

Seeing the Un-Scene: Learning Amodal Semantic Maps for Room Navigation

Medhini Narasimhan^{1,2*} Erik Wijmans^{3,1} Xinlei Chen¹ Trevor Darrell²
Dhruv Batra^{1,3} Devi Parikh^{1,3} Amanpreet Singh¹

¹Facebook AI Research ²University of California, Berkeley ³Georgia Institute of Technology

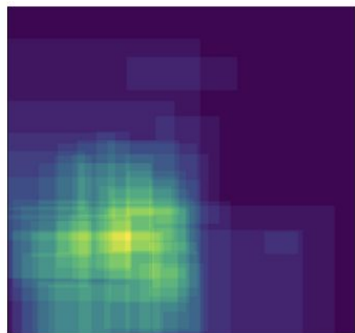
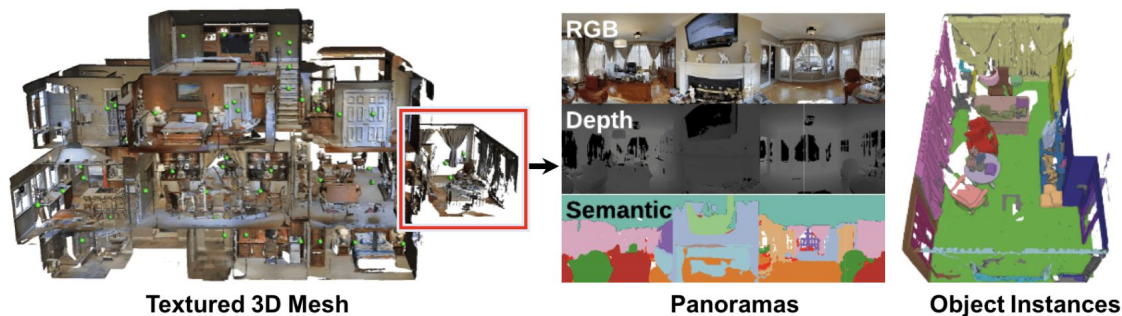




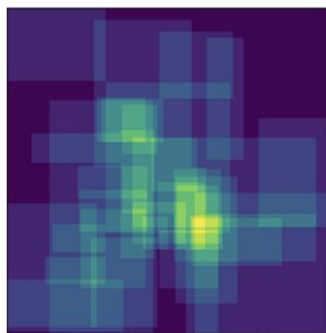
Motivation

Human priors on apartment layouts
aid visual navigation

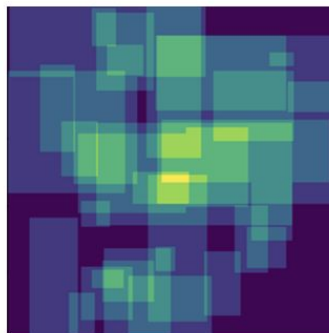
Matterport3D: Learning from RGB-D Data in Indoor Environments



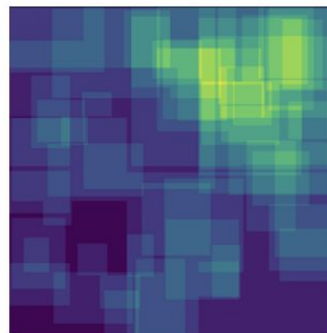
Kitchen



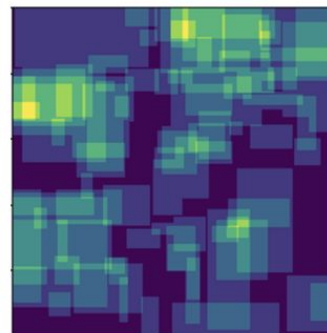
Dining Room



Living Room



Bedroom



Bathroom

Question

Can learning to predict semantic belief maps
improve visual navigation in unseen
environments?

Approach

Learn scene priors

(meta-learning)

+ visual exploration

+ actively update belief maps

**+ predict regions beyond
current field of view**

Improved navigation in unseen environments

(few-shot learning)

Background

Simultaneous Localization and Mapping (SLAM)

1. Map environment from sensory data
2. Locate agent in map
3. Plan path

Limitations

- Noisy data
- No meta-learning

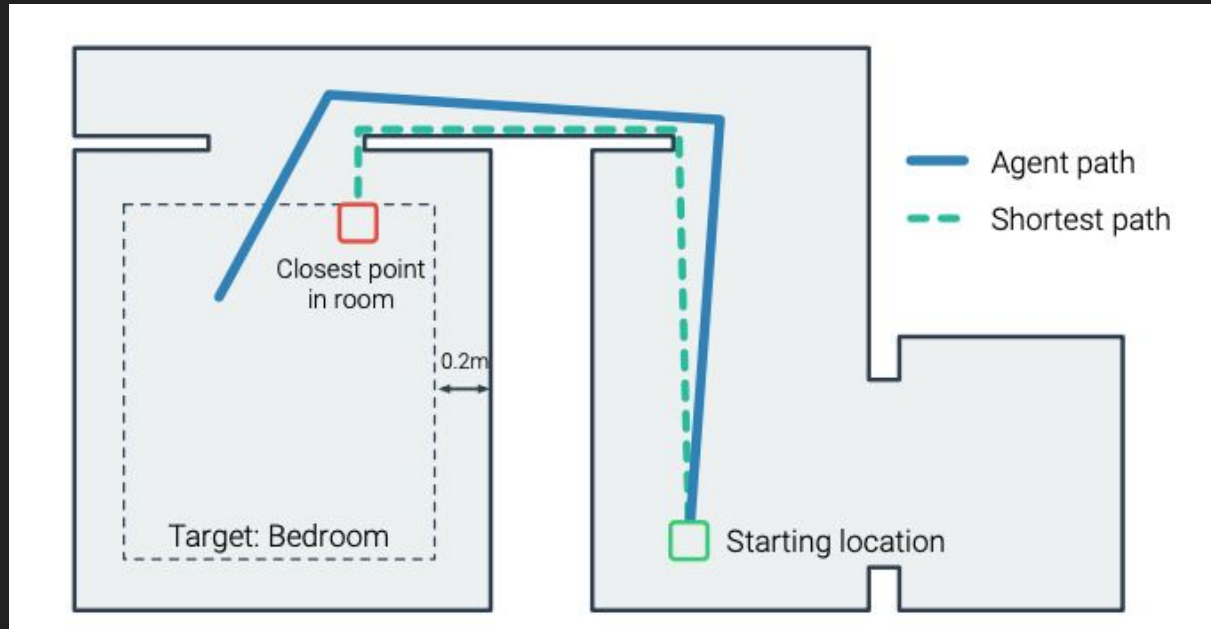
Learning based approaches

- End-to-end
- Learn goal-driven RL task with auxiliary depth prediction tasks
- Priors as knowledge graphs or probabilistic relationships

