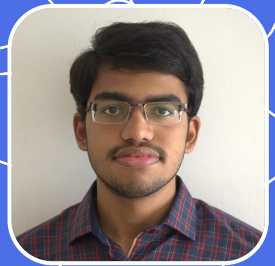

MULTIMODAL AI
SESSION #12
29th November 2024

About Us

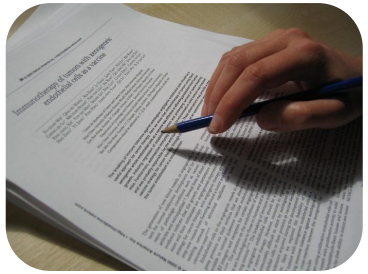
Henry Vo
@_lowkeyboi



Surya Guthikonda
@suryaguthikonda

What to expect?

#multimodal-ml - Channel
@multimodal - Role



Paper Reading Session

Guest Speaker Session

Coding Session

Second and Fourth Friday 10 AM ET

More Exciting Updates for 2025

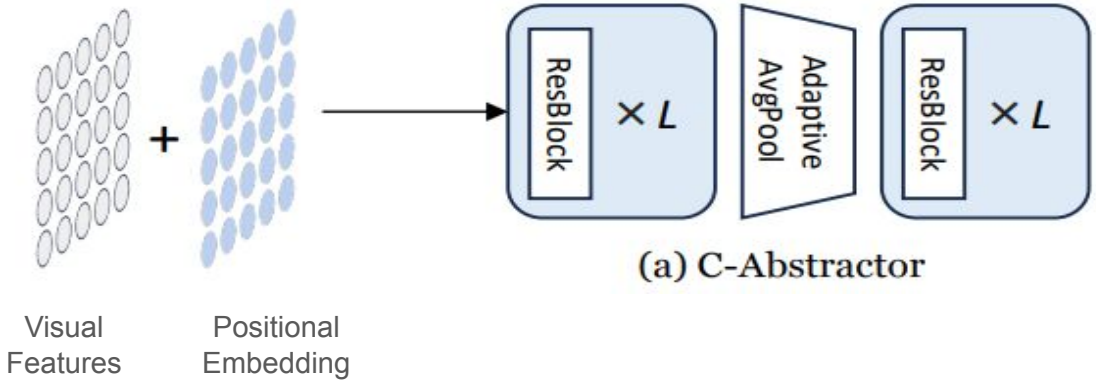
MM1.5: Methods, Analysis & Insights from Multimodal LLM Fine-tuning

Cohere For AI

Introduction

- Extends the MM1 family of models.
- MM1.5 focuses on improving the following capabilities
  - OCR
  - Visual Referring and Grounding
  - Multi-image reasoning and in-context learning.
- Introduces specialized MM1.5-Video and MM1.5-UI models.
- Analysis on data and architecture ablations.
- (Sep 2024) MM1.5 Models are SOTA open-source models in most of benchmarks.

