Multimodal Chain-of-Thought & Reasoning Models

W6 | Thursday, October 1

SEMINAL

Multimodal CoT Reasonsing

Zhang et al. - 2023

FRONTIER

Phi-4-reasoning-vision-15B

Aneja et al. - 2023

Presenters: Wahid Khan Hyder, Sai Pranav Komaragiri

Georgia Tech

Outline

- Intro
- CoT reasoning in LLMs
- Seminal:
  - Problem Statement
  - Approach
  - Results
- Open-ended Multimodal CoT
- Frontier:
  - Problem statement
  - Approach
  - Results
- What changed

Georgia Tech

Chain-of-Thought Prompting Can Mimic Reasoning in LLMs

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Is this truly reasoning?

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Why does Chain-of-Thought improve reasoning?

Georgia Tech

Problem Statement

- Chain-of-Thought boosts LLM reasoning, but has been studied almost entirely for the language modality.
- Previous works have shown that CoT reasoning is an emergent property in larger language models with over 100B parameters.
- This raises the question whether smaller models can reliably learn CoT.
- Multimodal reasoning introduces a new challenge: the model must reason from both language and visual features.
- Prior approaches convert images into text captions, but this compresses the image and can lose information needed for reasoning.

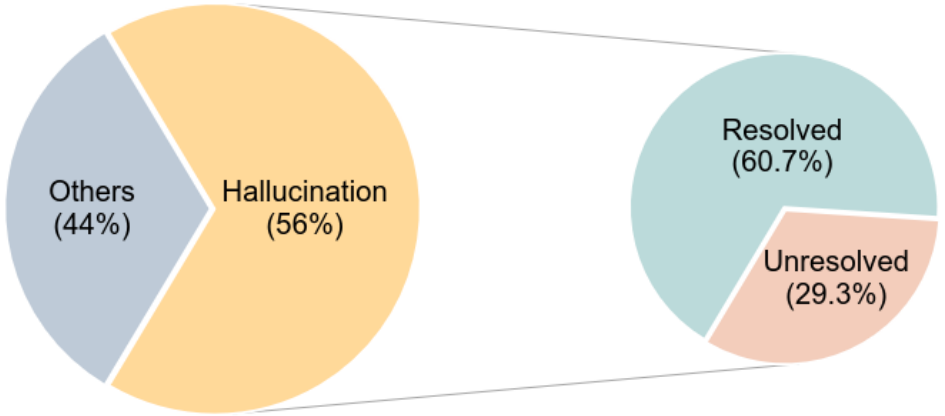
Problem Statement

- Naively adding CoT makes things worse.
- When testing a one-stage baseline with a small language model, the accuracy actually dropped by 12.31%, as compared to just guessing the answer directly.
- Error analysis revealed that 56% of these mistakes were because of hallucinations.
- Providing vision features solved 60% of these hallucination related errors.

Table 2: Effects of CoT in the one-stage setting.

Method	Format	Accuracy
No-CoT	QCM→A	81.63
Reasoning	QCM→RA	69.32
Explanation	QCM→AR	69.68

The accuracy actually goes down when using CoT (tested with 200M T5 based model: Flan-Alpaca)



(a) ratio of hallucination mistakes (b) correction rate w/ vision features

Georgia Tech

Related Works

CoT with in-context learning:

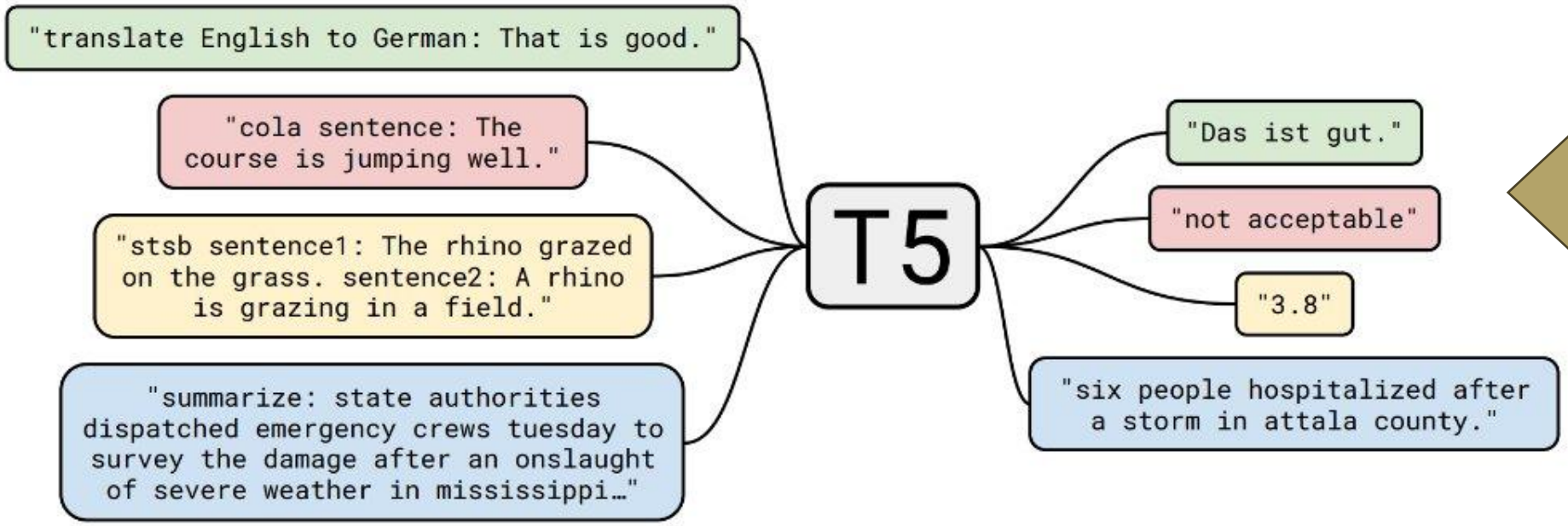
- Zero-Shot-CoT: LLMs are decent zero-shot reasoners when a prompt like “Let’s think step by step” is added after the test question [1].
- Few-Shot-CoT: Step-by-step demonstrations of reasonsing are used as conditioning during inference [2].
- Few-Shot-CoT has been explored with both manual [2] and automatic [3] generation of demonstrations.

Optimizing demonstration selection:

- Beyond manual/random selection of demonstration, related works try to optimize few shot CoT by selecting semantically similar examples [4], using RL to find optimal examples [3].
- Decomposing the problem into smaller sub problems [5], modelling reasoning steps as executable programs [6], and voting [7] to optimize few shot CoT are also related works.

[1] Kojima et al. (2022)
[2] Wei et al. (2022b)
[3] Zhang et al. (2023d)
[4] Rubin et al. (2022)
[5] Zhou et al. (2022)
[6] Chen et al. (2022)
[7] Wang et al. (2022b)

Related Works: CoT with T5



- T5 uses the standard attention Encoder-Decoder Architecture
- Generalises to all text-to-text tasks

CoT with fine-tuning models:

- Fine-tuning a T5 based model on a large scale dataset with CoT annotations results in a sharp drop in performance when the rationale is generated before the answer. [8]
- Instead, CoT is used as an explanation after the answer.

[8] Lu et al. (2022a)

Georgia Tech

Related Works: Prior Ideas in Multimodal CoT

- Transform other modalities into a unified modality and prompt LLMs
- To inject vision information, a simple way is to transform the image into a caption and then append the caption in the input of both stages. [8]
- However, this results in a marginal performance gain (0.8%).
- This is because captions do not a capture a rich enough representation of visual features.

[8] Lu et al. (2022a)

Georgia Tech

Challenge of Multimodal-CoT

Naive Multimodal CoT fails:

i. Rationale quality looks good, yet answer accuracy is below the No-CoT baseline.

ii. Why? The language model is being asked to reason about visual evidence it cannot directly access, so it fills in missing visual facts.

iii. The reasoning process can be logically valid but being built on an incorrect visual premise.