Limitations

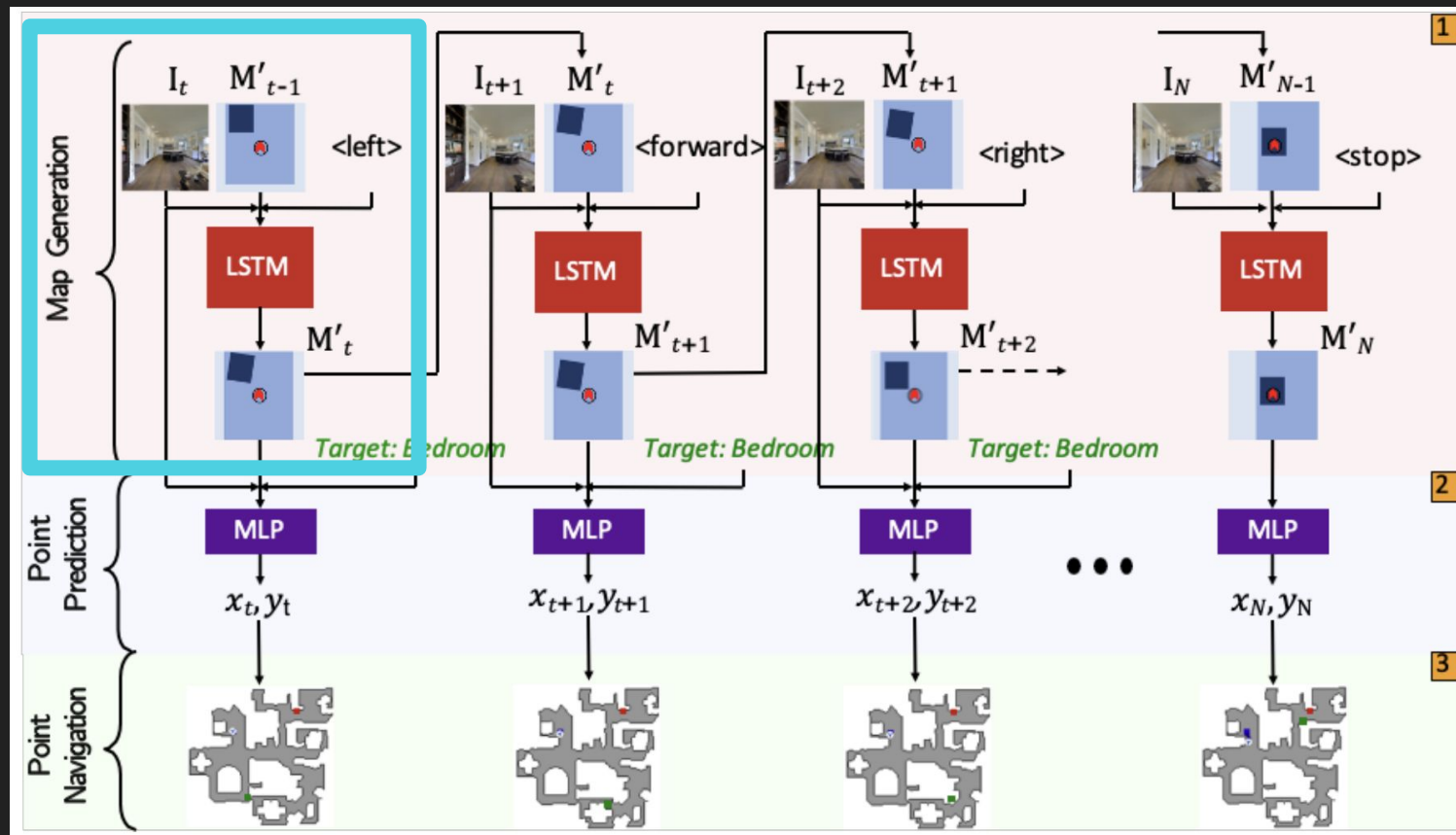
- No amodal mapping
- No semantic beliefs

Task



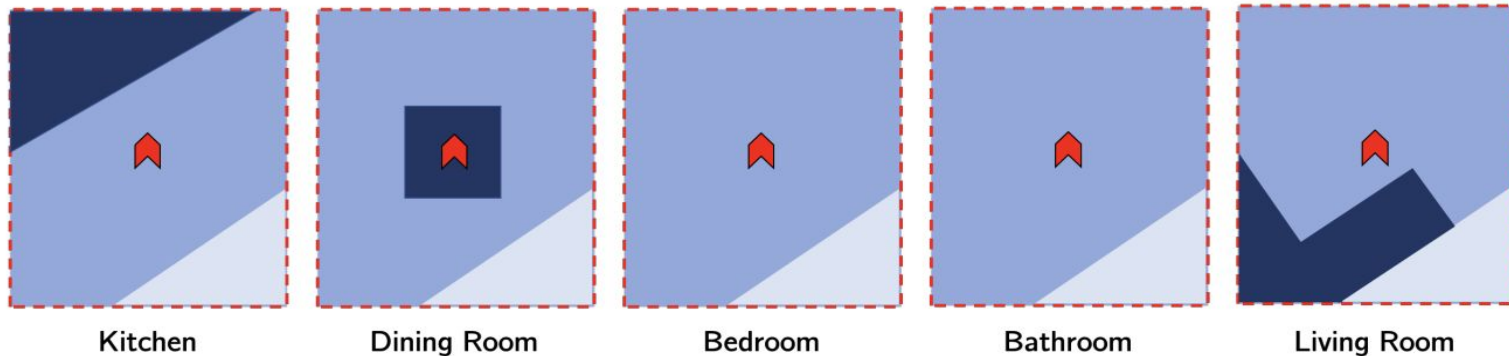
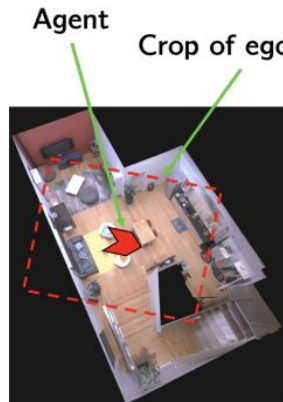
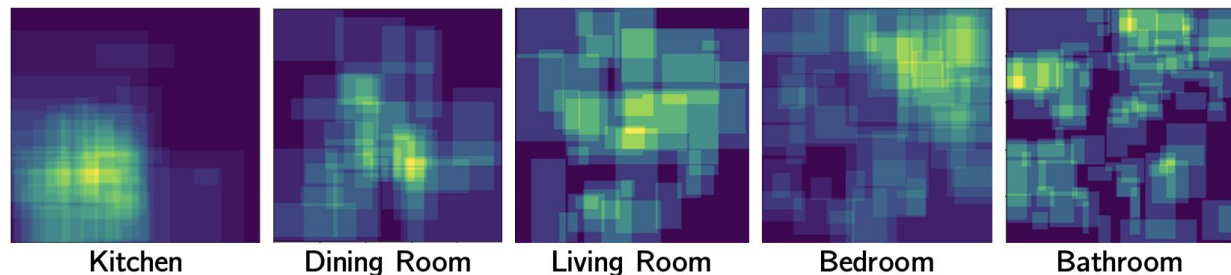
Room navigation

