Module 4: Generative AI

Deepak Subramani
Assistant Professor
Dept. of Computational and Data Science
Indian Institute of Science Bengaluru

Outline for Jan 25

- Part 1: Decoder only GPT Model
  - What are GPT-class Generative Large Language Models
  - Data preparation for GPT model training
  - GPT finetuning (Assignment)
- Part 2: LLMs and Interacting with them
  - Commercial and open source LLMs
  - What are the main issues in LLMs to be aware of?
  - Taxonomy of interaction with LLMs
  - Prompting Strategies – ZSL, FSL, CoT, ReACT, DSP
  - Parameter Efficient Fine Tuning (LoRA, QLoRA)

Outline for Feb 01

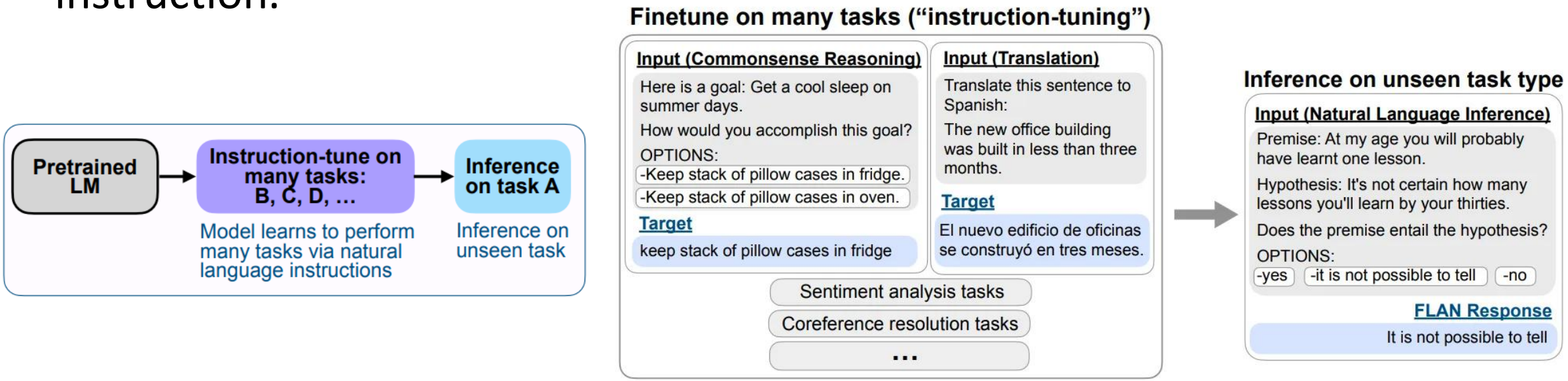
- Part 1:
  - Instruction Tuning
- Part 2:
  - Orchestration
  - Retrieval Augmented Generation
- Part 3:
  - LLM Guardrails
  - LLM Agents

List of Prompting Strategies

1. Chain of Thought Prompting (CoT)
2. Tree of Thought Prompting
3. Prompts with Instructions
4. Prompt Chaining
5. Directed Stimulus Prompting
6. ReAct

Prompts with Instructions

- Unlike open-ended prompts, prompts with instructions guide the model to perform specific tasks or follow certain rules within its response.
- Structured Responses: Instruction-based prompts lead to more structured and targeted outputs, adhering to the format or content specified in the instruction.



Wei et al., 2022, Instruction Tuning Vol. 1

Instruct Fine Tuning via RLHF

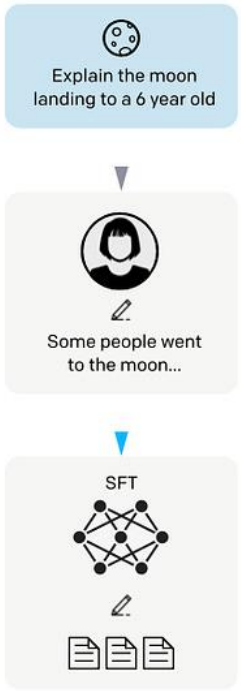
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



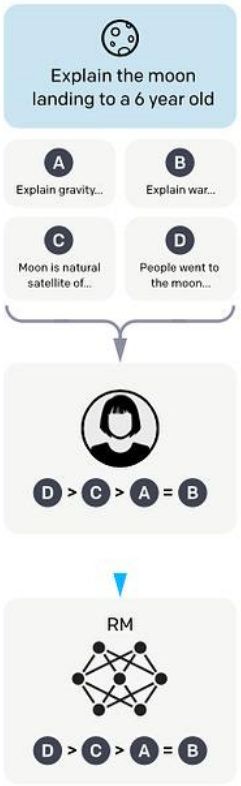
Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



Step 3

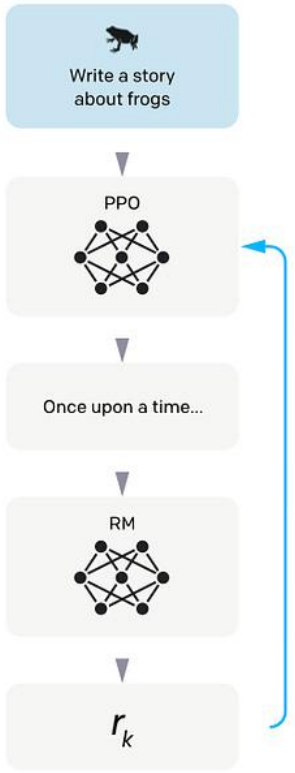
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

