From Multimodal LLM to Human-level AI

Modality, *Instruction*, *Reasoning*, *Efficiency* and **Beyond**

https://mllm2024.github.io/COLING2024

LREC-COLING 2024

CC BY 4.0 DEED
Attribution 4.0 International

This Keynote slide is licensed under a CC BY 4.0 International License.

NUS National University of Singapore

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

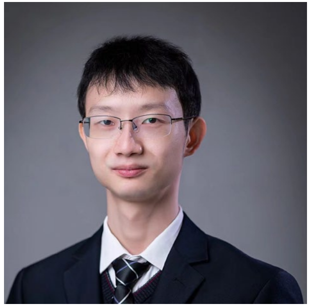
UNIVERSITY OF MARYLAND

Hao Fei

National University of Singapore

Yuan Yao

National University of Singapore



Zhuosheng Zhang

Shanghai Jiao Tong University

Fuxiao Liu

University of Maryland, College Park

Ao Zhang

National University of Singapore

Tat-Seng Chua

National University of Singapore

Part-II

Efficient MLLM

Ao Zhang

PhD Student

National University of Singapore

https://waxnkw.github.io/

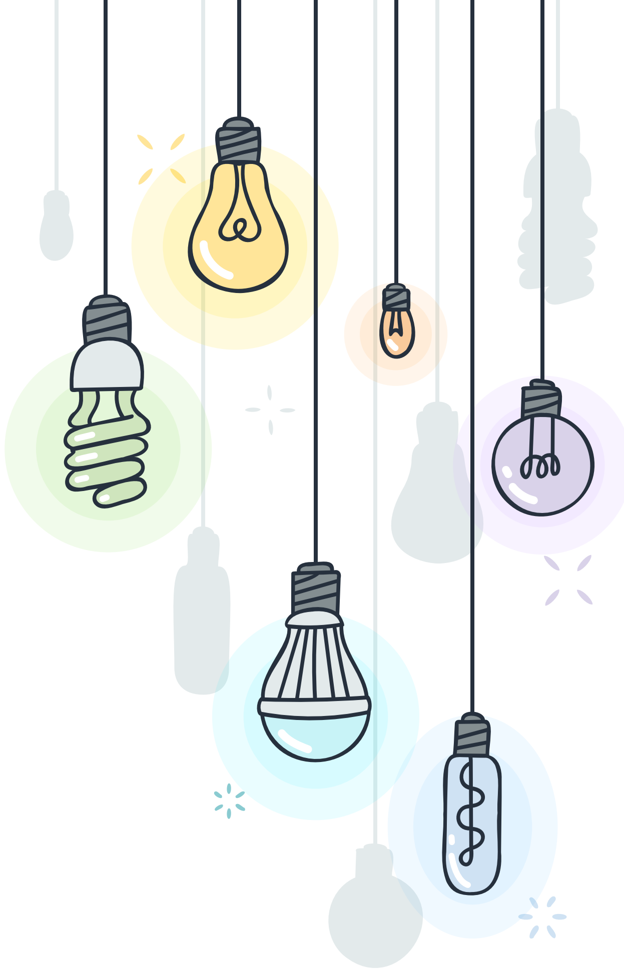


Table of Content

1 Efficient MLLM

1 Efficient MLLM

Overview

- ## What do you mean by saying efficient MLLM?

Given a target performance, we want to reduce the cost for **training** and **inference**.

Architecture: some architectures are more efficient.

Data: data source and arangement are important

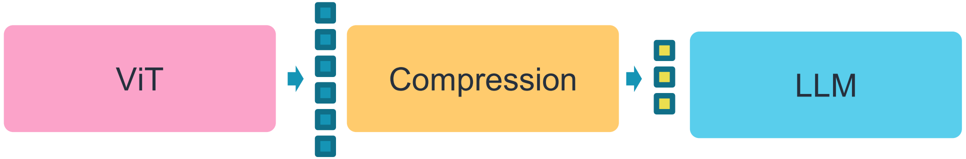
Training Strategy: use transfer learning or connect to pre-trained tools

Acceleration Techs: use Deepspeed for training acceleration

Architecture

- ## Visual encoding

High-resolution is a key factor for MLLM’s performance。
But high-res lead to significantly more tokens.



higher resolution → more tokens

higher resolution → key factor for performance

CLIP ViT-L/14

res.	#tokens
224x224	256
336x336	576
448x448	**1024**

Architecture

- ## Visual encoding

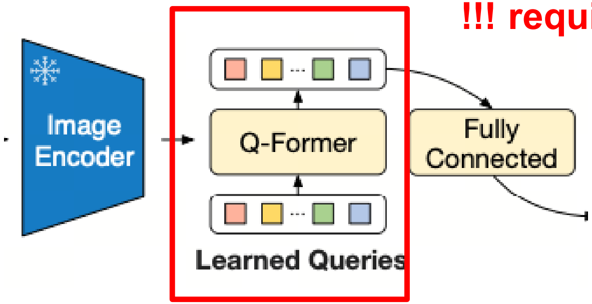
The innovation for efficiency in architecture mainly lies on visual encoding.

