# An Introduction to Multiagent Artificial Intelligence Systems in Radiology

RSNA Deep Learning Labs, 2025

**Presented by:**

Pouria Rouzrokh, MD, MPH, MHPE

Moein Shariatnia, MD

Melina Hosseiny, MD

# Disclosures

Melina Hosseiny receives grant support from the NIH T32 Institutional National Research Service Award; Grant Number: T32EB005970.

Melina Hosseiny receives grant support from the  RSNA Research Resident Grant Award; Grant Number: 7022577-Radiological Society of North America.

Pouria Rouzrokh and Moein Shariatnia have no relevant disclosures.

# Part 1. Generative AI

1. Quick Review of AI / ML / DL
2. A Recap on Discriminative AI
3. Demystifying Generative AI
4. Applications of Generative AI
5. Gen-AI Family of Models

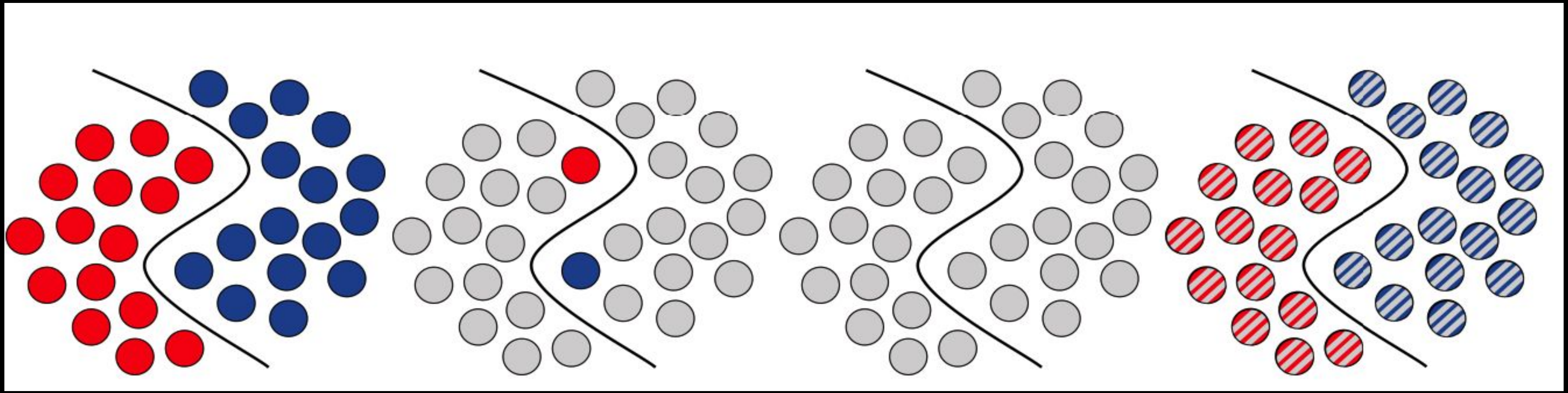
# How AI, ML, and Deep Learning Fit Together

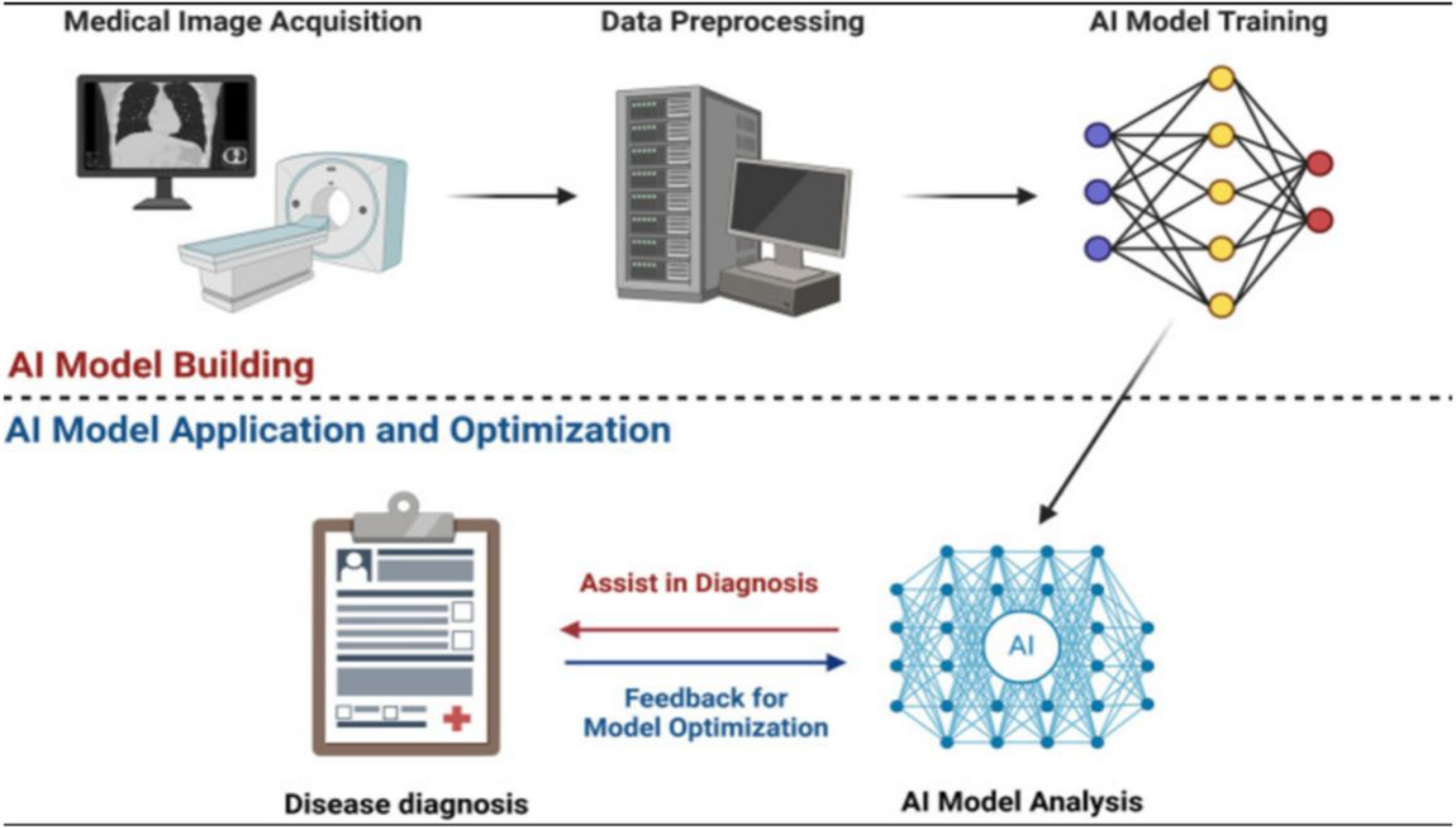
- **Artificial Intelligence:** Computational systems that approximate cognitive functions through learned or programmed decision rules.