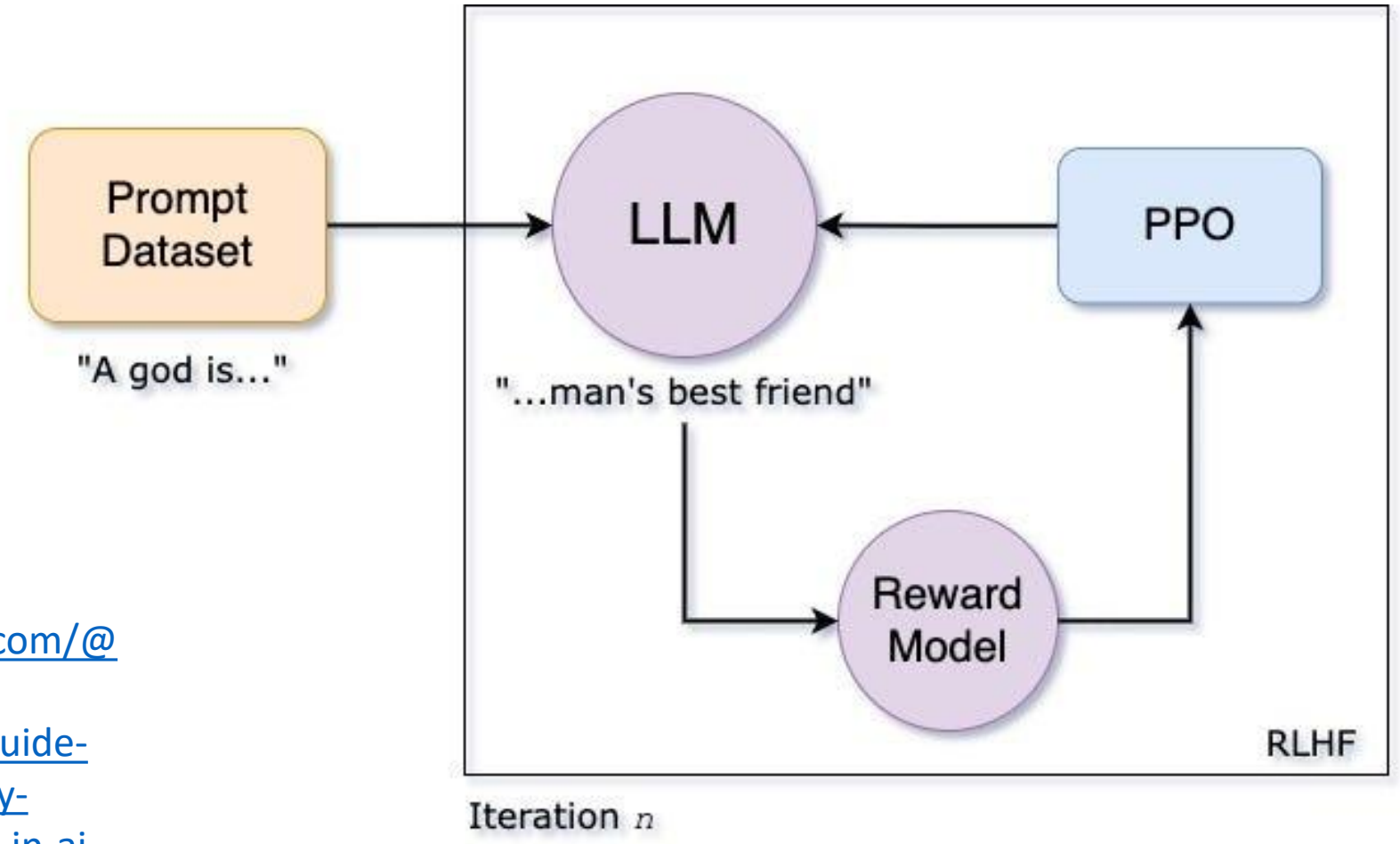
The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



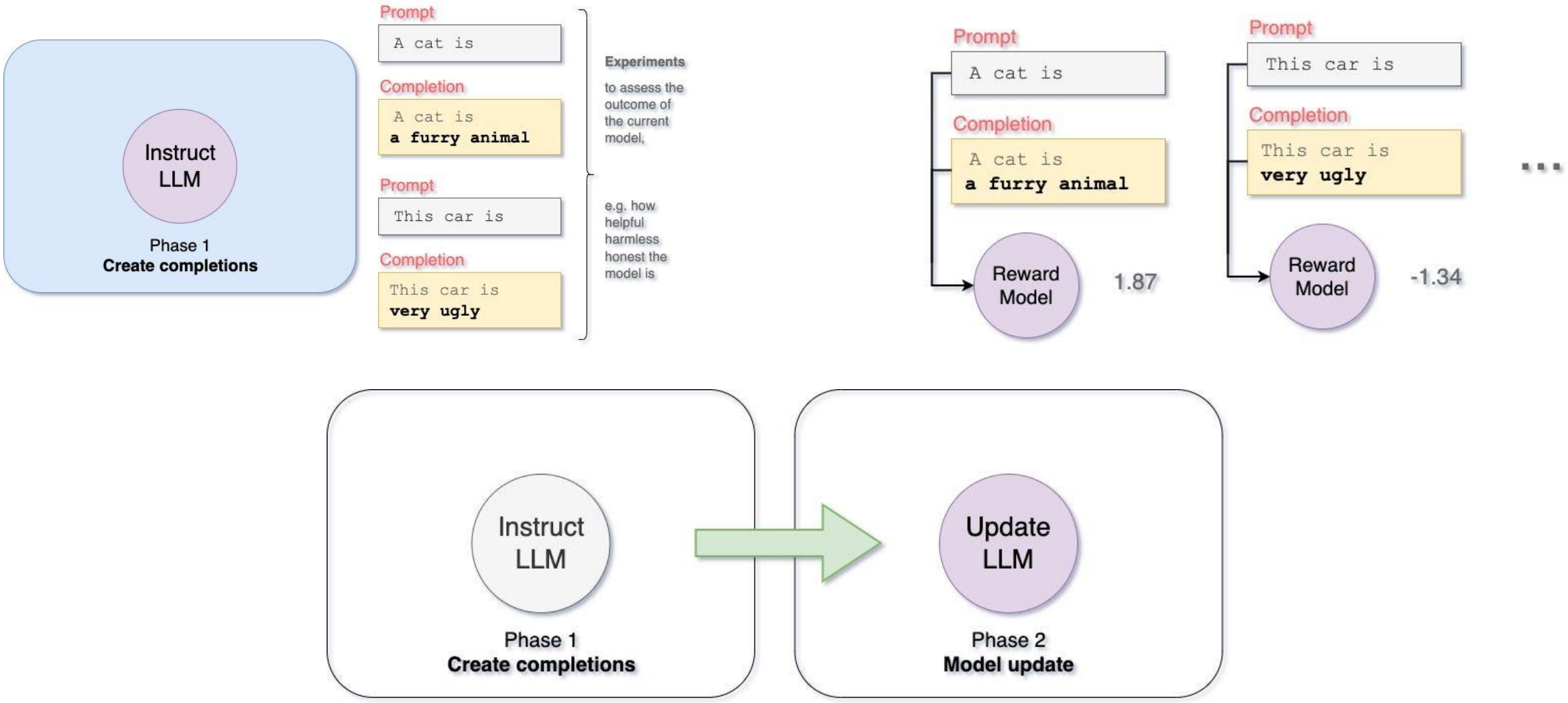
Deepak Subramani, deepakns@iisc.ac.in

RLHF via Proximal Policy Optimization (PPO)



Source: https://medium.com/@oleglatypov/a-comprehensive-guide-to-proximal-policy-optimization-ppo-in-ai-82edab5db200

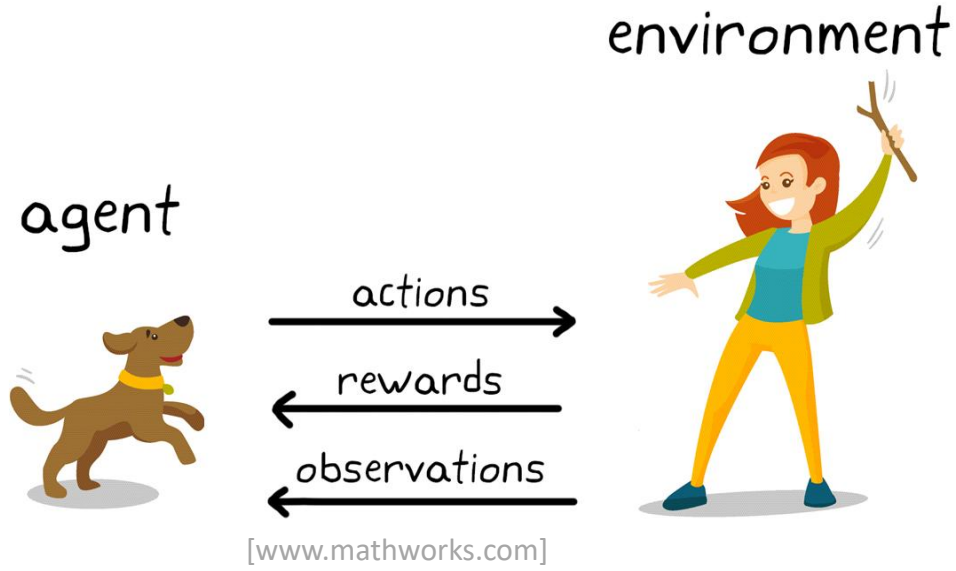
PPO Steps


Instruct LLM
Phase 1
Create completions
Prompt
A cat is
Completion
A cat is a furry animal
Prompt
This car is
Completion
This car is very ugly
Experiments
to assess the outcome of the current model,
e.g. how helpful harmless honest the model is
Prompt
A cat is
Completion
A cat is a furry animal
Reward Model
1.87
Prompt
This car is
Completion
This car is very ugly
Reward Model
-1.34
...
Instruct LLM
Phase 1
Create completions
Update LLM
Phase 2
Model update