Table 3: Two-stage setting of (i) rationale generation (RougeL) and (ii) answer inference (Accuracy).

Method	(i) QCM→ R	(ii) QCMR→ A
Two-Stage Framework	90.73	78.57
w/ Captions	90.88	79.37
w/ Vision Features	93.46	85.31

Georgia Tech

Challenge of Multimodal-CoT

Problem

Question: Will these magnets attract or repel each other?
Context: Two magnets are placed as shown. Hint: Magnets that attract pull together. Magnets that repel push apart.



Options: (A) attract (B) repel

Gold Rationale: Will these magnets attract or repel? To find out, look at which poles are closest to each other. The north pole of one magnet is closest to the south pole of the other magnet. Poles that are different attract. So, these magnets will attract each other.
Answer: The answer is (A).

Baseline

Generated Rationale: Will these magnets attract or repel? To find out, look at which poles are closest to each other. The south pole of one magnet is closest to the south pole of the other magnet. Poles that are the same repel. So, these magnets will repel each other.
Answer: The answer is (B).

Notice the model's logic is fine.
Hallucination is in the visual facts

+ Vision Features

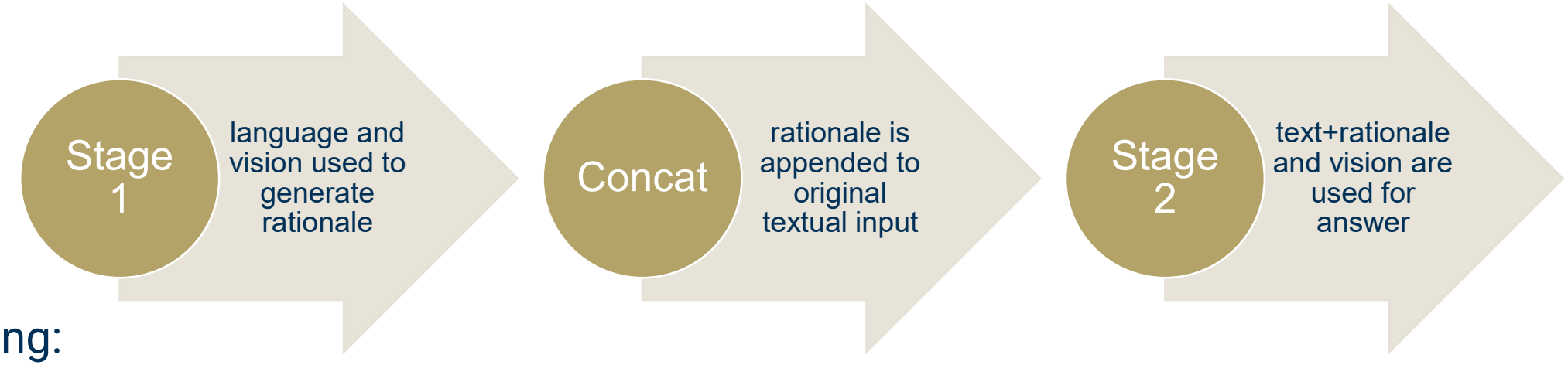
Generated Rationale: Will these magnets attract or repel? To find out, look at which poles are closest to each other. The north pole of one magnet is closest to the south pole of the other magnet. Poles that are different attract. So, these magnets will attract each other.
Answer: The answer is (A).

Georgia Tech

Approach: Overview

Split CoT into 2 stages:

Both the stages share the same model architecture but their inputs and outputs differ.



Training:

- Stage 1: Trained on language and vision input to predict rationale
- Stage 2: Trained on language, vision and rationale to predict the answer

Inference: Stage 1 model generates rationale, which is used by the Stage 2 model

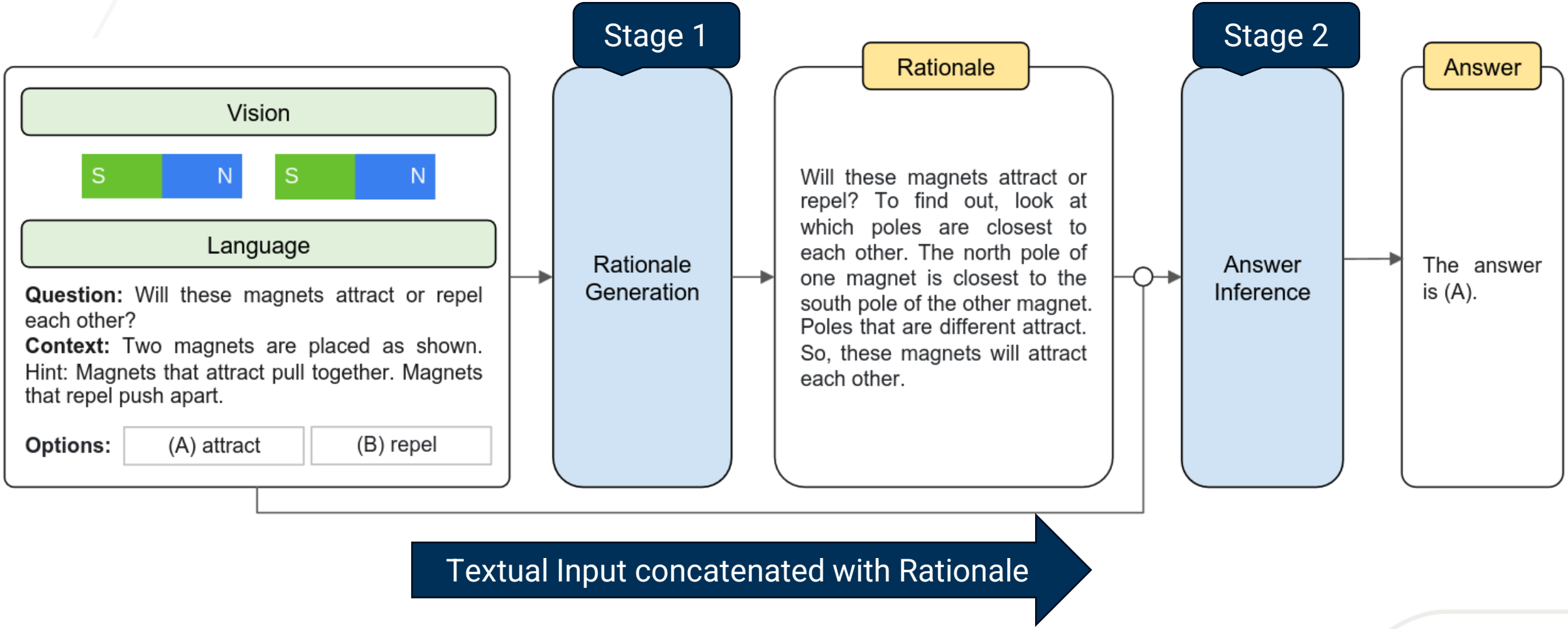
Georgia Tech

Approach: Notations

1. Rationale generation model is trained to predict R, given text and image as inputs
   - Input: $\{X^1_{language}, X_{vision}\}$
   - Output: $X^2_{language} = X^1_{language} . R$
2. Answer inference model is trained to predict the answer, given text with rationale and image as inputs
   - Input: $\{X^2_{language}, X_{vision}\}$
   - Output: $Y = A$

