Multi modal Scientific Question-Answering

Columbia COMS4705 Project Milestone
Keywords: *Vision Language Models, Scientific Question Answering, Hybrid Models, Chain of Thoughts (CoT)*

Reneth Raj Simon
Department of Computer Science
Columbia University
rs4761@columbia.edu

Sreeram Raghammudi
Department of Computer Science
Columbia University
sr4314@columbia.edu

Vishal Menon
Department of Data Science
Columbia University
vm2820@columbia.edu

Abstract

This paper studies multimodal scientific QA on the ScienceQA benchmark under compute constraints. We first benchmark strong baselines, including monolithic VLMs (e.g., LLaVA) and text-centric QA models (e.g., UnifiedQA, BERT), to establish performance bounds across modality settings. We then design and analyze Custom Multimodal Hybrid models (feature fusion, BiLSTM fusion, Cross-Attention), showing that specialized discriminative fusion can be more efficient while remaining competitive. Finally, we evaluate two Chain of Thought strategies: (i) instruction tuning with explicit CoT supervision using ground-truth rationales, and (ii) a decoupled pipeline where a frozen BLIP contextualizer converts images to text and a LoRA-tuned Qwen3-8B LLM performs structured CoT reasoning. The decoupled, prompt-aligned generative approach achieves the strongest results, highlighting the value of baseline-driven evaluation, hybrid fusion analysis, and structured rationale generation.

Keywords: Multimodal QA, ScienceQA, Chain-of-Thought (CoT), Decoupled Architecture, LoRA, Large Language Models (LLMs), Feature Fusion.

1 Key Information

- TA mentor: Chaitya Shah
- External collaborators: No
- Sharing project: No

2 Introduction

Many science exam questions are **multimodal**—combining text with diagrams or images—and require stepwise reasoning. We study this setting using the ScienceQA benchmark, where each question includes text, an image, and multiple-choice options. ScienceQA [1] is both educationally relevant and scientifically valuable because it probes whether models can connect visual understanding with structured reasoning rather than rely on text pattern matching.

Current systems show mixed performance: LLMs perform well when textual signals are strong, while many VLMs inconsistently use images. Caption-based pipelines (e.g., GIT + Qwen2.5-1.5B) perform surprisingly well but still treat reasoning implicitly.

We investigate whether **explicit chain-of-thought (CoT) supervision** improves multimodal scientific QA. After establishing text-only and caption-based baselines, we evaluate architectural variants: (i) different textual pooling strategies (CLS, max, learnable attention), (ii) a BiLSTM reasoning layer, (iii) cross-modal attention between textual queries and visual features, and (iv) a hybrid model integrating all components.

3 Related Work

ScienceQA by **Lu et al. (2022)** [2] introduced a benchmark requiring integration of images, questions, and lectures, showing that explicit **Chain-of-Thought** significantly improves accuracy. Unlike free-form explanations, we adopt a structured three-step CoT format for more predictable and consistent reasoning.

Because fine-tuning large multimodal LLMs is expensive, we use a modular design based on BLIP [3] as a frozen visual encoder. BLIP-generated captions are injected into the LLM prompt, enabling efficient LoRA fine-tuning while maintaining strong visual grounding.

Hybrid architectures are increasingly effective for multimodal reasoning. VLMo (Bao et al., 2022) [4] introduces a Mixture-of-Modality-Experts (MoME) transformer with modality-specific feed-forward experts and shared self-attention, improving generalization over traditional fusion-only VLMs.

Hybrid reasoning systems also show promise. Hybrid-DMKG (Tiwari et al., 2025) [5] combines dynamic multimodal knowledge graphs with LVLMs to support multi-hop symbolic + perceptual reasoning, achieving strong performance on scientific QA tasks.

Together, these works highlight two themes central to our study: (1) hybrid or modular architectures outperform naïve fusion, and (2) structured textual reasoning dominates performance even in multimodal settings. These findings align with our ScienceQA results, where caption-augmented and text-only models outperform larger VLMs, motivating our focus on modular multimodal-to-text reasoning enriched with structured CoT.

4 Approach

This section details your approach(es) to the problem. For example, this is where you describe the architecture of your neural network(s), and any other key methods or algorithms.

4.1 Initial Benchmarking and Model Comparisons

To establish a performance lower bound and isolate the contributions of different architectural components, we evaluated three distinct baseline strategies during our milestone project.

1. **Text-Only Linguistic Baseline:** We used the `bert-base-uncased` architecture for discriminative classification and /unifiedqa for generative QA. These models ignored the image input (x_{img}) to isolate the signal derived purely from the question, context, and options.
2. **Tightly Coupled Multimodal Baseline (VLM):** For the subset of questions including images, we fine-tuned a VLM (specifically, microsoft/git-base-textvqa) to jointly encode the image and text. This model directly generated the answer string, demonstrating the performance ceiling achievable via coupled multimodal encoding.
3. **Conditional Two-Model Pipeline:** This approach served as a competitive benchmark for our final methodology. It conditionally selects the model during inference: if an image is present ($x_{\text{img}} \neq \emptyset$), it uses the fine-tuned VLM (GIT); otherwise, it defaults to a fine-tuned text model (Qwen 1.5B) (code developed by the authors). This strategy tested a practical, performance-driven application of decoupled models.

4.2 Multimodal hybrid architectures

We now describe the multimodal hybrid architectures that we developed from scratched and evaluated in this work. Let $\mathcal{D} = \{(q_k, c_k, I_k, y_k)\}_{k=1}^{N}$ denote the dataset, where q_k is the question text, c_k an optional caption, I_k an optional image, and $y_k \in \{1, \ldots, C\}$ the discrete label (correct choice). Let $\mathbf{1}_{I_k}$ indicate the presence of an image for example k ($\mathbf{1}_{I_k} = 1$ if an image is available, else 0).