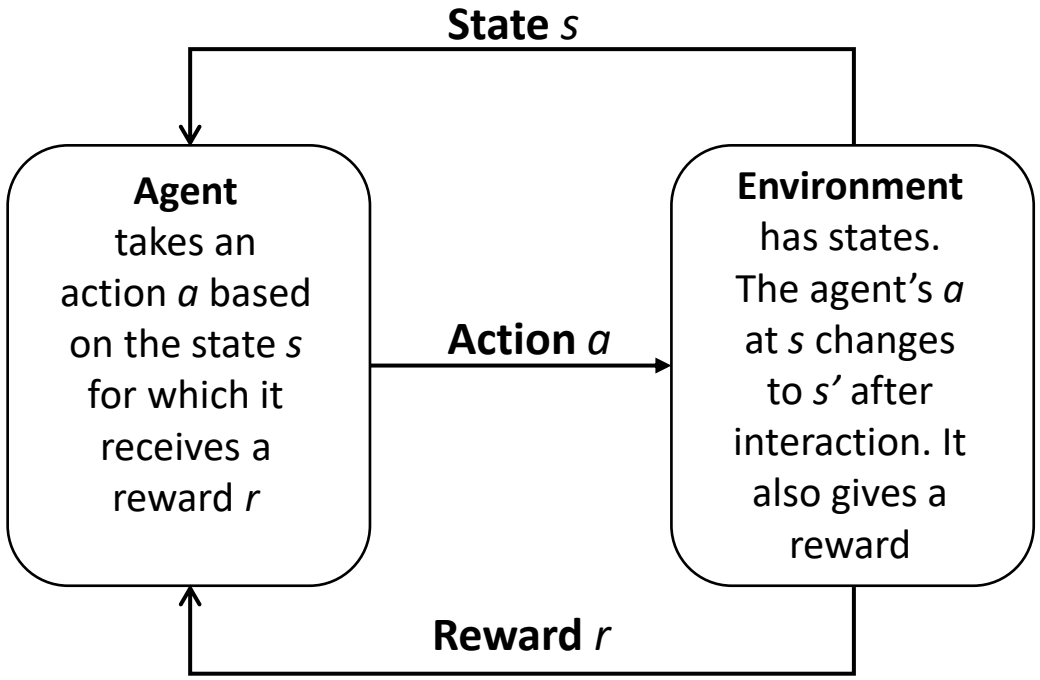
The RL Setting

- 5 Key Concepts

  Agent Environment State (Observation) Action Reward



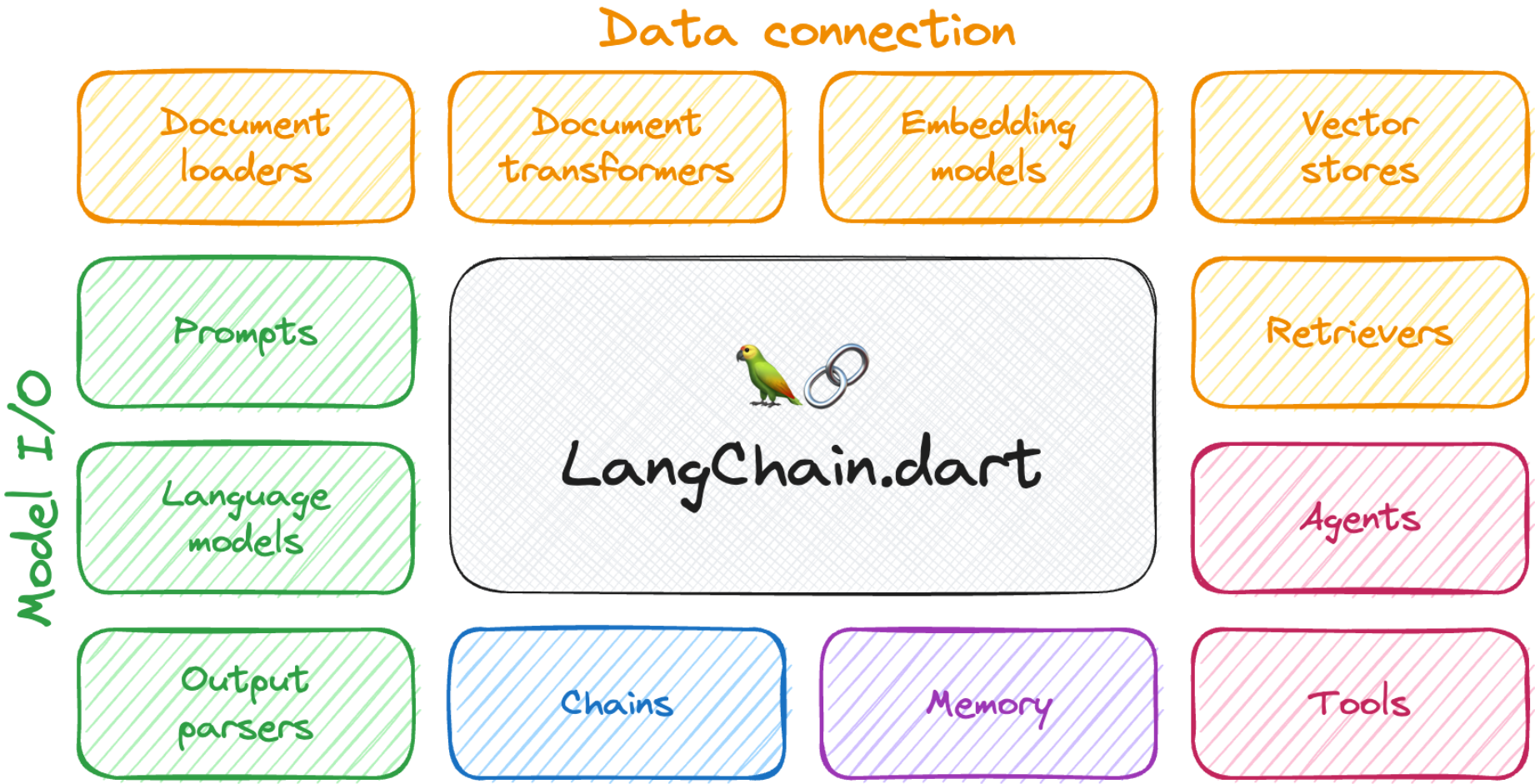
[www.mathworks.com]