Learning algorithm



time

Scene Priors / Semantic Belief Maps



1) Map Generation

Input image

current semantic map

action



1) Map Generation Alg

$$\mu \sim \text{Uniform}(0, 1) \quad (1)$$

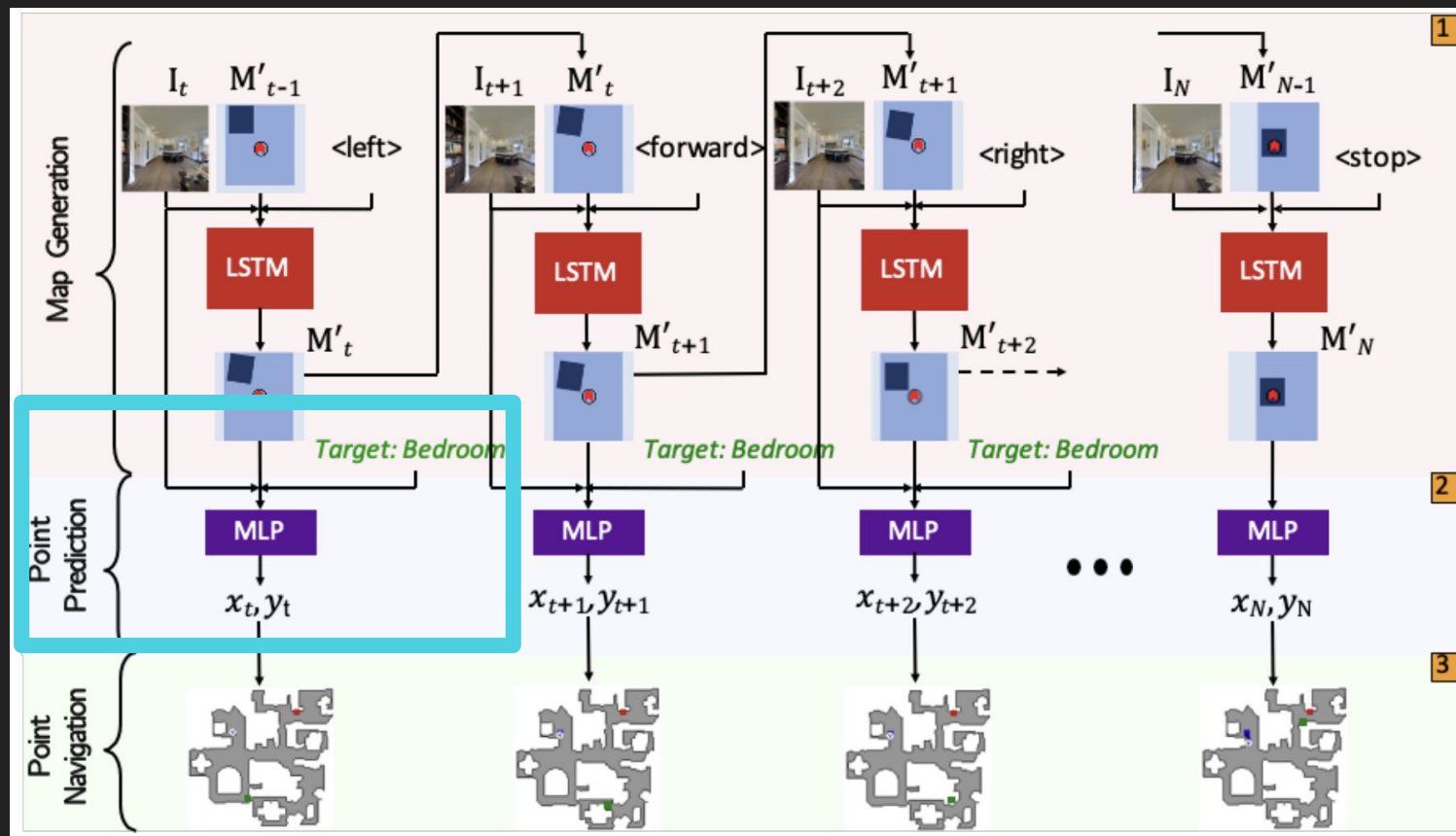
$$M_{t,r}^{\text{input}} = \begin{cases} M_{t-1,r}^{\text{GT}}, & \text{if } \mu > 0.5 \text{ or } t = 0, \\ M_{t-1,r}^{\text{pred}}, & \text{otherwise} \end{cases} \quad (2)$$

$$h_{t,r} = f_{\text{seq}}(f_m(M_t^{\text{input}}), f_{\text{act}}(a_{t-1}), f_i(I_t)) \quad (3)$$

