



CS791- Comprehensive viva¹

Multi-Agent System for Intelligent Query Processing²

Presented by,³

Mihir Modi- 202462001,⁴

MTech (3rd Semester) in DS⁵

Internal Guide,
Dr. Ravi Nahta

Guided by,
Mukesh Siroya⁶

01th October, 2025



Overview¹

Introduction

3

Problem Definition

Motivation

Workflow Diagram

Implementation Details

PDF Chatboard Extension

Multi-Modal Extension

Agentic RAG Extension

APIs List

Initial Results

Conclusion

Future Work

Reference



IIITV & IIITV-ICD

Introduction



- **Overview:** A Python-based multi-agent system integrating weather, GitHub, Q&A, PDF querying, image analysis, and agentic RAG for intelligent, multi-domain query processing.¹
- **Key Features:**²
 - Typo correction (e.g., "Ahmmedabad" → "Ahmedabad") using NVIDIA LLaMA-3³
 - Asynchronous processing with asyncio for speed
 - Modular agents: WeatherAgent, GitHubAgent, QAAgent
 - Extensions: PDF Chatboard, Multi-Modal Image Analysis, Agentic RAG with evaluation.
- **Objective:** Deliver scalable, accurate, user-friendly query resolution.⁴
- **Scope:** Leverages NVIDIA, Gemini, WeatherAPI, and GitHub API in Google Colab⁵

Problem Definition



- **Given:** Diverse user queries across domains (weather, repositories, general knowledge, documents, images)¹
- **Objective:** Provide structured, efficient responses with minimal user effort²
- **Challenges:**³
 - **Varied Query Styles:** Inconsistent inputs (e.g., "Patan temp" vs. "weather in Patan")⁴
 - **API Restrictions:** Rate limits and potential failures
 - **Scalability:** Handle multiple simultaneous queries quickly
 - **Accuracy:** Correct typos and resolve ambiguous inputs
- **Goal:** Build an AI-driven, typo-correcting, scalable query processing system⁵

Motivation



➤ **Why It Matters:** Fast, accurate information access is critical for decision-making in education, software development, and daily planning ²

➤ **Real-World Impact:** ³

- **Software Development:** Quick GitHub repo insights for collaboration ⁴
- **Education:** Reliable answers for students and researchers
- **Daily Use:** Real-time weather for planning (e.g., city storm response)