Recap: Convolutional-Abstractor



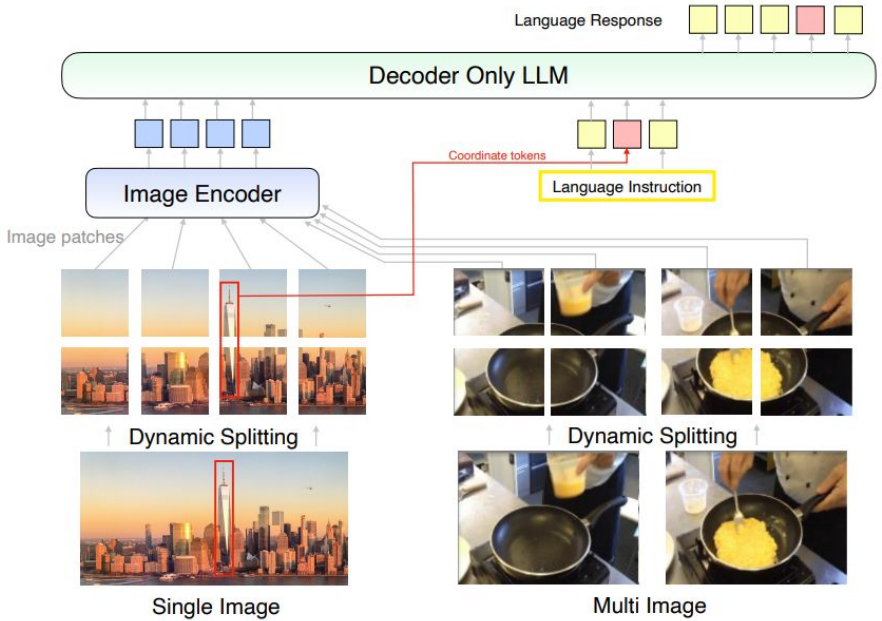
Recap: MM1 Architecture

- **Image Encoder:**
  - ViT-H (CLIP Loss on DFN-5B)
  - Image Size = 378x378
- **Vision-Language Connector:**
  - C-Abstractor with 144 Image Tokens
- **Pre-training Data**
  - Captioned Images (45%)
  - Interleaved Image-text documents (45%)
  - Text-Only (10%)
- **Language Model**
  - 3B, 7B and 30B Parameter Language Model

Recap: MM1 Insights

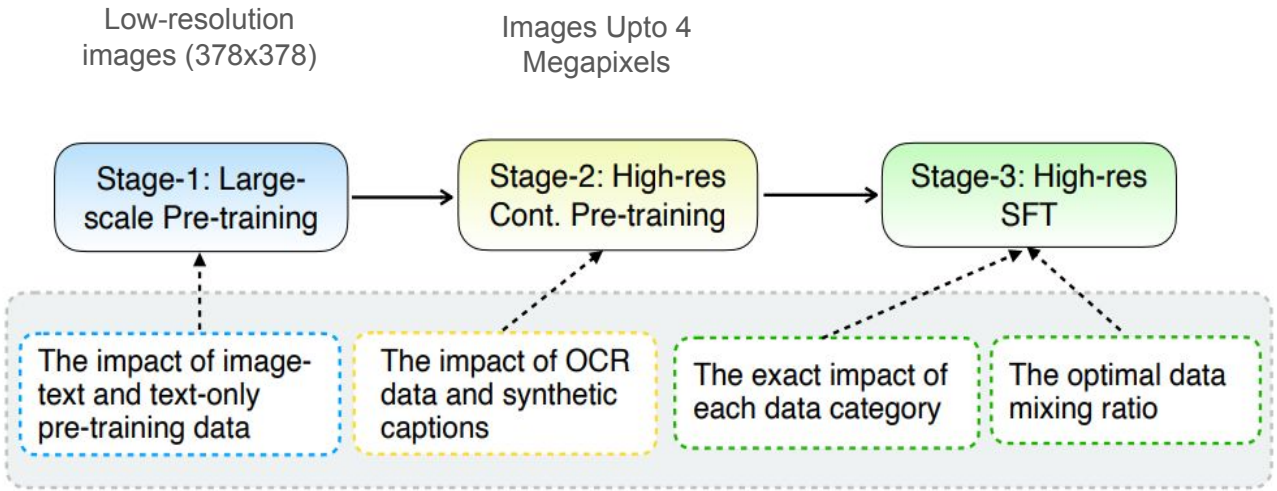
- Pre-training
  - Image resolution has the highest impact, followed by model size and training data composition.
  - Number of visual tokens and image resolution matters most, while the type of VL connector has little effect.
  - Interleaved data is instrumental for few-shot and text only performance, while captioning data lifts zero-shot performance.
  - Text-only data helps with few-shot and text-only performance.
  - Careful mixture of image and text data can yield optimal multimodal performance and retain strong text performance.
  - Synthetic data helps with few-shot learning
- SFT
  - With Increase in Image Resolution (upto certain point), the model performs better.
  - Pre-training the model with more data improves the model performance.

Model Overview



1. Text-rich image understanding with dynamic image splitting.
2. Visual Referring and Grounding with coordinate tokens
3. Multi-image reasoning

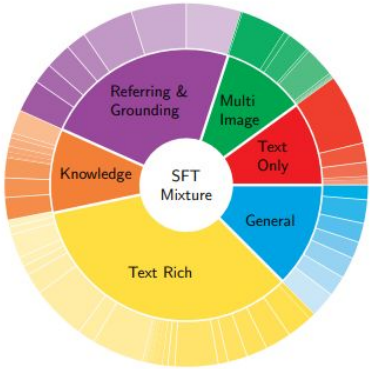
Model Training Stages



Empirical Setup for Ablations

- MM1 Architecture (3B Dense Model)
- Static Image Splitting with 4 sub-image splits + Overview Image (672x672)
  Applicable only for fewer than three images
- Batch Size of 256
- Ada Factor Optimizer with peak learning rate of 1e-5 and cosine decay of 0.
- Train max of 30k steps (Continual Pretraining) and 1 epoch (SFT)