Outline for Feb 01

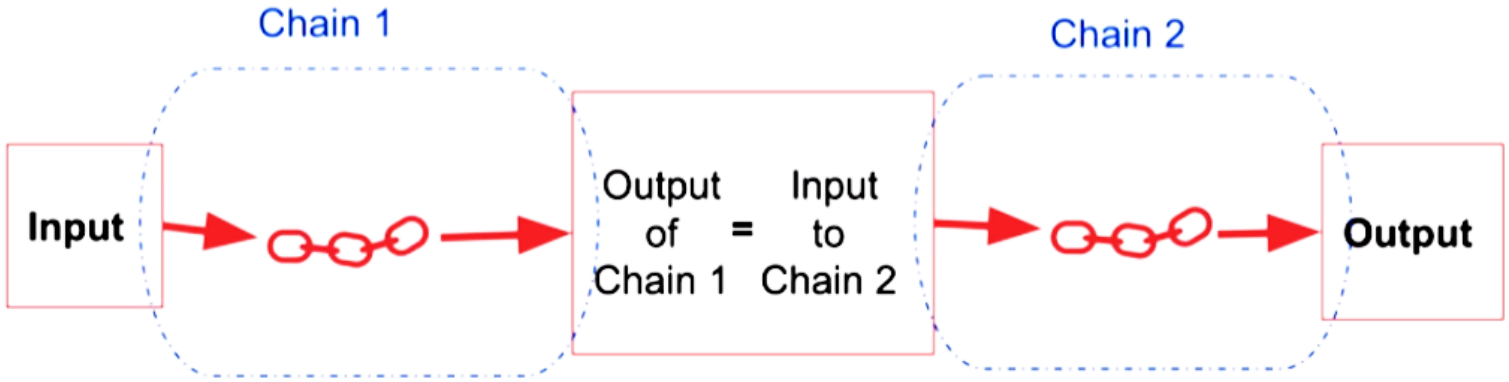
- Part 1:
  - Instruction Tuning
- Part 2:
  - Orchestration
  - Retrieval Augmented Generation
- Part 3:
  - LLM Guardrails
  - LLM Agents