➤ **Benefits:** ⁵

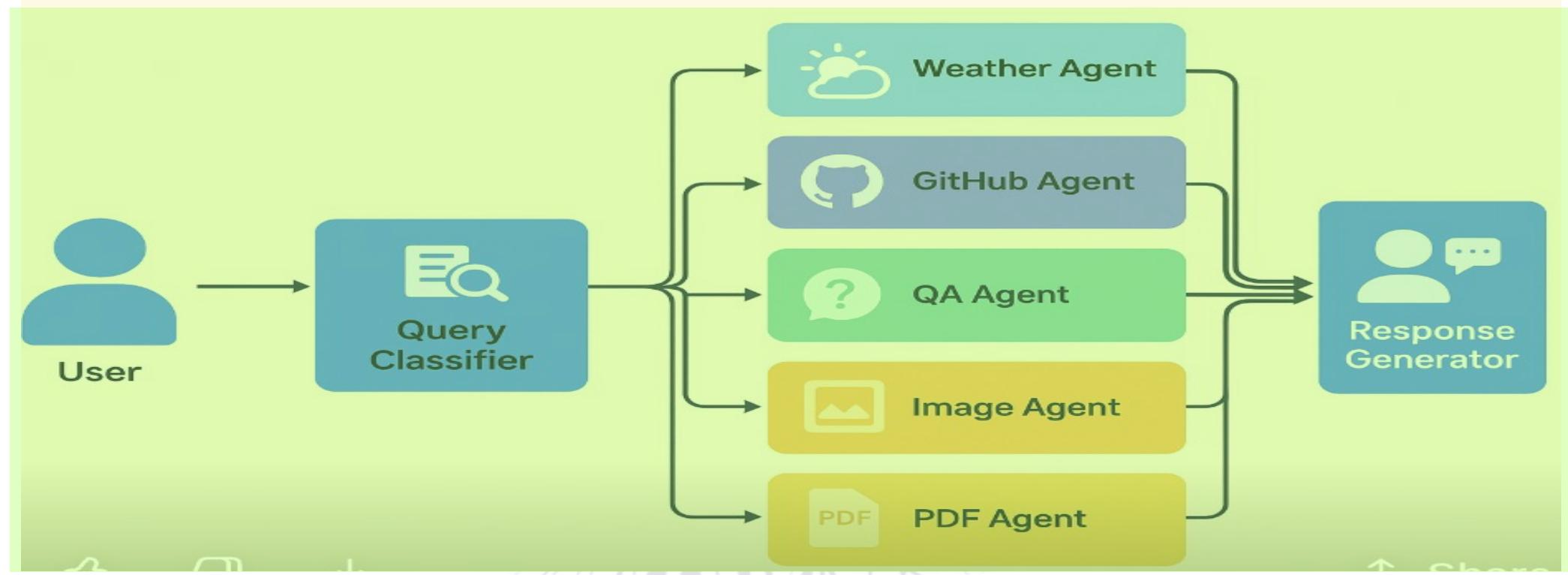
Modularity: Easy to add new agents ⁶

Efficiency: Async processing and caching

User-Friendly: Typo correction and suggestions

Extensibility: Supports new APIs and tasks

WorkFlow Diagram¹



Implementation Details¹

➤ **Environment:** Google Colab, Python 3.10+, asyncio for asynchronous processing.²

➤ **Key Libraries:**⁴

Core: langchain, langgraph, faiss-cpu, ragas, PyPDF2, google-generativeai.⁵

Support: requests, beautifulsoup4, pillow, pydantic, nest_asyncio, backoff.

➤ **Techniques:**⁶

Async I/O for parallel query handling.

FAISS vector store for efficient retrieval.

Pickle-based caching for API call reduction.

Backoff retries for robust API interactions.

➤ **Key Functions:**⁸

correct_spelling(): Typo correction via NVIDIA LLaMA-3.⁹

process_query(): Agent-specific logic.

main(): Async console interface.

PDF Chatboard Extension¹

- **What:** Console-based tool for uploading and querying PDFs.²
- **Why:** Enables document-specific Q&A for research and education³
- **How:**⁴
 - Uploads PDFs via google.colab.files
 - Extracts text with PyPDF2 and cleans using regex
 - Answers queries with NVIDIA LLaMA-3.1-8B-Instruct via OpenAI SDK.⁵
- **Key Functions:**⁶
 - save_pdf_to_local(): Saves uploaded PDFs
 - extract_text_from_pdf(): Processes text
 - generate_answer(): Delivers concise responses (1-2 sentences)⁸
- **Example:** Query “What is NLP?” on a PDF yields concise answer⁹

PDF Chatboard Output¹

Your query: what is svm , give me ans from svm.pdf²
I believe you meant: 'What is SVM? Give me an answer from SVM.pdf'

Answer (Source: PDF Chatboard (RAG on SVM.pdf)):³

Here is a concise answer:⁴

SVM stands for Support Vector Machines, which is a binary classification technique that aims to maximize the margin between two classes by finding the optimal hyperplane.⁵

Your query: what is svm , give me ans from svm.pdf⁶

Your query: what is svm , give me ans from svm.pdf I believe you meant: 'What is SVM?⁷
Give me an answer from SVM.pdf'



Answer (Source: PDF Chatboard (RAG on SVM.pdf)):⁸

Here is a concise answer: SVM stands for Support Vector Machines, which is a binary classification technique that aims to maximize the margin between two classes by finding the optimal hyperplane that separates them. The SVM decision function can also be represented as a linear programming problem, allowing for the generalization of the optimization problem to multi-class classification.⁹

Multi-Modal

- **What:** Analyzes .jpg/.png images using Gemini 2.5 Flash or local fallback ²
- **Why:** Provides structured visual analysis for diverse applications ³
- **How:**⁴
 - Uploads images via google.colab.files
 - Gemini API for detailed analysis (scene, objects, colors, environment)
 - Local PIL-based fallback for API failures (metadata analysis)
- **Key Functions:**⁶
 - `VisionAnalyzer.setup_gemini()`: Initializes Gemini model ⁷
 - `gemini_vision_analysis()`: Structured image analysis
 - `advanced_local_vision()`: Fallback for offline processing

Multi-Modal Output¹



²

Your query: how many boats in image_trial.png? ³
Analyzing image_trial.png...



