Reasoning QA Pipeline: Data, Models, Workflow, Deployment, and Results

August 11, 2025

1 Data

1.1 Data Sources

We assembled our dataset of high-school and JEE-level reasoning questions from two complementary streams:

Open-web repositories

- Crawled question banks and PDFs linked from educational websites (e.g., past year papers, forum archives).
- Targeted high-school (NCERT) and major JEE guides (Arihant, Allen, Cengage) covering mathematics, physics, biology, and chemistry.

"EM Data"

- Internal available dumps ($\sim$181k items) from online Q&A platforms, scraped via their APIs.

Across both streams we applied the same extraction pipeline (PDF $\rightarrow$ text + image $\rightarrow$ QA pairs), then de-duplicated and filtered for high-quality, full-context questions.

1.2 Preprocessing & Modality Split

Text extraction

- PDF parsing via PyMuPDF $\rightarrow$ raw text blocks.
- Our data is manually extracted from different books.
- Heuristic grouping into question / options / solution.

Image extraction

- Manually took screenshots of every image question from different sources like books, exams, etc.

Name / Modality	Text Only	Image Based	Total
EM Data	175,185	5,815	181,000
Ours (Math & Physics)	145,550	18,910	164,460
Ours (Biology)	8,592	3,682	12,274
Ours (Chemistry)	6,574	1,643	8,217

Table 1: Dataset split by modality. Biology and Chemistry values are illustrative.

Final split

1.3 Reasoning-Level Annotation

To gauge the complexity of each question, we adopted a five-level taxonomy:

Level	Description
L-1	Basic Arithmetics
L-2	Elementary Problem Solving
L-3	Moderate Conceptual Thinking (e.g., combine two concepts/formulae)
L-4	Advanced Reasoning
L-5	Complex problem mainly requires reasoning (complex derivation / proofs)

Table 2: Reasoning levels.

Note. Biology & Chemistry questions proved predominantly factual and fell into L-1/L-2; hence we focus our reasoning analysis on the Maths & Physics subset.

Dataset	L-1	L-2	L-3	L-4	L-5
EM Data	44,969	29,976	74,210	22,444	9,412
Ours (Math & Physics)	22,594	16,063	91,056	27,998	11,849
Ours (Biology & Chemistry)	—	—	—	—	—

Table 3: Level-wise counts by dataset (as provided).

Dataset	L-1	L-2	L-3	L-4	L-5
EM Data	899	901	2,968	673	376
Ours (Math & Physics)	2,537	2,884	12,759	3,651	1,607
Ours (Biology & Chemistry)	—	—	—	—	—

Table 4: Text based classification

2 Prompt: Question Difficulty & Subject Classifier

Purpose This prompt instructs an LLM to classify a given question into two attributes: (i) Difficulty Level—a number from 1 to 5, based on complexity, required concepts, and reasoning depth; and (ii) Subject Category—one of a predefined set of subject labels.

[0]Numbers in this row are reproduced as provided.

Prompt Text

Role & Task:
You are a question difficulty classifier. Your task is to assign a difficulty level from 1 to 5 to any given question based on its complexity, required concepts, and reasoning depth. Additionally, classify the subject category.

Difficulty Levels:
Level 1 – Basic Arithmetic
Direct, single-step calculations (addition, subtraction, multiplication, division). No reasoning or multi-step logic.
Examples: What is 6 + 4?; Multiply 3 and 7.

Level 2 – Elementary Problem Solving
Slightly more involved; may require interpreting short text, combining 2 steps, or applying elementary math (perimeter, averages). No abstract reasoning.
Examples: A pencil costs INR 5. How much do 3 pencils cost?; What is the average of 10, 20, and 30?

Level 3 – Moderate Conceptual Thinking
Requires basic algebra, geometry, or logic. Multiple steps or concepts; straightforward steps but needs some reasoning.
Examples: Solve for x: $2x + 3 = 11$; Find the area of a triangle with base 6 cm and height 4 cm.

Level 4 – Advanced Reasoning / Multi-Step Logic
Chaining multiple concepts; may include algebraic manipulation, conditional reasoning, interpreting diagrams. Not solvable at a glance.
Examples: A train leaves station A at 9:00 AM and another from station B at 9:30 AM...; If $4x - 2 = 3y$ and $x + y = 10$, what is x?