Old models

Model	Compression	# tokens
BLIP-2	QFormer	32
LLaVA-1.5	No	256
LLaVA-1.6	**No**	**576**

too much params (BERT)

!!! requires pre-training

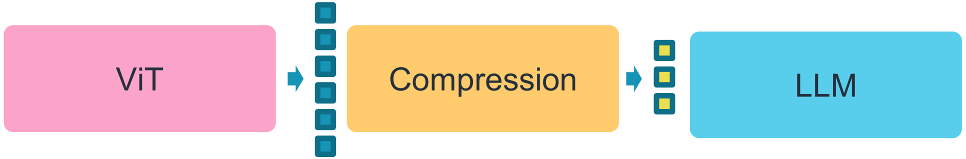


[1] BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. 2023
[2] Visual Instruction Tuning. 2023

Architecture

- Visual encoding

Solution: light-weight compression layer.

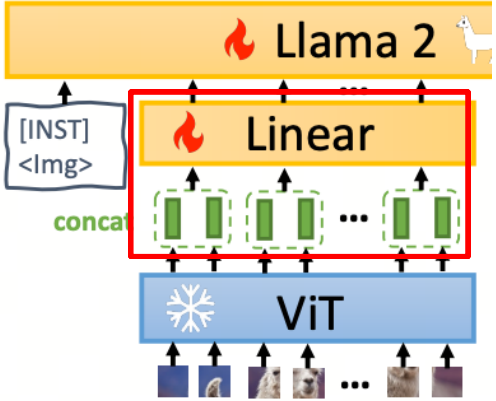


new models

Qwen-VL
MiniCPM-V2 } **1 layer** cross-attention

MiniGPT-v2 merge adjacent tokens with **Linear**

CogAgent low-res feature as queries



MiniGPT-v2

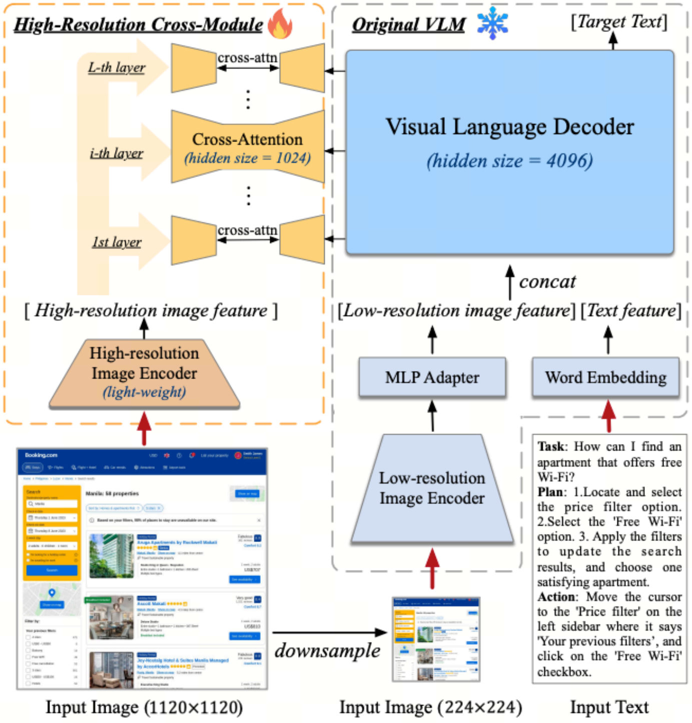
Architecture

- ## Visual encoding

  Solution: light-weight compression layer.

low-res feature as queries

CogAgent

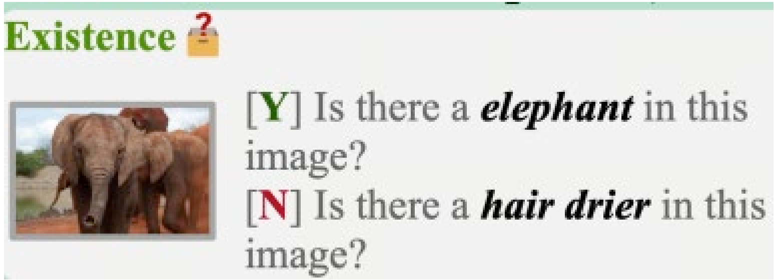


Data

- ## High quality data

The usage of human-annotated data signicantly boost the MLLM's ability.