Answer (Source: GEMINI 2.5 VISION ANALYSIS): ⁴

There are **2** boats visible in the image. ⁵



IIITV & IIITV-ICD

Agentic RAG¹

- **What:** Retrieval-Augmented Generation with evaluation²
- **Why:** Enhances Q&A with context-aware, evaluated responses⁴

➤ How RAG Works⁵

- **Retrieval Step:**⁶

When a user queries a PDF (e.g., “Explain backpropagation in SVM.pdf”), the system:⁷

- Extracts text using PyPDF2.
- Splits it into manageable overlapping chunks using RecursiveCharacterTextSplitter.
- Encodes each chunk into numerical embeddings using NVIDIA nv-embedqa-e5-v5 embeddings.
- Stores and searches these embeddings in a FAISS vector database to find top relevant content.⁸

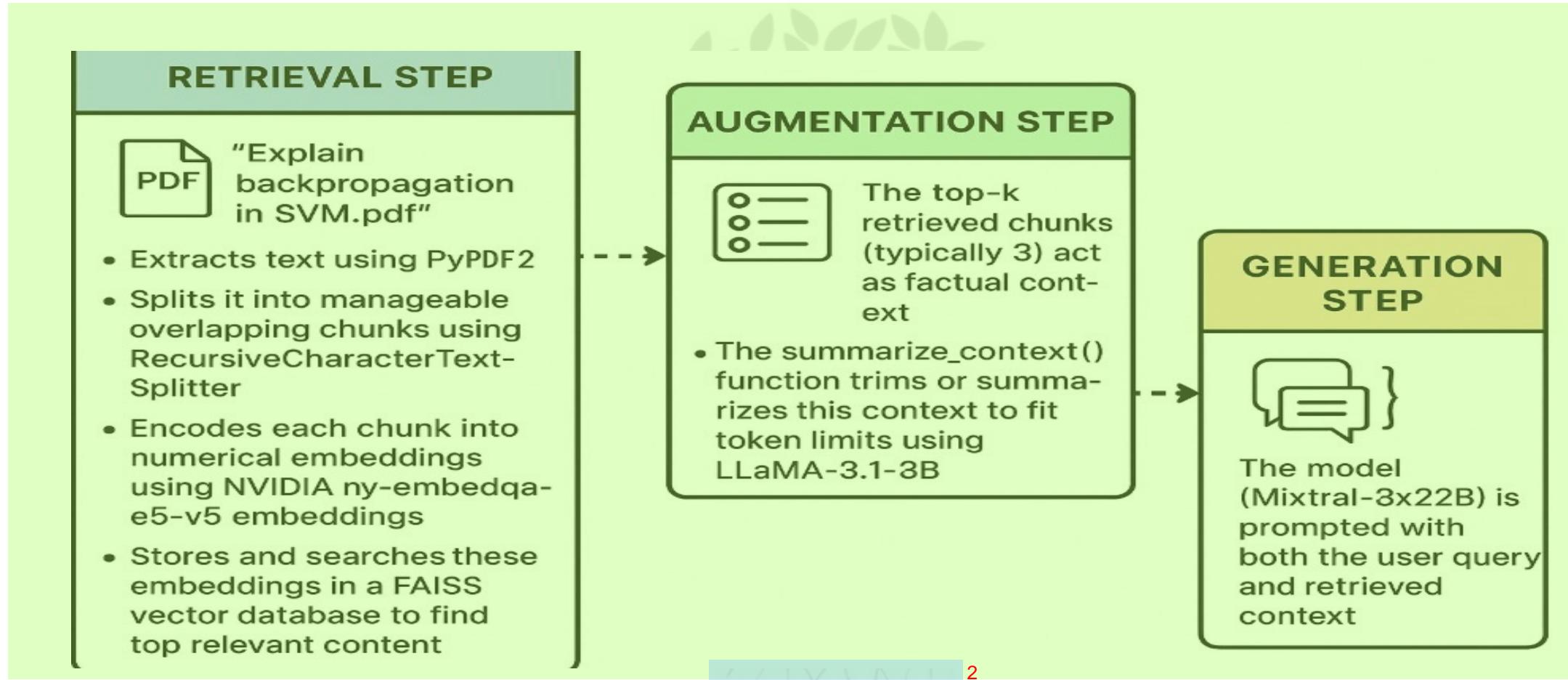
- **Augmentation Step:**⁹

- The top-k retrieved chunks (typically 3) act as factual context.
- The summarize_context() function trims or summarizes this context to fit token limits using LLaMA-3.1-8B.¹⁰

- **Generation Step:**¹¹

- The model (Mixtral-8x22B) is prompted with both the *user query* and *retrieved context*.¹²
- It produces an informed, concise answer consistent with the underlying document

Agentic RAG Work_flow



2

IIITV & IIITV-ICD



2

API	Source	Data Provided
WeatherAPI	weatherapi.com	Temperature, humidity, wind (JSON)
GitHub API	github.com	Repository metadata (stars, language)
