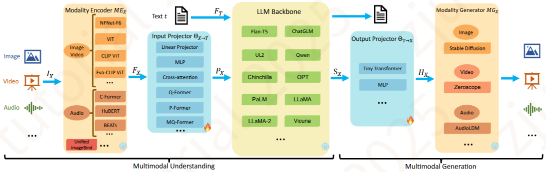


Figure: An illustration of a multimodal LLM that can accept different input modalities (audio, text, images, and videos) and returns text as the output modality.

- Notable Models: LLaVA, Flamingo, BLIP-2, ImageBind, Gemini, Qwen2-VL, InternVL2.5/3.

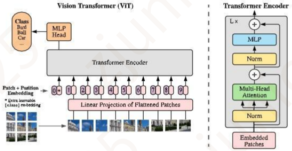
Core Components of MLLMs



A. Modality Encoder: Pre-trained unimodal encoders. (e.g., ViT, CLIP ViT)

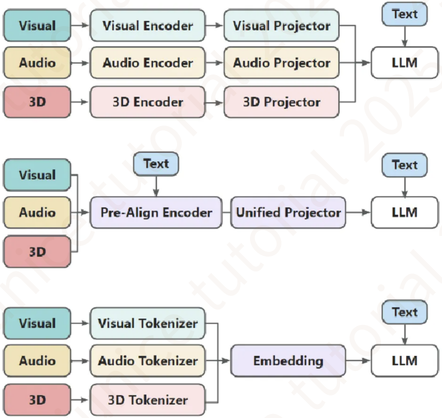
B. LLM Backbone: Pre-trained Large Language Model for reasoning. (e.g., LLaMA, Flan-T5)

C. Modality Interface / Connector: Connects encoder outputs to LLM input space. **Critical Component**.



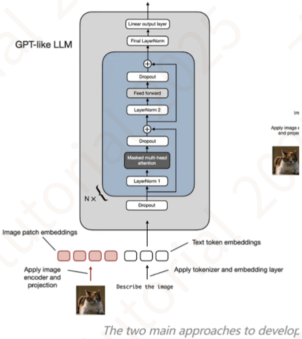
Modality Alignment Philosophies

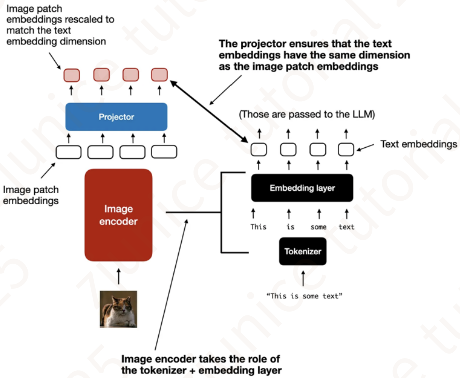
MLLMs must align features by mapping the outputs of Modality Encoders into the LLM's embedding space. This ensures information from different modalities can be understood.



1. **Multi-Branch Projection**: Separate encoders and projectors per modality.
2. **Single-Branch Unified Projection**: Align modalities to a shared space pre-LLM. (e.g., ImageBind)
3. **Discrete Tokenization / Embedding**: Convert modalities into discrete tokens for LLM. (e.g., Gemini)

Architectural Paradigm A: Unified Embedding Decoder

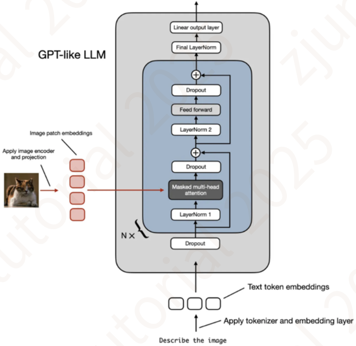


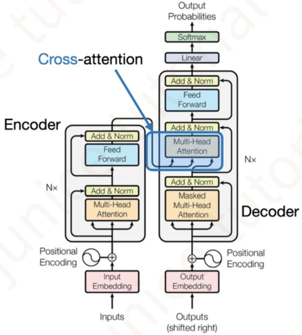


A: Unified Embedding Decoder Architecture

- **Concept**: Image as a "Foreign Language"
- **Mechanism**: Concatenate visual and text embeddings and feed them into the LLM.
- **Key Component**: "Projector" for Dimensionality Alignment