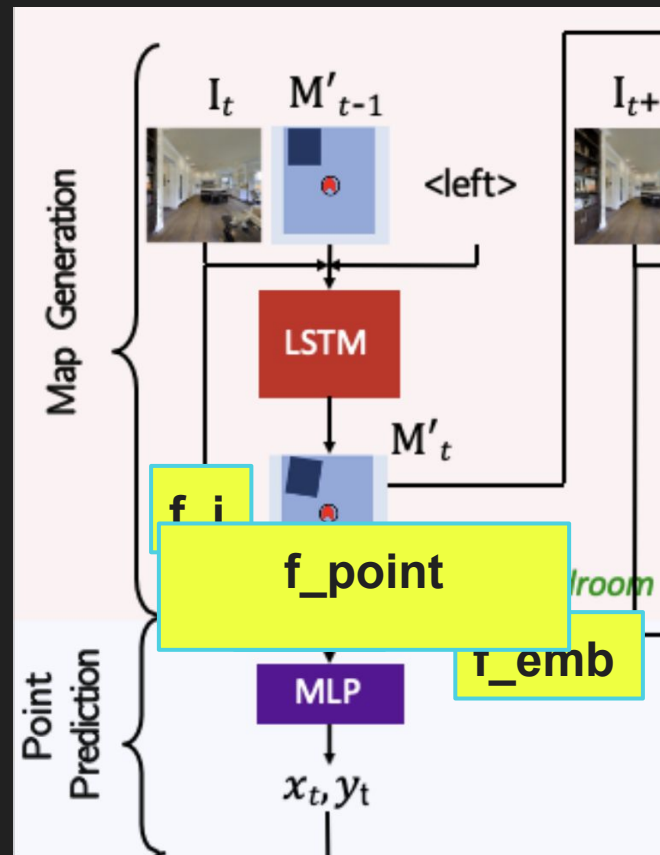
$$M_{t,r}^{\text{pred}} = f_{\text{dec}}(h_{t,r}) \quad (4)$$

$$\mathcal{L}_{\text{map}} = \sum_{r \in \mathcal{R}} \text{CrossEntropy}(M_{t,r}^{\text{pred}}, M_{t,r}^{\text{GT}}) \quad (5)$$

Learning algorithm



2) Point Prediction



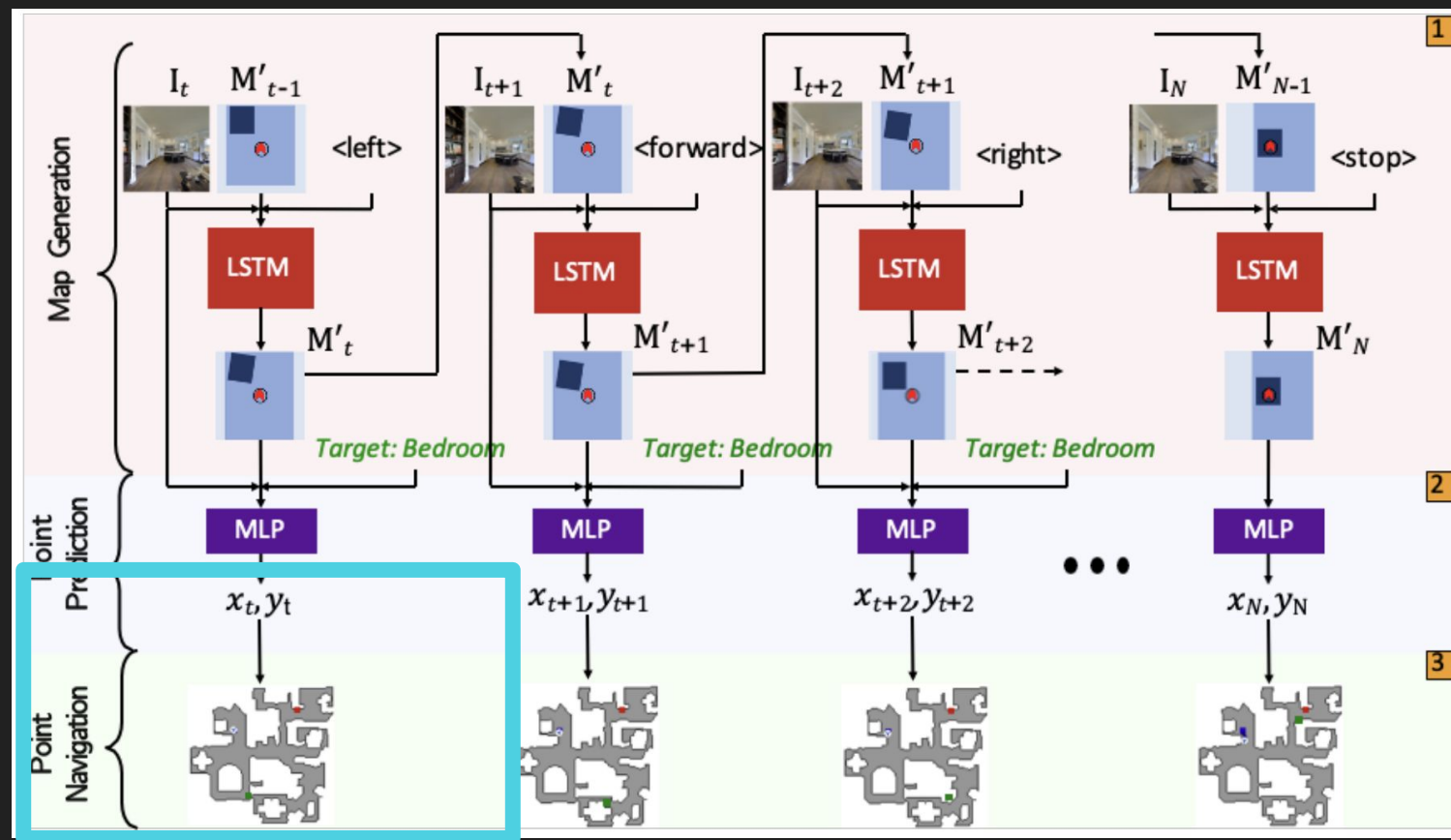
2) Point Prediction Alg

$$g_M = f_m(M_{t,r}^{\text{pred}}) \odot f_{emb}(tr) \quad (6)$$

$$P_t^{\text{pred}} = (x'_t, y'_t) = f_{point}(g_M, f_i(I_t), f_{emb}(tr)) \quad (7)$$

$$\mathcal{L}_{point} = \text{MSELoss}(P_t^{\text{pred}}, P^{\text{GT}}) \quad (8)$$

Learning algorithm



time

3) Point Navigation

- Proximal Policy Optimization (PPO) ¹
- Pre-trained a navigation policy on Habitat² dataset