Type	Examples	Data	MME
Old models	BLIP-2, VL-Vicuna...	Captioning	1293.84
New models	InstructBLIP, LLaVA-1.5	VQAv2, GQA...	1531.31



Example of MME

[1] MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. 2023

Data

- ## High quality data

Recommendation of some high-quality datasets.

General VQA	OCR	Instruction Tuning	Region Understanding	Pure Text
VQAv2 GQA A-OKVQA OK-VQA ScienceQA VGQA	TextCaps OCR-VQA DocVQA TextVQA ArxivQA	ShareGPT4V LRV ALLaVA All-Seeing V2 LLaVA-Instruct	RefCOCO series Flickr-30K VCR All-Seeing V2 LVIS GranD PSG ADE20K	ShareGPT Ultra-Chat

Data

High quality data

Data ratio is important but limited works on how to set it.

An empirical experience is: higher ratio for **data with long text, VQA, and OCR (or other ability you want)**

open source VS closed source

Benchmark	InternVL 1.5	Grok-1.5V	GPT-4V	Claude-3 Opus	Gemini Pro 1.5
MMMU Multi-discipline	45.2%	53.6%	56.8%	**59.4%**	58.5%
MathVista Math	**53.5%**	52.8%	49.9%	50.5%	52.1%
AI2D Diagrams	80.7%	88.3%	78.2%	**88.1%**	80.3%
TextVQA Text reading	**80.6%**	78.1%	78.0%	-	73.5%
ChartQA Charts	**83.8%**	76.1%	78.5%	80.8%	81.3%
DocVQA Documents	**90.9%**	85.6%	88.4%	89.3%	86.5%
RealWorldQA Real-world understanding	66.0%	**68.7%**	61.4%	49.8%	67.5%

Intern-VL

task	ratio	dataset
Captioning	53.9%	Laion-EN (en) [93], Laion-ZH (zh) [93], COYO (zh) [10], GRIT (zh) [90], COCO (en) [17], TextCaps (en) [99]
Detection	5.2%	Objects365 (en&zh) [97], GRIT (en&zh) [90], All-Seeing (en&zh) [119]
OCR (large)	32.0%	Wukong-OCR (zh) [29], LaionCOCO-OCR (en) [94], Common Crawl PDF (en&zh)
OCR (small)	8.9%	MMC-Inst (en) [61], LSVT (zh) [105], ST-VQA (en) [9] RCTW-17 (zh) [98], ReCTs (zh) [137], ArT (en&zh) [19], SynthDoG (en&zh) [41], COCO-Text (en) [114], ChartQA (en) [81], CTW (zh) [134], DocVQA (en) [82], TextOCR (en) [101], PlotQA (en) [85], InfoVQA (en) [83]

(a) Datasets used in the pre-training stage.

[1] InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. 2023

Data

High quality data

Data ratio is important but limited works on how to set it.

An empirical experience is: higher ratio for **data with long text, VQA, and then OCR (or other ability you want).**

Table 8: Results on general multimodal benchmarks.

Model	Size	Open-Compass	MME	MMB dev(en)	MMB dev(zh)	MMMU val	Math-Vista	LLaVA Bench	Object HalBench
Proprietary									
Gemini Pro	-	63.8	2148.9	75.2	74.0	48.9	45.8	79.9	-
GPT-4V	-	63.2	1771.5	75.1	75.0	53.8	47.8	93.1	86.4 / 92.7
Open-source 6B~34B									
Yi-VL-6B	6.7B	49.3	1915.1	68.6	68.3	40.3	28.8	51.9	-
Qwen-VL-Chat	9.6B	52.1	1860.0	60.6	56.7	37.0	33.8	67.7	56.2 / 80.0
Yi-VL-34B	34B	52.6	2050.2	71.1	71.4	45.1	30.7	62.3	-
DeepSeek-VL-7B	7.3B	55.6	1765.4	74.1	72.8	38.3	36.8	77.8	-
CogVLM-Chat	17.4B	52.5	1736.6	63.7	53.8	37.3	34.7	73.9	73.6 / 87.4
Open-source 2B~3B									
DeepSeek-VL-1.3B	1.7B	46.0	1531.6	64.0	61.2	33.8	29.4	51.1	-
MobileVLM V2	3.1B	-	1440.5(P)	63.2	-	-	-	-	-
Mini-Gemini	2.2B	-	1653.0	59.8	-	31.7	-	-	-
MiniCPM-V1	2.8B	47.6	1650.2	67.9	65.3	**38.3**	28.9	51.3	78.4 / 88.5
MiniCPM-V2	2.8B	**55.0**	**1808.6**	**69.6**	**68.1**	38.2	**38.7**	**69.2**	**85.5 / 92.2**

Table 7: Results on OCR-specific benchmarks.