- **Machine learning:** Data-driven function approximation where models learn parameters that minimize a loss over observed examples.

- **Deep learning:** Multi-layer neural networks that learn hierarchical, nonlinear representations directly from raw data through end-to-end training.

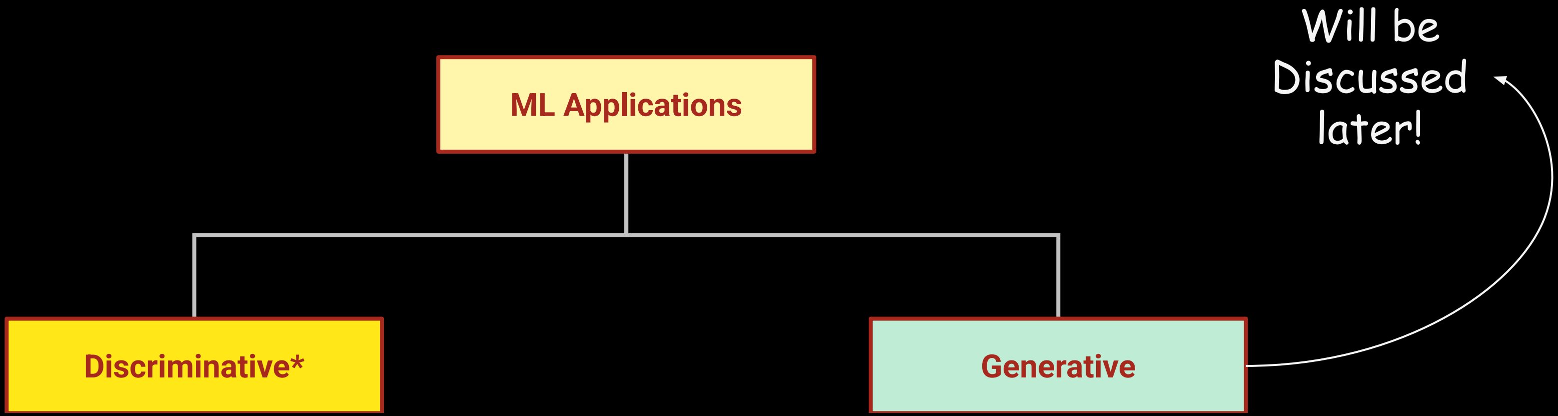
# ML classification by "Learning Schema"





Medical Image Acquisition
Data Preprocessing
AI Model Training
AI Model Building
AI Model Application and Optimization
Assist in Diagnosis
AI
Feedback for
Model Optimization
Disease diagnosis
AI Model Analysis

# ML classification by Application:



# ML applications: Classification

Machine learning classification models assign an **input** to one of several predefined categories.

**Examples of classification tasks in medicine**

- Disease vs no disease (e.g., pneumonia on CXR)
- Multi-class diagnosis (e.g., stroke subtype)
- Prognostic classification (e.g., risk categories)
- Treatment response prediction

Interpretable deep learning framework for COVID-19 screening on chest X-rays

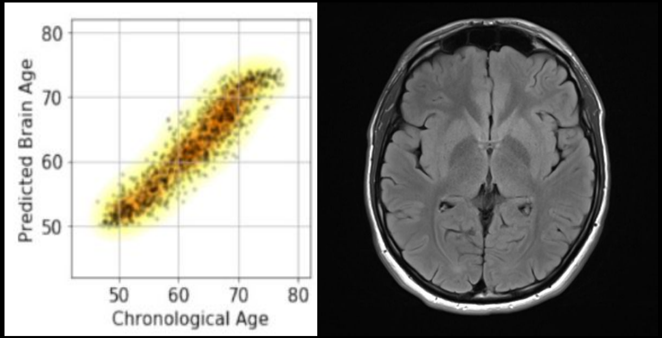
# ML applications: Regression

**Regression models predict a continuous output variable rather than a discrete class.**

**The model learns a function** $f(x) \rightarrow y$ that maps input features to a numeric target by minimizing a loss function over training examples.

**Examples in medical imaging include predicting physiologic or quantitative variables** such as patient age, tumor volume, ADC values, ejection fraction, or risk scores.

Typical evaluation metrics:  MSE (Mean Squared Error), MAE (Mean Absolute Error), and RMSE.

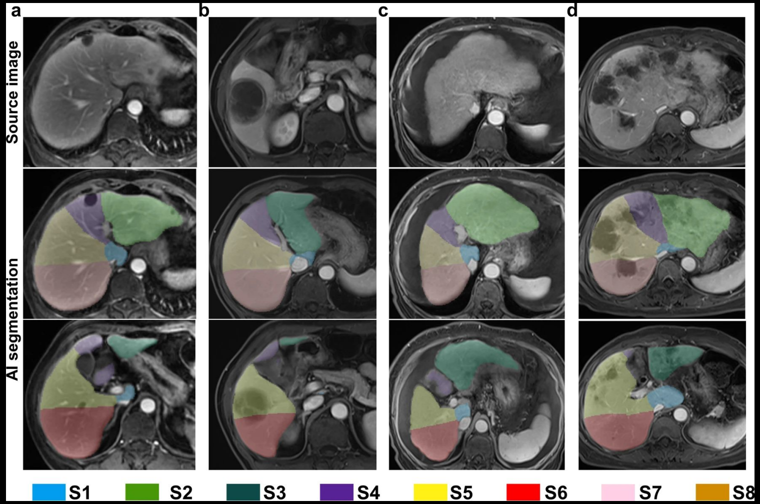



Accurate brain-age estimation using lightweight deep neural network regression models.