Architectural Paradigm B: Cross-Modality Attention



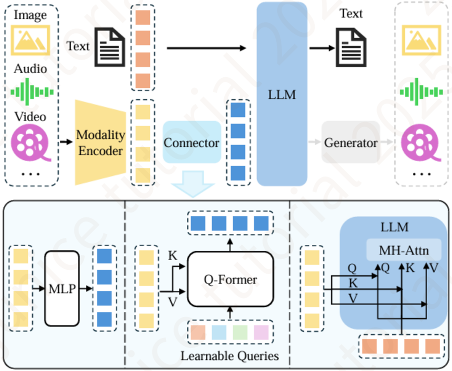


B: Cross-Modality Attention Architecture

- **Concept**: Inject Vision into LLM Layers
- **Mechanism**: Fuse Modalities via Cross-Attention inside LLM
- **Key Component**: Text (Query) + Image (Key/Value)

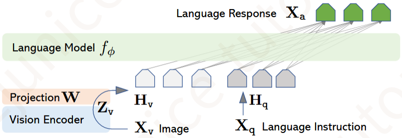
Key Technique: Modality Interface Architectures

Core Task: Feature Space Alignment



Common Implementations:

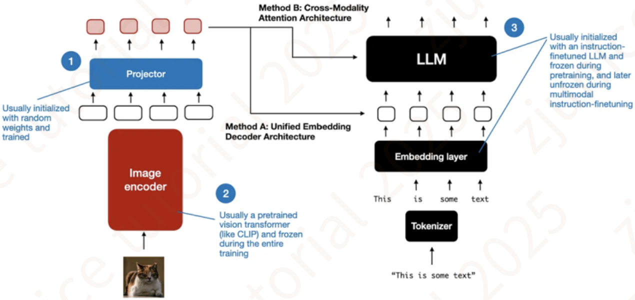
1. **Simple Projection** (e.g., LLaVA1.5, Deepseek-VL): Direct dimensional mapping. Parameter-efficient.



2. **Cross-Attention** (e.g., Flamingo, Qwen-VL): Compresses features into fixed-length queries.
3. **Q-Former** (e.g., BLIP-2, InstructBLIP): A transformer-based module to extract visual features relevant to the text.

Yin, Shukang et al., "A survey on multimodal large language models," National Science Review, 2024.

Training Method

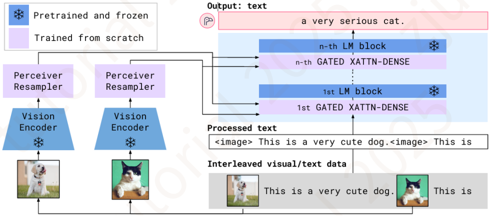


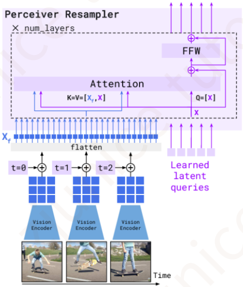
An overview of the different components in a multimodal LLM. The components numbered 1-3 can be frozen or unfrozen during the multimodal training process.

1. **Pre-training: Alignment**
   - **Goal**: Align vision and language.
   - **Method**: Train Projector (1) only. Freeze Encoder (2) and LLM (3).
2. **Instruction Finetuning: Capability**
   - **Goal**: Follow complex, specific instructions.
   - **Method**: Unfreeze LLM (3).
3. **Alignment Tuning: Preference**
   - **Goal**: Align with human values (safety, helpfulness).
   - **Method**: Refine using RLHF or DPO.

Classical and Advanced Models

Feature-level fusion (Flamingo)






```python
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
```

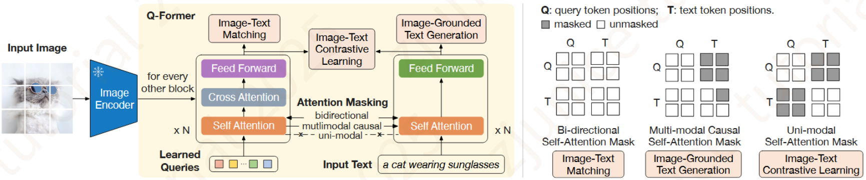
- **Visual Encoder**: Normalizer-Free ResNet (F6) with CLIP loss.
- **Perceiver Resampler**: Compresses large feature maps to a few visual tokens using cross-attention.
- Adopts a Cross-Modality Attention architecture.

Alayrac, Jean-Baptiste et al., “Flamingo: a visual language model for few-shot learning,” Advances in Neural Information Processing Systems, vol. 35, pp. 23716–23736, 2022.