Level 5 – Intense Multi-Concept Reasoning
Deep understanding, abstraction, combining multiple areas (e.g., number theory, combinatorics, geometry, logic). Tricky even for experienced students.
Examples: Given $f(x) = 2x^2 + 3x + 1$, find the smallest positive integer for which $f(f(x)) = 0$; Complex spatial/algebraic reasoning problems.

Subject Categories:
`general`, `maths`, `physics`, `biology`, `chemistry`, `prover`

Classification Rules:
If the question explicitly asks to "prove", "show that", "demonstrate", "verify", "establish", or "derive" → classify as `prover`. Focus on cognitive complexity and reasoning depth. Consider the number of steps/concepts involved. Evaluate if the solution is immediate or requires deeper thought.

Input:
Question to classify: `{question}`

Explanation of the Prompt This prompt is designed for consistent and explainable classification of questions based on difficulty and subject. The 1–5 scale captures a clear progression from basic, single-step calculations (Level 1) to high-complexity, multi-disciplinary reasoning (Level 5). Having predefined subject tags ensures uniform categorization across all classified questions. The rules prevent ambiguity by setting explicit conditions (e.g., proof detection) and guiding the classification toward reasoning depth rather than just topic difficulty.

3 Models

3.1 Overview & Selection Criteria

We evaluated over 50 different LLMs—both open- and closed-source—across a balance of utility, model size, and inference cost. From this pool, we report:

- **Closed-source:** only OpenAI models (o3, 4.1, 4.1-mini), since these were used in our implementation and consistently outperformed others like Claude for educational Q&A.
- **Open-source:** the top candidates spanning text-only and image-based reasoning, selected for their task-aligned pretraining (PnM, CnB, SST).

Note: We did not evaluate Google Gemini Flash 2.5—although it is cost-effective and strong, it was outside our current compute budget.

3.2 Open-Source Models

Name	**Usage**	**Modality**	**Size**
Qwen QvQ	Solves PnM Qs up to L-3	Image-Based	72B (∼50 GB)
Deepseek R1 Distill (Qwen 8B)	Solves PnM Qs up to L-3	Text-Based	8B (∼10 GB)
Google Med-Gemma 4B	CnB Qs	Image-Based	4B (∼8 GB)
Google Med-Gemma 27B	CnB Qs	Text-Based	27B (∼30 GB)
Llama 3.3 70B	SST Qs (with RAG)	Text-Based	70B (∼38 GB)
DeepSeek R1-8528	PnM > L-3 Qs	Text-Based	685B (∼200 GB)

Table 5: Open-source models (as used/reported).

3.3 Closed-Source Models & Pricing

Name	**Usage**	**Modality**	**Price**
OpenAI o3	PnM > L-3 Qs	Both	$1.5 / $6[*]
OpenAI 4.1	PnM L-3 + other domain Qs (agent per domain)	Both	$1.5 / $6[†]
OpenAI 4.1-mini	Context handling & chaining	Both	$0.4 / $1.6[*]

Table 6: Closed-source models and pricing (per 1K tokens).

[*] Price = input / output per 1K tokens. [†] Same pricing as 4.1, but billed per agent invocation.

4 Workflow

4.1 Overview

Our pipeline ingests any incoming question—whether it's a text-only prompt or contains an embedded figure—and routes it through a sequence of classifiers and solvers, leveraging both local open-source models and, when needed, high-level reasoning APIs. The two sub-flows ("Text vs. Image" and "Subject-&-Level" routing) are fully integrated, ensuring every question is handled by the most appropriate model given its modality, reasoning complexity, and subject domain.