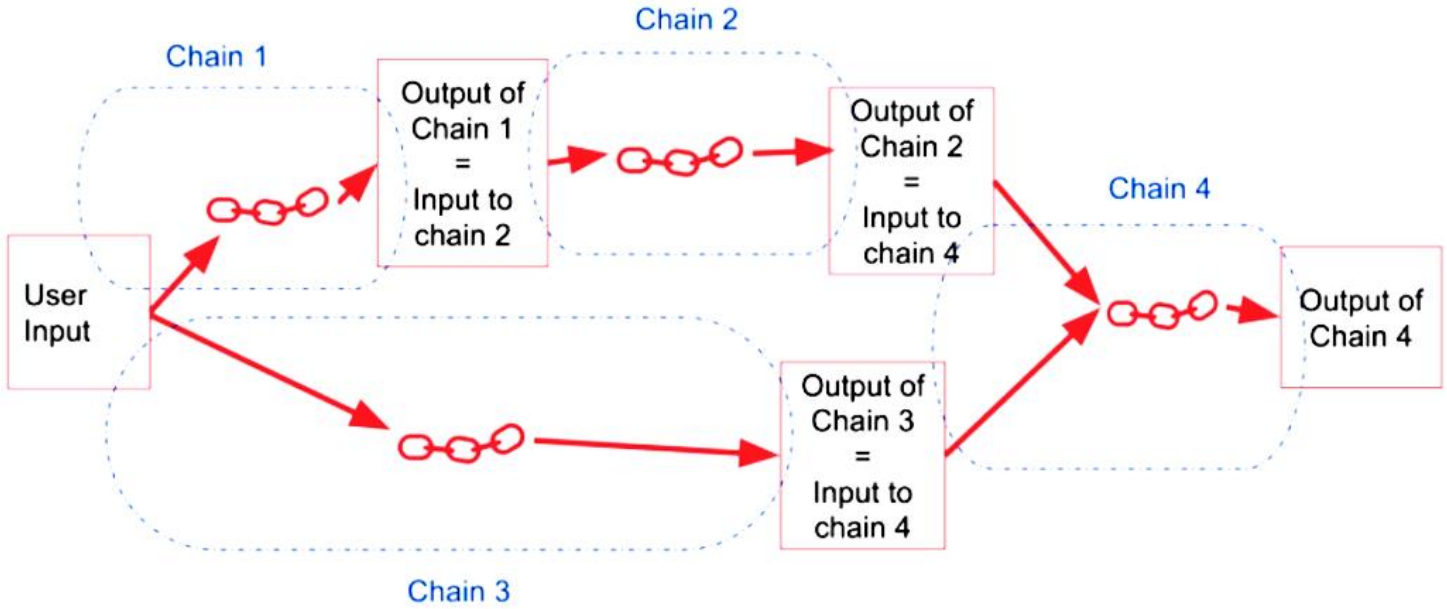
LangChain



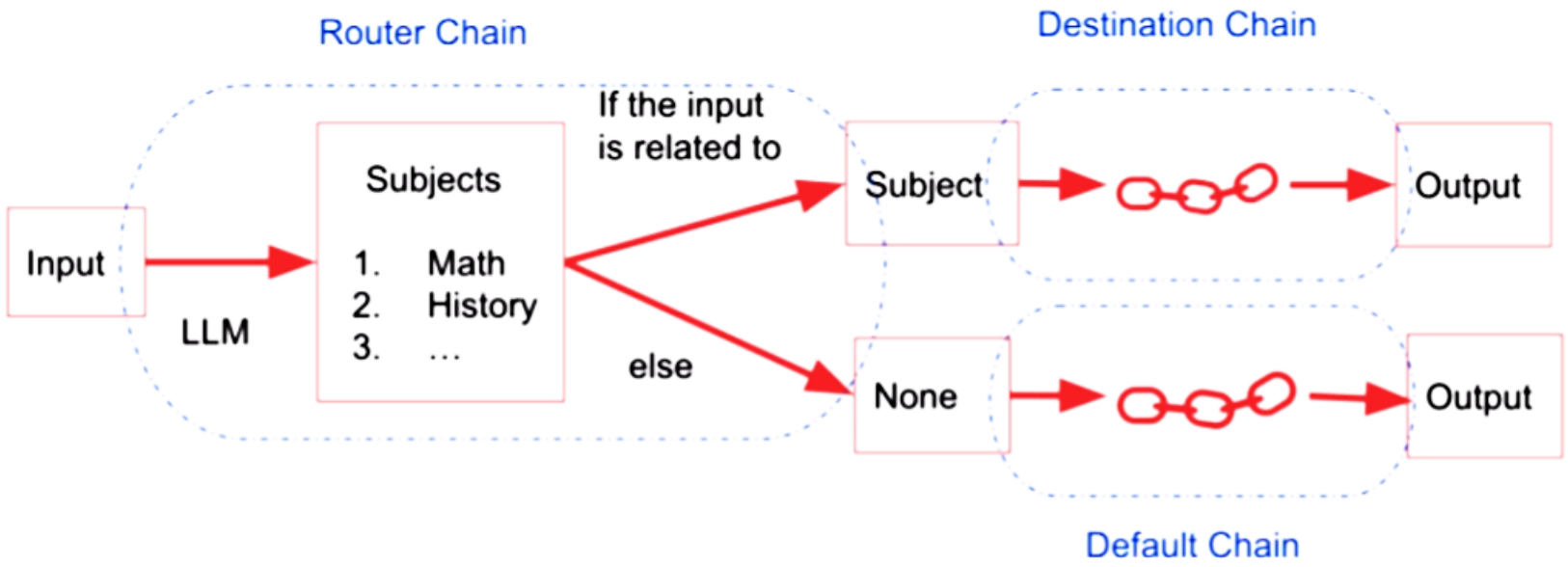
Deepak Subramani, deepakns@iisc.ac.in

Orchestrating a Solution



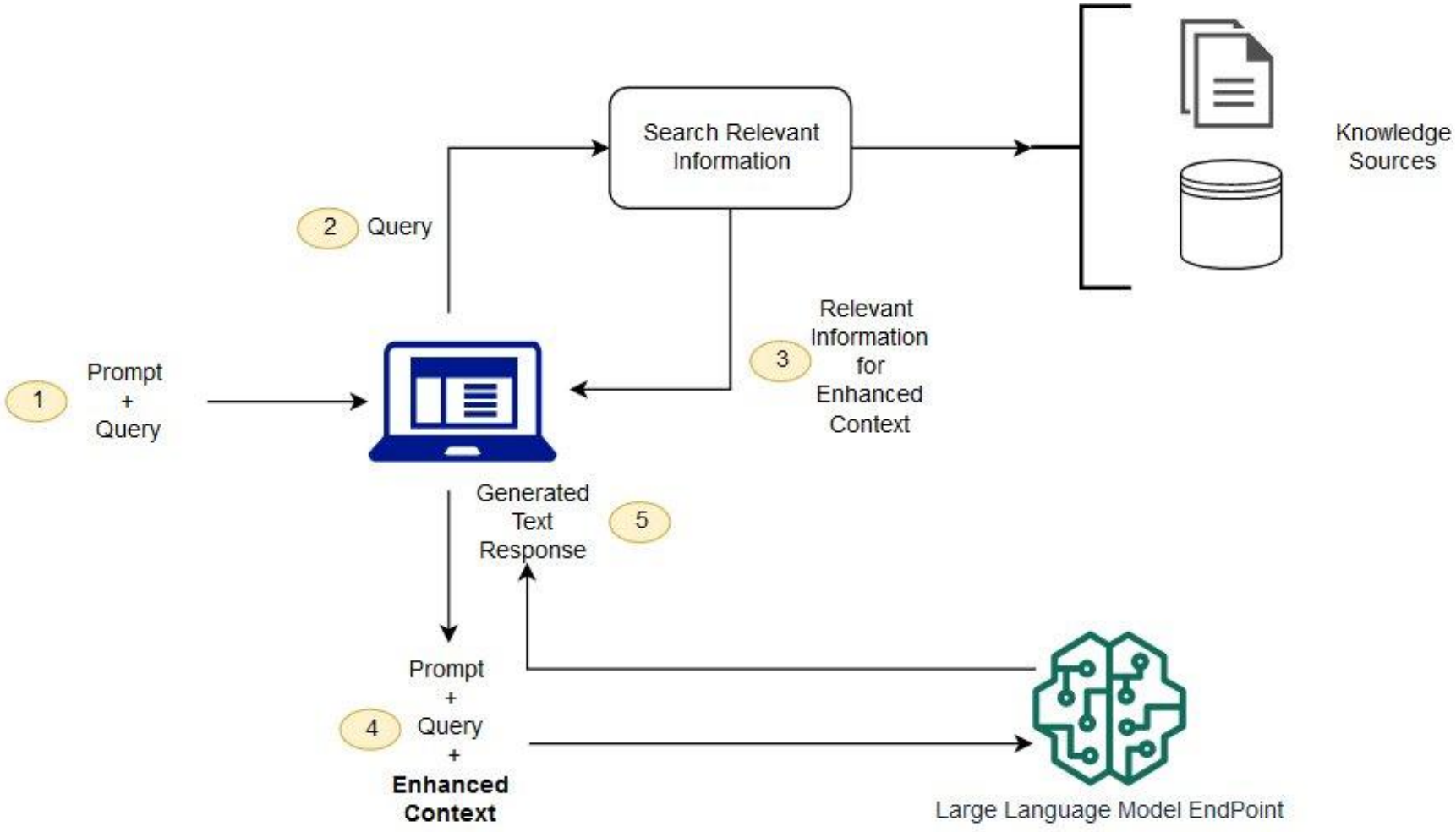


Router Chain



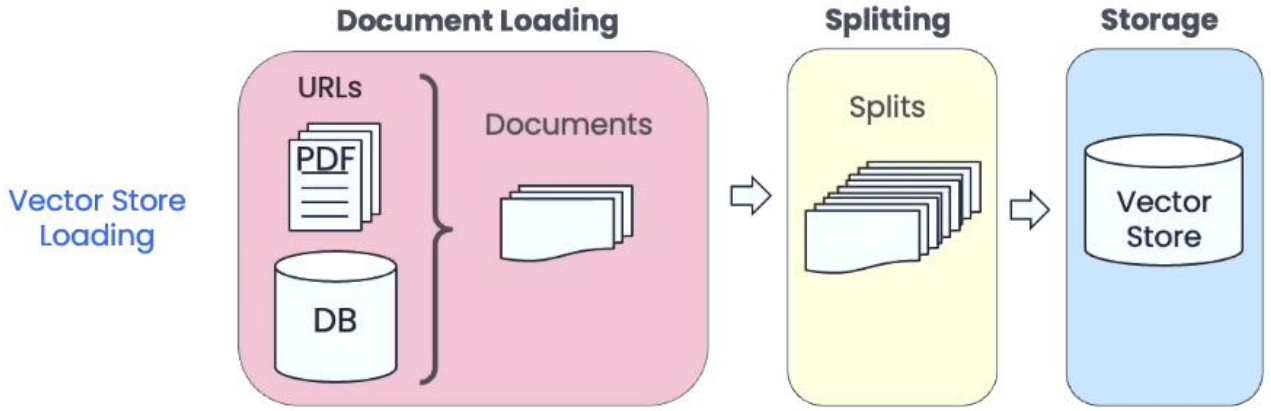
Deepak Subramani, deepakns@iisc.ac.in

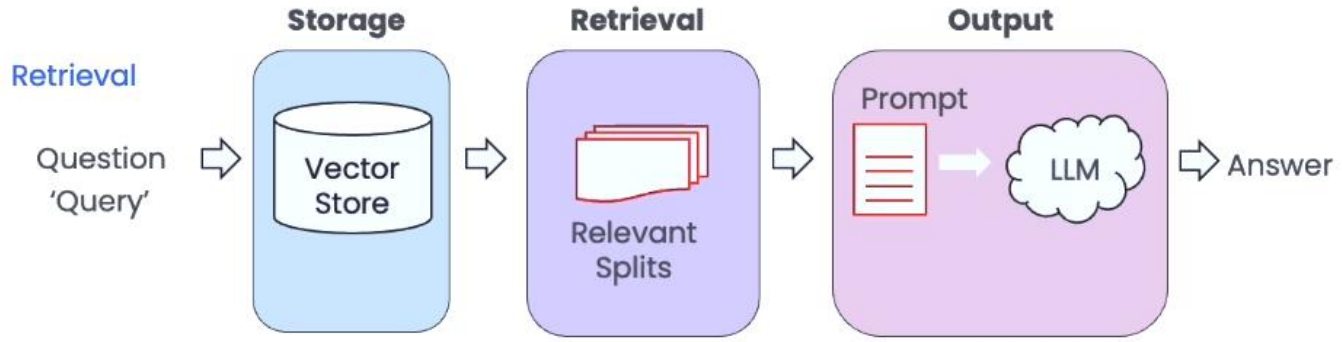
Retrieval Augmented Generation



Deepak Subramani, deepakns@iisc.ac.in