Georgia Tech

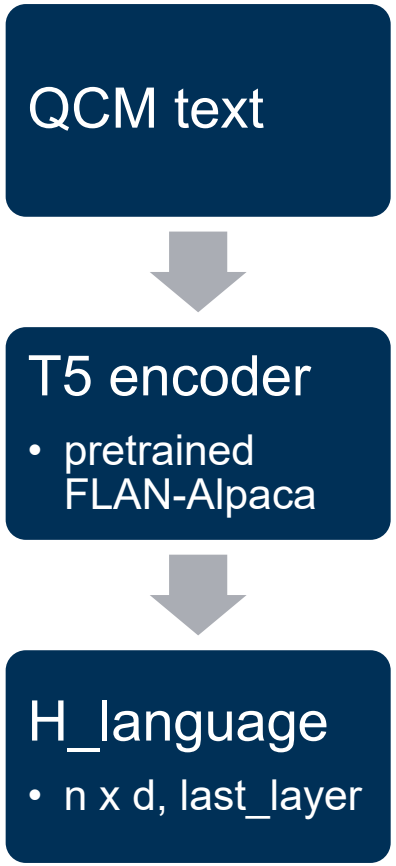
Approach: Overview

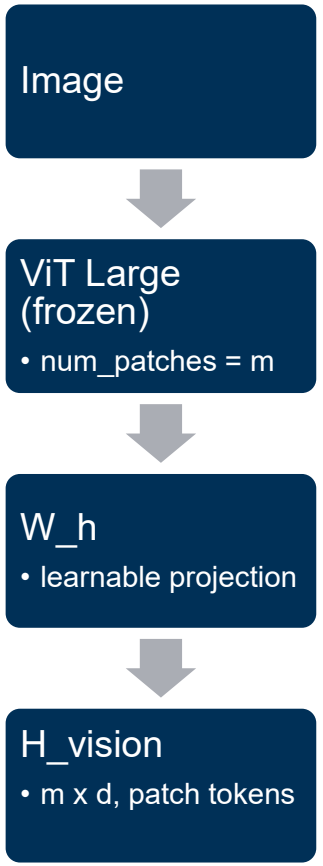


Approach: Encoding

The model takes both language and vision inputs and obtains text representation $H_{language}$ and image feature H_{vision}.

$$H_{\text{language}} = \text{LanguageEncoder}(X_{\text{language}}), \qquad H_{\text{vision}} = W_h \cdot \text{VisionExtractor}(X_{\text{vision}}),$$

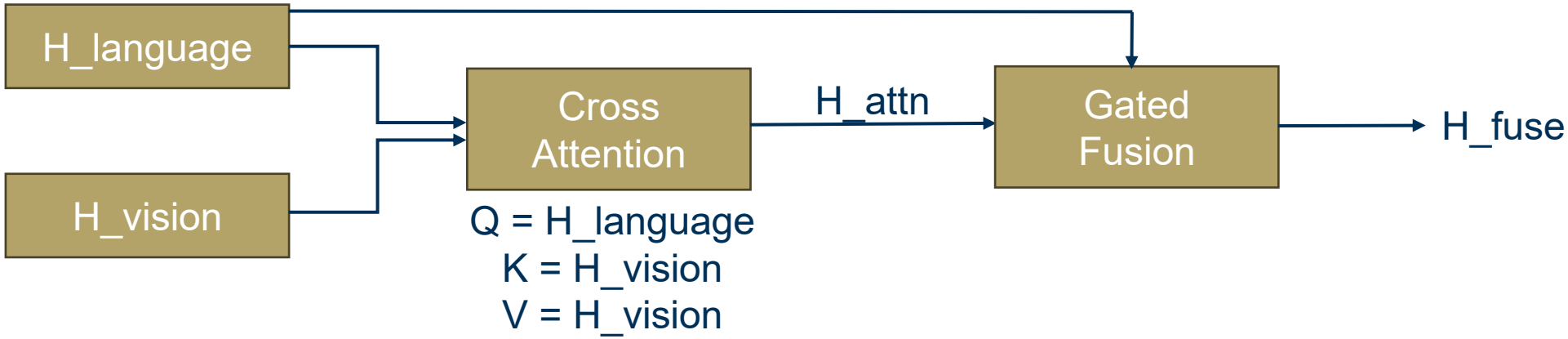




Georgia Tech

Approach: Interaction and Gated Fusion

- Interaction: A single head attention layer correlates text tokens with image patches.
- Gated fusion: $H_{language}$ and H_{vision} then undergo gated fusion.

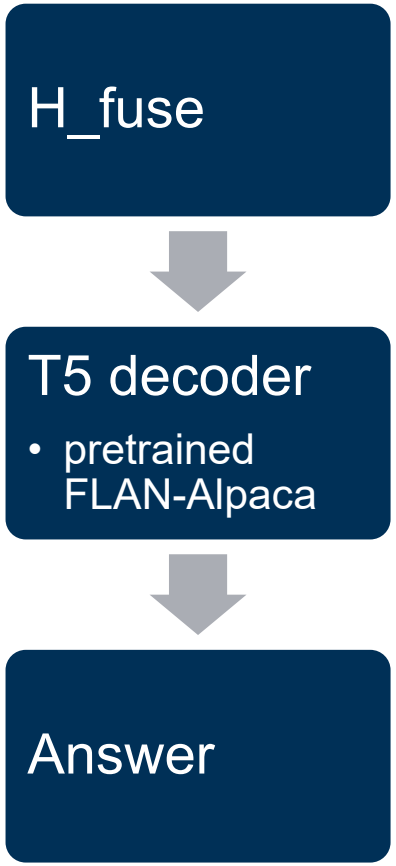


$$\lambda = \text{Sigmoid}(W_l H_{\text{language}} + W_v H^{\text{attn}}_{\text{vision}}),$$
$$H_{\text{fuse}} = (1-\lambda) \cdot H_{\text{language}} + \lambda \cdot H^{\text{attn}}_{\text{vision}},$$

Georgia Tech

Approach: Decoding

- The fused output H_{fuse} is fed into the transformer decoder to predict the target Y. (rationale in Stage1, answer in Stage2)



Georgia Tech

Approach: Training Setup

Setting	Value
Epochs	20
Batch size	8
Learning rate	5e-5
Max context (Rationale)	512
Max context (Answer)	64
Hardware	8 x NVIDIA V100 32GB GPUs
T5 backbone (FLAN-Alpaca)	Base: 200M Large: 700M
ViT variants	vit_large_patch32_384

Georgia Tech

Dataset

- ScienceQA is a large-scale multimodal science question dataset with annotated lectures and explanations,
  - 3 subjects, 26 topics, 127 categories, and 379 skill
  - 2k, 4k, and 4k questions in the training, validation, and test splits
- A-OKVQA is a knowledge-based visual question answering benchmark.
  - Provides multiple-choice and direct answer evaluation settings. Multiple choice was used to keep it consistent with ScienceQA
  - 17k, 1k, and 6k questions in the training, validation, and test splits

Results: ScienceQA & A-OKVQA

ScienceQA — key rows from Table 4:

Model (params)	Avg. Accuracy
Human	88.40%
UnifiedQA (223M)	74.11%
GPT-4 (few-shot)	83.99%
LLaVA-13B	90.92%
MM-CoT Base (223M)	**85.31%**
MM-CoT Large (738M)	**90.45%**

A-OKVQA:

Model	Accuracy
Language-only	47.86%
MM-CoT Base	**50.57%**

- Multimodal-CoT-Large (738M) beats the prior published best (86.54%) by +3.91 pts, and is competitive with LLaVA-13B (90.92%) using ~18x fewer parameters
- On A-OKVQA, grounding in real vision features improves +2.71 pts over a language-only baseline

Georgia Tech

Ablation Study

Component ablation:

Model	Base	Large
Multimodal-CoT	85.31	90.45
w/o Two-Stage Framework	82.62	84.56
w/o Vision Features	78.57	83.97

Vision feature choice:

Feature	Feature Shape	Accuracy
ViT	(145, 1024)	85.31
CLIP	(49, 2048)	84.27
DETR	(100, 256)	83.16
ResNet	(512, 2048)	82.86

- Removing either the two-stage framework or the vision features hurts accuracy at both scales. So both design choices must be load-bearing, not just one
- Two-stage + vision also converges faster across training epochs than the one-stage or vision-free variants
- ViT patch features outperform CLIP, DETR, and ResNet as the vision extractor

Georgia Tech

Generalization & Data Efficiency

MMMU zero-shot generalization:

Model (params)	Accuracy
Kosmos-2 (1.6B)	24.4%
Fuyu (8B)	27.9%
OpenFlamingo-2 (9B)	28.7%
MiniGPT4-Vicuna (13B)	26.8%
Multimodal-CoT (738M)	**28.7%**
GPT-4V	56.8%
Gemini Ultra	59.4%

Rationale source:

Rationale source	Avg. Accuracy
Human-annotated	90.45%
LLM-generated	**87.76%**

- Multimodal-CoT (738M) matches/beats 8-13B open models on MMMU, but trails GPT-4V and Gemini Ultra by ~28-31 pts. It is competitive with similarly-resourced models, not frontier proprietary LLMs
- Training with LLM-generated pseudo-rationales (InstructBLIP + ChatGPT, no human annotation needed) recovers 87.76% vs. 90.45% with human ones — the two-stage + vision framework is the load-bearing idea, not the specific dataset

Georgia Tech

Does chain-of-thought always help?