Model	Size	OCRBench	TextVQA val	DocVQA test
Proprietary				
Gemini Pro	-	680	74.6	88.1
GPT-4V	-	645	78.0	88.4
Open-source 6B~34B				
Yi-VL-6B	6.7B	290	45.5*	17.1*
Qwen-VL-Chat	9.6B	488	61.5	62.6
Yi-VL-34B	34B	290	43.4*	16.9*
DeepSeek-VL-7B	7.3B	435	64.7*	47.0*
TextMonkey	9.7B	558	64.3	66.7
CogVLM-Chat	17.4B	590	70.4	33.3*
Open-source 2B~3B				
DeepSeek-VL-1.3B	1.7B	413	58.4*	37.9*
MobileVLM V2	3.1B	-	57.5	19.4*
Mini-Gemini	2.2B	-	56.2	34.2*
MiniCPM-V1	2.8B	366	60.6	38.2
MiniCPM-V2	2.8B	**605**	**74.1**	**71.9**

MiniCPM-V

Data

- ## High quality data

Data ratio is important but limited works on how to set it.

An empirical experience is: higher ratio for **data with long text, VQA, and then OCR**

Category		Sources	Size	Ratio
Part-1	Short Caption	Flickr-30K [75], COCO [56]	560K	10.4%
	VQA	FM-IQA [29], VGQA [47], IconQA [64], GQA [39], VQAv2 [5] CLEVR [42], VizWiz [33], Visual7W [110], COCO-QA [77]	1430K	26.6%
	Knowledge	OKVQA [67], A-OKVQA [80], KVQA [81], ScienceQA [65]	60K	1.1%
	Grounding	RefCOCO [100]	570K	10.6%
	Reasoning	COMVINT [27], VCR [103], NLVR [87], LRV [57]	135K	2.5%
	Math	GeoQA [17], SMART-101 [21]	125K	2.3%
	OCR	DocVQA [69], TextVQA [84], OCR-VQA [72], ST-VQA [10], VisualMRC [89], DVQA [43] FigureQA [44], ChartQA [68], DeepForm [88], TabFact [20], InfographicsVQA [70] Kleister Charity [86], WikiTableQuestions [73], Real-CQA [2], AI2D [45], In-House-OCR	1720K	32.0%
	Chat	FSVQA [83], Visual-Dialog [25]	780K	14.5%

MiniCPM-V

Data

High quality data

Data ratio is important but limited works on how to set it.

An empirical experience is: higher ratio for **data with long text, VQA, and then OCR**

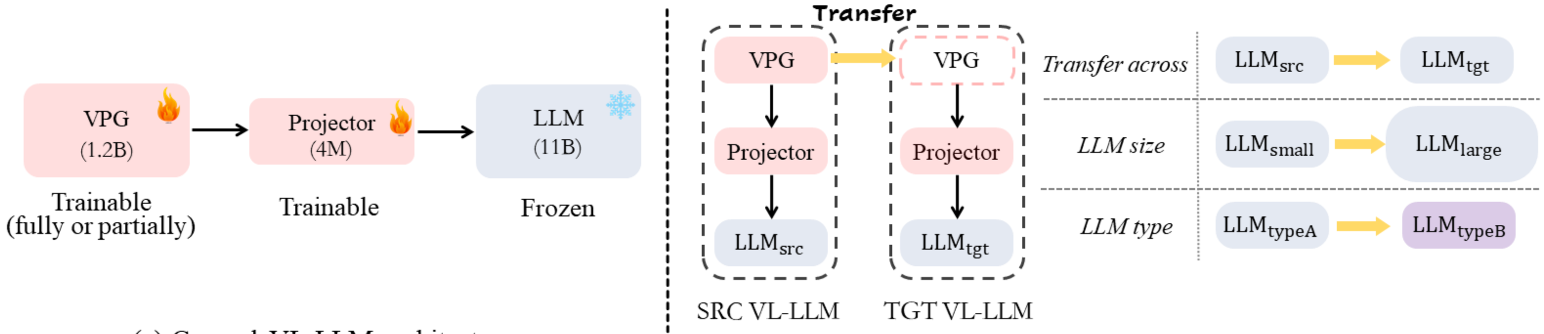
Part-2	OCR-Short	DocVQA, TextVQA, OCR-VQA, VisualMRC, ChartQA, AI2D	190K	8%
	OCR-Detail	In-House-Web, ArxivQA [53], LLaVAR [106], TextOCR-GPT4V [14], In-House-GPT4V	500K	18%
	Part-1	sample from part-1 data	400K	8%
	Instruct	SVIT [107], LLaVA-Instruct-150K [58], UniMM-Chat [101], ShareGPT4V [19] LVIS [31], ALLaVA [16]	2000K	56%
	Text-Only	Ultra-Chat [26], Alpaca [90], ShareGPT [108], BELLE [9] OpenOrca [55], OpenHermes [92], In-House-MiniCPM-SFT	-	10%