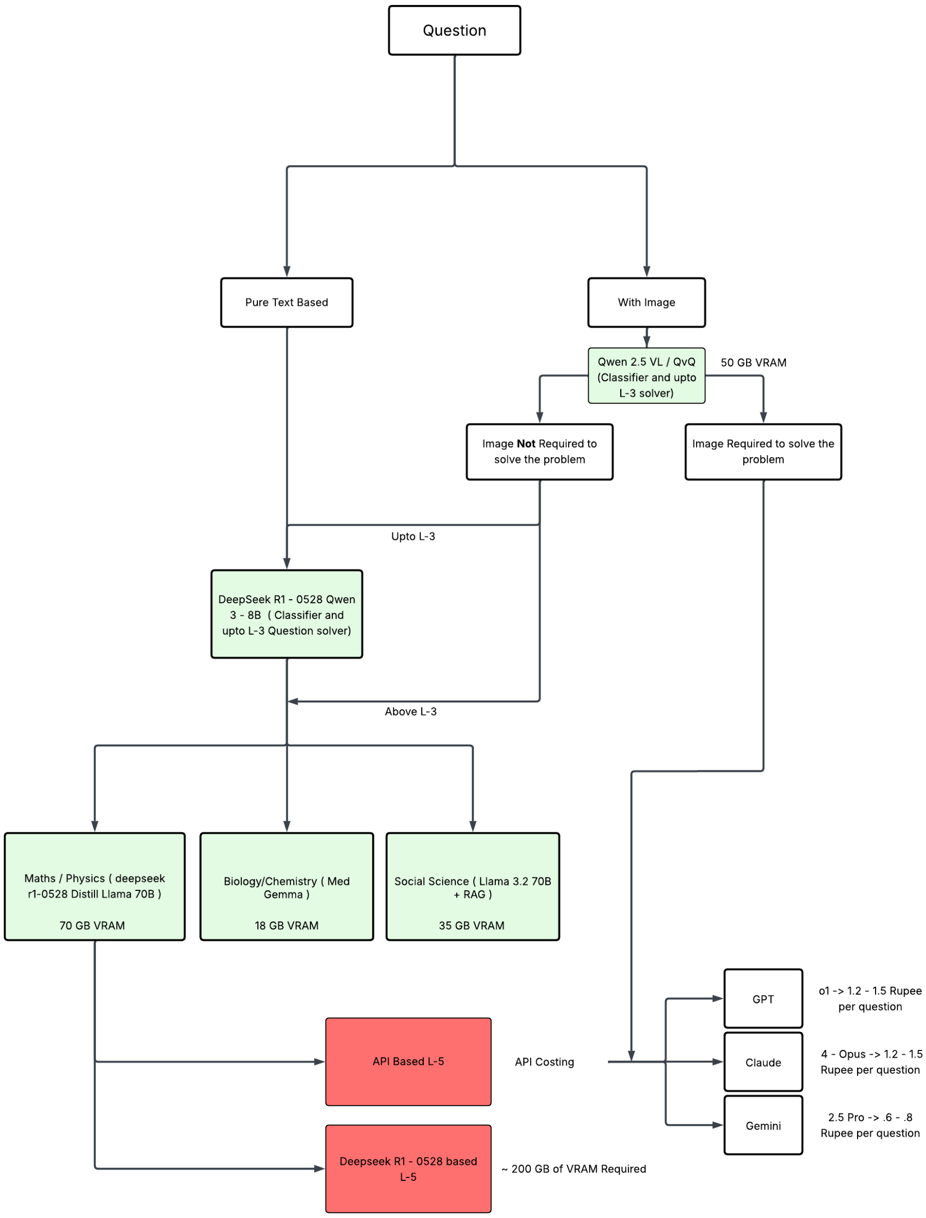

Figure 1: Pipeline

4.2 Ingestion & Modality Split

Receive Question Every request arrives with (i) question text (stem + any options), and (ii) optional `question_image` (diagram, graph, circuit, etc.).

Detect Modality

- If there is no attached image → Pure Text branch.
- If there is an image → With Image branch.

2A. Pure Text Branch

Model: DeepSeek R1-0528 Qwen 8B (text-only classifier & solver).
Action:

1. DeepSeek reads the text and internally estimates reasoning level L_{est}.
2. If $L_{\text{est}} \leq 3$ → DeepSeek directly solves and returns the answer.
3. If $L_{\text{est}} > 3$ → escalate to High-Level Reasoning.

2B. With Image Branch

Model: Qwen QvQ (multimodal classifier & L-3 solver).
Action:

1. QvQ ingests both text + preprocessed image and outputs: `requires_image` (yes/no), $L_{\text{est}} \in [1, 5]$.
2. If `requires_image` = no → fallback to Pure Text Branch (DeepSeek R1).
3. Else if `requires_image` = yes:
   - If $L_{\text{est}} < 3$ → QvQ solves and returns the answer.
   - If $L_{\text{est}} > 3$ → escalate to High-Level Reasoning (L-4/L-5).

4.3 Escalation to High-Level Reasoning

Whenever either branch finds a question that requires more than 3 reasoning steps, we forward it for an L-4/5 solution using:

- **Local L-5:** DeepSeek R1-8528 full model (≈200 GB VRAM, zero token cost).
- **API L-5:** OpenAI (o3/4.1/4.1-mini), Claude, or Gemini (INR 0.4–1.5 per question).