- Multimodal-CoT shows that forcing a rationale before the answer can hurt accuracy even in a narrow task (81.6% QCM-A vs. 69.3% QCM-RA).
- *Fix:* A gated fusion deciding, per token, how much to trust vision before reasoning
- But how would this scale from narrow tasks to general-purpose VLM tasks?
- General-purpose tasks span far more domains and formats, so the field needed a strategy that didn't depend on hand-designing a gate for each new task type
- *Field's dominant strategy:* Scale parameters and training tokens to push general-purpose accuracy



QUERY

Split this bill for me. I had the pizza, an peach smash, my friend had the steak and wine, and we split the lemon. It's fine to split the tax evenly. Total up mine and my friends portions return them at the end as a json.

MODEL OUTPUT

Here's the breakdown of the bill split between you and your friend:

Your portion:
- Mushroom Pizza: $26.00 - PeachSmash$:11.00
- Half of Lemon: $11.00 - HalfofTax$:4.90
Total: $52.90 Friend'sportion : -HangerSteak$:43.00
- 1 GL WP Sauv Blanc: $15.00 - HalfofLemon$:11.00
- Half of Tax: $4.90 Total$:73.90

Here's the result in JSON format:

```
{
"your_portion": 52.90,
"friend_portion": 73.90
}
```

Georgia Tech

Issues with Scaling Parameters

- Even within the same scaled-up family, which mode wins depends on the task, not just on size
- Scaling leads to more expensive training and inference, impeding usability for downstream deployment, especially in resource-constrained or interactive settings

- **Phi-4-reasoning-vision-15B:** Targets this open efficiency problem
- **Core idea:** Reason selectively so accuracy and cost can both improve together, rather than trading one for another

Benchmark	Qwen3-VL-32B Instruct-4K	Qwen3-VL-32B Thinking-4K	Change
ChartQA	**84.3**	78.5	−5.8
OCRBench	**88.5**	83.7	−4.8
ScreenSpot-v2	**93.7**	83.1	−10.6
AI2D	84.8	**86.9**	+2.1
MathVista	82.5	**83.9**	+1.4

Georgia Tech

Motivation: Phi-4-Reasoning-Vision-15B

The Trend:

- VLMs keep growing in parameters and in consumed and generated tokens
- Qwen3-VL, Kimi-VL, Gemma3, etc., use >1T training tokens
- This raises training costs and inference latency, hurting resource-constrained and interactive use.

Phi family:

- Bet that careful design and data curation can substitute for scale.

Goal:

- Can compact open-weight models be competitive on general vision-language tasks, math/science reasoning, and computer use (GUI understanding), with far less data and inference compute

Main Ideas of the paper:

- Reasoning-capable backbone
- Data quality over data quantity
- Hybrid <think>/<nothink> training, so the model reasons only when useful

Georgia Tech

Efficiency metrics scopes: Training and Inference

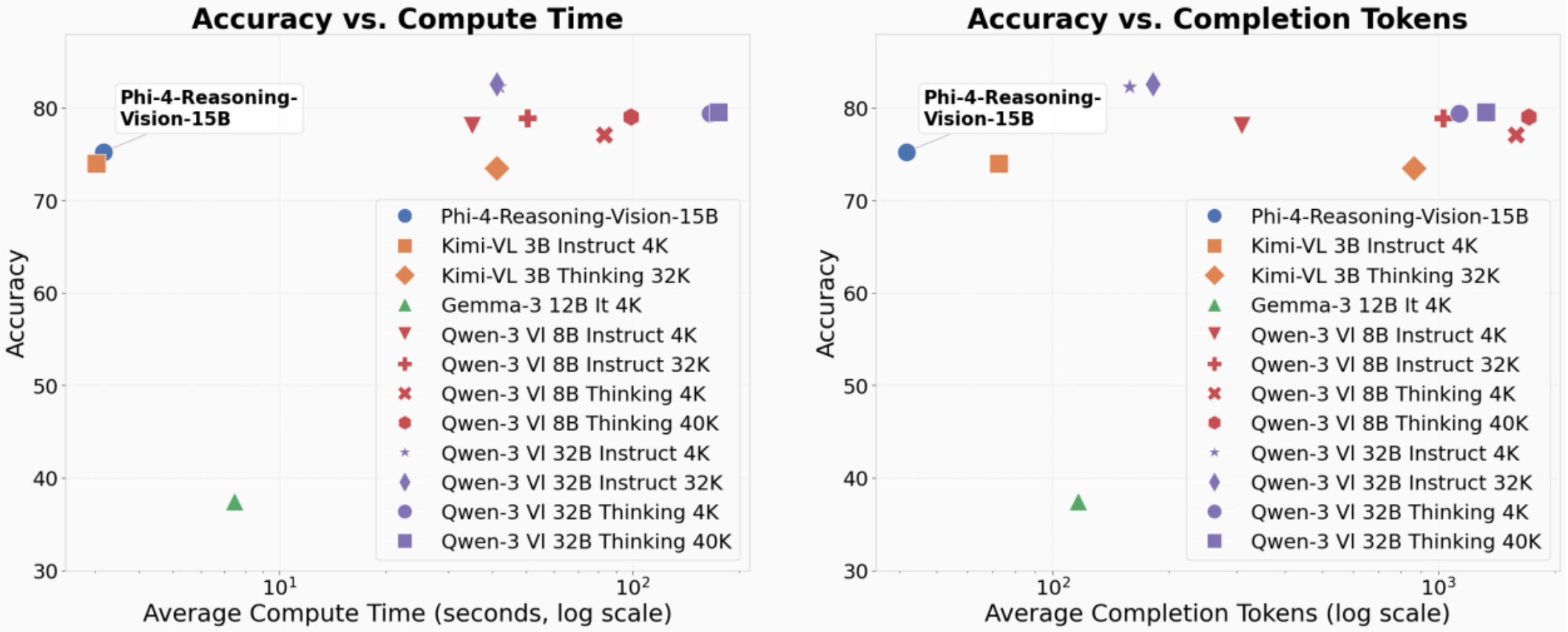
- **Training:**
- Training tokens: Total cumulative tokens processed during training
- Prior methods used for training multimodal models like Qwen 3 VL, Kimi-VL, and Gemma3 use more than 1 trillion tokens
- Phi-4-reasoning-vision-15B:
  - Takes a prior checkpoint (400B pretraining) and adds 200B more tokens of multimodal (image+text) data on top
  - Architecture, along with improved data quality, allows this to be achieved.