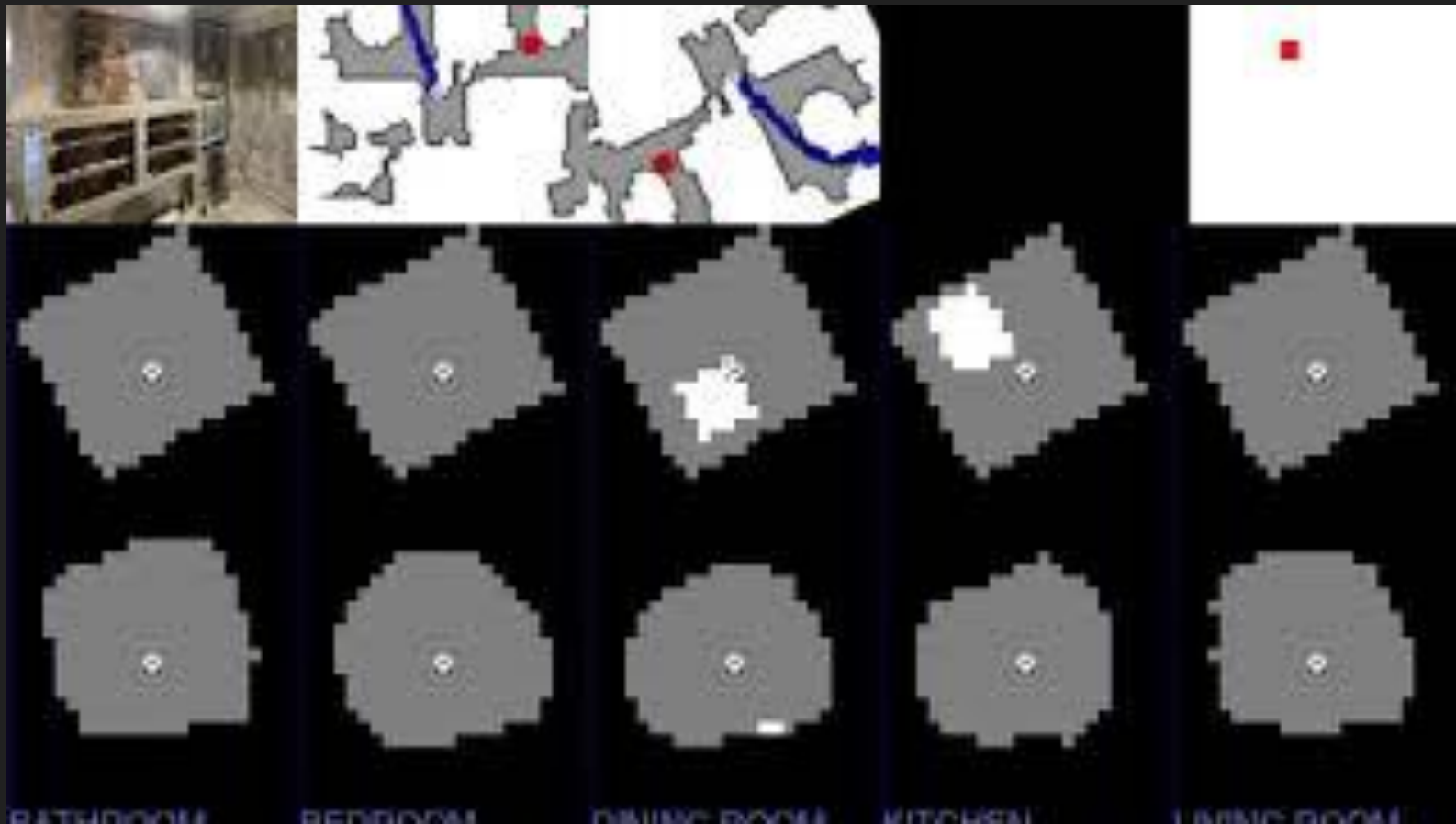
$$a_t = \pi_{nav}(P_t, f_d(D_t), a_{t-1}) \quad (9)$$

- Fine-tuned on points predicted by their model

[1] Wijmans et al.: DD-PPO: Learning near-perfect point goal navigators from 2.5 billion frames. In: International Conference on Learning Representations (ICLR) (2020)

[2] Savva et al.: Habitat: A platform for embodied ai research. arXiv preprint arXiv:1904.01201 (2019)

In action



Eval Metrics

RoomNav Success
weighted by (normalized
inverse) Path Length
(**RoomNav SPL**)

Success: indicator for
whether agent stops 0.2m
inside query room

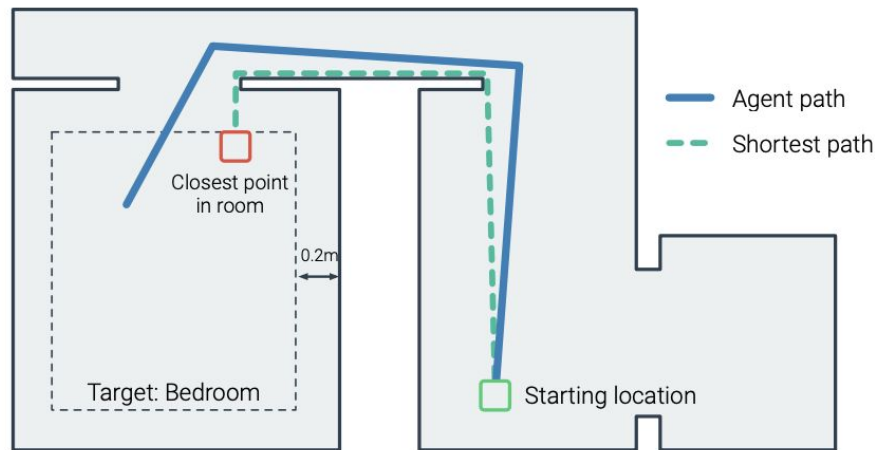


Fig. 5: **SPL** for **Room Navigation**.