SFT Data Categorization



General
- LLaVA v1.5 VQAv2 OKVQA [114, 100] (91.8k)
- LLaVA v1.5 A-OKVQA [133] (66.2k)
- LLaVA Complex Reasoning [102] (76.6k)
- LLaVA 1.5 conversation [100] (56.7k)
- LLaVA v1.5 GQA [50, 100] (72.1k)
- Coco Captions [20] (82.8k)
- ShareGPT-4V [18] (96.1k)

Text Rich
- Synthdog-En [62] (500k)
- ScreenQA [45] (80.8k)
- WikiTQ [126] (38.2k)
- TabMWP [108] (22.7k)
- ST-VQA [12] (17.2k)
- VisText [147] (10k)
- DeepForm [145] (7k)
- HiTab [25] (2.5k)
- ChartQA [115] (66.7k)
- InfoVQA [116] (52.3k)
- TabFact [19] (91.6k)
- KleisterCharity [140] (27.7k)
- TextVQA [138] (34.6k)
- TextCaps [137] (22k)
- ArxivQA [86] (100k)
- WikiSQL [189] (75k)
- Chart2Text [124] (27k)
- TextVQA [138] (34.6k)
- RenderedText [1] (10k)
- FinQA [23] (5.3k)
- TAT-QA [192] (2.2k)
- DVQA [57] (200k)
- DocVQA [117] (128.5k)
- WikiTable (35.8k) [126]
- VisualMRC [146] (24.5k)
- OCRVQA [121] (80k)

Knowledge (Math/Science/Code)
- RAVEN [180] (42k)
- AI2D [60] (3.2k)
- DaTikZ [10] (48k)
- CLEVR [56] (70k)
- Inter-GPS [106] (1.3k)
- ScienceQA [107] (5k)
- Design2Code [136] (0.5k)
- IconQA [109] (27.3k)
- GeomVerse [58] (9.3k)
- WebSight [69] (10k)

Referring & Grounding
- GRIT-Flickr30k [170, 128] (200.8k)
- GRIT-Visual Genome [170, 64] (267.6k)
- GRIT-Refcoco [170, 98, 173, 123] (210k)
- GRIT-Region reasoning [170] (76.6k)
- GRIT-Spatial Negative Mining [170, 134] (213.6k)

Multi-image
- DreamSim [53, 34] (15.9k)
- NLVR2 [53, 141] (86.4k)
- Star [53, 158] (3k)
- Birds-to-Words [53, 31] (2.6k)
- IconQA [53, 109] (64.5k)
- Spot-the-diff [53, 52] (8k)
- ICL-Instruct† (0.4k)
- Coinstruct [53, 159] (132.3k)
- MultiVQA [53] (3k)
- NExT-QA [53, 161] (3.9k)
- Coco instruct interleaved† (2.1k)

Text
- OrcaMath [122] (200k)
- Dolly [120] (11k)
- OpenOrca [94] (994k)
- WizardCoder [110] (43k)
- MathInstruct [178] (262k)
- OpenCodeInterpreter [188] (66k)