4.2.1 Encoders and basic representations

We denote a pretrained text encoder (BERT) parameterized by ϕ_{text} which maps a token sequence to a sequence of contextual embeddings:

$$H_k = \text{TextEnc}_{\phi_{\text{text}}}(q_k \oplus c_k) \in \mathbb{R}^{L \times d}, \tag{1}$$

where L is the input token length and d the hidden dimensionality. The i-th token embedding is $H_{k,i} \in \mathbb{R}^d$ and $H_k = (H_{k,1}, \ldots, H_{k,L})$.

A pretrained vision encoder (ResNet) with parameters ϕ_{vis} produces an image feature vector:

$$v'_k = \text{VisEnc}_{\phi_{\text{vis}}}(I_k) \in \mathbb{R}^{d_v}, \tag{2}$$

which we project into the text space with a learned linear projection $W_v \in \mathbb{R}^{d \times d_v}$ and bias b_v:

$$v_k = W_v v'_k + b_v \in \mathbb{R}^d. \tag{3}$$

When no image is available ($\mathbf{1}_{I_k} = 0$) we set $v_k = \mathbf{0} \in \mathbb{R}^d$ (equivalently multiply by $\mathbf{1}_{I_k}$).

4.2.2 Pooling strategies

We consider three pooling functions $\mathcal{P} : \mathbb{R}^{L \times d} \to \mathbb{R}^d$ that map token sequences to a fixed-length text vector t_k:

CLS pooling. Assuming the first token is a special `[CLS]` token,

$$\mathcal{P}_{\text{CLS}}(H_k) = H_{k,1}. \tag{4}$$

Max pooling. Element-wise max over the sequence dimension (padding masked by attention mask $m_k \in \{0,1\}^L$):

$$\mathcal{P}_{\max}(H_k, m_k)_j = \max_{1 \le i \le L} \left(m_{k,i} \cdot H_{k,i,j}\right) \quad \text{for } j = 1, \ldots, d. \tag{5}$$

Learnable attention pooling. A learned scalar scorer $a(h) = w_a^\top h$ produces scores which are normalized with softmax:

$$s_{k,i} = w_a^\top H_{k,i}, \tag{6}$$

$$\alpha_{k,i} = \frac{\exp(s_{k,i}) m_{k,i}}{\sum_{j=1}^{L} \exp(s_{k,j}) m_{k,j}}, \tag{7}$$

$$\mathcal{P}_{\text{att}}(H_k, m_k) = \sum_{i=1}^{L} \alpha_{k,i} H_{k,i}. \tag{8}$$

We denote the chosen pooling function generically by $\mathcal{P}_\pi$, where $\pi \in \{\text{CLS}, \max, \text{att}\}$.

Thus the pooled text vector is

$$t_k = \mathcal{P}_\pi(H_k, m_k) \in \mathbb{R}^d. \tag{9}$$

4.2.3 LSTM reasoning module

To inject a sequential inductive bias, we optionally pass the token embeddings through a bidirectional LSTM with parameters ϕ_{LSTM}. Let the BiLSTM produce final forward and backward hidden states $h_k^{\rightarrow}, h_k^{\leftarrow} \in \mathbb{R}^{d_r}$. We define the LSTM text representation as the concatenation:

$$\ell_k = \text{BiLSTM}_{\phi_{\text{LSTM}}}(H_k) = [h_k^{\rightarrow}; h_k^{\leftarrow}] \in \mathbb{R}^{2d_r}. \tag{10}$$

When the LSTM is used we replace (or complement) t_k with ℓ_k; in practice we project ℓ_k to dimension d via a linear layer if $2d_r \neq d$.

4.2.4 Cross-modal attention

We implement text-to-image cross-attention so that textual tokens can query visual features. For simplicity denote the set of key/value visual features as $V_k \in \mathbb{R}^{P\times d}$ (in our implementation $P = 1$ when using a global image vector; patch-level $P > 1$ is also supported). Using multi-head attention with parameters ϕ_{XAttn}, the attended text representations are

$$\widetilde{H}_k = \mathrm{CrossAttn}_{\phi_{\mathrm{XAttn}}}\big(Q = H_k,\ K = V_k,\ V = V_k\big) \in \mathbb{R}^{L\times d}. \tag{11}$$

Concretely, a single-head scaled dot-product attention (per head) is

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\Big(\tfrac{QK^\top}{\sqrt{d_h}}\Big)V, \tag{12}$$

and multi-head attention concatenates several such heads followed by an output projection. After cross-attention we either pool $\widetilde{H}_k$ to produce $t_k = \mathcal{P}_\pi(\widetilde{H}_k, m_k)$ or feed $\widetilde{H}_k$ into the LSTM module, depending on the configuration.

4.2.5 Fusion and classification

We fuse the (possibly LSTM-transformed) text vector and the projected visual vector by concatenation and a small MLP classifier. Let z_k denote the final text representation used (either t_k or a projected ℓ_k) and let v_k be the projected visual vector. We apply masking for missing images by multiplying v_k by $\mathbf{1}_{I_k}$. The fusion input is

$$u_k = [\, z_k;\ v_k \cdot \mathbf{1}_{I_k}\,] \in \mathbb{R}^{d_f}. \tag{13}$$

The classifier $f_{\phi_{\mathrm{clf}}}$ is an MLP:

$$h_k^{(1)} = \mathrm{LayerNorm}\big(W^{(1)}u_k + b^{(1)}\big), \tag{14}$$

$$h_k^{(2)} = \mathrm{ReLU}\big(W^{(2)}h_k^{(1)} + b^{(2)}\big), \tag{15}$$

$$\mathrm{logits}_k = W^{(o)}h_k^{(2)} + b^{(o)} \in \mathbb{R}^C. \tag{16}$$

Predicted probabilities are obtained via softmax:

$$\hat{p}_\phi(y \mid q_k, c_k, I_k) = \mathrm{softmax}(\mathrm{logits}_k), \tag{17}$$

where ϕ denotes the collection of all multimodal parameters $\{\phi_{\text{text}}, \phi_{\text{vis}}, W_v, \phi_{\mathrm{LSTM}}, \phi_{\mathrm{XAttn}}, \phi_{\mathrm{clf}}, \dots\}$ (only the chosen modules are active for a given variant).