Results

RoomNav Model	Validation		Test	
	SPL	Success	SPL	Success
Baseline: Random	0.00	0.00	0.00	0.00
Baseline: Vanilla Room Navigation Policy	0.10	0.15	0.10	0.11
Baseline: Map Generation + Room Navigation Policy	0.16	0.17	-	-
Baseline: Point prediction + PointNav	0.17	0.20	-	-
Baseline: Point prediction + PointNav + Fine-tune	0.21	0.23	-	-
Our: Map Generation + Point Prediction + PointNav	0.31	0.35	-	-
Our: Map Generation + Point Prediction + PointNav + Fine-tune	0.35	0.38	0.33	0.36
Oracle: GT Maps + Room Navigation Policy	0.54	0.56	-	-
Oracle: GT Maps + Point Prediction + PointNav	0.61	0.64	-	-
Oracle: GT Maps + Point Prediction + PointNav + Fine-tune	0.67	0.68	-	-
Oracle: GT Point Selection + PointNav	0.83	0.84	0.79	0.82

Conclusion

Yes! Predicting amodal semantic belief maps of room layouts
improves navigation in unseen environments

Limitations

- Diversity of room types
- Continuous map predictions
- Less modularity
- Performance gap from ground truth maps indicates room to improve!



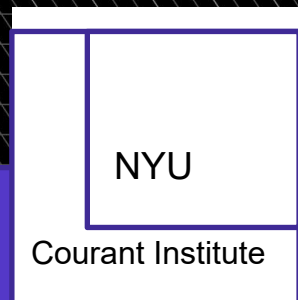
Questions/Comments

Few-shot Learning from Representation to Problem-solving: A Conversation

(Few-shot Learning + Diffusion)

Xu Zhang (xz4863)

栩



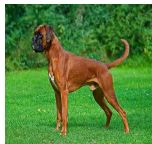
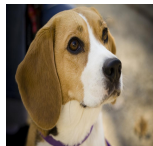
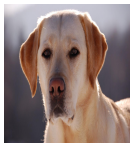
Few-shot Learning Task Definition

Learn a new concept with only a few examples

e.g. a new image class

Few-shot Learning Task Definition

Train base model on
Support Set

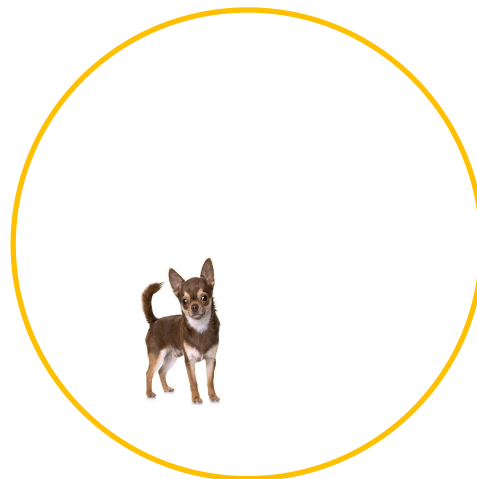


Labrador Retriever
5000 img's

Boxer
5000
imgs

5000 imgs
Beagle

Make sure it performs well in
Query Set

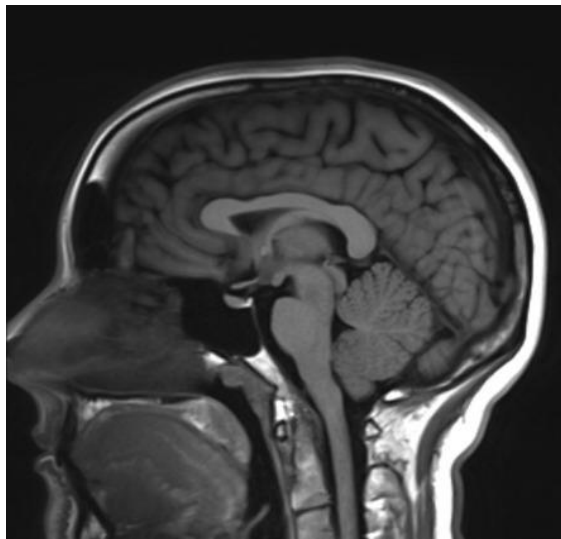


chihuahua Only 3 images
(not present in support)

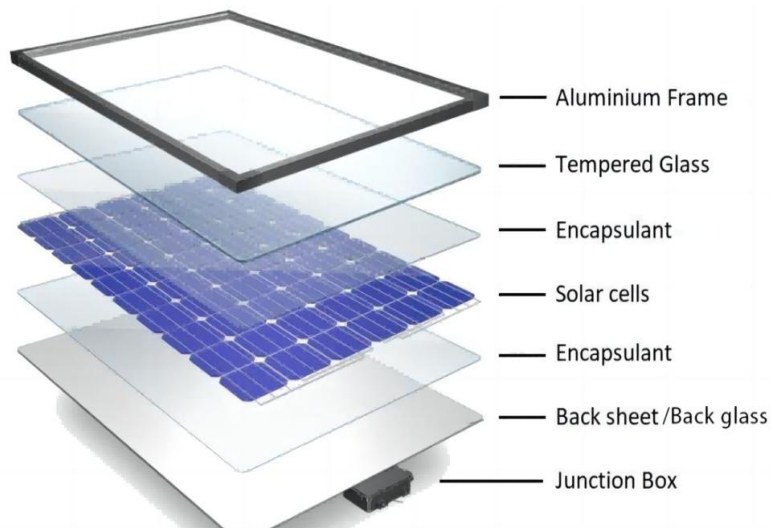
Few-shot Learning Task Motivation

Application:

Scene 1: Cancer doctor
detecting rare cancer



Scene 2: Solar panel Factory defect



Human Don't learn from tons of example >> Few-shot learning ability is critical characteristic of human intelligence

The Chinese room argument

Few-shot Learning as a test to machine's intelligence

Few-shot Learning Task Definition

- Example: Bruce Li's kung fu skills are so good, he's such a **YYDS!**

yyds

yyds 永远的神 means **eternal** god in **English** it is used to describe something or someone you love and nothing can **beat it**

A Chinese **dialectical** abbreviation. Used to abbreviate “**yong yuan de shen**”, literally translated into “forever god”. Used when praising a majority-agreed awesome thing. Compare with GOAT.

what is yyds

ChatGPT

"YYDS" is an abbreviation commonly used in Chinese internet slang. It stands for "Yǒu Yú Dìng Shū" (有娱定神), which translates to "Entertainment controls the mind." This phrase is often used to express the idea that entertainment, such as movies, TV shows, or games, can captivate and distract people's attention from their daily worries or troubles. It's frequently used in online discussions related to entertainment topics.

🔊 📄 ↺ 🔒

Hallucination

Wrong with old models

Correct with new models:

what is yyds

📄 ✎ Edit



"YYDS" is a popular internet slang term in Chinese, short for "永远的神" (yǒngyuǎn de shén), which translates to "**Forever God**" or "**Eternal Legend**" in English.