# ML applications: Segmentation

- **Semantic segmentation assigns a class label to every pixel** in the image, producing a pixel-wise map of anatomical structures or pathologies.

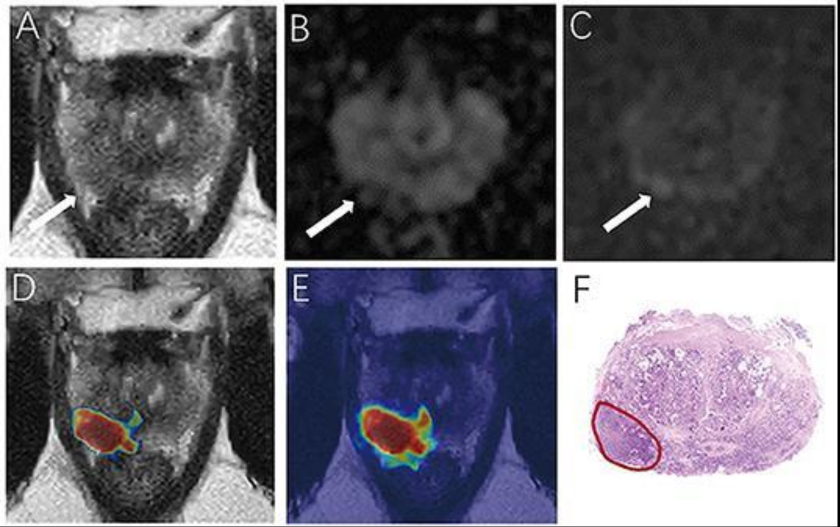
- **All pixels belonging to the same class are grouped**, meaning semantic segmentation does not distinguish between separate instances of the same object.




Examples of liver segmentation across cysts, abscesses, cirrhosis, and metastatic disease, shown with original images and ML-generated masks.

# ML applications: Detection

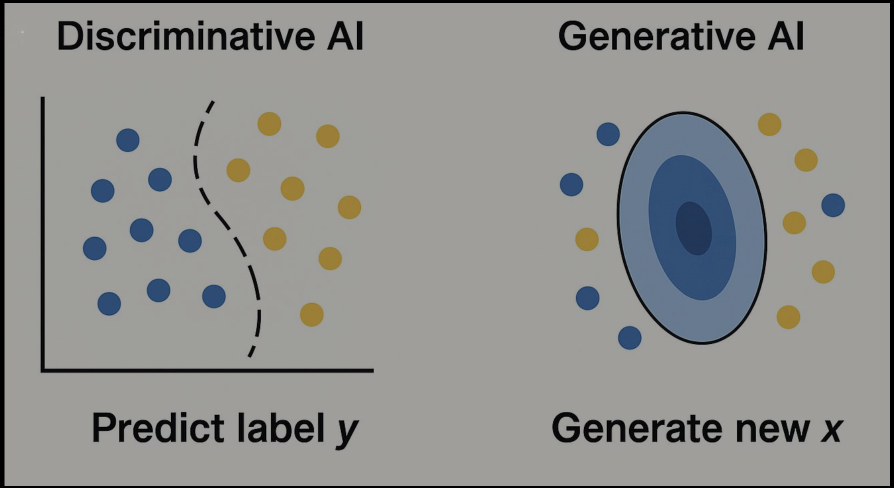
- **Object detection identifies and localizes specific objects within an image.**
- **Models output bounding boxes and class labels** for each detected object.
- **Detection networks learn both *what* the object is and *where* it is**, unlike classification which only predicts a label.




Deep learning highlights a prostate cancer focus missed by radiologists, showing overlap between predicted and true cancer regions.

# Discriminative AI vs. Generative AI

- **Discriminative models learn the conditional distribution** to classify or analyze input data by estimating decision boundaries between classes.
- **Generative models learn the data distribution** or the conditional distribution, enabling them to **sample new data** consistent with the learned distribution.
- **Discriminative models predict labels**, like “normal” or “pneumonia.”
- **Generative models create new data**, like synthetic images, text, or signals that resemble real examples.



# Main Applications of Generative AI

## General Applications

- Text generation (chatbots, summarization, translation)
- Image generation (design, art, synthetic data)
- Video and audio synthesis
- Code generation and software automation
- Data augmentation for training ML models
- Virtual assistants and multimodal reasoning systems

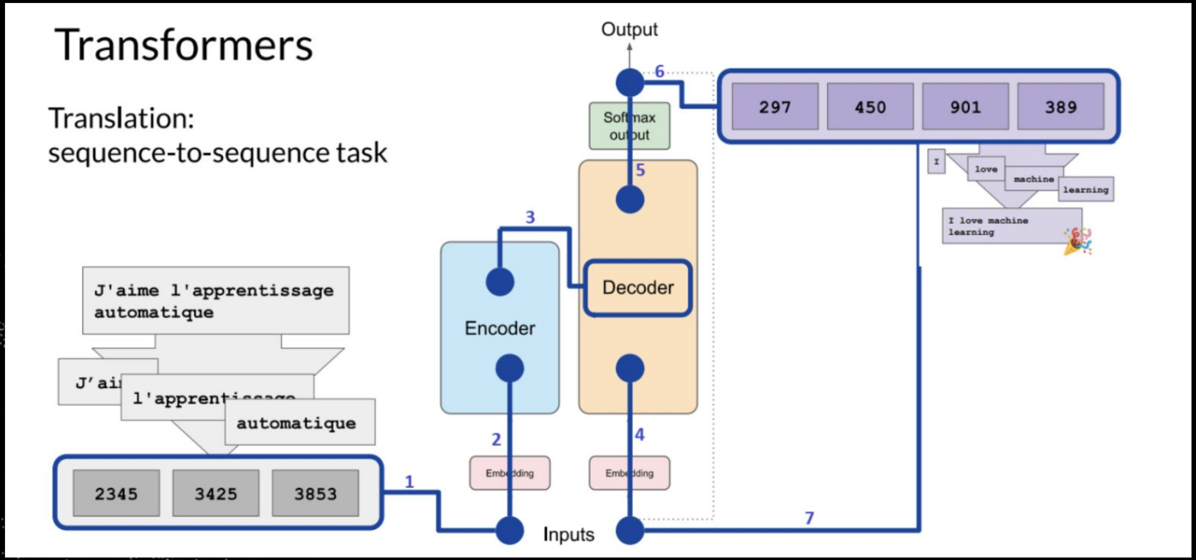
## Medical Applications

- Image reconstruction
- Image synthesis
- Cross-modality translation (MRI→CT, PET→CT)
- Super-resolution
- Segmentation assistance
- Report generation
- Summarization (EMR → clinical indication)
- Disease simulation for education and rare-pathology modeling
- Data harmonization (removing scanner or protocol differences)