4.4 Subject & Level Routing

For questions of level 3 and below, our pipeline calls open-source models. A particular model is used per subject: If a question is classified as L-4 or L-5, it is escalated to the high-level

Subject	Classifier Source	Model & VRAM	Notes
Math / Physics	Deepseek R1-0528	Distill–Llama 70B (≈70 GB VRAM)	Handles L-3 loc
Biology / Chemistry	Med-Gemma	Google Med-Gemma 4B (≈8 GB VRAM)	Handles L-3 loc
Social Science	Llama 3.2 70B + RAG	Llama 3.3 70B (≈38 GB VRAM)	Handles L-3 loc

Table 7: Subject-specific routing for L-3 and below.

reasoning stage.

5 High-Level (L-5) Reasoning

For Level 5 questions, there are two choices. First, using an open-source model like DeepSeek R1—but these models are large and require around 200 GB VRAM (and batching increases VRAM use). Second, using API-based inference.

5.1 Local L-5 Solver

DeepSeek R1-0528 full (≈685B)
Requirements: ∼200 GB VRAM
Trade-off: zero token cost at the expense of massive hardware.

5.2 API-Based L-5 Solver

OpenAI GPT-4.1 / 4.1-mini / o3
Pricing (per Q): GPT-o3/4.1 ∼ INR 1.2–1.5; 4.1-mini ∼ INR 0.4–0.6; Claude 4 Opus ∼ INR 1.2–1.5; Gemini 2.5 Pro ∼ INR 0.6–0.8.
Automatically selected based on cost / latency requirements.

6 Compute vs. Cost

Path	**Model**	**VRAM Req.**	**Token Cost**
Text-only L-1–L-3	Deepseek Distill Llama 70B	70 GB	0
Image-&-Text L-1–L-3	Qwen QvQ + Med-Gemma / Llama	8–38 GB	0
Local L-5	DeepSeek R1-8528 (full)	200 GB	0
API-based L-5	OpenAI / Claude / Gemini	N/A	INR 0.4–1.5 / Q

Table 8: Compute vs. cost trade-offs.

7 Available Architectures

We support three deployment modes, each trading off capital expenditure (GPU infrastructure) against per-query API costs and operational complexity.

7.1 Purely Local Deployment

Description All models—text-only solvers (DeepSeek R1-0528 Distill Llama 70B), multimodal classifiers (Qwen QvQ), subject-specific solvers (Med-Gemma, Llama 3.3), and the full L-5 engine (DeepSeek R1-8528)—run on a single on-premise "Gin" server. No external API calls are ever made.

Infrastructure & Server Requirements

- GPUs: NVIDIA H100 or H200
- Total VRAM: ≈1.2 TB (spread across multiple GPUs)
- Networking: NVLink / Infiniband fabric for inter-GPU sharding
- Storage: High-I/O NVMe for model checkpoints and cache

Pros

- Lowest inference latency (no network hops)
- Deterministic throughput (∼64 concurrent queries)

Cons

- Very high capex (∼\$100K+ for GPUs alone)

7.2 Hybrid (Local + API Keys)

Description Two compact, general-purpose models locally (e.g., Qwen QvQ and an 8B text solver) handle all L-1 to L-3 on-premise. L-4/L-5 or out-of-coverage domains fall back to external APIs.

Infrastructure & Server Requirements

- GPUs: NVIDIA H100 or H200
- Total VRAM: ≈150 GB (enough to host two small models)
- Networking: Standard Ethernet to cloud endpoints

Pros

- Local, zero-token cost for ≈80% of queries
- Reduced GPU footprint → lower capex
- Graceful degradation if an API is unavailable

Cons

- API latency (∼100–200 ms) for high-level reasoning

Additional Costs API usage (pay-per-token, typically \$0.4–1.5 per 1K tokens).

7.3 Purely API-Based

Description No on-premise GPUs. A lightweight CPU server orchestrates all inference via external LLM APIs. Ideal for rapid prototyping or low-volume use.

Infrastructure & Server Requirements

- CPU: 4–8 vCPUs
- Memory: 16–32 GB RAM
- Network: Reliable internet with low latency to provider endpoints

Pros

- Zero capital investment in hardware
- Automatic access to the latest model versions

Cons

- Highest per-question cost at scale
- Subject to API rate limits and occasional outages

7.4 Summary Comparison

8 Deployment

Below we describe the migration from a GPU-bound HuggingFace setup to a lean, CPU-friendly Ollama/llama.cpp deployment, and its implications for throughput, memory, and ops.