MiniCPM-V

Training Strategy

- ## Training Strategy

Transfer learning for efficient MLLM building.
Idea: transfer the visual part across LLMs.



(a) General VL-LLM architecture (representatively with a 11B LLM)

(b) VPG transfer across LLMs

[1] VPGTrans: Transfer Visual Prompt Generator across LLMs. 2023

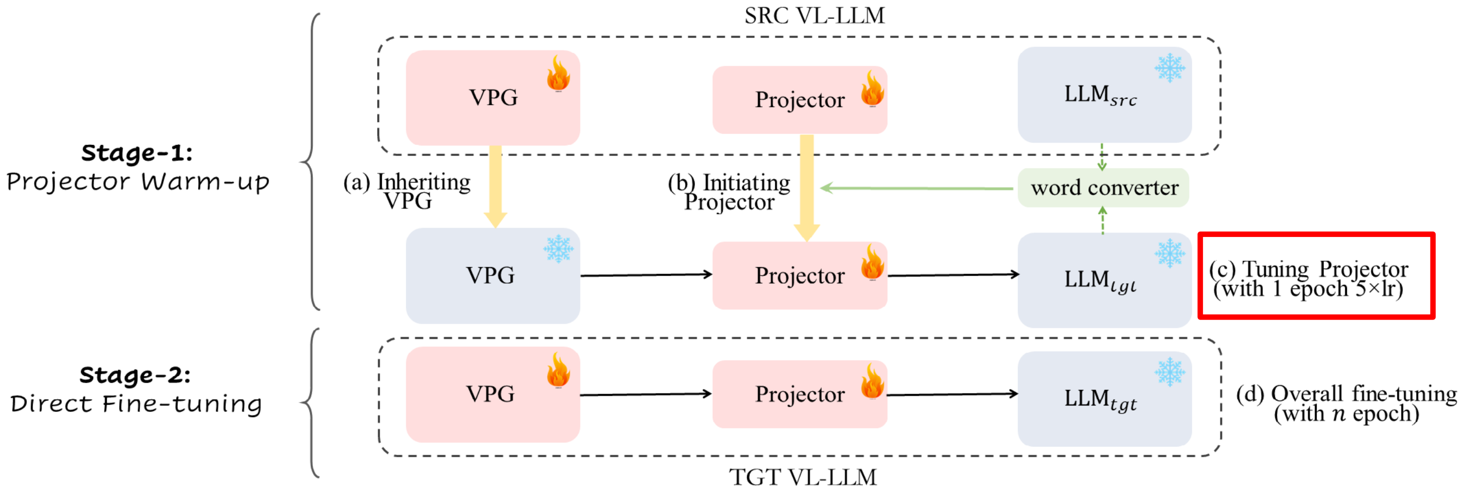
Training Strategy

- ## Training Strategy

VPGTrans:

(1) train projector with large lr

(2) normal training



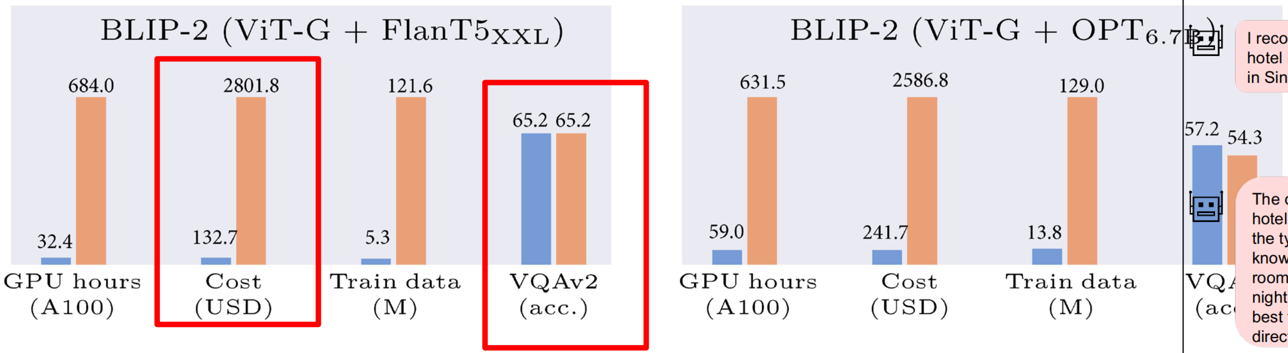
VPGTrans

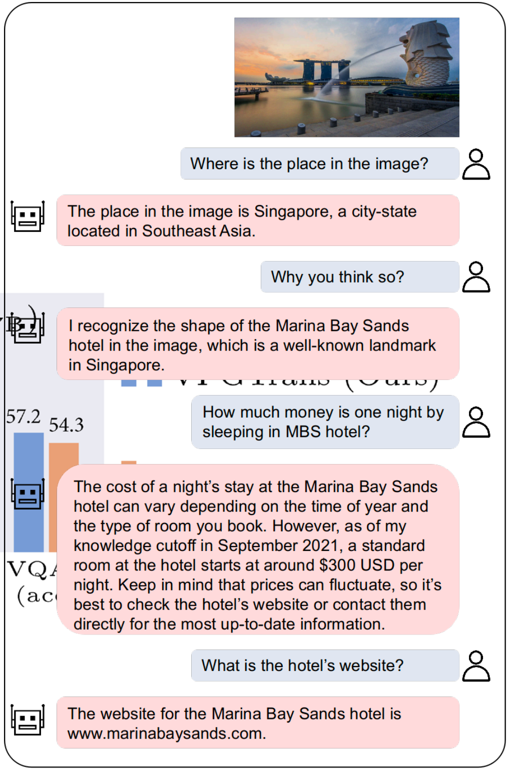
Training Strategy

- ## Training Strategy

We first validate it on BLIP-2 training.

We then transfer the visual encoder from OPT to Vicuna.

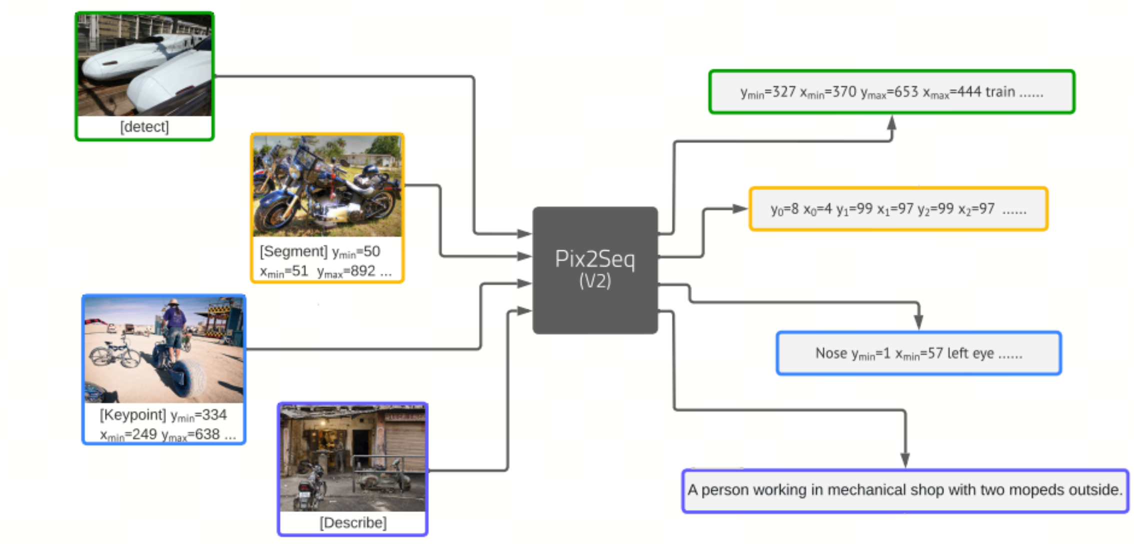




Training Strategy

- ## Training Strategy

  Pix2seq: let the MLLM to output everything as text, like **bounding boxes (detection)** and **object boundary point (segmentation)**.