NVIDIA API	integrate.api.nvidia.com	Query processing and response generation
Google Gemini API	genai.googleapis.com	Multi-modal image Analysis

3

Challenges and Solutions

➤ Challenges:²

- **API Rate Limits:** Frequent API calls lead to throttling³
- **Typo Handling:** Misspelled queries (e.g., "Mehasana" vs. "Mehsana")
- **Async Complexity:** Managing concurrent tasks in Colab
- **Data Scalability:** Processing large PDFs or image datasets

➤ Solutions:⁴

- **Caching:** Pickle-based response caching to reduce API calls⁵
- **Typo Correction:** NVIDIA LLaMA-3 for spelling fixes
- **Asyncio:** Parallel query processing with nest_asyncio
- **Modular Design:** FAISS and LangGraph for scalable retrieval

Results¹



- **Efficiency:** Python's asyncio saving time and resources for all queries.²
- **Speed:** Delivers fast responses using asynchronous processing, ensuring quick results for any task.³
- **Accuracy:** Achieves high reliability, with the Weather Agent being more accurate for cities worldwide.⁴
- **Smart Correction:** Using Pre_train(LLaMA-3) model's API calls to fixes typos (e.g., "temp" to "temperature") to improve results for any query.⁵
- **Evaluation Results:** {'faithfulness': 0.8333, 'answer_relevancy': 0.4270, 'context_precision': 0.6667, 'context_recall': 0.4444, 'answer_correctness': 0.7224}⁶