4.2.6 Training objective

All model variants are trained with the standard cross-entropy loss:

$$\mathcal{L}(\phi) = -\frac{1}{N}\sum_{k=1}^{N} \log \hat{p}_\phi\big(y_k \mid q_k, c_k, I_k\big). \tag{18}$$

When instruction-tuned CoT supervision is combined with the multimodal classifier (e.g., training a decoder to produce both reasoning and classification), one may use a multitask objective that jointly optimizes reasoning token likelihood and classification. Let θ denote parameters of a generative decoder used for CoT. A simple joint objective is

$$\mathcal{L}_{\text{joint}}(\phi, \theta) = \lambda\, \mathcal{L}_{\text{CE-clf}}(\phi) \ +\ (1-\lambda)\, \mathcal{L}_{\text{CoT}}(\theta), \tag{19}$$

where $\lambda \in [0, 1]$ balances the classification cross-entropy $\mathcal{L}_{\text{CE-clf}}(\phi)$ and the instruction-tuning next-token loss over CoT tokens

$$\mathcal{L}_{\text{CoT}}(\theta) = -\sum_{k=1}^{N}\sum_{t=1}^{T_k} \log p_\theta\big(r_{k,t} \mid x_k, r_{k,<t}\big), \tag{20}$$

with r_k the supervision rationale for example k and x_k the prompt (see your Instruction-Tuned CoT formalism). In experiments where the classifier is trained alone (no generative decoder), set $\lambda = 1$.

4.2.7 Model variants and ablations

We denote the following variants compactly:

$$\begin{aligned}
\text{Baseline}_{\text{CLS}}1: &\ \text{use } \mathcal{P}_{\text{CLS}},\ \text{no LSTM},\ \text{no XAttn},\\
\text{Baseline}_{\text{att}}2: &\ \text{use } \mathcal{P}_{\text{att}},\ \text{no LSTM},\ \text{no XAttn},\\
\text{LSTM-Enhanced}4: &\ \text{use BiLSTM } \ell_k \text{ (replace/augment } t_k),\\
\text{CrossAttn}3: &\ \text{use } \mathrm{CrossAttn} \text{ before pooling},\\
\text{Hybrid}5: &\ \text{LSTM + CrossAttn + attention pooling (all components)}.
\end{aligned}$$

Each variant is trained with $\mathcal{L}(\phi)$ (or $\mathcal{L}_{\text{joint}}$ when combined with CoT supervision). Ablation experiments compare performance under component removal to quantify each module's contribution.

The model architecture diagrams are included in the Appendix and can be accessed using the reference number (Appendix 5).

4.2.8 Notes on implementation

- When global image features are used ($P = 1$), cross-attention still provides benefit by conditioning textual token representations on the image vector; for finer alignment, P can be increased using patch or region features.
- Missing modalities are handled deterministically via masking (zeroing) to avoid distribution shift between examples with and without images.
- All projection and classifier weights are trained from scratch while pretrained encoders can be either frozen or fine-tuned depending on experimental setup; we denote the tunable parameter subset explicitly in implementation details.

4.3 Chain-of-Thought Instruction Tuning for ScienceQA

We explore two complementary ways of incorporating **chain-of-thought (CoT)** into instruction tuning for ScienceQA. The first is **explicit CoT supervision**, where the model is trained directly on rationale tokens. The second is a **decoupled multimodal prompting pipeline**, where a frozen vision model converts images into text and the LLM is fine-tuned to output answers under a structured CoT prompt (with loss masking).

4.3.1 (A) Explicit CoT-Supervised Instruction Tuning

We extend a base autoregressive language model M_θ via supervised instruction tuning with explicit CoT supervision. We construct a dataset of instruction–reasoning–answer tuples:

$$\mathcal{D} = \{(i_k, q_k, r_k, a_k)\}_{k=1}^{N}, \tag{21}$$

where i_k is a high-level instruction, q_k is the task input (question), r_k is the chain-of-thought rationale, and a_k is the final answer.

Input / Output Formatting. For each example, we linearize the components into a single instruction-style prompt x_k and target completion y_k.

Input (conditioning). We prepend a fixed system prompt s and use role-style markers:

$$x_k = \texttt{<bos>}\ s \parallel \text{“User:”}\ (i_k \oplus q_k) \parallel \text{“Assistant: Let's reason step by step.”}, \tag{22}$$

where $\oplus$ denotes string concatenation, and $\parallel$ denotes formatting boundaries.

Target (supervision). The target sequence explicitly contains the CoT followed by a clearly delimited final answer:

$$y_k = r_k \oplus \text{“\textbackslash nFinal answer:”} \oplus a_k \oplus \texttt{<eos>}. \tag{23}$$

In our implementation, the data pipeline (i) adds system/user/assistant prefixes, (ii) inserts the cue “Let's reason step by step.” before the CoT, and (iii) inserts the delimiter “Final answer:” before a_k. This teaches a consistent pattern: generate r_k first, then output a parsable a_k.

Training Objective. Let $x_k = (x_{k,1}, \ldots, x_{k,m})$ and $y_k = (y_{k,1}, \ldots, y_{k,T_k})$ be the tokenized input and target. We fine-tune with standard next-token cross-entropy over the assistant tokens:

$$\mathcal{L}(\theta) = -\sum_{k=1}^{N} \sum_{t=1}^{T_k} \log p_\theta\big(y_{k,t} \mid x_k, y_{k,<t}\big), \tag{24}$$

where $y_{k,<t} = (y_{k,1}, \ldots, y_{k,t-1})$.