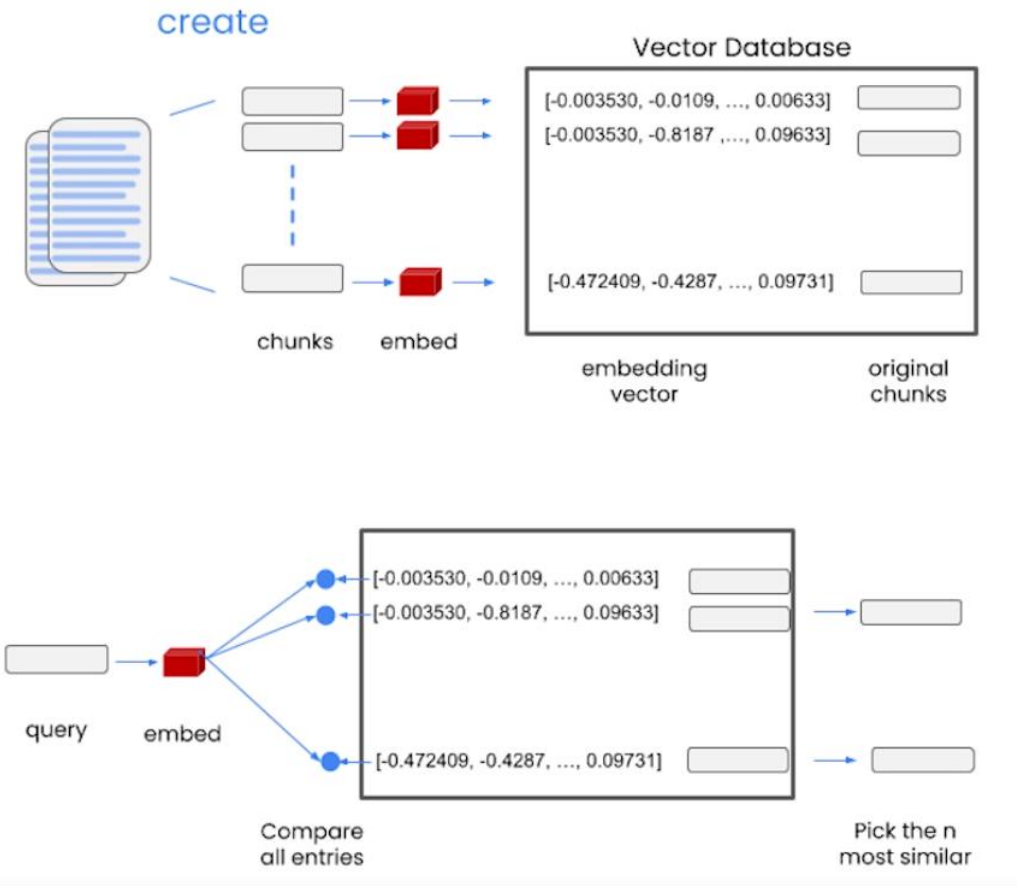
Vector Store and Retrieval





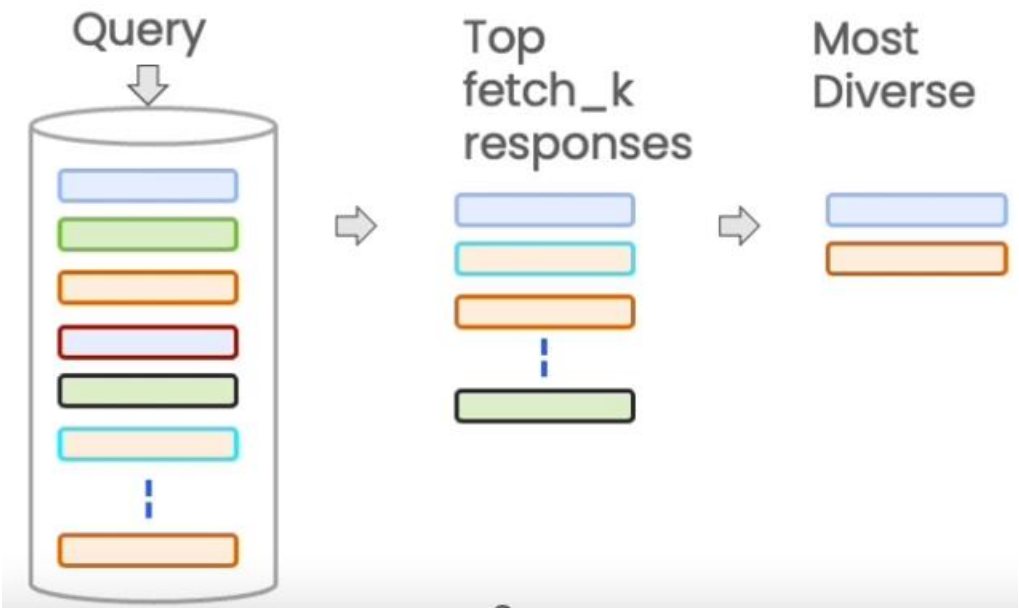
Deepak Subramani, deepakns@iisc.ac.in

Retrieval in Action



- Data is chopped into chunks and embedded in the vector database
- When a query comes in, the query embedding is compared to all entities and top similar, yet diverse matches are selected

Maximum Marginal Relevance(MMR)