- **Inference:**
- Completion Tokens: The number of tokens the model actually generated in the model's response.
- Computation Time (Latency): Total end-to-end execution time, including prefill and decode.
- The paper achieves lower computation time through thin/nothink tags and a visual encoder with dynamic tokens

Georgia Tech

Computation Times vs Completion Tokens

$$Compute\ Time \approx T_{prefill_per_token} \times N_{inp_tokens} + T_{decode_const} \times N_{completion_tokens}$$



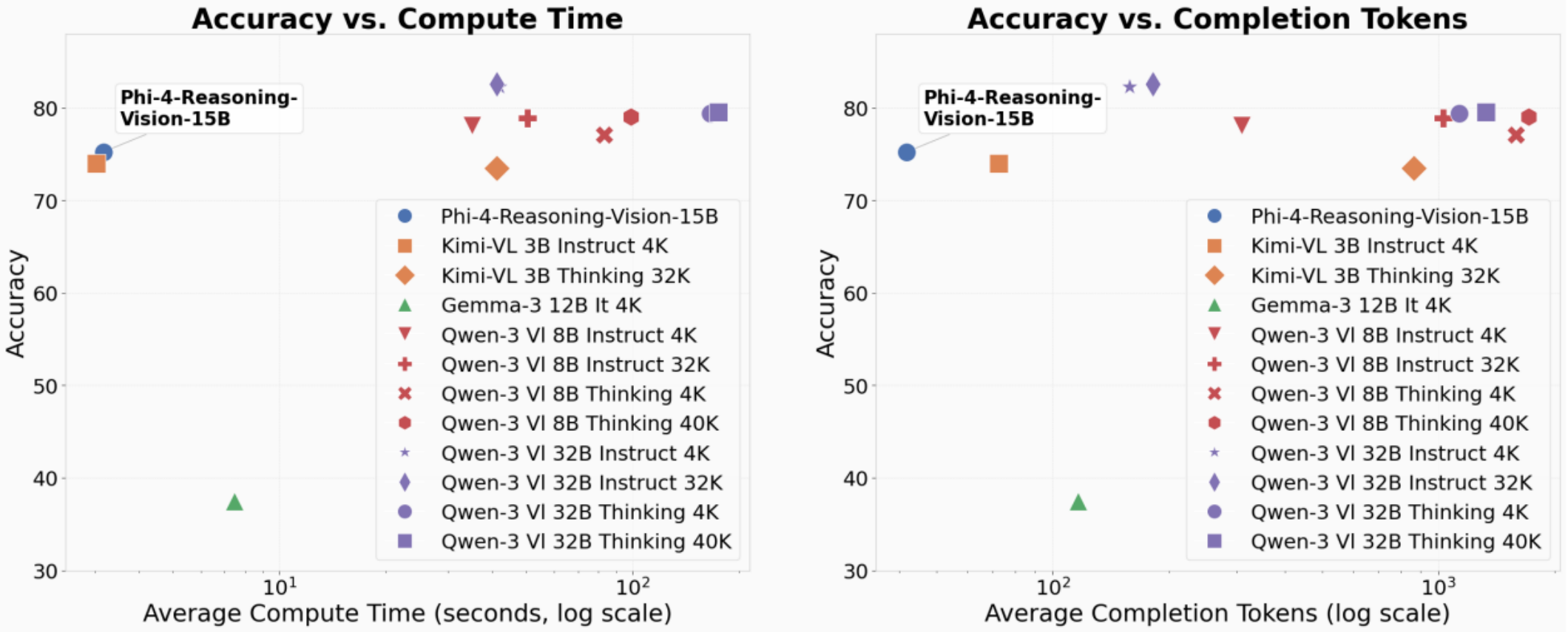
These values were computed by averaging accuracy, time, and output token counts for a subset of 4 benchmarks: ChartQATEST, MathVistaMINI, MMMUVAL, and ScreenSpot_v2

Georgia Tech

Computation Times vs Completion Tokens

$$Compute\ Time \approx T_{prefill_per_token} \times N_{inp_tokens} + T_{decode_const} \times N_{completion_tokens}$$

Why is this $\mathcal{O}(N)$?



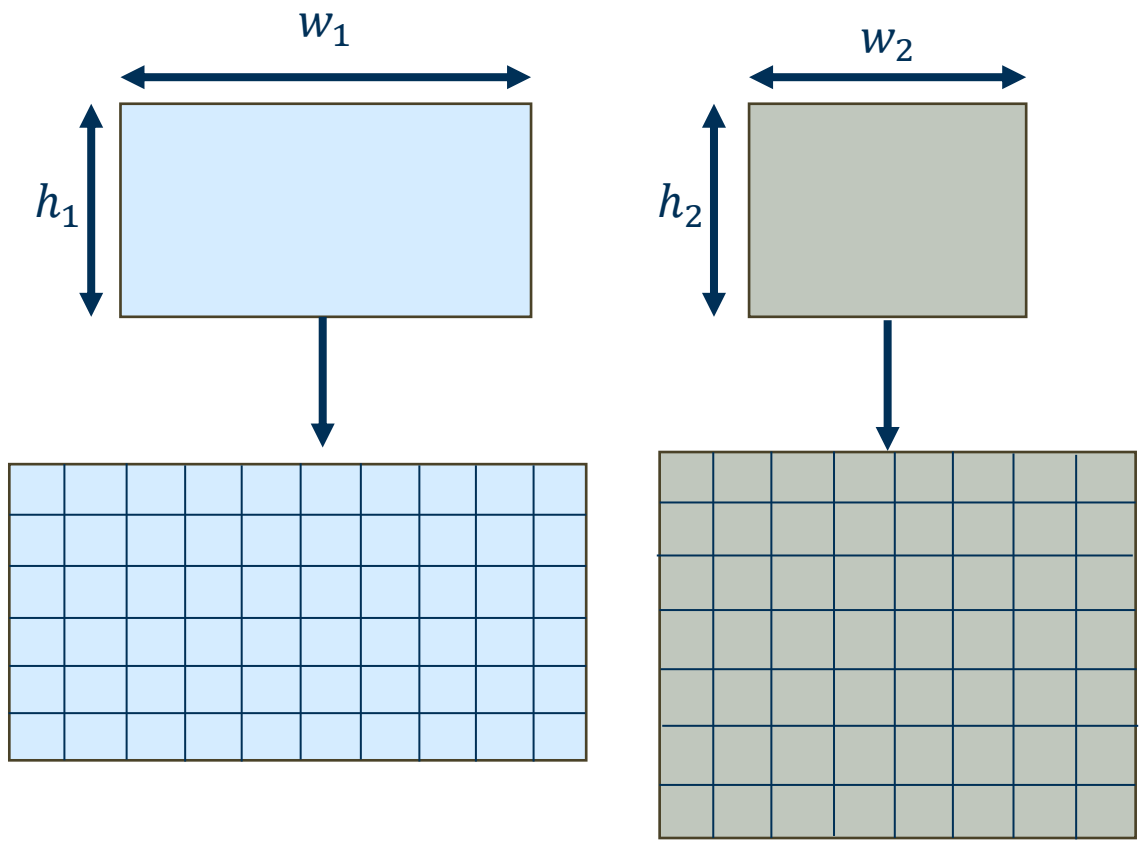
These values were computed by averaging accuracy, time, and output token counts for a subset of 4 benchmarks: ChartQATEST, MathVistaMINI, MMMUVAL, and ScreenSpot_v2

Georgia Tech

SIGLIP-2 NaFlex Recap

- Turns pixels into patch embeddings that the projector maps into the LLM's token space
- Image-text pretrained encoder (sigmoid loss, plus added objectives that improve localization and dense features)
- NaFlex variant: native aspect ratio, variable number of patches
- More tokens mean finer perception but a longer prefill and larger KV cache