ICL-Instruct
and
Coco instruct interleaved
are in-house datasets

Evaluation Benchmarks and Metrics

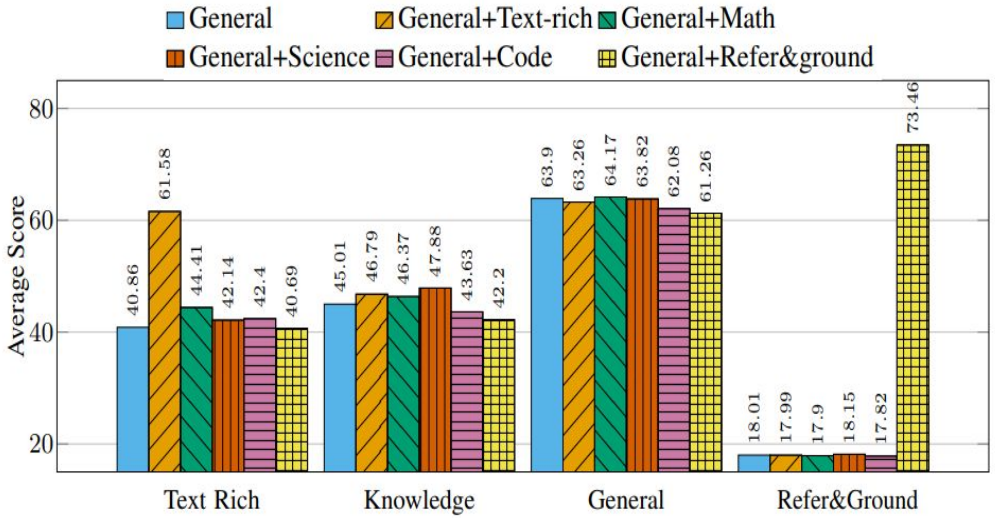
Category	Benchmark	Metric
General	MME [32]	Normalized Accuracy
	SEED [75]	Seed-IMG
	POPE [92]	Average of random, popular and adversarial
	LLaVA-Bench (Wild) [102]	GPT-assisted score
	MM-Vet [174]	GPT-assisted score
	RealWorldQA [160]	Accuracy
Text-rich	WTQ [126]	Accuracy
	TabFact [19]	Accuracy
	OCRBench [103]	Accuracy
	ChartQA [115]	Accuracy
	TextVQA [138]	VQA Open Flamingo Accuracy
	DocVQA [117]	ANLS Score
	InfoVQA [116]	ANLS Score
Refer&Ground	Flickr30K [172]	Recall (IoU>0.5, any protocol)
	LVIS_Ferret [40, 170]	Accuracy
	Refcoco [59]	Recall@1 (IoU>0.5)
	Refcoco+ [59]	Recall@1 (IoU>0.5)
	Refcocog [59]	Recall@1 (IoU>0.5)
	Ferret-Bench† [170]	GPT-assisted score
Knowledge (Math/Science/Code)	AI2D [61]	Accuracy
	ScienceQA [107]	Accuracy-IMG
	MathVista [105]	GPT-assisted score
	MMMU [177]	Accuracy
Multi-image	Qbench2 [186]	Accuracy
	Mantis [53]	Accuracy
	NLVR2 [141]	Accuracy
	BLINK [35]	Accuracy
	MVbench [85]	Accuracy
	Muirbench† [155]	Accuracy

Category Average Score

MMBase Score

SFT Ablation:
Impact of Different Data Categories

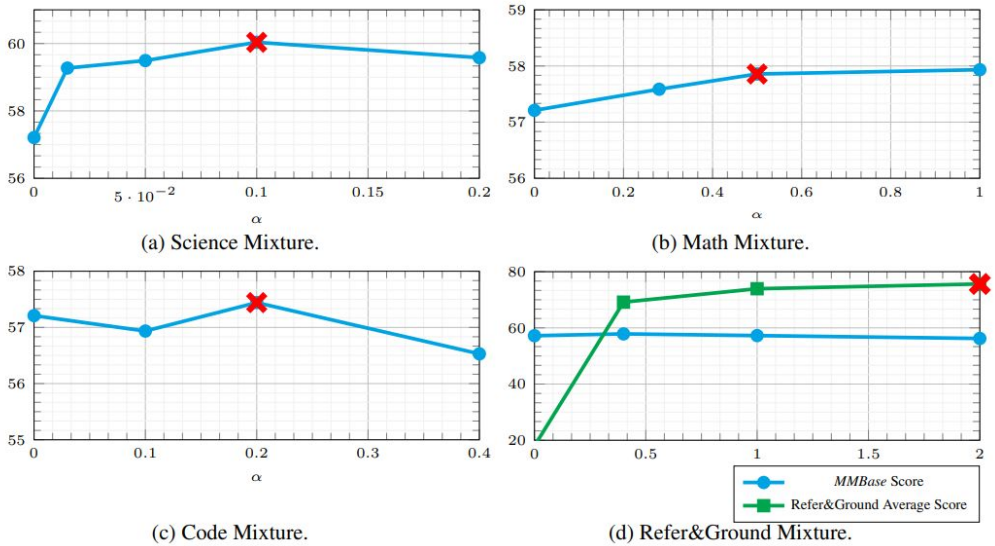
Single-Image Data Category & Category Average Score



- Text-rich data addition improved performance on text-rich and knowledge benchmarks significantly
- Math data showed similar improvements but with less impact on text-rich scores
- Science data enhanced knowledge benchmarks and slightly improved text-rich performance
- Code data marginally increased text-rich scores without other benchmark improvements
- Refer & ground data added new capabilities but caused slight regressions across other categories