# Key Architectural Innovations in Modern Generative AI

## Transformers

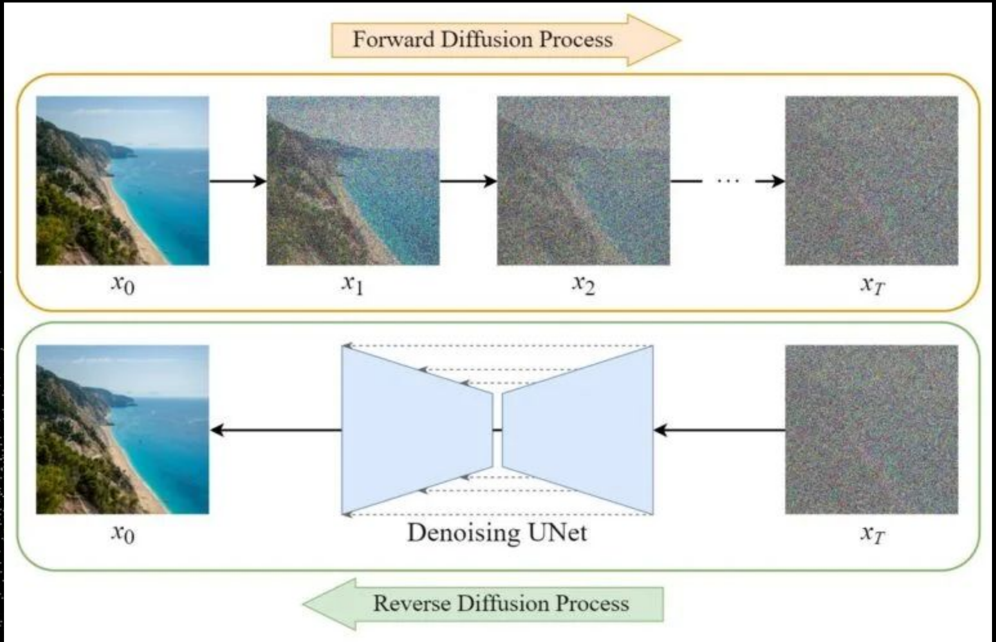
- Self-attention for long-range relationships
- Scales with data and compute → foundation of LLMs
- Enables multimodal alignment (text ↔ image ↔ audio)



# Key Architectural Innovations in Modern Generative AI

## Diffusion Models

- Generate images by iterative denoising
- Extremely stable training compared to GANs
- Produce high-fidelity medical images with realistic noise/texture



# Key Architectural Innovations in Modern Generative AI

### 3. Cross-Modal Attention

- Links image and text representations
- Core to text-to-image models (e.g., Stable Diffusion, Imagen)
- Enables radiology applications like image→report or report→image

### 4. Latent Space Models (VAEs / Latent Diffusion)

- Compress data into a structured latent space
- Enables efficient sampling and editing
- Used in stable diffusion’s latent architecture

### 5. Foundation Model Scaling

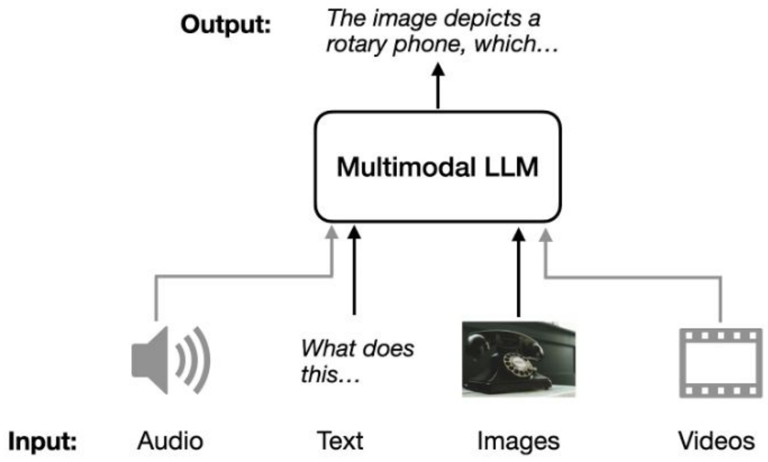
- Massive datasets + large model capacity
- Allows few-shot / zero-shot capabilities
- Makes generative AI useful across many imaging tasks