NaFlex encoder has two fixed parameters:
Patch Size, Target Sequence Length



Resize $h \rightarrow h'$ and $w \rightarrow w'$ by ensuring:

$$h', w' \text{ are divisible by } PatchSize$$

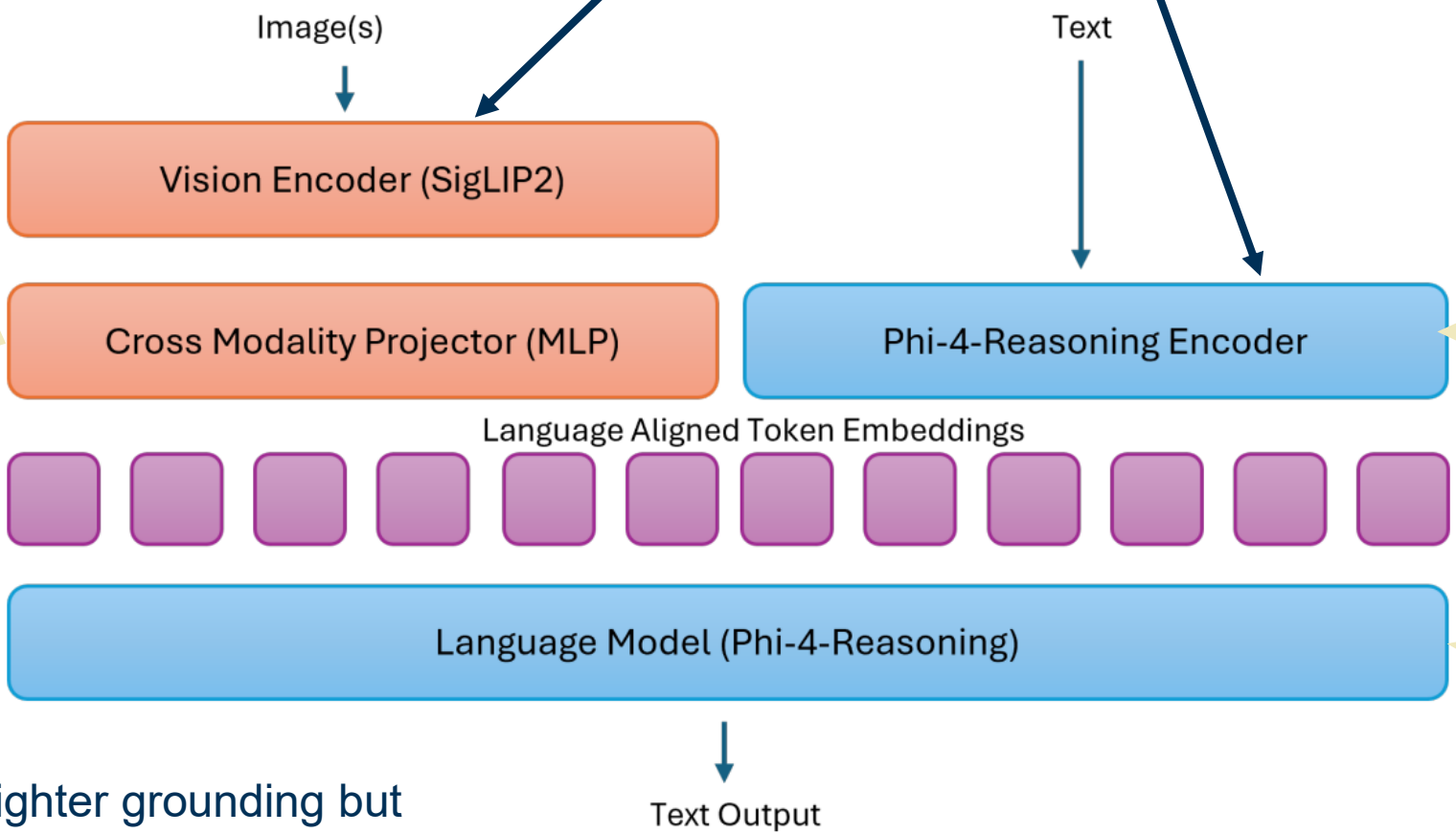
$$\frac{h' \times w'}{PatchSize \times Patchsize} \leq Target\ Sequence\ Len$$

Georgia Tech

Architecture Overview

- Separate Pretrained Unimodal Encoders
- Mid-fusion reuses components pretrained on billions of tokens

The MLP adapts, and the LLM fuses: cross-modal reasoning happens in the LLM's self-attention at every layer



- Pretrained on 400B unique tokens
- Additional fine-tuning with 16B more tokens to teach CoT-style reasoning

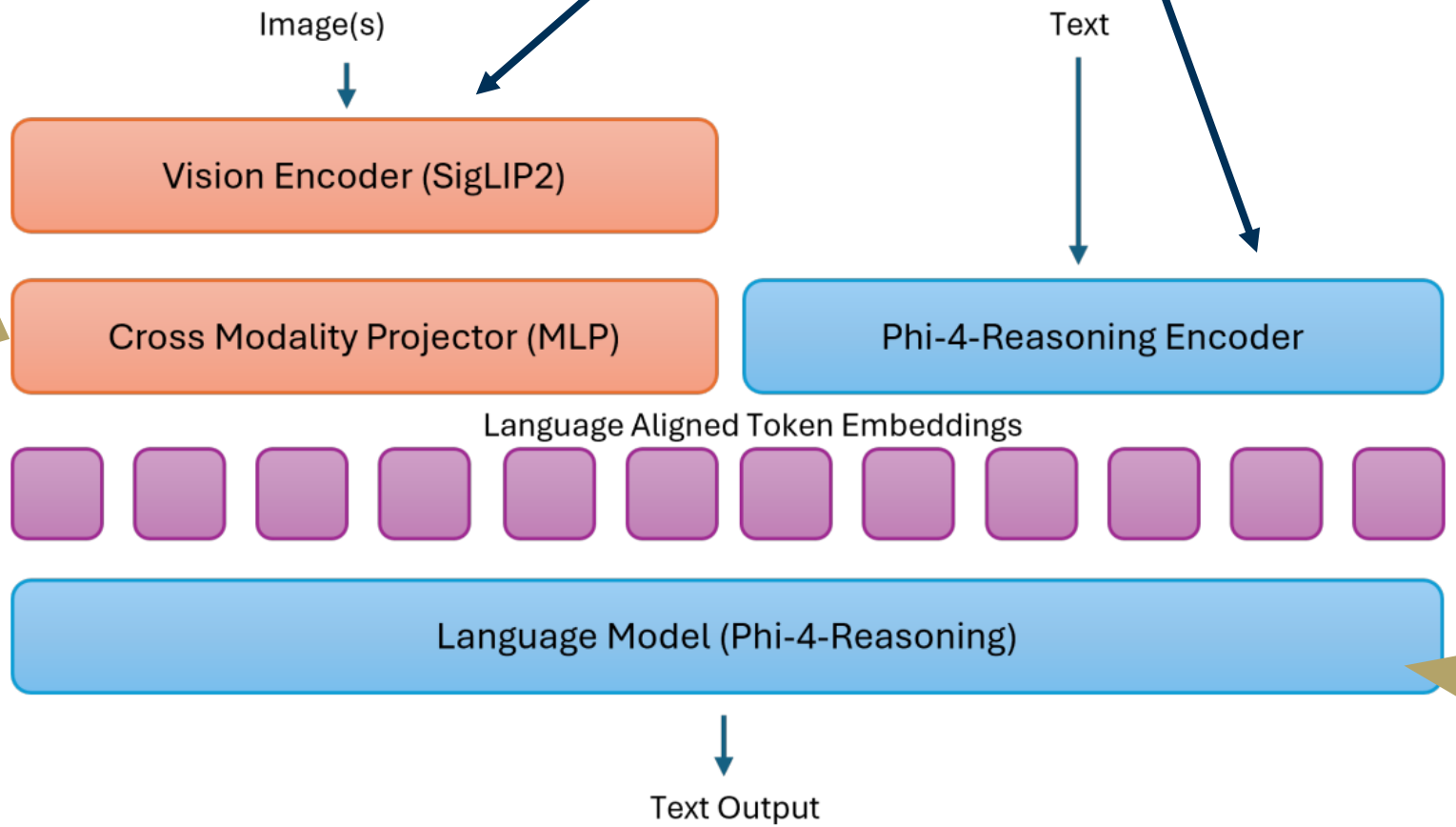
Early fusion has tighter grounding but "significantly increases compute, memory, and data requirements"

Georgia Tech

Architecture Overview

- Dynamic vision encoder
- No cross attention

Compare to Multimodal CoT this work does not fuse vision and language, the tokens are simply interleaved



Fusion happens through self-attention in the LLM itself

Georgia Tech

Vision Encoder Ablations

- 5B test variant, 10M mostly GUI/computer-use pairs
- Dynamic S^2 (Liu et al., 2025a): similar to S^2 , but resizes to a rectangular resolution chosen to minimize distortion while admitting a tiling by 384 × 384 squares
- Multi-crop: crops the image into (potentially overlapping) 384 × 384 squares; sends each through the vision encoder and concatenates features on the token dimension.
- Multi-crop with S^2 : similar to multi-crop but uses S^2 to broaden the receptive field, i.e., crops the image into (potentially overlapping) 1536 × 1536 squares, performs S^2 , and concatenates features on the token dimension.
- Dynamic resolution: a natively dynamic resolution vision encoder; primarily used the NaFlex variant of the SigLIP-2 encoder (Tschannen et al., 2025) and adjusted the minimum and maximum number of patches.