SFT Ablation:
Data Mixture Ratio Study

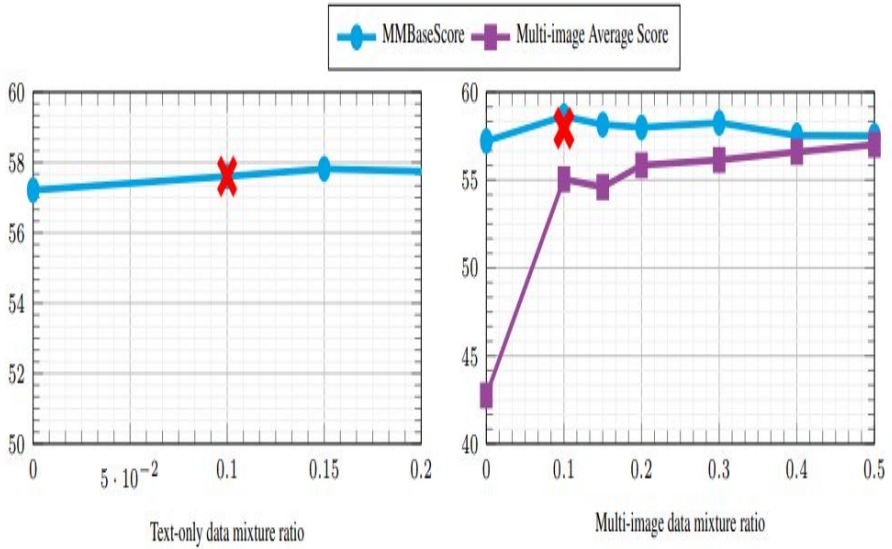
Mixture of Single Image Data (General : Target = 1: α)



- Optimal α ratios: 0.1 for science, 0.5 for math, and 0.2 for code categories
- For refer&ground data, α = 2.0 was chosen as the optimal balance between MMBase scores (which decreased slightly) and Refer&Ground scores (which increased significantly)
- Selection process prioritized maximizing Refer&Ground performance while minimizing negative impact on MMBase metrics

SFT Ablation:
Data Mixture Ratio Study

Mixture of single-image, multi-image, and text-only data. $w_{\text{single}} + w_{\text{multi}} + w_{\text{text}} = 1$

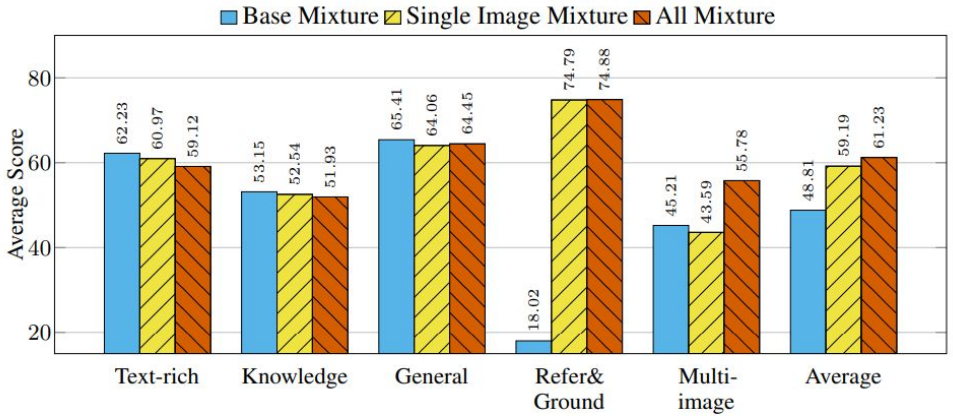


- Varying w_{text} showed minimal impact on base capabilities; w_{text} = 0.1 was chosen to prioritize single-image data performance
- For multi-image data, higher sampling ratios decreased MMBase scores but improved multi-image performance metrics
- w_{multi} = 0.1 was selected as it significantly boosted multi-image average scores while limiting negative impact on base capabilities

SFT Ablation:
Data Mixture Ratio Studies

Mixing Multiple Categories

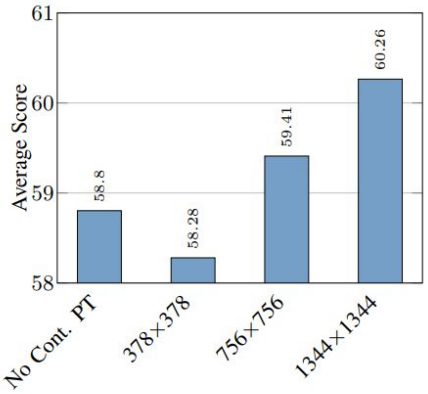
Base - General, Text-Rich, Science ($\alpha_{science}$ = 0.1), math (α_{math} = 0.1) and code(α_{code} = 0.1).
Single Image - Base + Refer&Ground (α_{rg} = 2.0)
All - w_{single} = 0.8 + w_{multi} = 0.1 + w_{text} = 0.1