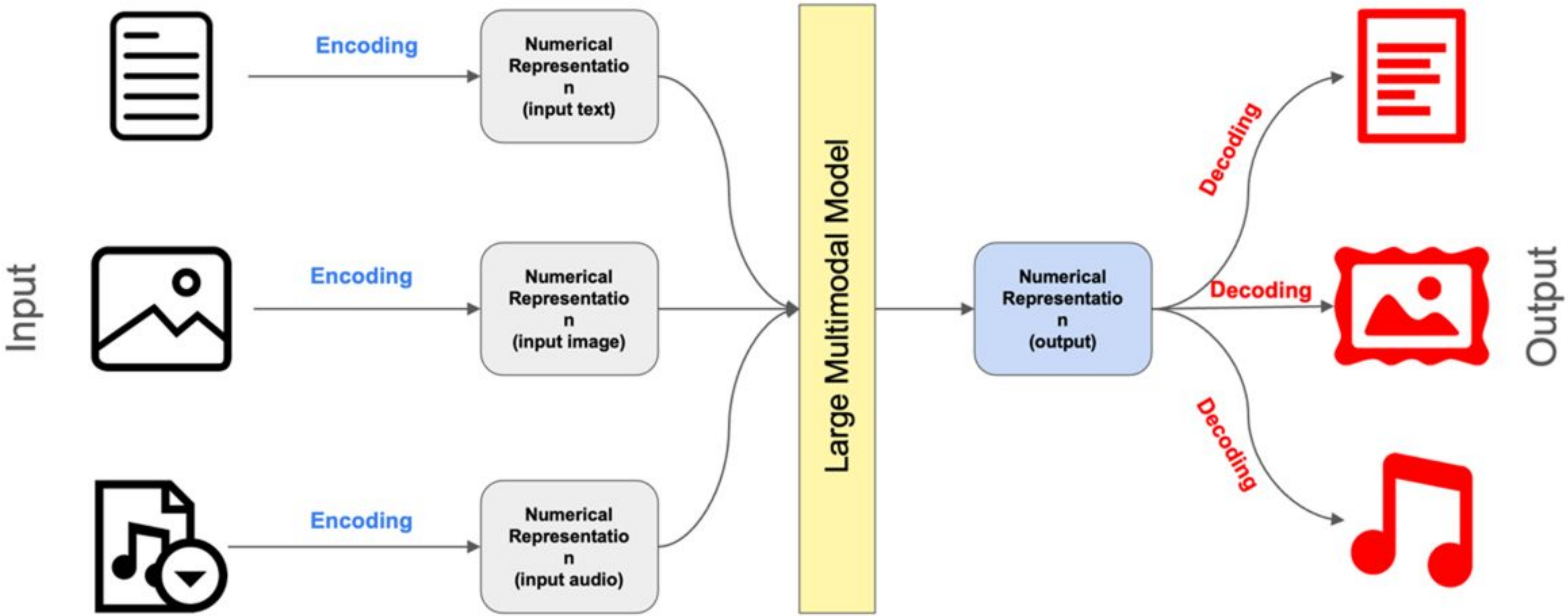
# Large Language Models (LLMs)

• **Transformer-based models that generate text** by predicting the next token, trained on massive text corpora.

• **Capture long-range relationships with self-attention**, enabling reasoning, summarization, translation, and instruction following.

• **Can be multimodal**, supporting text ↔ image tasks such as report generation, image interpretation, and vision-language prompting.




Input
Encoding
Numerical Representation (input text)
Encoding
Numerical Representation (input image)
Encoding
Numerical Representation (input audio)
Large Multimodal Model
Numerical Representation (output)
Decoding
Decoding
Decoding
Output

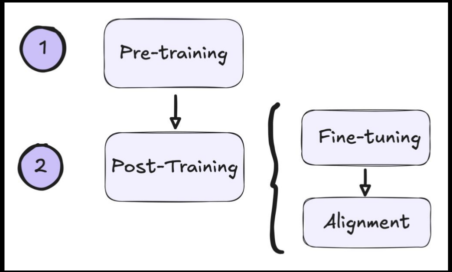
# Part 2. Agentic AI

1. Quick Review LLMs
2. On LLMs Limitations
3. Augmented LLMs and Limitations
4. AI Agents
5. Multi-agent systems: Design patterns and examples

# LLMs

- **What are LLMs, very briefly?**
  - **A cognitive core**
  - **+ factual knowledge** learned from Internet

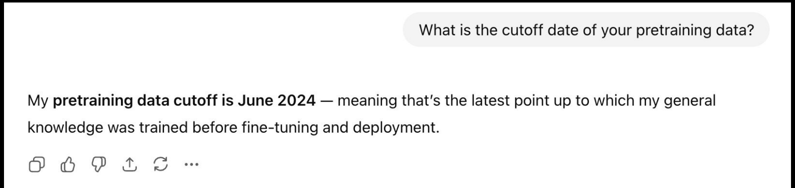
- **Multiple steps in training LLMs**
  - Pre-training
  - Post-training: SFT, (RLHF, DPO, ...) → alignment

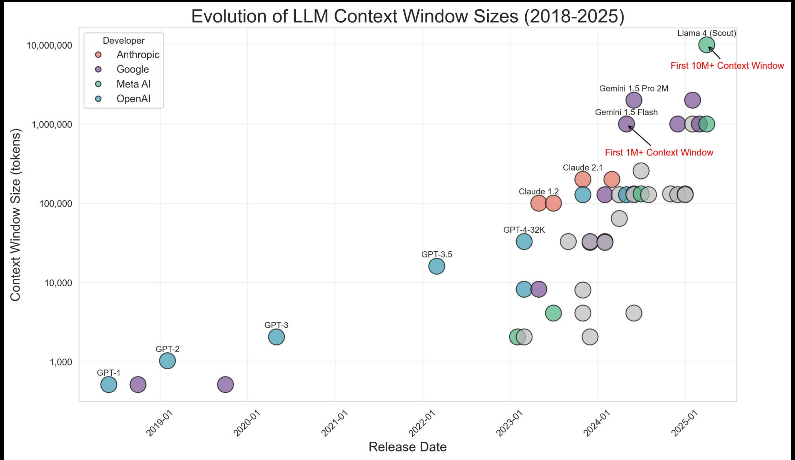



- **Pre-training usually done once in a while, and it's very costly:**
  - "... Claude 3.5 Sonnet is a mid-sized model that cost a few **$10M's** to train (I won't give an exact number)." ***Dario Amodei, Anthropic***

# LLMs Limitations

- **Static Knowledge & Hallucination**
- **Context Limitation**
- **Lack of access to tools**
- **Operational Gaps**



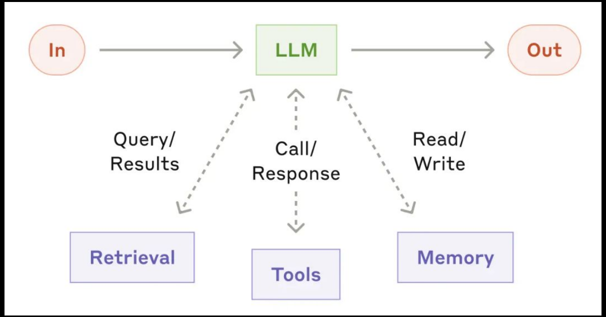




Plot from meibel.ai (url)