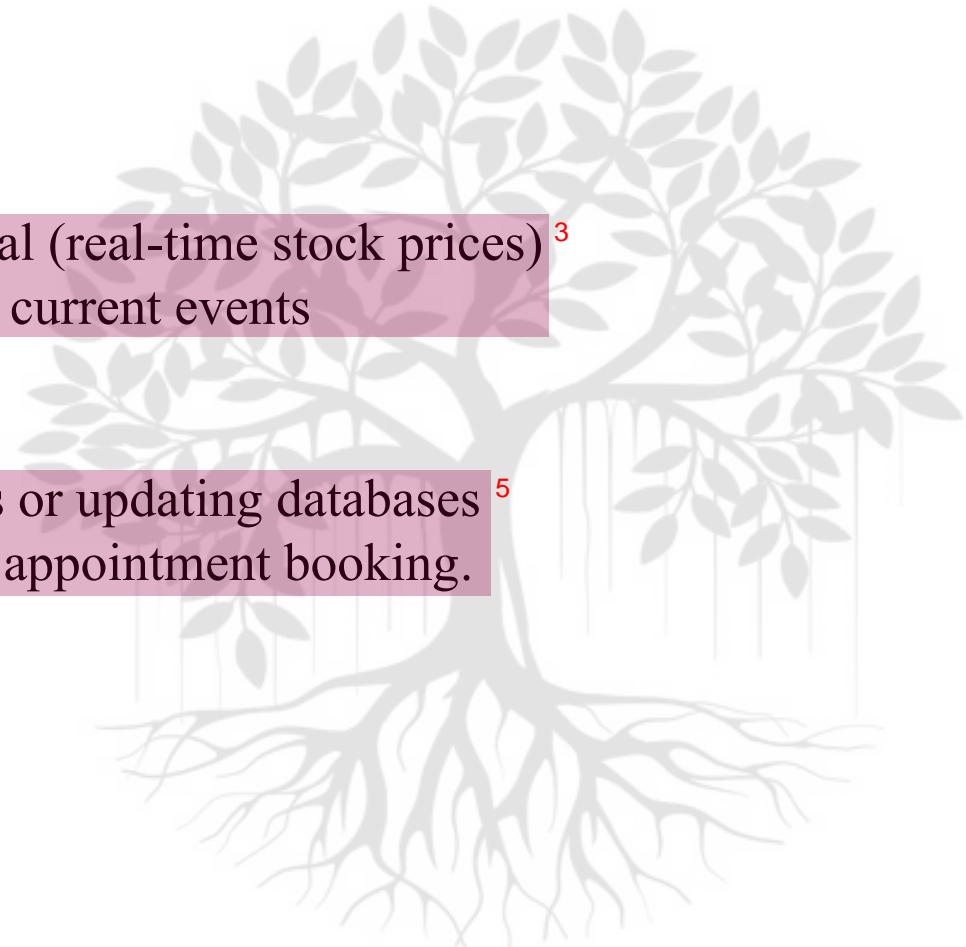
→ Multi-Agent System Ready (Efficient Edition). Enter your query (e.g., 'weather in London', 'temp of Ahmedabad', GitHub URL, or any question):
Your query: weather in patan
I believe you meant: 'weather in Patan'
Multiple matches; using Patan, Gujarat, India. Alternatives: Patan, Gujarat, Patuncan, Veracruz-Llave
The weather in Patan, Gujarat, India is Light rain shower with a temperature of 25.3°C (feels like 28.0°C). Humidity: 87%, Wind: 22.0 kph. (Updated: 2025-10-01 08:30)
Your query: temp of gandhinagar
I believe you meant: 'temperature of Gandhinagar'
The weather in Gandhinagar, Gujarat, India is Mist with a temperature of 24.4°C (feels like 26.8°C). Humidity: 83%, Wind: 20.9 kph. (Updated: 2025-10-01 08:30)
Your query: quit
Exiting. Goodbye!

Conclusion



- **Key Achievements:**²
 - Integrated WeatherAPI, GitHub API, NVIDIA API, and Gemini API ³
 - Scalable multi-agent system with typo correction and async processing
 - Extended with PDF querying, image analysis, and evaluated RAG
- **Impact:**⁴
 - Enhances real-time access for education, development, and planning ⁵
 - Demonstrates robust, modular AI system design
- **Takeaway:** Unified platform for intelligent, multi-domain query processing⁶

Future Work



➤ Additional Agents:²

- Stock market data retrieval (real-time stock prices)³
- News summary agent for current events

➤ Actionable Agents:⁴

- Tasks like sending emails or updating databases⁵
- Example: Weather-based appointment booking.

References¹

- LangChain Academy. (n.d.). *Intro to LangGraph*. <https://academy.langchain.com/courses/intro-to-langgraph> ²
- ByteByteGo. (n.d.). *GenAI System Design Interview: Introduction and Overview*. <https://bytebytogo.com/courses/genai-system-design-interview/introduction-and-overview> ³
- Karpathy, A. (n.d.). *YouTube Playlists*. <https://www.youtube.com/@AndrejKarpathy/playlists> ⁴
- YouTube Video. (n.d.). *LangChain Tutorial*. <https://youtu.be/wMVzCctmtLI> ⁵
- Fan, Y., Ma, X., Wu, R., Du, Y., Li, J., Gao, Z., & Li, Q. (2025). VideoAgent: A Memory-augmented Multimodal Agent for Video Understanding. <https://arxiv.org/abs/2410.00000> ⁶
- Llama Team. (2024, July 23). *The Llama 3 Herd of Models (Version 3)*. *arXiv preprint arXiv:2407.21783*. <https://arxiv.org/abs/2407.21783> ⁷



Thank You!²

IIITV & IIITV-ICD