Architecture	GPUs & VRAM	API Cost	Capex / Opex	Notes
Purely Local	H100/H200, ∼1.2 TB VRAM	None	High capex + server mgmt	Ultra-
Hybrid	H100/H200, ∼150 GB VRAM	Moderate (APIs)	Moderate capex + API fees	Best b
Purely API	CPU only	High (APIs)	Zero capex, high variable opex	Easies

Table 9: Deployment architecture comparison.

8.1 Initial HuggingFace–Transformers Setup

Models & Formats Initially we loaded all solvers (Qwen QvQ, DeepSeek Distill Llama 70B, Med-Gemma, Llama 3.3 70B, R1-8528 full) using HuggingFace safetensors.

Hardware Requirements

- Multiple NVIDIA H100/H200 GPUs with >1.2 TB VRAM in aggregate.
- Even text-only models (e.g., Llama 70B) consumed ≈38 GB VRAM each.

Drawbacks

- Large disk footprint: each safetensors checkpoint was tens of gigabytes.
- GPU-only inference: no practical CPU fallback—cold starts and inference failed on CPU.
- Cost: the GPU cluster was underutilized for simple L-1/L-3 queries, driving up opex.

8.2 Migration to Ollama & llama.cpp

Why Ollama / llama.cpp + gguf

- **gguf format:** built-in 8-bit (and lower) quantization; model files shrink by 3–4× vs. full-precision safetensors.
- **CPU inference:** models can run on commodity x86 servers (no GPU); enables "burst" capacity on existing CPU fleets.
- **Easier ops:** single binary (ollama or llama.cpp) per model; simpler container images.

8.3 Batching & Throughput Constraints

- **Max in-flight queries:** 64. Beyond this, host RAM consumption spikes.
- **Memory growth:** 1× batch (64 queries) on a 70B model uses ≈64 × (quantized size + working memory) ≈ 200 GB RAM. Doubling to 128 in-flight causes non-linear growth (swap thrashing or OOM).
- **Implication:** cap concurrency at 64 and use a request queue for backpressure; for peaks, throttle or failover to API-based L-5 (OpenAI).

8.4 Deployment Topology

Component	Software	Resources
Text & Image Solver	Ollama / llama.cpp	8 CPU cores, 128 GB RAM, gguf model
Classifier (QvQ)	Ollama / llama.cpp	4 CPU cores, 64 GB RAM
L-5 Fallback	OpenAI / Claude API	–
Orchestrator	Python + FastAPI	2 CPU cores, 16 GB RAM

Table 10: Deployment components and resource profiles.

Containerization Each Ollama model runs in its own Docker container with dedicated CPU and RAM limits.

Autoscaling For non-peak hours, we spin down extra CPU nodes; during exam season, we scale to multiple replicas (each capped at 64 in-flight).

8.5 Pros & Cons of the Ollama/llama.cpp Deployment

✓ Lower disk usage (gguf $\ll$ safetensors)
✓ CPU inference—no GPU fleet required
✓ Simplified ops (single binary, fewer deps)
✓ Cost-effective for L-1 through L-3 queries
× Concurrency cap at 64 requests
× Memory per batch still high ($\sim$200 GB for 70B)
× Throughput limited compared to GPU deployment

9 Results

End-to-end accuracy broken down by reasoning level (L-1 through L-5) and deployment architecture. For L-3 and above we also report text-only vs. text+image.

Architecture	Level	Overall Acc.	Text-Only	Text+Image
	L-1	∼100%	—	—
	L-2	∼100%	—	—
OURS (Local)	L-3	∼100%	∼100%	∼98%
	L-4	93%	94.5%	78%
	L-5	76%	78%	48%
	L-1	∼100%	—	—
	L-2	∼100%	—	—
OURS + API	L-3	∼100%	∼100%	∼98%
	L-4	98%	98%	93%
	L-5	93%	93%	86%
	L-1	∼100%	—	—
	L-2	∼100%	—	—
API Only	L-3	∼100%	—	—
	L-4	98%	98%	93%
	L-5	93%	93%	86%

Table 11: Accuracy by reasoning level and architecture (as provided).