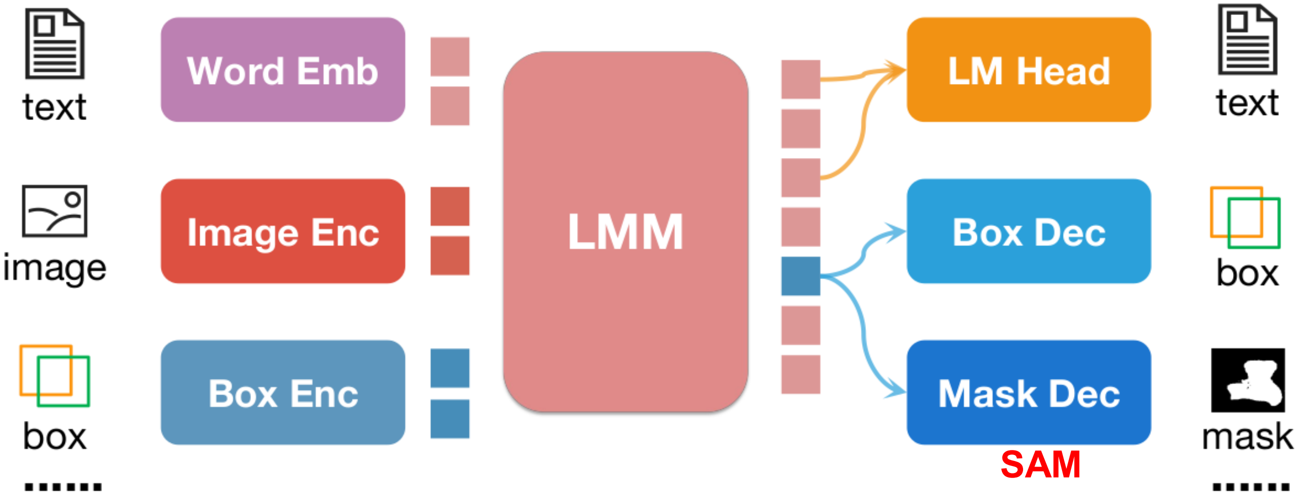
costly training!!!

[1] A Unified Sequence Interface for Vision Tasks. 2022

Training Strategy

- ## Training Strategy

Pix2emb: connecting LLM and tools with emb. for efficient function extension.

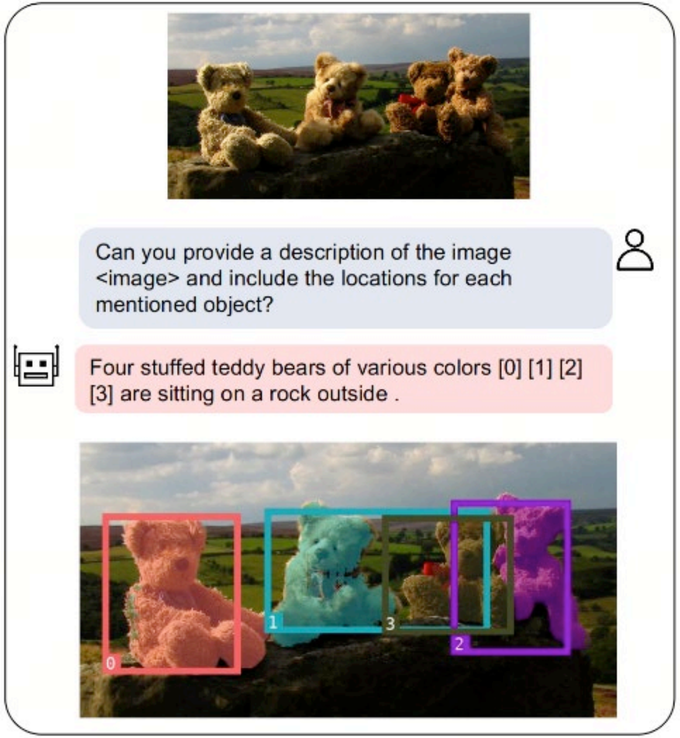


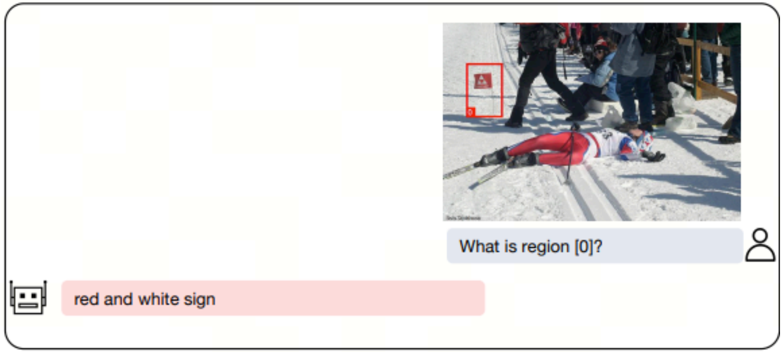
[1] *NExT-Chat: An LMM for Chat, Detection and Segmentation. 2023*