Baseline: Direct-Answer Instruction Tuning. As a baseline, we train a direct-answer instruction-tuned model using the same inputs but without CoT supervision. The prompt omits the reasoning cue, and the target contains only the final answer:

$$y_k^{\text{base}} = a_k \oplus \texttt{<eos>}, \tag{25}$$

with loss

$$\mathcal{L}_{\text{base}}(\theta) = -\sum_{k=1}^{N} \sum_{t=1}^{T_k^{\text{base}}} \log p_\theta\big(y_{k,t}^{\text{base}} \mid x_k^{\text{base}}, y_{k,<t}^{\text{base}}\big). \tag{26}$$

This isolates the effect of explicit CoT supervision: the underlying model and optimization remain unchanged, and only the target formatting differs.

4.3.2 (B) Decoupled CoT Prompting with Multimodal Context (BLIP + LLM)

We also implement a **decoupled, prompt-based multimodal pipeline** that separates **visual understanding** from **language reasoning**. Raw multimodal inputs (image I_k and lecture context L_k) are converted into a single, information-rich textual prompt P_k suitable for LLM fine-tuning.

Visual Contextualization (BLIP). We use a frozen BLIP captioning model [3] (`Salesforce/blip-image-captioning-base`) as a visual contextualizer to generate a natural language description C_k of image I_k. This is done via a custom `caption_images_batch` method (code developed by the author), which **processes images in parallel batches** to significantly accelerate data preparation

$$C_k = \text{BLIP}(I_k). \tag{27}$$

Prompt Construction. For each example k, the prompt is constructed as:

$$P_k = [\text{Context: } L_k,\ \text{Image Description: } C_k,\ \text{Question: } Q_k,\ \text{Choices: } \mathcal{C}_k,\ \text{CoT Template}], \tag{28}$$

where Q_k is the question and $\mathcal{C}_k$ are the multiple-choice options.

Structured CoT Template. To encourage consistent rationales (building on [2]), we include a fixed, structured CoT template:

```
Let's think step by step:
1. What is the question asking?
2. What information is relevant?
3. Which choice is correct?

Answer:
```

Example, attached here

Efficient Fine-Tuning with LoRA and Loss Masking

Base Model and LoRA. We fine-tune `Qwen/Qwen3-8B` [6] using LoRA [7]. For a weight matrix W, LoRA parameterizes the update as a low-rank factorization:

$$\Delta W = BA, \quad W_{\text{new}} = W_{\text{base}} + BA, \tag{29}$$

with rank $r = 32$, reducing trainable parameters while preserving expressiveness.

Training Implementation. Training is executed on the Tinker platform via the Tinker SDK [8], using custom remote execution primitives (e.g., `forward_backward` and `optim_step`).

Training Objective with Loss Masking. We optimize causal language modeling (CLM) with **loss masking** so that only the answer tokens contribute to the loss. Let T_k denote the full token sequence (prompt + answer) and $w_{k,t}$ be a binary weight:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{k=1}^{N} \sum_{t=1}^{|T_k|} w_{k,t} \log p_\theta(t_{k,t} \mid t_{k,<t}), \tag{30}$$

$$w_{k,t} = \begin{cases} 0.0 & \text{if } t \in P_k \\ 1.0 & \text{if } t \in A_k, \end{cases} \tag{31}$$

where A_k denotes the ground-truth answer tokens. This trains the model to produce the correct answer following the CoT template without penalizing prompt tokens.

Baseline and Evaluation. A pretrained `Qwen/Qwen3-8B` model serves as the baseline. Zero-shot accuracy is measured using identical structured prompts P_k (including the CoT template) for both baseline and fine-tuned models, isolating gains attributable to supervised fine-tuning.

5 Experiments

5.1 Data

We use the SCIENCEQA [9]benchmark via its HuggingFace release `derek-thomas/ScienceQA`[1]. Each instance consists of an image, an optional textual context, a question, answer options, and a rationale. We model every example as an input triplet $(x_{\text{img}}, x_{\text{text}}, q)$ with target $y = (answer, \textit{explanation})$, and evaluate models on predicting y.

5.2 Evaluation Method

We follow the evaluation protocol of the original SCIENCEQA benchmark [1]. Since each example is a multiple–choice question with a single correct option, we use **classification accuracy** as our main metric, defined as the percentage of test questions for which the predicted option matches the correct answer.

To align with prior work and analyze performance in more detail, we report:

- **Overall accuracy**: over all test questions.
- **By subject**: separate accuracies for natural science (NAT), social science (SOC), and language (LAN).
- **By context type**: questions with text only (TXT), image only (IMG), both (TXT+IMG), or no context (NO).
- **By grade level**: grouped into grades 1–6 and 7–12.

All metrics are computed on the held–out test split and reported as percentages.