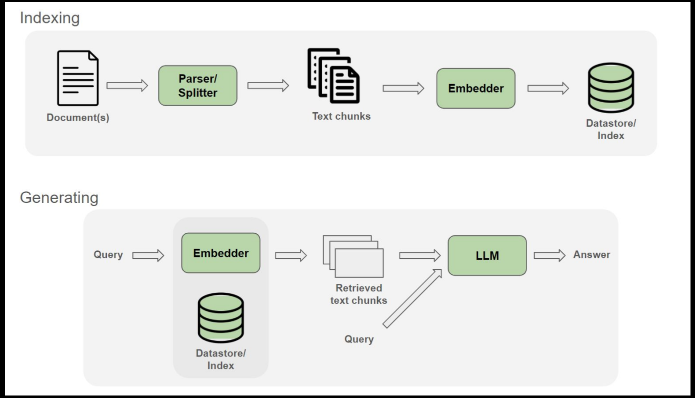
# Augmented LLM

- 🔧 Tools use and Function calling
- 📄 Retrieval Augmented Generation (RAG)
- 📝 Extended Context / Memory
- 🚧 Guardrails and constraints

from Anthropic blog on building effective agents, url




from Nvidia docs, url



# Augmented LLM Limitations

- **Single pass processing, with fixed workflows**
  - No iteration or refinement loop; w/o sophisticated workflows
  - No support for continuous human-in-the-loop feedback
- **Limited Goal-directed behavior**
  - Can't break down complex tasks autonomously
- **Limited Self-assessment**
  - Unless chained to other LLMs, cannot evaluate its own responses

# AI Agents

Systems where LLMs dynamically direct their own processes and tool usage—understanding complex inputs, reasoning and planning, using tools reliably, and recovering from errors—while maintaining control over how they accomplish tasks.

**by Anthropic**

An autonomous application that attempts to achieve a goal by observing the world and acting upon it using the tools that it has at its disposal.

**by Google**

Systems that independently accomplish tasks on your behalf, allowed by 2 core characteristics:

- Autonomous Workflow Management
- Dynamic & Safe Tool Utilization

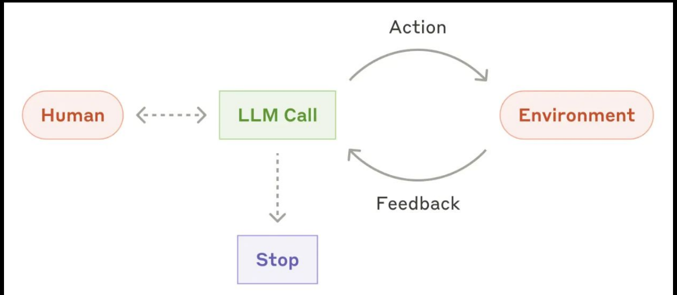
**by OpenAI**

An agent is an **autonomous application** that continuously interacts with its **environment** by **observing inputs**, **planning** responses, **acting** to accomplish tasks independently, and recovering from errors using prior context or memory, repeating this **observe–plan–act** cycle until the objectives are achieved.

# Agents vs Augmented LLMs

| Aspect | Augmented LLM | AI Agent |
|---|---|---|
| **Task Execution** | Single Pass | Iterative |
| **Human Involvement** | Constant | Oversight |
| **Goal oriented** | No | Yes |
| **Self-correction** | No | Yes |

from Anthropic blog on building effective agents, url




- Augmented LLM
  - "Analyze this imaging study" → provides single analysis
- Agent
  - "Work up this patient for suspected PE" → queries history, orders relevant labs, analyzes imaging, calculates Wells score, recommends anticoagulation protocol, flags for peer review

# Agent Components

- **A Language Model**
  - Providing the cognitive core: doing reasoning, planning and action taking

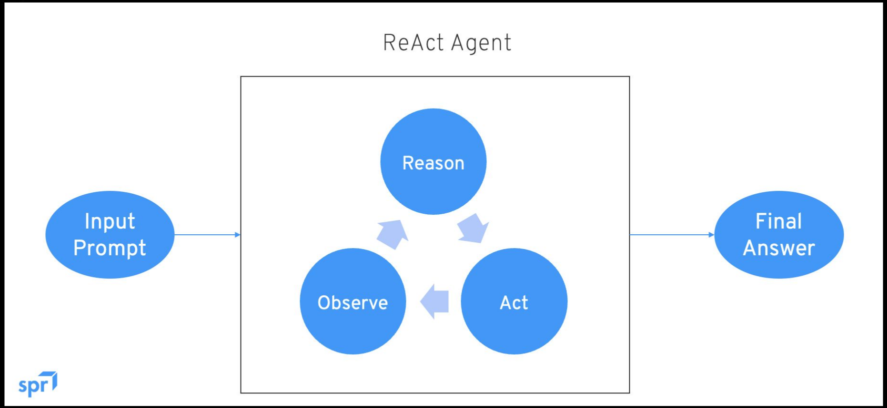
- **An Orchestration Layer:**
  - **Profile:** Describing agent's identity, goals, and instructions to follow
  - **Memory**
    - Short-term
    - Long-term
  - **Guardrails:**
    - Layers safety mechanisms that help control risk in LLM systems

- **Environment:**
  - A pre defined toolset and resources
  - Protocols and interfaces