Method	Max Tokens	MathVista	ScreenSpot	ScreenSpot-Pro	V*Bench
Dynamic-S^2	3096	42.9	78.4	9.4	52.9
Multi-crop	3096	43.4	67.8	5.4	51.8
Multi-crop with S^2	2048	43.4	**79.1**	**10.6**	**57.1**
Dynamic resolution	2048	**45.2**	**81.5**	9.2	51.3
Dynamic resolution	3600	**44.9**	79.7	**17.5**	**56.0**

Georgia Tech

Training Recipe

Stage 1 — Only MLP is trained on Image-caption alignment

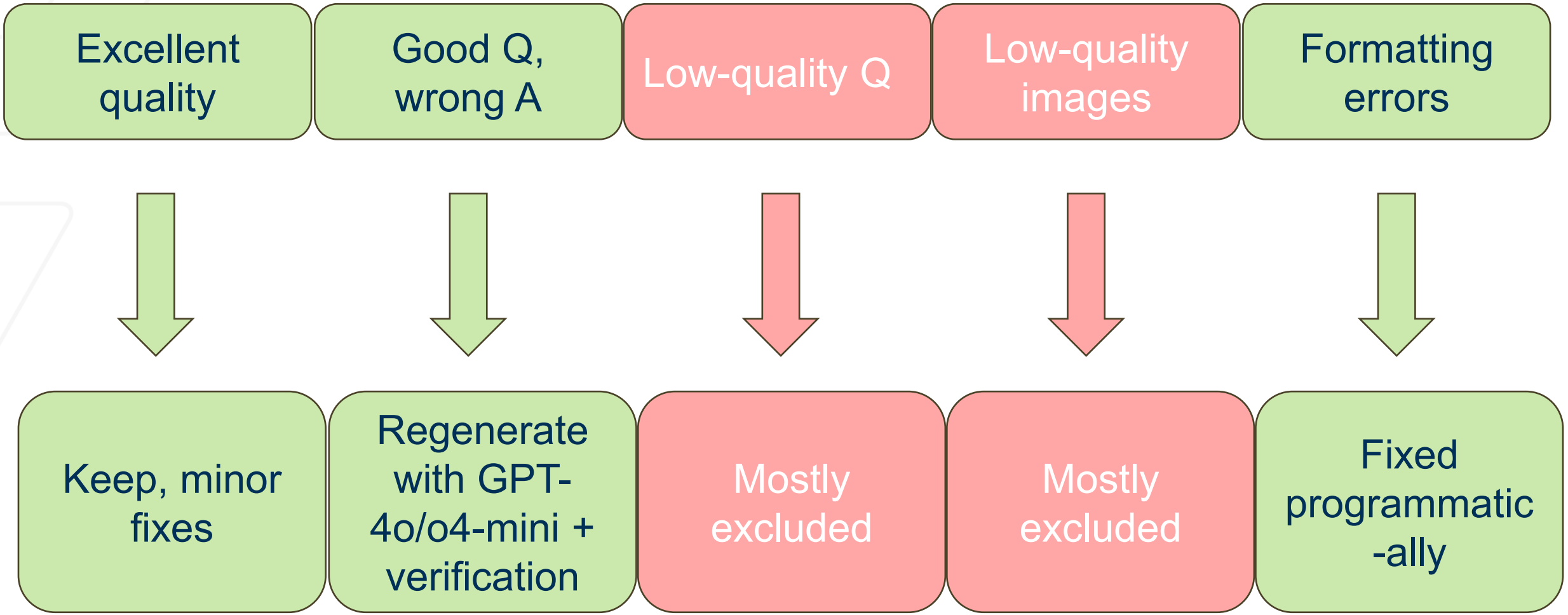
Stage 2 — Everything is trained on single-image instruction tuning

Stage 3 — Everything is trained on long documents, multi-image, safety (RAI)

Hyperparameter	Stage 1	Stage 2	Stage 3
Learning rate	1×10^{-3}	2×10^{-5}	7×10^{-7}
LR schedule	Cosine	Cosine w/ min LR	Cosine w/ min LR
Min LR ratio	–	0.1	0.1
Warmup	3% of steps	500 steps	500 steps
Weight decay	0	10^{-4}	10^{-4}
Adam (β_1, β_2)	(0.9, 0.999)	(0.9, 0.95)	(0.9, 0.95)
Max grad norm	1.0	1.0	1.0
Global batch size	1024	1920	1920
Max sequence length	2048	8192	16384
Training samples	2.0M	62.8M	3.2M
Training tokens	1.4B	188.5B	12B

Georgia Tech

Data Quality: Manual Review



Data Quality Improvement

- **Getting more milage from datasets**
  - Captions for every math/science image, so the LLM can reason over a text description
  - Ground-truth QA rewritten to also teach output-format instruction following ("double duty")
  - Multi-image data from captions: scrambled ordering and caption matching
  - "What's changed?" data from sequential screenshots, for computer-use scenarios
  - Varied, natural prompts instead of over-engineered ones
- **Coordinate normalization:** all spatial outputs in [0, 1] relative to image size, so grounding is resolution-independent

Two domains: Math vs Computer Data

Can one model be strong at both?

General	Math	CUA	Total	MMMU-CoT	MathVista	ScreenSpot-V2
1M	150K	450K	1.6M	44.0	37.4	48.2
1M	150K	850K	2.0M	44.1	37.3	60.0
1M	450K	450K	1.9M	45.3	36.0	48.3
1M	450K	850K	2.3M	43.4	38.9	**63.1**
1M	150K	150K	1.3M	44.2	36.9	29.8
1M	150K	250K	1.4M	**45.4**	37.4	37.7

Math: NuminaMath, LLaVA-OneVision, MMR1, and Eedi

CUA: AGUVis, MultiUI, Pixmo, CoSyn, BigDocs, and SeeClick

- A 5B test model, trained on six different mixes of general, math, and computer-use data, then scored on one benchmark per domain
- Tripling math data (150K→450K) left ScreenSpot-V2 essentially flat (48.2→48.3)
- The Phi-Ground dataset (the 850K CUA tier) gave the largest ScreenSpot-V2 jump
- Caveats: 5B model, small scale, mild imbalance (worst case 7.5% math); math and MMMU differences look noisy
- **Neither domain hurt the other at this scale**

Georgia Tech

How Can a Text Model Learn to Reason?

- **Supervised Fine-Tuning (SFT):**
  - The model learns by studying thousands of high-quality examples of step-by-step logic.
  - Straightforward to implement but requires a massive, expensive dataset.
  - Excellent at replicating the structure of training data but struggles to adapt to unseen problems.
  - Relatively faster during inference.