[1] `https://huggingface.co/datasets/derek-thomas/ScienceQA`

5.3 Experimental details

Table 1: Model Training Configuration Parameters

Model	Params (M)	Batch Size	LR	Epochs	Time (hrs)	Arch. Type	Modality	Other Details
Standard Baseline Architectures								
`unifiedqa-t5-small`	60	8	5×10^{-5}	3	1.5	Enc-Dec (Seq2Seq)	Text	Max Len 128
`bert-base-uncased`	110	4	5×10^{-5}	3	1.5	Enc-only (Discrim.)	Text	Max Len 256
`Qwen2.5-1.5B-Instruct`	1,500	4	5×10^{-5}	3	5.0	Dec-only (Gen. LLM)	Text	Max Len 256
`blip-vqa-base`	220	8	5×10^{-5}	3	4.0	Vision-Language (VQA)	Text + Vision	Max Len 128
`llava-v1.6-mistral-7b`	7,300	1	5×10^{-5}	3	12.0	Vision-Enc. + Dec.	Text + Vision	Max Len 128
`git-base-textvqa`	120	8	5×10^{-5}	3	3.0	Vision-Language (VLM)	Vision	Max Len 128
Instruction-Tuned Chain-of-Thought								
`UnifiedQA`	223	4	5×10^{-5}	5	4	Dec-only (LLM)	Text + Vision	ViT Pre-Captioning, CoT
Decoupled Instruction Tuning								
`Qwen3-8B` (LoRA)	8,000	128	3×10^{-5}	5	3	Dec-only (LLM + LoRA)	Text + Vision	BLIP Pre-Captioning, CoT, $r = 32$
Custom Multimodal Hybrid Architectures								
`baseline_cls`	135.48	32	2×10^{-5}	10	1.0	Transformer-based	Text + Vision	CLS Pooling
`baseline_attention`	135.49	32	2×10^{-5}	10	1.0	Transformer-based	Text + Vision	Attention Pooling
`lstm_attention`	139.03	32	2×10^{-5}	10	1.0	LSTM-Augmented	Text + Vision	Attention + LSTM
`cross_attn`	137.85	32	2×10^{-5}	10	1.0	Cross-Attention	Text + Vision	Cross-Attn
`hybrid`	141.40	32	2×10^{-5}	10	1.0	Full Hybrid	Text + Vision	LSTM + Cross-Attn
`hybrid_v2`	141.40	32	2×10^{-5}	15	1.5	Full Hybrid	Text + Vision	LSTM + Cross-Attn

Note: All models use weight decay 0.01 *and a linear warmup schedule. Time here is training time.*

6 Results

Table 2: Accuracy (%) for all models, overall and by subject, context type, and grade level.

Model	Overall	NAT	SOC	LAN	Context Type			Grade Level		
					TXT	IMG	NO	TXT+IMG	G1–6	G7–12
Baseline Models										
`Qwen2.5-1.5B-Instruct + git-base-textvqa`	59.00	60.63	33.33	69.57	93.33	22.73	71.67	38.36	57.75	62.07
`UnifiedQA (fine-tuned)`	57.23	56.26	58.72	58.00	63.88	—	59.23	—	59.91	52.41
`UnifiedQA (pre-trained)`	41.38	42.01	36.45	44.09	46.64	—	43.14	—	43.21	38.10
`LLaVA-v1.6-Mistral-7B (pre-trained)`	40.20	41.65	26.43	48.36	43.85	25.92	44.88	41.21	40.31	40.01
`BLIP (pre-trained)`	2.66	2.31	0.90	4.82	2.79	1.05	4.39	1.59	2.61	2.77
`BLIP (fine-tuned)`	0.02	0.00	0.11	0.00	0.00	0.00	0.00	0.08	0.00	0.07
`BERT (fine-tuned)`	42.75	43.87	32.62	48.64	42.75	—	45.44	—	43.61	41.20
Custom Multimodal Hybrid Architectures										
`baseline_cls`	65.60	67.17	66.99	61.36	75.51	65.56	60.00	65.31	66.98	63.13
`baseline_attention`	62.00	64.15	64.08	56.06	72.45	60.00	55.15	63.95	61.37	63.13
`lstm_attention`	64.60	64.53	67.96	62.12	70.41	65.56	61.82	63.27	66.04	62.01
`cross_attn`	65.40	65.66	66.02	64.39	74.49	61.11	63.64	63.95	64.17	67.60
`hybrid`	60.80	60.75	64.08	58.33	71.43	60.00	58.18	57.14	61.99	58.66
`hybrid_v2`	67.40	70.19	65.05	63.64	79.59	61.11	64.24	66.67	67.29	67.60
Instruction-Tuned Chain-of-Thought										
`UnifiedQA CoT (ViT image captioning)`	48.20	41.60	40.38	36.89	47.73	37.76	41.11	40.14	45.45	47.51
Decoupled Instruction Tuning										
`Qwen3-8B (LoRA) (BLIP pre-captioning)`	78.00	76.90	74.44	84.11	87.07	79.17	88.89	67.30	81.98	69.23

We report **accuracy (%)** overall and along three axes: (i) **subject** (NAT, SOC, LAN), (ii) **context type** (TXT, IMG, NO, TXT+IMG), and (iii) **grade band** (G1–6, G7–12). Training configurations are summarized in Table 1, and main accuracy results are shown in Table 2.

6.1 Overall Performance

Across all compared methods, the **decoupled instruction-tuning pipeline** (`Qwen3-8B (LoRA)` with **BLIP pre-captioning** and a structured CoT template) achieves the best overall accuracy of **78.00%** (Table 2). This is a substantial improvement over both (i) the best **standard baseline architecture** (`Qwen2.5-1.5B-Instruct + git-base-textvqa`, **59.00%**) and (ii) the strongest **custom multimodal hybrid** (`hybrid_v2`, **67.40%**). Notably, a direct **instruction-tuned CoT** baseline (`UnifiedQA CoT (ViT image captioning)`) underperforms at **48.20%**, indicating that CoT prompting alone is insufficient without a strong multimodal prompting and fine-tuning strategy.

6.2 Subject-wise Trends

The decoupled LoRA approach improves performance consistently across subjects, reaching **76.90%** on NAT, **74.44%** on SOC, and **84.11%** on LAN (Table 2). Among the hybrid models, `hybrid_v2` is the strongest overall and is particularly competitive on NAT (**70.19%**), but remains behind the decoupled method across all subjects.