Meaning & Usage:

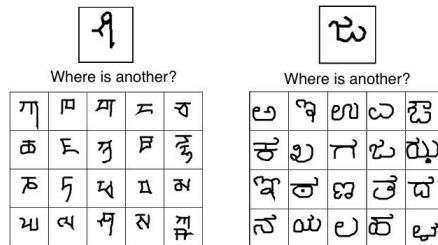
- It's used to express extreme admiration, praise, or worship for someone or something (e.g., a celebrity, athlete, game, food, etc.).
- Similar to saying "**GOAT**" (**Greatest of All Time**) in English.

Examples:

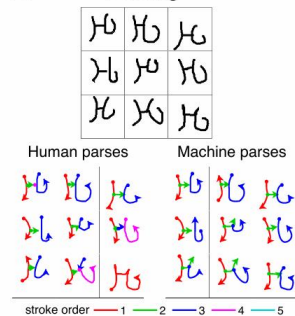
- "梅西, YYDS!" → "Messi, the GOAT!"
- "这家火锅店 YYDS!" → "This hot pot restaurant is legendary!"

Example of Benchmark: Omniglot

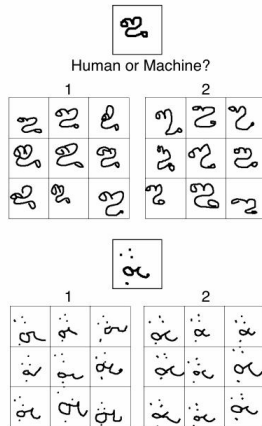
A One-shot classification



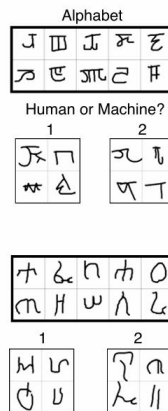
B Parsing



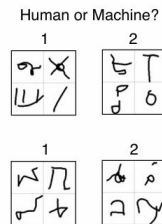
C Generating new exemplars



D Generating new concepts (from type)



E Generating new concepts (unconstrained)



credit: Lake, Brenden M., Ruslan Salakhutdinov, and Joshua B. Tenenbaum. "The Omniglot challenge: a 3-year progress report." *Current Opinion in Behavioral Sciences* 29 (2019): 97-104.

Classic Few-shot Learning Methods

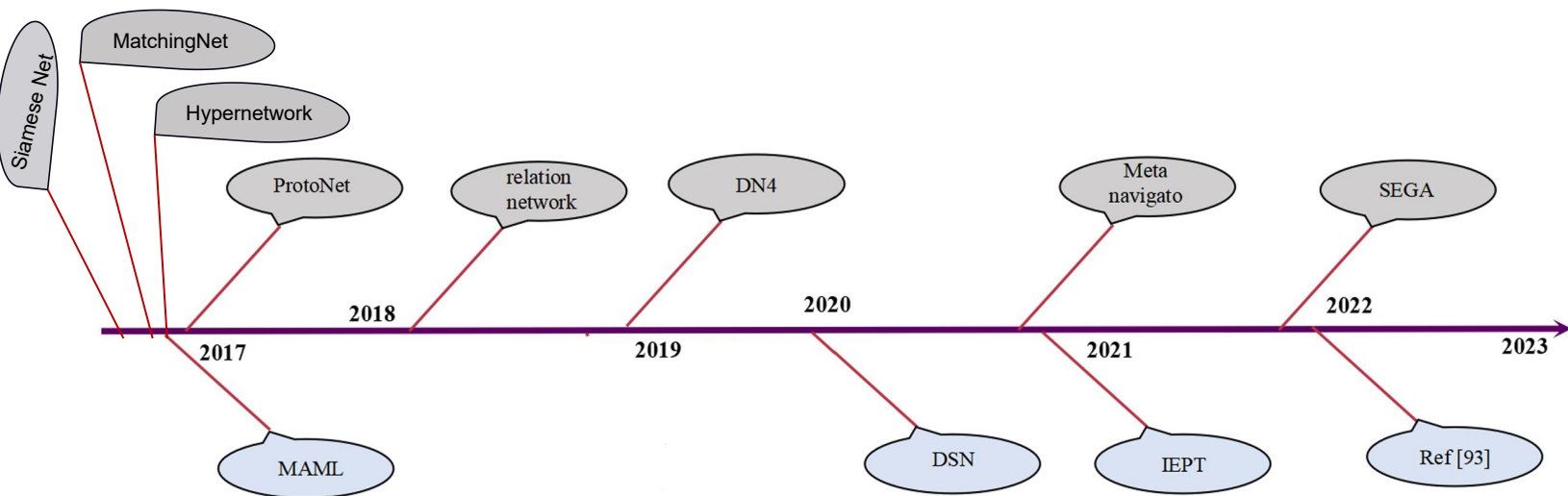
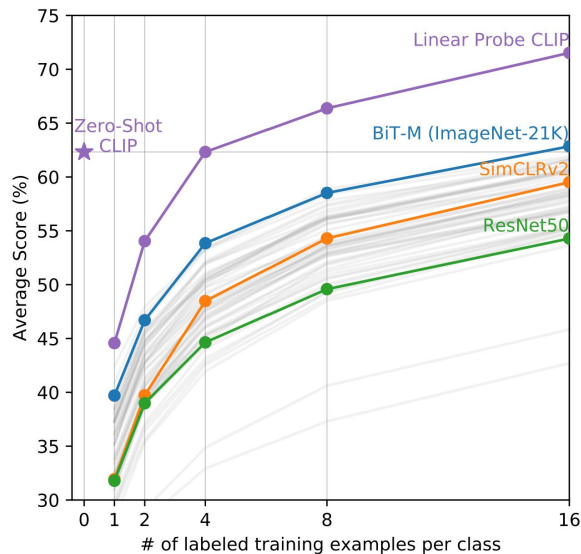


Figure 3. Some mainstream and advanced FSL methods in recent years.

credit: Zeng, Wu, and Zheng-Ying Xiao. "Few-shot learning based on deep learning: A survey." Mathematical biosciences and engineering : MBE vol. 21,1 (2024): 679-711. doi:10.3934/mbe.2024029