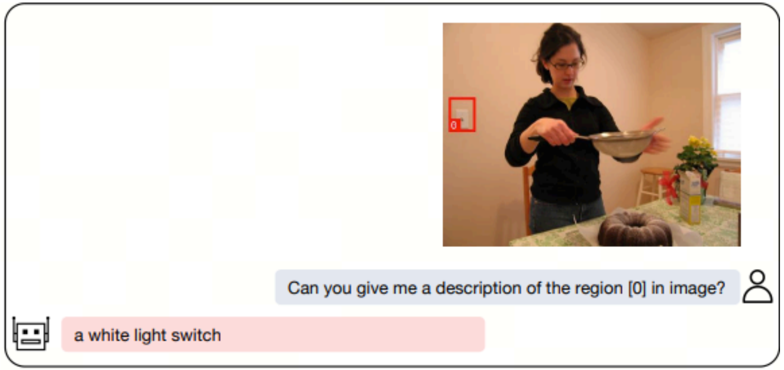
Training Strategy

- ## Training Strategy

Pix2emb: connecting LLM with tools for efficient function extension.



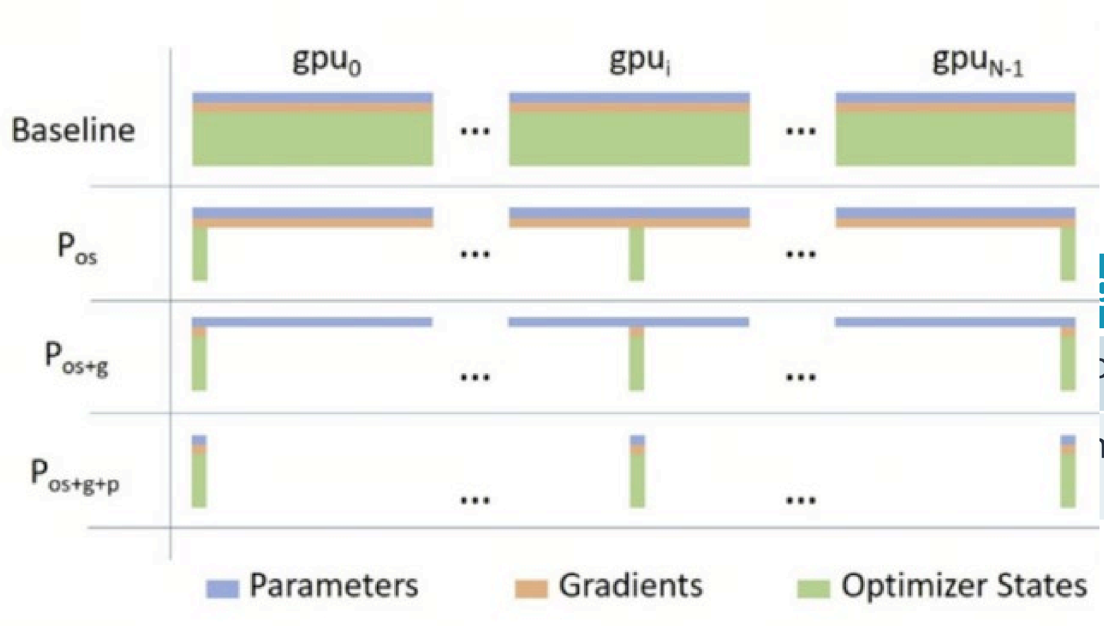




Techniques

- ## Acceleartion Techniques

DeepSpeed or FSDP: optimizer state, gradient, model parameters partitioning



ges
or LoRA
nstallation

[1] https://www.deepspeed.ai/getting-started/

Techniques

- ## Acceleartion Techniques

Other Widely Used Practice

use bfloat16

gradient checkpointing for training

quantization for inference

Data Loading

Parquet or TSV: save data items in large files for faster loading.

Pre-fetch: pre-fetch the batch before forward.

Packing: pack multiple data items into a pre-defined max length.

No packing

batch 1
Item1: XXXXXXXXXXXXXXXXXXXXXX
Item2: YY

batch 2
Item1: XXXXXXXXXXXXXXXXXXXXXX
Item2: YYYYYYYYYYYYYYYYYYYYYY

Packing

Item1+2: XXXXXXXXXXXXXXXXXXXXXXYYYYYYY

Summary

Model Architecture

high-resolution + light-weight compression layer

Data

high-quality data

high data ratio for VQA, Long-text data, data for ability you want (OCR)

Training Strategy

transfer learning, high learning rate for adaption layer (e.g. projector).

pix2emb for function extension

Techniques

Deepspeed

quantization, gradient checkpointing, bf16

parquet to avoid small files, pre-fetch, packing

Thanks!

Any questions?