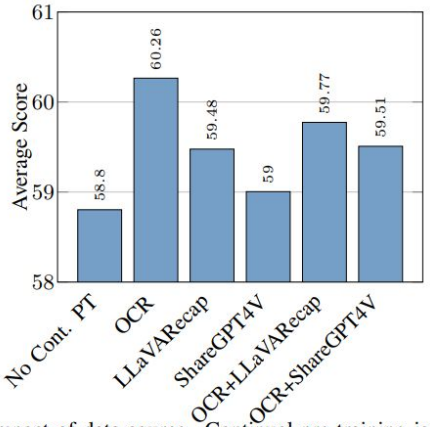
- Including refer&ground and multi-image data slightly decreased performance in text-rich, knowledge, and general benchmarks
- Refer&ground data significantly improved referring/grounding capabilities, while multi-image data enhanced multi-image benchmark performance
- The optimized data mixture achieved the best overall performance by balancing improvements across all benchmark categories

Continual Pre-training Ablation

Continual Pre-training - OCR Data (PDFA, IDL, Rendered-text and DocStruct-4M)
SFT - Base Mixture



(a) Impact of input resolution. OCR data is used for all the continual pre-training experiments.

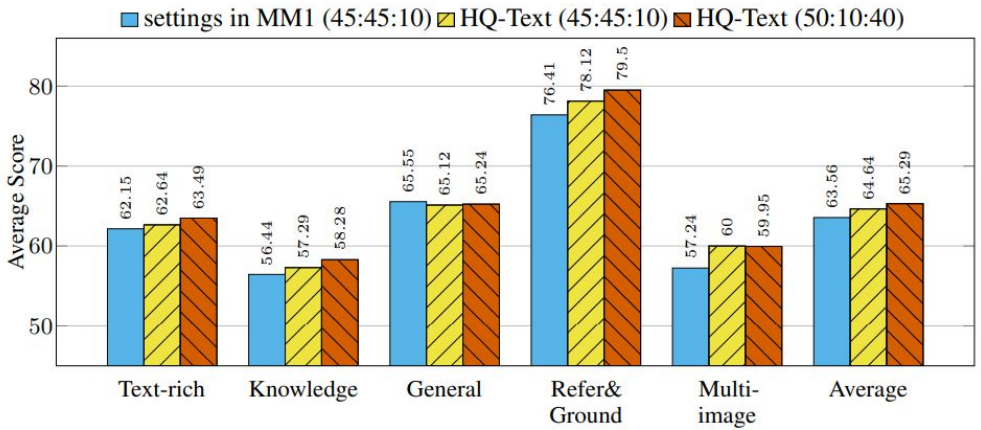


(b) Impact of data source. Continual pre-training is conducted in the high-resolution (1344×1344) setting.

- Higher image resolution (1344×1344) during continual pre-training achieved best performance, with lower resolutions showing consistent decline - 378×378 performed worse than no pre-training
- All continual pre-trained models outperformed the baseline, though synthetic captions showed inconclusive benefits compared to OCR data

Pre-training Ablations

Continual pre-training - OCR Data
SFT - All Mixture
HQ-Text : Higher Quality and More diverse set of text-only Datasets.
Captioned Images + Interleaved Image-Text + Text-Only Datasets



- New data composition improved knowledge scores by 0.85, with optimal ratio of 50:10:40 versus MM1's original 45:45:10
- Updated mixture significantly improved multiple metrics: text-rich (+0.85), knowledge (+0.99), refer&ground (+1.4), with minor decline in multi-image (-0.05)
- Evaluation strategy shifted from few-shot pre-training metrics to post-SFT downstream benchmarks for more reliable performance assessment.

Dynamic Image Splitting

The method begins with a set of **candidate grids** $G = \{(n_h, n_w)\}$, where n_h and n_w are the number of rows and columns, respectively.

Images are resized such that the **longer side** of the image aligns with the sub-grid size, preserving aspect ratio.

Case 1: If a grid exists that can cover the image without excessive scaling or padding, the grid minimizing **padding** is chosen.

Case 2: If no such grid exists, the grid that minimizes **resolution loss** (due to scaling) is selected. This ensures that image content is not overly compressed or distorted.

The optimal grid $g^* = (n_h, n_w)$ is selected by minimizing the difference between the grid resolution and the resized image:

$$g^* = \arg \min_{(n_h, n_w) \in G} n_h n_w r^2 - h_g w_g$$

Example: Dynamic Image Splitting

Example Setup

- **Input image resolution**: 800×1200 (height × width).
- **Vision encoder resolution** (r): 400 (each sub-image should ideally have a resolution of 400×400).
- **Allowed sub-images**: Between $n_{\min} = 1$ and $n_{\max} = 4$ sub-images.
- **Candidate grids**:
  Possible grids $G = \{(1 \times 1), (1 \times 2), (2 \times 1), (2 \times 2)\}$, where n_h and n_w are the grid's rows and columns.