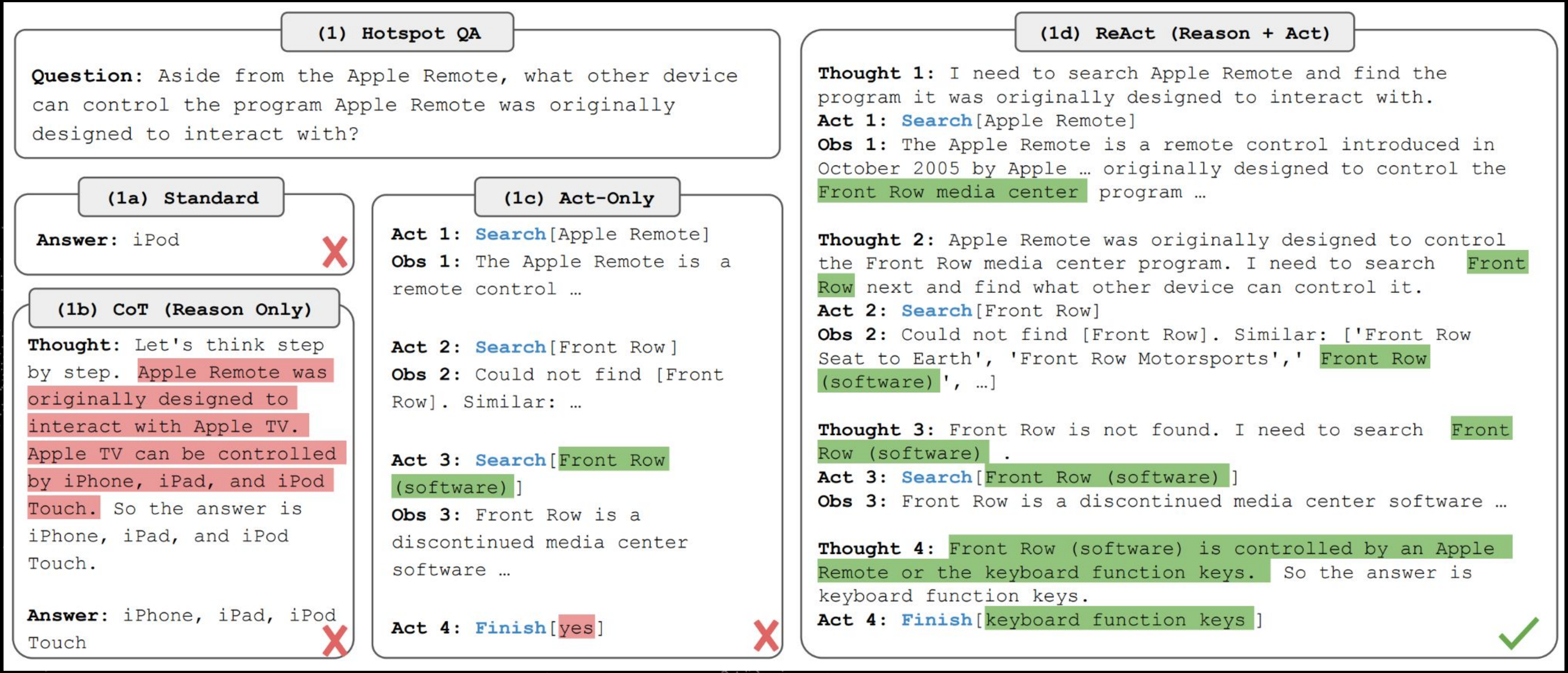
# Agent Characteristics

- **Tool Use & Environment Interaction**
  - Tool selection and chaining
  - Affecting environment via APIs
- **Autonomy**
  - Self-directed action with minimal human intervention
- **Reasoning and Planning**
  - Task decomposition + planning

From spr.com (url)



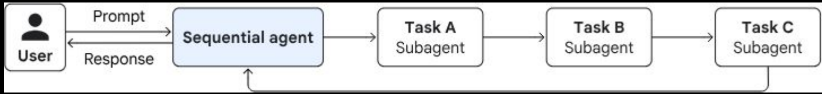
# ReAct Framework Example



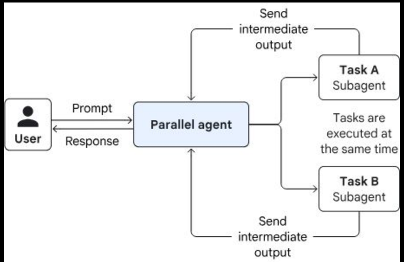
# Multi-agent Systems

- Why multiple agents?
  - Specialization
  - Parallelization / Scalability
  - Verification
  - Emergent intel. ➔ the sum is greater than the parts

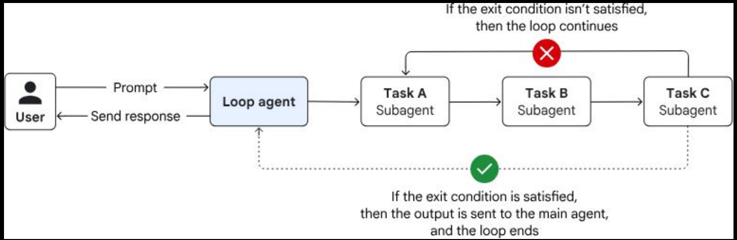
**Sequential Pattern**




**Parallel Pattern**




**Loop Pattern**






Source: Google Cloud Doc (URL)

# Multi-agent Systems: Examples

- **Landmark paper on multi-agent applications**
  - **25 agents** living in a Sims-inspired sandbox town
  - **Valentine's Day party**: One agent decides to throw a party → agents autonomously spread invitations, make new friends, ask each other on dates, and coordinate attendance

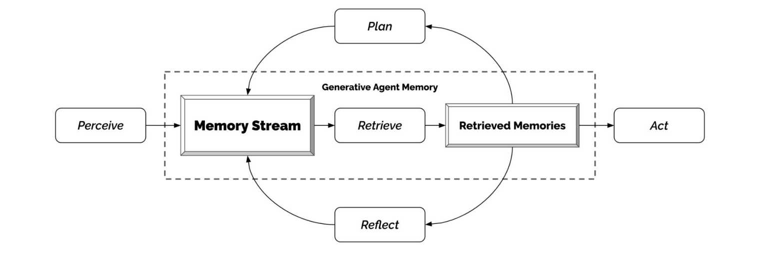
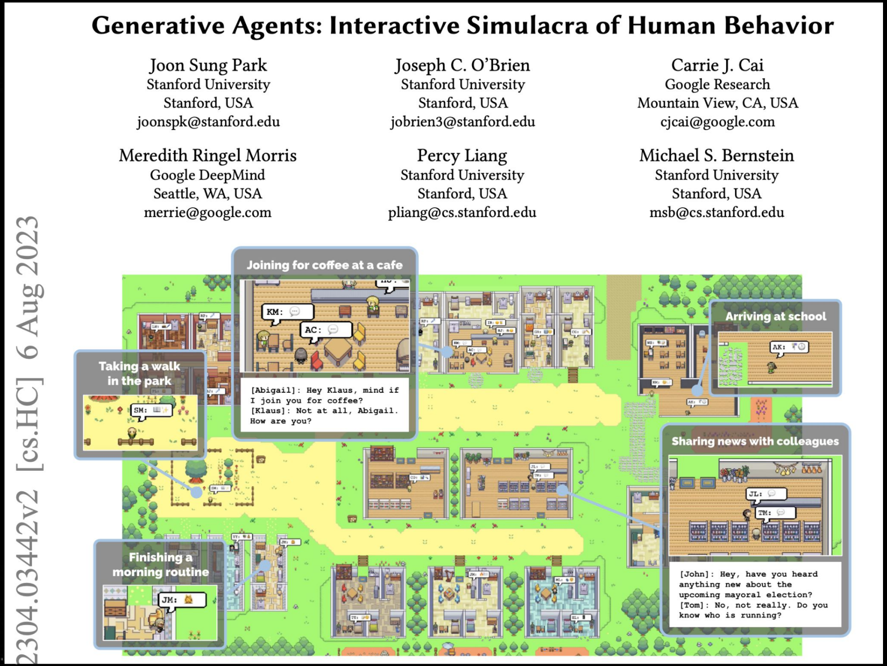