6.3 Robustness Across Context Types

A key differentiator is robustness to context type. The best standard baseline (`Qwen2.5 + git`) performs very well on TXT (**93.33%**) but degrades sharply on IMG (**22.73%**) and TXT+IMG (**38.36%**), suggesting strong dependence on textual cues and limited visual grounding (Table 2). In contrast, the decoupled approach maintains high accuracy across all settings, achieving **87.07%** (TXT), **79.17%** (IMG), **88.89%** (NO), and **67.30%** (TXT+IMG). This supports the benefit of **caption-based visual contextualization** and **structured CoT prompting** during fine-tuning.

6.4 Grade-level Performance

Performance is higher for G1–6 than G7–12 across most systems (Table 2). The decoupled approach reaches **81.98%** on G1–6 but drops to **69.23%** on G7–12, indicating that later-grade questions remain more challenging, likely due to increased compositional reasoning and domain knowledge demands.

6.5 Efficiency Considerations

Custom hybrid architectures are comparatively lightweight ($\sim$135–141M parameters) and train quickly (about 1–1.5 hours in our setup; Table 1), achieving up to **67.40%** overall. The decoupled method uses a larger LLM backbone (`Qwen3-8B`) but leverages **LoRA** to keep fine-tuning efficient while delivering the strongest accuracy. Overall, these results suggest a favorable trade-off: **caption-based decoupling + LoRA fine-tuning** yields the best end-to-end performance, while **hybrid models** provide a strong, compute-efficient alternative.

7 Analysis

7.1 Component Behavior

Ablations show structured reasoning is essential. Adding a BiLSTM improves performance (62.00% $\rightarrow$ 64.60%), suggesting long-range textual dependencies matter. Cross-attention also outperforms simple concatenation (65.40% vs. 62.00%), indicating more precise visual grounding.

7.2 Visual Bottlenecks

Split modality based pipelines (e.g., Qwen+GIT, 38.36% on image questions) lose critical spatial detail. In contrast, Hybrid v2 (66.67%) benefits from direct feature fusion, avoiding this information bottleneck.

7.3 Domain Performance

Hybrid v2 excels in Natural Science questions (70.19%), where diagrams align well with ImageNet-pretrained encoders, but struggles in Social Science tasks (65.05%), which require culturally grounded visual understanding.

7.4 Chain-of-Thought Models

CoT methods provide an upper bound for reasoning performance. CoT fine-tuning improves accuracy by **3–4%** and state-of-the-art models exceed **78%** 2[1]. CoT is particularly effective for multi-step reasoning but prone to hallucination if early rationale steps are incorrect.

7.5 Analysis: Chain-of-Thought

CoT improves interpretability and allows easy debugging via generated rationales. However, it requires longer sequences, higher latency, and typically larger models. Effective multimodal CoT must integrate visual information during rationale generation to avoid hallucinated explanations.

8 Conclusion

Our results show that compact multimodal discriminative models can rival or exceed much larger generative systems on ScienceQA, offering strong accuracy with minimal compute. However, they remain limited in multi-step reasoning compared to CoT-based approaches. Future work may combine efficient feature fusion with grounded multimodal CoT generation to balance accuracy, interpretability, and efficiency.

9 Contributions

All authors contributed equally to this work. The research, analysis, and report writing were carried out collaboratively, and the final manuscript was completed jointly.

- **Reneth Raj Simon**: Benchmark baselines and CoT model
- **Sreeram Raghammudi**: Hybrid Models + baseline fintuning
- **Vishal Menon**: Benchmark baslines and CoT model

References

[1] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[2] Pan Lu, Swaroop Mishra, Ahsaas Bajpai, Abhinav Anand, Saneem Ahmed Chemmengath, Chitta Baral, Tony K. Chen, Mark Johnson, Tanmay Shrotriya, Ashwin Kuplish, Bhavana Sachdeva, Shailesh Saluja, Moontae Lee, Romila Majumder, Roy Fox, and Robert Balog. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 25017–25030, 2022.

[3] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *International Conference on Machine Learning (ICML)*, volume 162, pages 12888–12903, 2022.

[4] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts, 2022.

[5] Nitya Tiwari, Parv Maheshwari, and Vidisha Agarwal. Cross domain evaluation of multimodal chain-of-thought reasoning of different datasets into the amazon cot framework, 2025.

[6] Alibaba Cloud. Qwen Large Language Model. https://qwenlm.github.io/.

[7] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2022.

[8] Thinking Machines Lab. Tinker SDK Documentation. https://tinker-docs.thinkingmachines.ai/, 2025. Accessed: [Insert Month and Year of Access].

[9] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering, 2022.

A Appendix

A.1 Chain of Thought Sample

```
=== Formatted Example ===

INPUT:
Context: an aerial view of a painting of a forest
Which of these states is farthest north?
Options: (A) West Virginia (B) Louisiana (C) Arizona (D) Oklahoma

TARGET:
The answer is (A).

BECAUSE:
Maps have four cardinal directions, or main directions...
West Virginia is farthest north.
```

A.2 GitHub Repository

The code for this project is available at:
https://github.com/Sreeram-Ragha/NLP-Final-Project-COMS4705

A.3 Hybrid Model Figures

Next page onward.