Token-level fusion (BLIP-2)

Bridge Modality Gaps via Comprehensive Pre-training

Key Technique: Two-Stage Pre-training with Q-Former



Stage 1: Representation Learning

- Image-Text Contrastive (ITC)
- Image-Text Matching (ITM)
- Image-grounded Text Generation (ITG)

$$\mathcal{L}_{itc} = \mathcal{H}(\mathbf{p}^{itc}(I), \mathbf{y}^{itc}(I)) + \mathcal{H}(\mathbf{p}^{itc}(T), \mathbf{y}^{itc}(T))$$

$$\mathcal{L}_{itm} = \mathcal{H}(\mathbf{p}^{itm}(I, T), \mathbf{y}^{itm}(I, T))$$

$$\mathcal{L}_{itg} = -\mathbb{E}_{I,t} \log P(T|I)$$

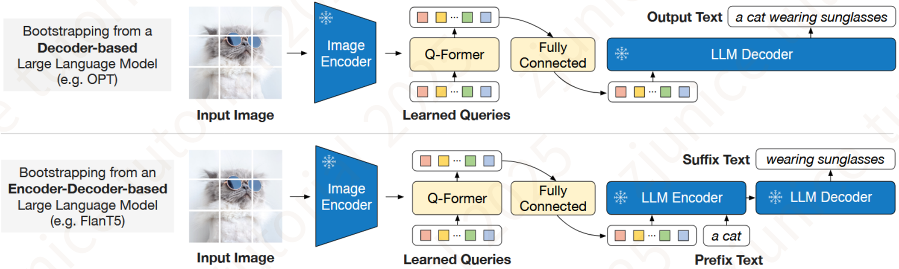
Li, Junnan et al., “BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” in International Conference on

Token-level fusion (BLIP-2)

Bridge Modality Gaps via Comprehensive Pre-training

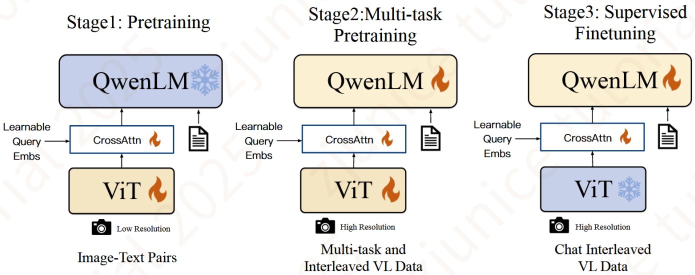
Stage 2: Generative Learning

- Connects the trained Q-Former to a frozen LLM to bootstrap vision-language generative capabilities.



Li, Junnan et al., “BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” in International Conference on Machine Learning (ICML), pp. 19730–19742, 2023.

Efficient Processing (Qwen-VL)



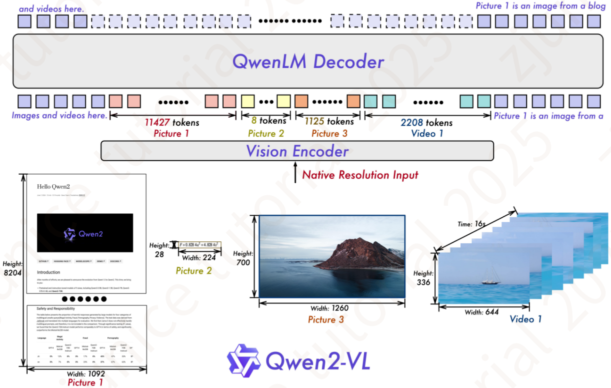
Cross-Modality Attention architecture
Synthesizing Pre-training Robustness with SFT Alignment

- **Stage 1 - Pre-training:** General feature alignment on large-scale web data.
- **Stage 2 - Multi-task Pre-training:** Developing specific capabilities on high-quality annotated data.
- **Stage 3 - Supervised Fine-tuning (SFT):** User alignment with instruction data.

Bai, Jinze et al., "Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond," arXiv:2308.12966, 2023.

Efficient Processing (Qwen2-VL)

Objective: Enhance Perception of the World at Any Resolution



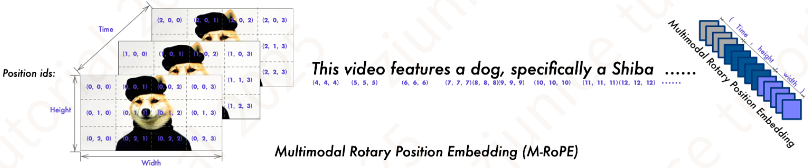
Key Innovations:

- Naive Dynamic Resolution
- Multimodal Rotary Position Embedding (M-RoPE)
- Unified Image and Video Paradigm

Wang, Peng et al., “Qwen2-VL: Enhancing vision-language model’s perception of the world at any resolution,” arXiv:2409.12191, 2024.