Deepak Subramani, deepakns@iisc.ac.in

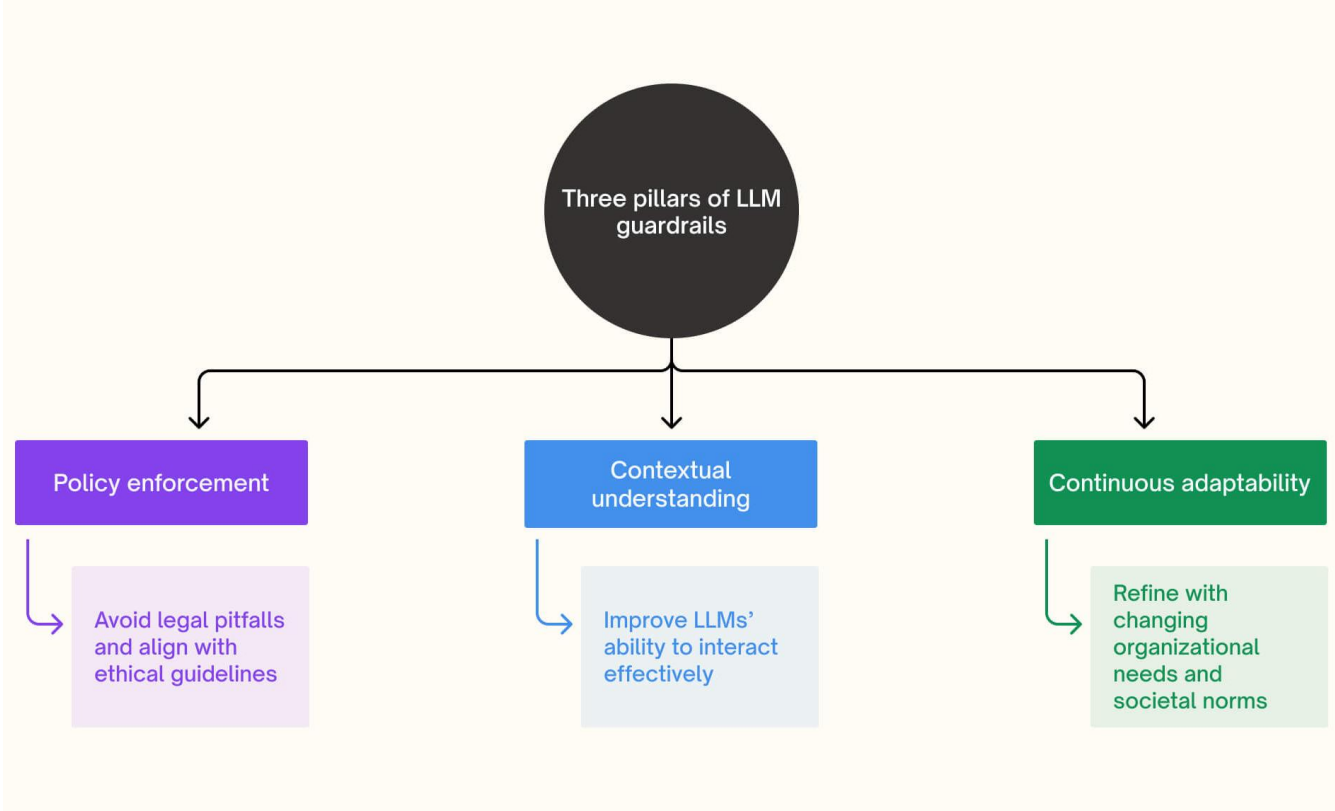
Vector Store Vendors

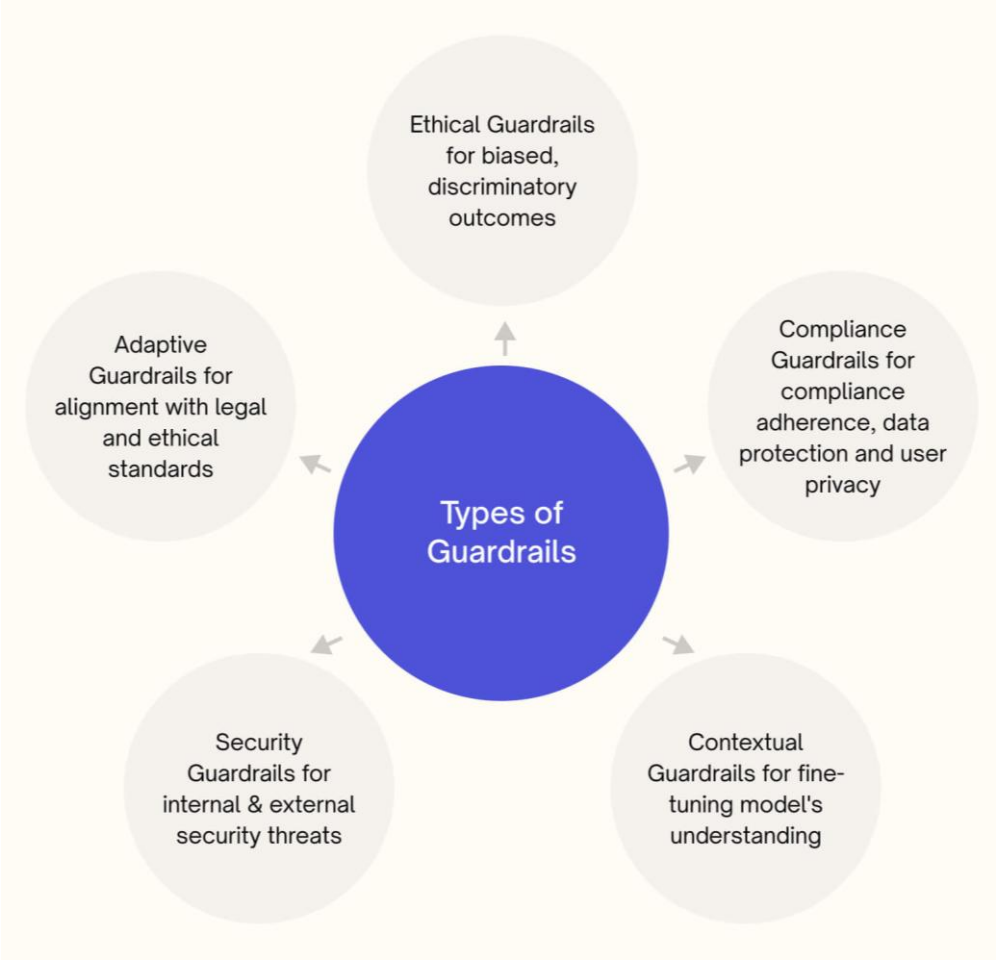
	Released	Billion-scale vector support	Approximate Nearest Neighbor Algorithm	LangChain Integration
Open-Sourced				
Chroma	2022	No	HNSW	Yes
Milvus	2019	Yes	FAISS, ANNOY, HNSW	
Qdrant	2020	No	HNSW	
Redis	2022	No	HNSW	
Weaviate	2016	No	HNSW	
Vespa	2016	Yes	Modified HNSW	
Not Open-Sourced				
Pinecone	2021	Yes	Proprietary	Yes

Outline for Feb 01

- Part 1:
  - Instruction Tuning
- Part 2:
  - Orchestration
  - Retrieval Augmented Generation
- Part 3:
  - LLM Guardrails
  - LLM Agents

Why LLM Guardrails are Needed?