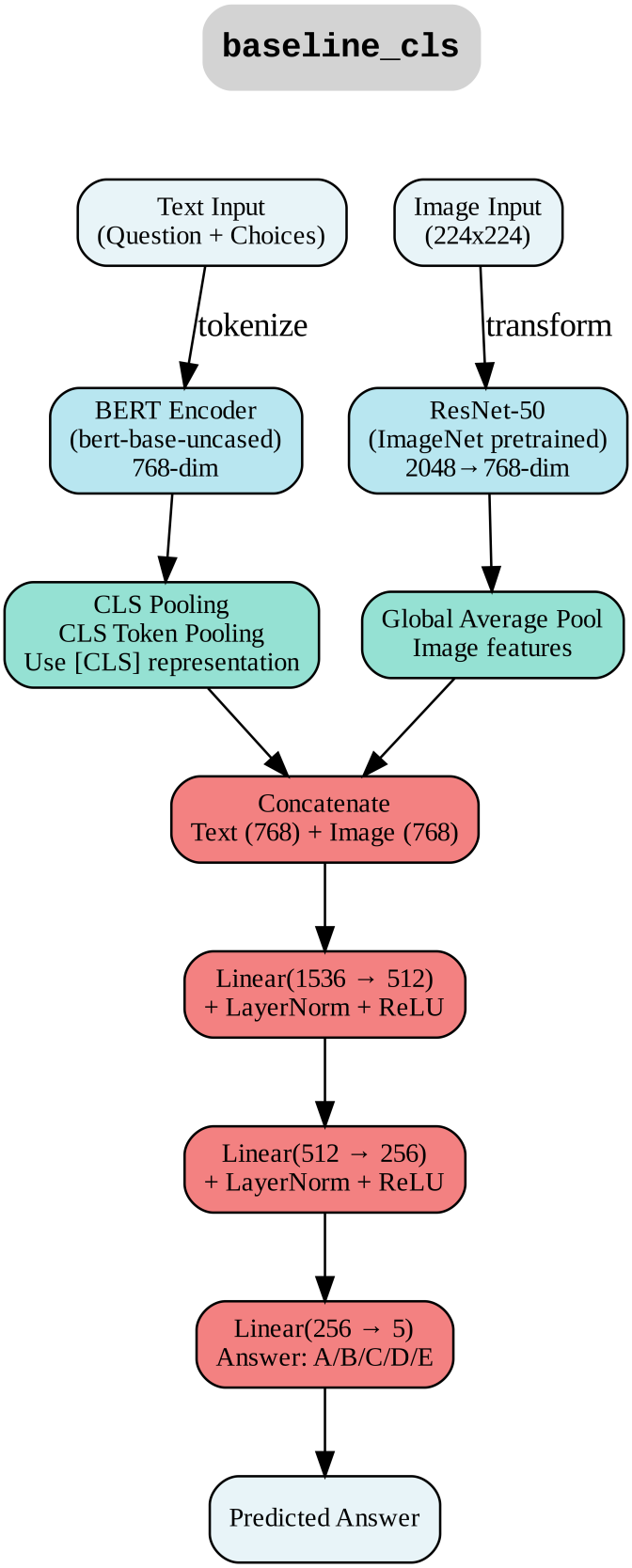

Figure 1: Model architecture of baseline$_{cls}$

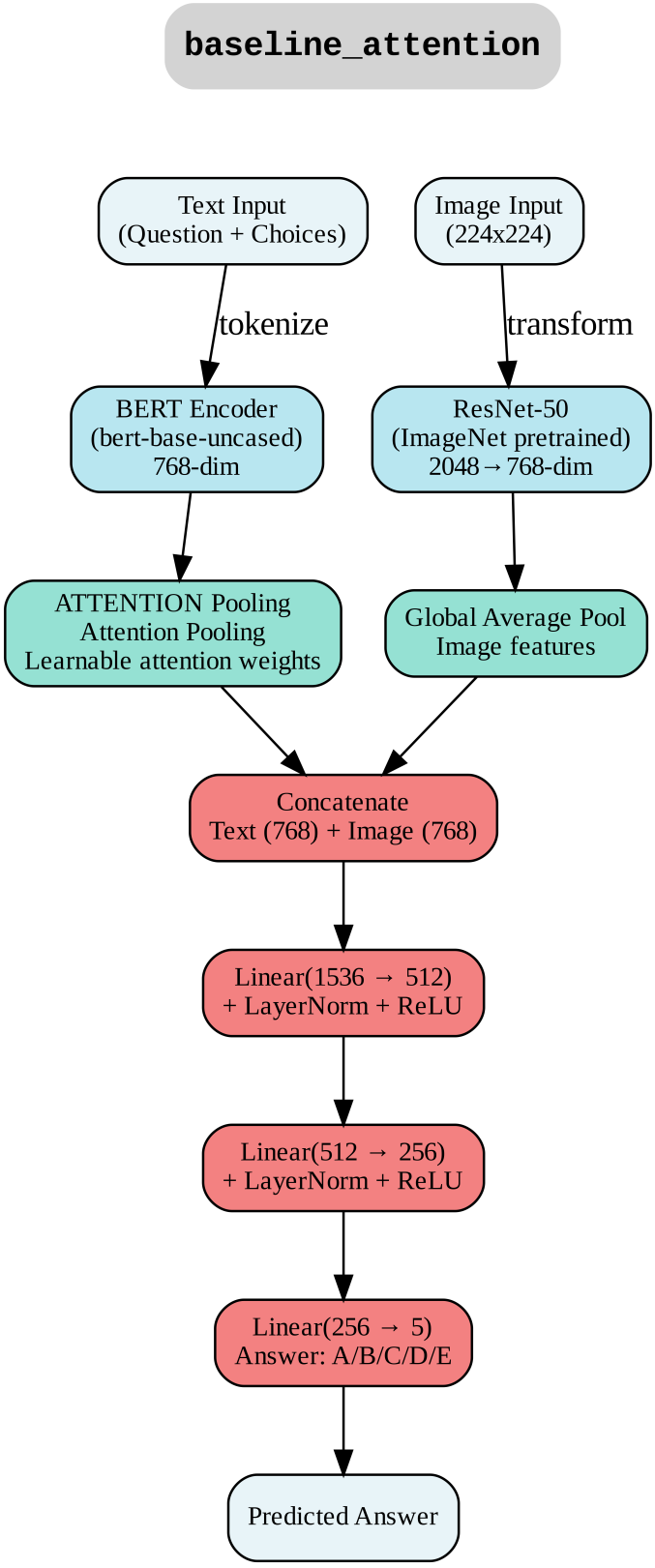

Figure 2: Model architecture of baseline$_{attention}$

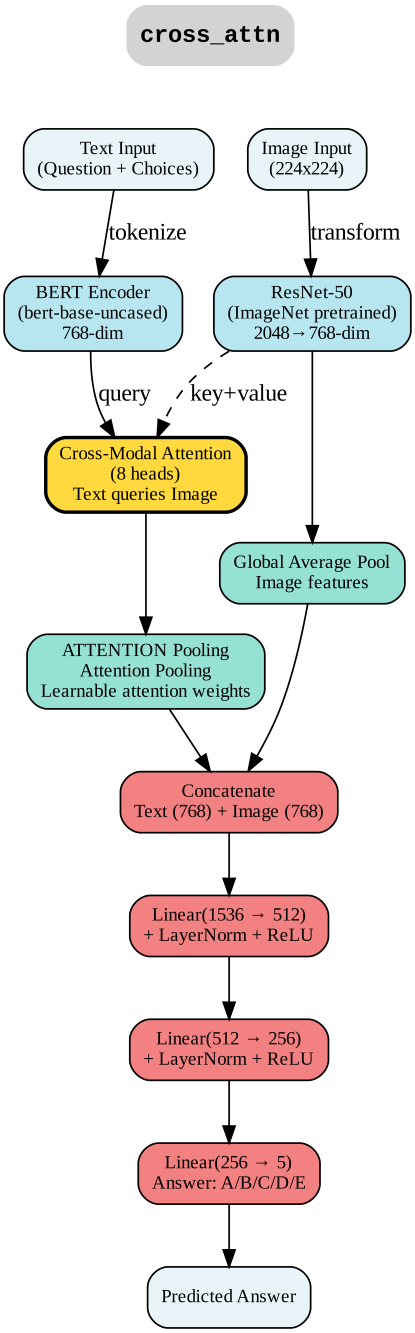