We want a grid that:

1. Covers the image's height ($h = 800$) and width ($w = 1200$).
2. Minimizes either **padding** or **resolution loss**.

Example: Dynamic Image Splitting

Case 1: Grid (2 × 2)

- Sub-image dimensions after splitting:
$$h_g = \frac{h}{n_h} = \frac{800}{2} = 400, \quad w_g = \frac{w}{n_w} = \frac{1200}{2} = 600$$
- Problem: The sub-images will have dimensions 400×600, larger than 400×400. This causes **padding** for unused space.

Case 2: Grid (1 × 2)

- Sub-image dimensions after splitting:
$$h_g = \frac{h}{n_h} = \frac{800}{1} = 800, \quad w_g = \frac{w}{n_w} = \frac{1200}{2} = 600$$
- Problem: The height 800 exceeds $r = 400$, requiring **scaling down** to 400×300. This causes **resolution loss**.

Case 3: Grid (2 × 1)

- Sub-image dimensions after splitting:
$$h_g = \frac{h}{n_h} = \frac{800}{2} = 400, \quad w_g = \frac{w}{n_w} = \frac{1200}{1} = 1200$$
- Problem: Width 1200 exceeds $r = 400$, requiring **scaling down**, leading to significant **resolution loss**.

Case 4: Grid (1 × 1)

- Sub-image dimensions: The entire image is treated as one sub-image:
$$h_g = h = 800, \quad w_g = w = 1200$$
- Problem: The entire image is downscaled to fit $r = 400$. This causes **severe resolution loss** across the whole image.

The (2×2) grid minimizes **padding**, as the image fits neatly into 4 sub-images with less empty space compared to other grids.

Dynamic splitting adapts the grid dynamically based on the input image size:

- For small images: Avoids unnecessary splitting (e.g., uses 1×1).
- For large images: Finds the grid that balances padding and resolution loss.
  This makes it more efficient and flexible than static splitting, which blindly applies a fixed grid (e.g., 2×2) regardless of image characteristics.

Dynamic Image Splitting : Setup

- **Global-Local Format**
  - Before: Overview + Sub-Images
  - After : Sub-Images + Overview
- **Sub-image Position Indicator**
  - Index: (k, i, j)
    - k - zero-indexed image number in the example
    - i - one-index row
    - j - one-index column
  - Seps: “:” - Overview Image Indicator, “,” is the column separator and “<n>” - row separator.
- **Inference for Higher Resolution**
  - (n_{min}, n_{max}): Train = (4, 9) and Inference = (4, 16)

Dynamic Image Splitting Ablation
Image Resolution and Image Tokens

Single Image Mixture
MM1 pre-trained checkpoint without continual pretraining

Row #	Mode	n	#image tokens (per sub-img / total)	Image Enc. Resolution	Effective Resolution	Text-rich	Knowledge	General	Refer & Ground	Average
1	Static	1	144/144	672×672	0.45MP	49.4	53.6	62.6	71.3	59.2
2		5	144/720	672×672	1.8MP	57.7	53.8	64.4	74.8	62.7
3	Dynamic	5	144/720	672×672	1.8MP	58.6	53.7	64.1	74.0	62.5
4		10	81/810	378×378	1.3MP	57.6	53.3	62.9	74.0	62.0
5		10	81/810	672×672	4.1MP	58.3	53.8	64.3	74.9	62.8
6		10	144/1440	378×378	1.3MP	58.5	54.0	63.2	74.5	62.6
7		10	144/1440	672×672	4.1MP	59.8	54.0	64.5	75.2	63.3

- Dynamic image splitting outperformed static splitting for text-rich tasks, with both methods using 5 sub-images maximum
- Text-rich performance improved with both higher number of image tokens and higher effective resolution, while other tasks showed minimal sensitivity
- Optimal configuration found: 10 sub-images at 672×672 resolution with 144 tokens per sub-image