- **Reinforcement Learning (RL):**
  - The model learns by trial and error, getting rewarded for arriving at correct answers.
  - This requires less human-generated data but demands massive computational power and complex engineering
  - Discovers robust, adaptable strategies that transfer well across new tasks and changing rules
  - Requires immense inference-time computation

Design Space for Multimodal Reasoning AI

Approach	Starting Base	Training Pipeline	Primary Trade-Off
Simultaneous	Basic LLM	Vision + Logic trained at once	Demands massive reasoning/visual datasets.
Sequential	Basic LLM	Vision first, Logic second	High risk of catastrophic forgetting of visual skills.
Rigid Logic	Reasoning LLM	Pure Reasoning Multimodal	Inefficient, generates forced, slow logic for simple visual tasks.
Hybrid (Proposed)	Reasoning LLM	Mixed (Logic + Direct Answer)	Optimal, model learns when to reason and when to react.

- Reasoning from scratch is too fragile or data-heavy.
- The best approach is to inherit a strong reasoning base and feed it mixed training data to adapt its compute to the complexity of the prompt

Georgia Tech

Mixed Non-Reasoning and Reasoning: Implementation

- ***The problem:*** Always reasoning wastes time on easy tasks; never reasoning hurts hard ones.
- ***Implementation:***
  - The model is trained with SFT
  - Reasoning samples (~20% of the mix) include tagged <think>...</think>, sections with chain-of-thought reasoning before the final answer, covering domains like math and science.
  - Non-reasoning samples are tagged to start with a token, signaling a direct response, and cover perception-focused tasks such as captioning, grounding, OCR, and simple VQA
- ***User override:*** either mode can be forced at inference by including the <think>/<nothink> tokens in the prompt.

CoT has seperate rationale stage, answer stage. Here one model has two modes

Georgia Tech

Mixed Non-Reasoning and Reasoning: Results

- The balance between modes is a direct function of design choices
- Default beats forcing either mode, on average (MathVista): 75.2% default vs. 68.7% forced <nothink> vs. 74.1% forced <think>
- **However, default is not always right:**
  - Forcing reasoning *helps* MathVerse (44.9 to 53.1) and MMMU slightly. The default is nothink here
  - Forcing reasoning *hurts* AI2D (84.8 to 79.7). The default's instinct to skip reasoning was correct
- **Limitation:**
  - The reasoning/non-reasoning boundary is learned implicitly from data, not from an explicit rule.
  - The 20/80 split is a design heuristic, not a proven optimum.

Evaluation: Acc vs Non Thinking Models

Benchmark	Phi-4-reasoning-vision-15B	Phi-4-reasoning-vision-15B –force nothink	Phi-4-mm-instruct	Kimi-VL-A3B-Instruct	gemma-3-12b-it	Qwen3-VL-8B-Instruct-4K	Qwen3-VL-8B-Instruct-32K	Qwen3-VL-32B-Instruct-4K	Qwen3-VL-32B-Instruct-32K
$AI2D_{TEST}$	84.8	84.7	68.6	84.6	80.4	82.7	83	84.8	85
$ChartQA_{TEST}$	83.3	76.5	23.5	87	39	83.1	83.2	84.3	84
HallusionBench	64.4	63.1	56	65.2	65.3	73.5	74.1	74.4	74.9
$MathVerse_{MINI}$	44.9	43.8	32.4	41.7	29.8	54.5	57.4	64.2	64.2
$MathVision_{MINI}$	36.2	34.2	20	28.3	31.9	45.7	50	54.3	60.5
$MathVista_{MINI}$	75.2	68.7	50.5	67.1	57.4	77.1	76.4	82.5	81.8
$MMMU_{VAL}$	54.3	52	42.3	52	50	60.7	64.6	68.6	70.6
MMStar	64.5	63.9	45.9	60	59.4	68.9	69.9	73.7	74.3
OCRBench	76	75.6	62.6	86.5	75.3	89.2	90	88.5	88.5
$ScreenSpot_{v2}$	88.2	88.3	28.5	89.8	3.5	91.5	91.5	93.7	93.9

- Trails Qwen3-VL-8B/32B on most benchmarks, sometimes substantially
- It wins clearly only against the smaller/older comparators: Phi-4-mm-instruct and Gemma3-12b-it (collapses on ChartQA and ScreenSpot-v2, 39 and 3.5).
- Against Kimi-VL-A3B (only ~3B active params), Phi-4-rv is close but doesn't consistently win Kimi-VL beats it on ChartQA and OCRBench.
- The "force nothink" column is usually a few points below default, confirming the mixed-mode default is doing useful work, not just adding latency

Georgia Tech

Evaluation: Acc vs Thinking Models

Benchmark	Phi-4-reasoning-vision-15B	Phi-4-reasoning-vision-15B –force thinking	Kimi-VL-A3B-Thinking	gemma-3-12b-it	Qwen3-VL-8B-Thinking-4K	Qwen3-VL-8B-Thinking-40K	Qwen3-VL-32B-Thinking-4K	Qwen3-VL-32B-Thinking-40K
$AI2D_{TEST}$	84.8	79.7	81.2	80.4	83.5	83.9	86.9	87.2
$ChartQA_{TEST}$	83.3	82.9	73.3	39	78	78.6	78.5	79.1
HallusionBench	64.4	63.9	70.6	65.3	71.6	73	76.4	76.6
$MathVerse_{MINI}$	44.9	53.1	61	29.8	67.3	73.3	78.3	78.2
$MathVision_{MINI}$	36.2	36.2	50.3	31.9	43.1	50.7	60.9	58.6
$MathVista_{MINI}$	75.2	74.1	78.6	57.4	77.7	79.5	83.9	83.8
$MMMU_{VAL}$	54.3	55	60.2	50	59.3	65.3	72	72.2
MMStar	64.5	63.9	69.6	59.4	69.3	72.3	75.5	75.7
OCRBench	76	73.7	79.9	75.3	81.2	82	83.7	85
$ScreenSpot_{v2}$	88.2	88.1	81.8	3.5	93.3	92.7	83.1	83.1

- It beats Qwen3-VL-32B-Thinking on ScreenSpot-v2 (88.2 vs. 83.1)
- Forcing thinking mode helps Phi-4-rv on MathVerse and MMMU, but hurts AI2D and ChartQA slightly. consistent with the paper's claim that the learned default mode beats forcing either mode on average.
- Kimi-VL-A3B-Thinking, despite ~3B active parameters, beats Phi-4-rv on HallusionBench, MathVerse, MathVision, MathVista, MMMU, MMStar, OCRBench.
- Phi-4-rv is overall best for latency but not completely for accuracy

Georgia Tech

Limitations

- Trails larger proprietary and open models on broad, unconstrained benchmarks.
- *The paper states:* Competitive with similarly-sized open-weight models and state-of-the-art relative to training/inference compute, not state-of-the-art in absolute accuracy.
- Reasoning-mode switching is imperfect.
- The model sometimes reasons when a direct answer would do, or skips reasoning when it would help. (Users can override with explicit <think>/<nothink> prompting.)
- Fine-grained visual understanding has limits.
- Users should verify critical outputs, especially detailed visual claims — a limitation shared with essentially every VLM of this scale.

Georgia Tech

Similarities

- Both try to address hallucination which comes from missing visual grounding and not reasoning failure.
- The perception is stated as the bottleneck by both papers in multimodal reasoning settings.
- Both try to separate reasoning from perception although in different ways.
- Both establish that naive CoT can actually hurt the performance.
- Both works achieve efficiency compared higher parameter models.

Changes from Seminal paper

	Multimodal-CoT	Phi-4-reasoning-vision
Vision encoder	Frozen ViT-large, ~145 patches, static	SigLIP-2, variable resolution, up to thousands of tokens (unfrozen in Stages 2–3)
Language model	T5 encoder-decoder (200M / 738M)	Phi-4-Reasoning
Merge mechanism	Cross-attention plus learned gate, once	MLP projector, then joint self-attention in every layer
Vision features addition	Folded into text-length features	Extra tokens in the sequence
Interaction depth	One layer, one head, one direction	Every layer, bidirectional across visual and text tokens (subject to causal masking)
Reasoning setup	Two separately trained models (rationale, then answer)	One model with <think>/<nothink>