Figure 3: Model architecture of cross$_a ttn$

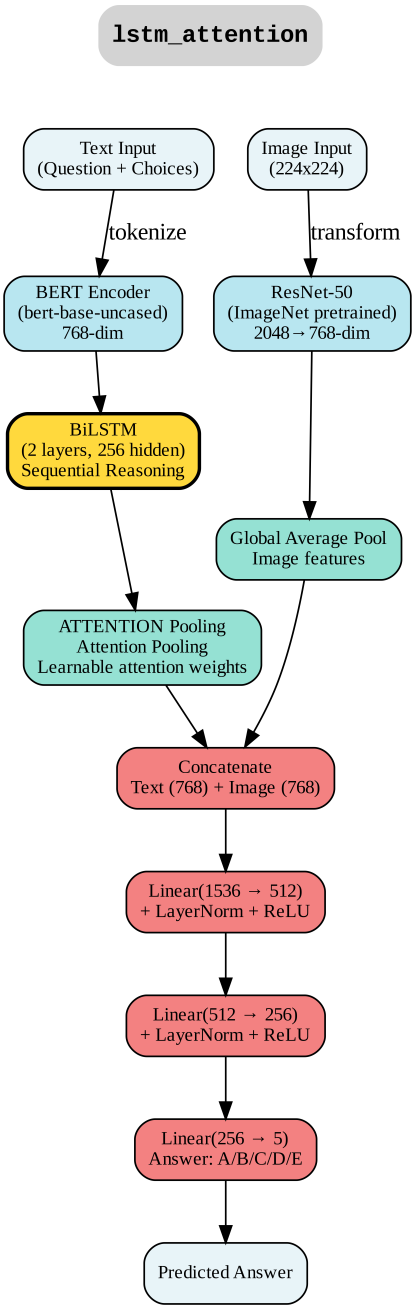

Figure 4: Model architecture of lstm$_{attention}$

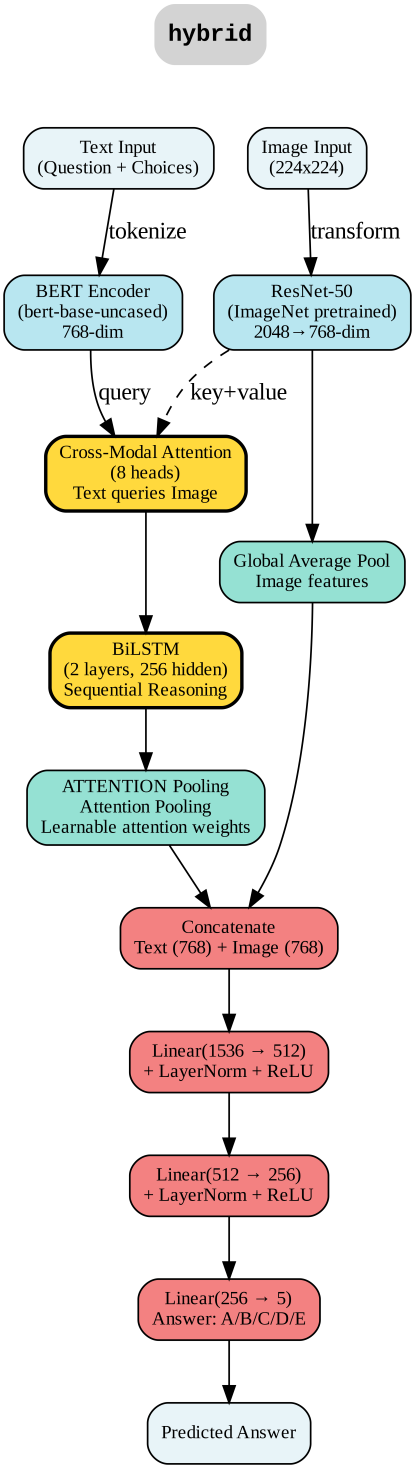

Figure 5: Model architecture of hybrid