Dynamic Image Splitting Ablation
Image Grid Configuration

Single Image Mixture
MM1 pre-trained checkpoint without continual pretraining

Row #	(n_{min}, n_{max}) Train	Inference	DocVQA	InfoVQA	Text-rich	Knowledge	General	Refer & Ground	Average
			3B Model Comparison						
1	(4, 4)	(4, 4)	73.2	48.3	58.6	53.3	64.1	74.0	62.5
2	(4, 9)	(4, 9)	75.7	53.8	60.0	54.0	63.9	74.6	63.1
3	(4, 16)	(4, 16)	76.3	55.2	60.7	53.4	64.0	73.8	63.0
4	(1, 9)	(1, 9)	76.2	54.1	60.4	53.7	62.5	71.5	62.0
5	(4, 4)	(4, 9)	73.4	52.9	59.7	53.5	63.8	74.0	62.8
6	(4, 4)	(4, 16)	72.3	53.5	59.6	53.8	63.5	74.0	62.7
7	(4, 4)	(1, 9)	73.5	52.7	59.8	50.7	62.6	24.5	49.4
			7B Model Comparison						
8	(4, 4)	(4, 4)	77.0	54.3	64.5	61.1	66.8	77.7	67.5
9	(4, 9)	(4, 9)	81.7	62.1	67.4	60.1	66.6	78.0	68.0
10	(4, 16)	(4, 16)	83.3	64.1	68.0	58.7	67.7	77.2	67.9

- Dynamic image splitting with higher nmax significantly improved document understanding: DocVQA (+3.1-6.3 points) and InfoVQA (+6.9-9.8 points), with larger gains seen in 7B vs 3B models
- Training specifically for higher sub-image counts performed better than just increasing them during inference, though both improved performance
- Grounding performance was highly sensitive to minimum grid size changes, particularly when modifying minimum sub-images from 4 to 1 during inference

Dynamic Image Splitting Ablation
Sub-Image and Overview Image Position

Single Image Mixture
MM1 pre-trained checkpoint without continual pretraining

Row #	Sub-img pos. indicator	Overview image pos.	DocVQA	InfoVQA	Text-rich	Knowledge	General	Refer & Ground	Average
1	none	before	73.2	48.3	58.6	53.5	64.1	74.0	62.5
2	seps	before	74.3	49.7	58.8	53.0	63.8	74.5	62.5
3	index	before	73.4	48.6	58.6	52.7	63.4	74.8	62.4
4	none	after	73.3	49.7	59.2	54.3	64.1	73.8	62.8

- Position indicators were not essential overall, though they helped with referring/grounding tasks and showed benefits for DocVQA and InfoVQA
- Placing overview image after sub-images improved performance due to decoder attention mask enabling overview image to attend to all sub-images
- Index position indicators specifically enhanced referring and grounding tasks, but minimal impact on text-rich tasks

Final Model and Training Recipe

- MM1 Architecture
- **Pre-training Data** (50:10:40) - 2B Image-Text Pairs, 600M interleaved image-text documents with 1B Images in Total and 2T tokens of text-only data
- **Continual Pre-training**: 45M OCR Data
- **SFT**: 80% Single Image Data, 10% Multi Image Data and 10% Text Only Data.
- **Dynamic high-resolution**: (nmin, nmax) = (4, 9), index-based sub-image position indicators, overview image after sub-images; supports up to 4 Megapixels resolution (2016×2016 square or 6048×672 long)
- Image Encoder and LLM Backbone are unfrozen.
- **Pre-training** (Same learning rate schedule as MM1, 200k training steps with 4096 sequence length.
- **Continual pre-training** - peak learning rate of 1e-5 with the cosine decay and 30k training steps
- **SFT** - peak learning rate of 2e-5 and 23k training steps
- Trained using **AXLearn** Framework

Results

Evaluated on 35 multimodal benchmarks using fork of lm-evalharness

- MM1.5 represents a major upgrade over MM1
- MM1.5-1B is the state-of-the-art model at the 1B scale
- MM1.5-3B outperforms MiniCPM-V 2.0 and is competitive with InternVL2 and Phi-3-Vision
- MM1.5-30B is a stronger generalist model than Cambrian-34B
- MM1.5 excels in visual referring and grounding
- MM1.5 excels in multi-image reasoning and in-context learning

Conclusion

- MM1.5 achieves significant improvements through optimized data ratios (50:10:40 for image-text:interleaved:text-only) and dynamic image splitting capabilities
- Dynamic high-resolution processing supports up to 4 Megapixels, with improved handling of document understanding and unusual aspect ratio
- Text-rich performance shows sensitivity to both resolution and sub-image count, with optimal results at 672×672 resolution using 144 tokens per sub-image
- Position indicators enhance referring/grounding tasks but aren't crucial for overall performance, while placing overview images after sub-images improves attention mechanisms
- Final training pipeline combines three stages (pre-training, continual pre-training with OCR data, and SFT) with carefully balanced data mixtures, achieving superior performance across diverse tasks while maintaining strong language understanding.


THANK YOU :)