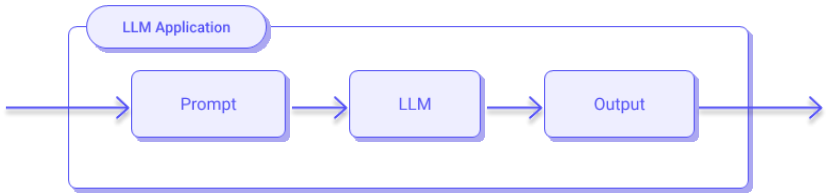




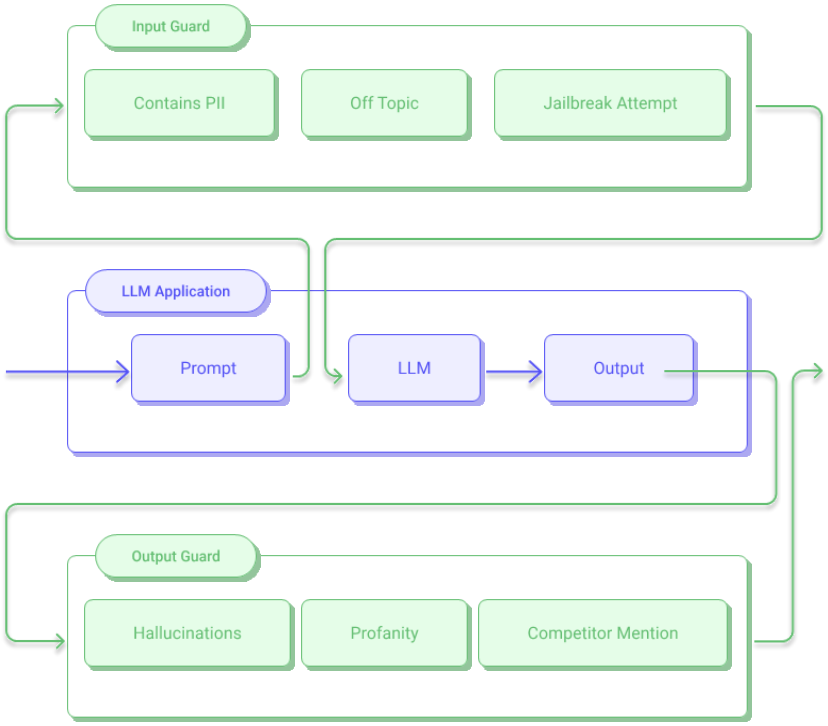
Deepak Subramani, deepakns@iisc.ac.in

LLM Guardrails

Without Guardrails



With Guardrails



Deepak Subramani, deepakns@iisc.ac.in

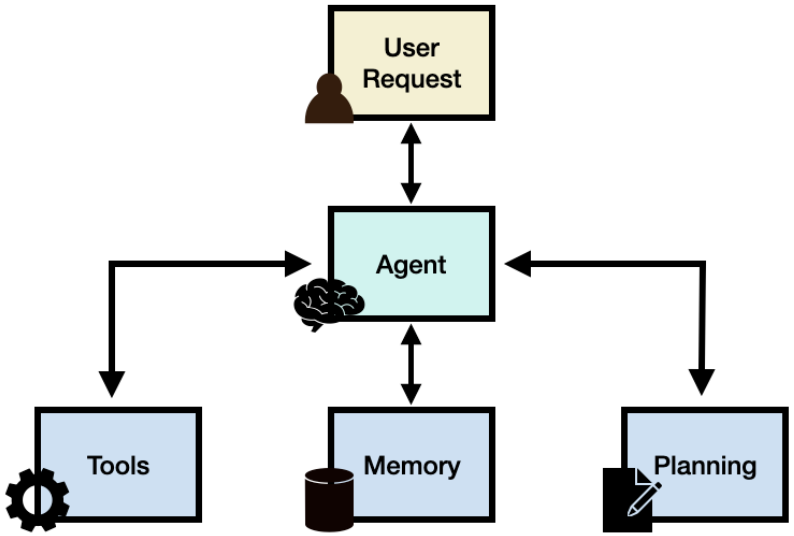
How to Implement Guardrails in your Orchestration?

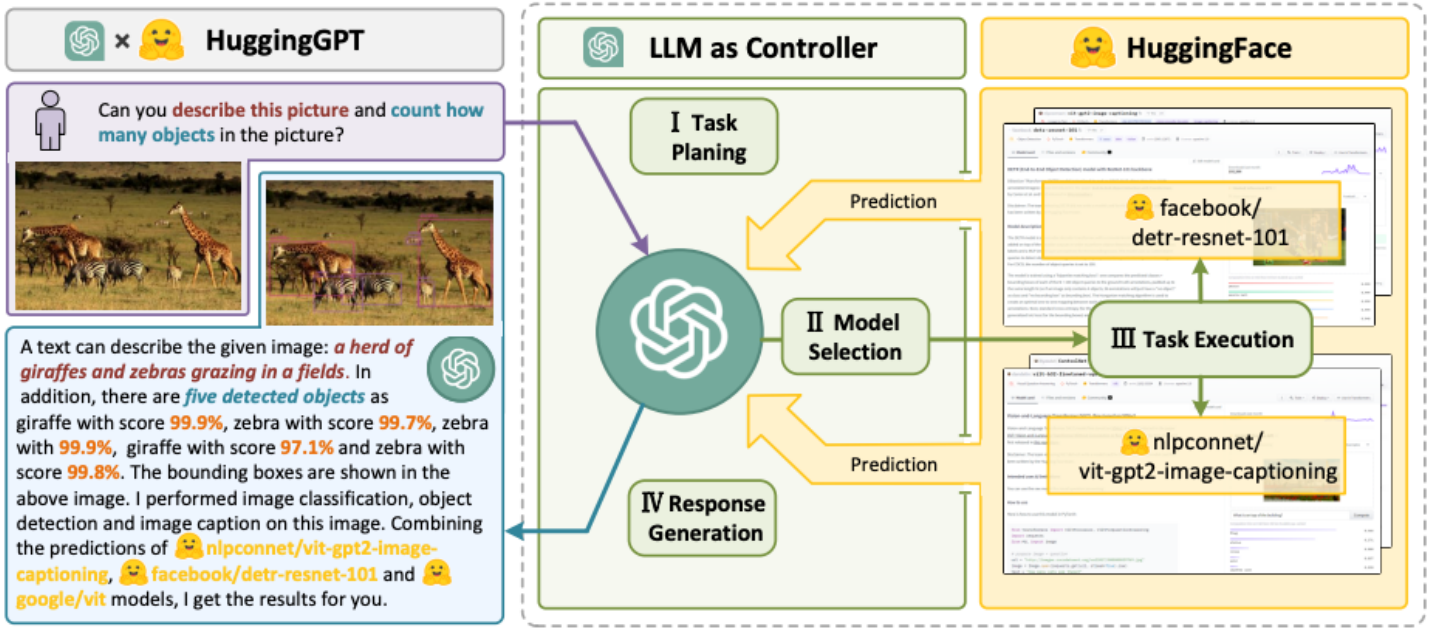
- Use Guardrails AI
  - https://hub.guardrailsai.com/
  - https://github.com/guardrails-ai/guardrails?tab=readme-ov-file
- OpenAI's Moderation AI
  - https://platform.openai.com/docs/guides/moderation
- NVIDIA's NeMo Guardrails
  - https://github.com/NVIDIA/NeMo-Guardrails

LLM Agents

- LLM Agents are advanced AI systems that use LLMs to understand and generate human language.
- LLM Agents go beyond simple text generation.
  - They can maintain the thread of a conversation,
  - recall previous statements, and
  - adjust their responses accordingly with different tones and styles.
- Diverse Use Cases: customer service, copywriting, data analysis, education, healthcare
- Caution: do not understand nuanced human emotions, and are subject to the risk of misinformation, bias, privacy data leaks and toxicity

LLM Agent Framework





Deepak Subramani, deepakns@iisc.ac.in

Mixture of Experts Model

- **Gating network** : decides what expert to use

  $g_1, g_2, \ldots g_k$ - gating functions