Efficient Processing (Qwen2-VL): M-RoPE

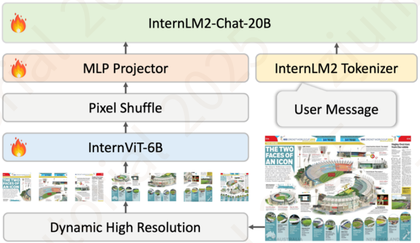
Key Innovation: Multimodal Rotary Position Embedding (M-RoPE)

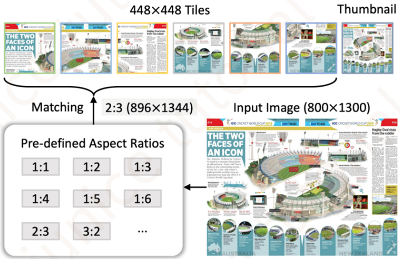


- **Problem**: Traditional 1D position embeddings (like RoPE) cannot effectively model the spatio-temporal nature of video.
- **Solution**: M-RoPE deconstructs the rotary embedding into three components: temporal, height, and width.
  - For images, the temporal ID is constant.
  - For videos, the temporal ID increments for each frame.

Wang, Peng et al., "Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution," arXiv:2409.12191, 2024.

Efficient Processing (InternVL1.5)





Key Features

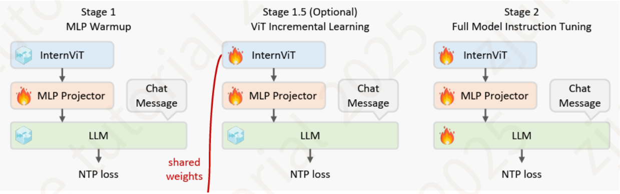
- **Simplified Architecture**: ViT-MLP-LLM with a simple MLP as the vision-language bridge.
- **Dynamic High-Resolution**: Supports high-resolution (448x448) input. Large images are tiled into smaller patches and processed sequentially.
- **Two-Stage Training**: Pre-trains the ViT and MLP connector, then fine-tunes all parameters for full-capability alignment.

Chen, Zhe et al., "How Far Are We to GPT-4V? Closing the Gap to Commercial Multimodal Models with Open-Source Suites," arXiv:2404.16821, 2024.

Efficient Processing (InternVL 2.5)

InternVL 2.5: Optimized Training

- **Refined 3-Stage Training**:
  - Stage 1: Train MLP only.
  - Stage 1.5: Train InternViT + MLP.
  - Stage 2: Train all parameters.
- **Data Optimization**: Expanded fine-tuning dataset and implemented data cleaning.
- **Test-Time Scaling**: Introduces Chain-of-Thought (CoT) prompting at inference time.



Chen, Zhe et al., "Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling," arXiv:2412.05271, 2025.

Efficient Processing (InternVL 3)

InternVL 3: Advanced Techniques

- **Core Tech 1: Flexible Long-Sequence Handling with V2PE**
  - **Challenge**: Long visual sequences can exceed an LLM's context window.
  - **Solution (V2PE)**: Assigns smaller, fractional position increments (δ) to visual tokens, allowing more visual information to fit within the context window.
- **Core Tech 2: Mixed Preference Optimization (MPO)**
  - **Objective**: Enhance complex reasoning and align with human preferences.
  - **Methodology**: A combined loss function using preference, quality, and generation losses.

Zhu, Jinguo et al., "InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models," arXiv:2504.10479, 2025.

Conclusion

Conclusion

- We explored the two mainstream MLLM paradigms: the Unified Embedding Decoder and the Cross-Modality Attention architecture. We also delved into multi-stage training strategies.
- Key trends identified from models like Qwen-VL and InternVL include:
  - **Architectural Simplification** (e.g., using simple MLP projectors).
  - **Enhanced Input Processing** (e.g., support for dynamic, high-resolution, and long visual sequences).
  - **Continuous Optimization of Training Strategies**.

Thank you

Networked Intelligence for Comprehensive Efficiency (NICE) Lab
College of Information Science and Electronic Engineering
Zhejiang University
https://nice.rongpeng.info