Figure 5: Our generative agent architecture. Agents perceive their environment, and all perceptions are saved in a comprehensive record of the agent's experiences called the memory stream. Based on their perceptions, the architecture retrieves relevant memories and uses those retrieved actions to determine an action. These retrieved memories are also used to form longer-term plans and create higher-level reflections, both of which are entered into the memory stream for future use.

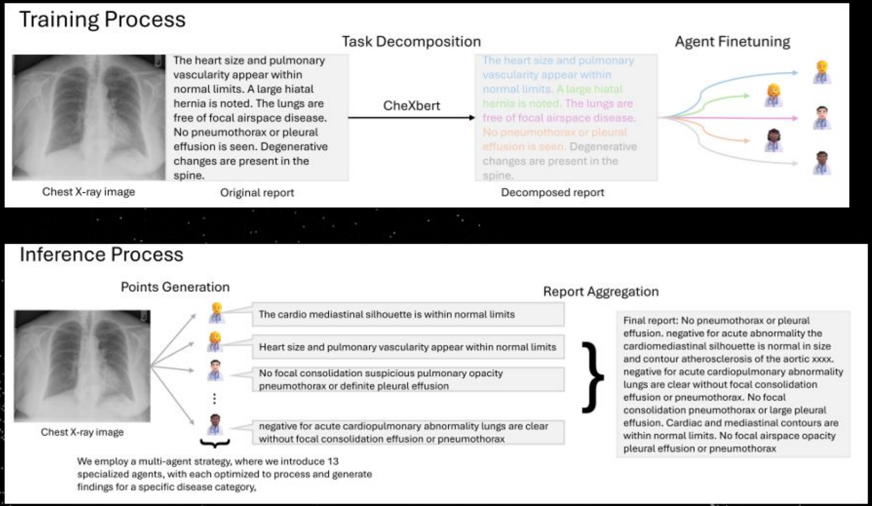


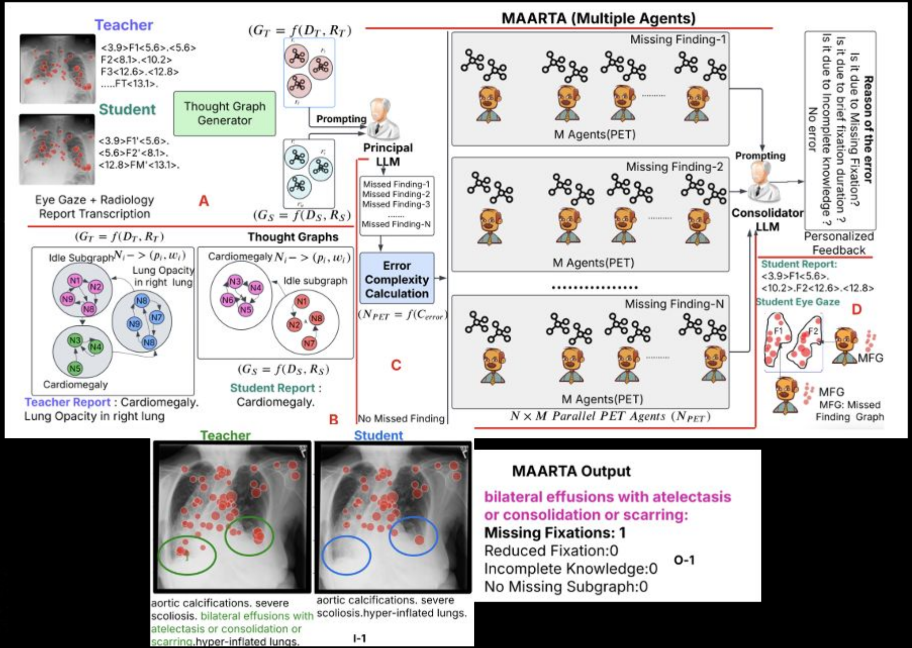
**Generative Agents: Interactive Simulacra of Human Behavior**

Joon Sung Park, Stanford University, Stanford, USA, joonspk@stanford.edu
Joseph C. O'Brien, Stanford University, Stanford, USA, jobrien3@stanford.edu
Carrie J. Cai, Google Research, Mountain View, CA, USA, cjcai@google.com
Meredith Ringel Morris, Google DeepMind, Seattle, WA, USA, merrie@google.com
Percy Liang, Stanford University, Stanford, USA, pliang@cs.stanford.edu
Michael S. Bernstein, Stanford University, Stanford, USA, msb@cs.stanford.edu

2304.03442v2 [cs.HC] 6 Aug 2023

# Multi-agent Systems: Examples




**MRGAgents** (Medical Report Generation Agents) by Wang et al

**MAARTA (Multi-Agentic Adaptive Radiology Teaching Assistant)** by Awasthi et al.

# Thank you

Please open the RSNA 2025 GitHub Repository to continue:

**https://github.com/RSNA/AI-Deep-Learning-Lab-2025**

## Contact


@PRouzrokh;
in/pouria-rouzrokh
po.rouzrokh@gmail.com

@MelinaHosseiny;
in/melina-hosseiny-m-d-0a12b7a1
melina2.hosseiny@gmail.com

@MoeinShariatnia;
/in/moein-shariatnia
moein.shariatnia@gmail.com