Few-shot Learning with Foundation Models

CLIP



Flamingo

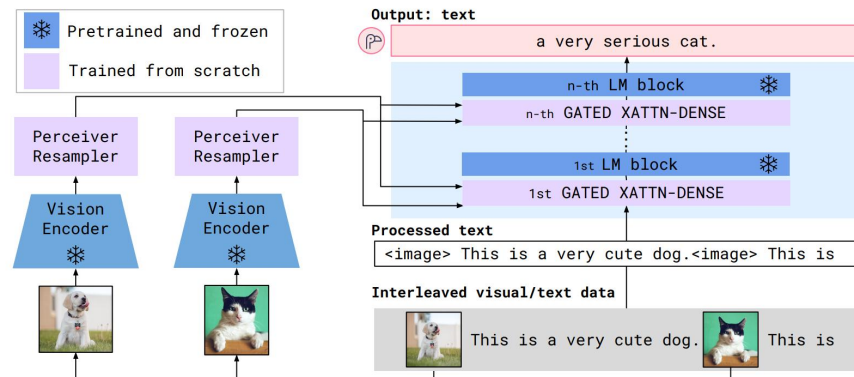


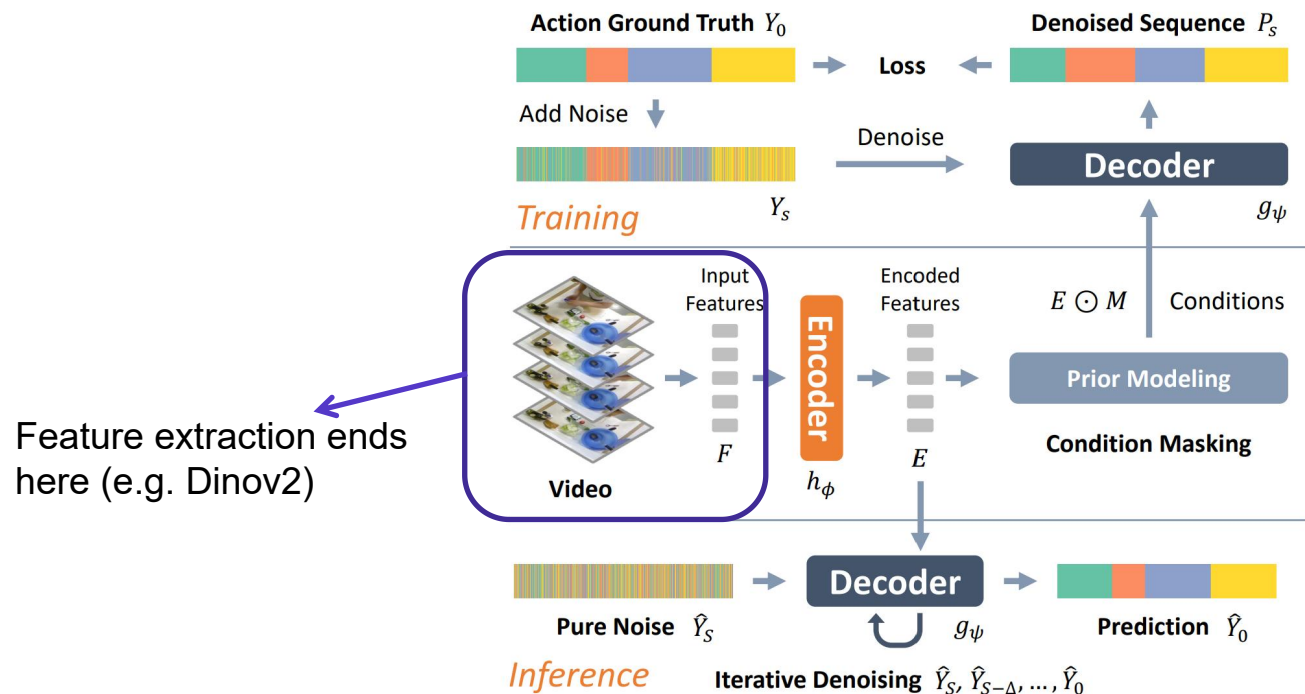
Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.

Few-shot Learning Methods that Focuses on Representation

- What's in common:
both focuses on feature/representation
- Assume small 'head' for downstream tasks:
can be problematic

Few-shot Learning for Harder Task: an example on Video Segmentation

Diffusion Action Segmentation



Few-shot Learning with Diffusion

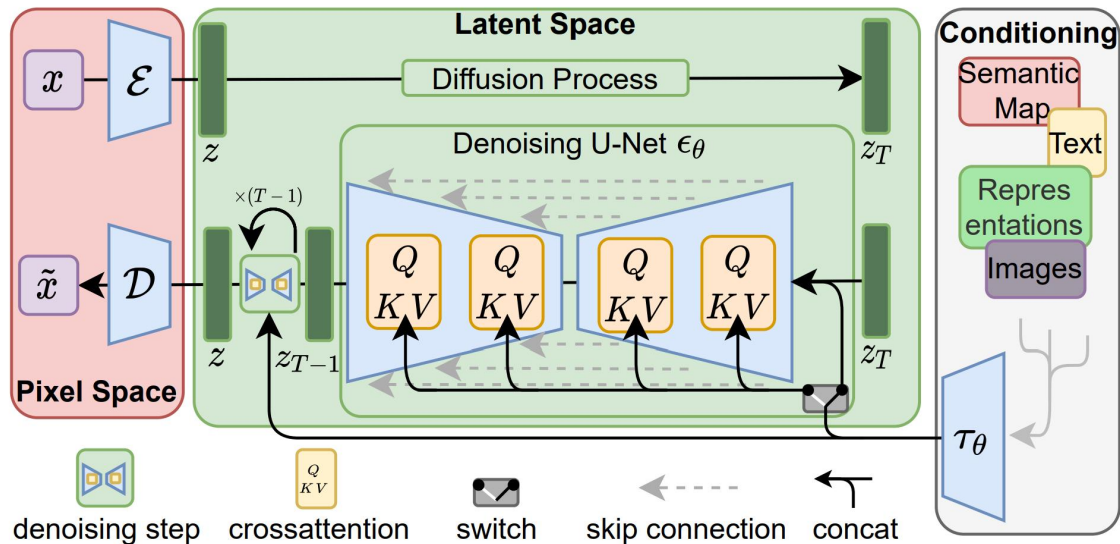
1. Case Study:

Multi-Concept Customization of Text-to-Image Diffusion

2. Why Few-shot learning working well on Diffusion?

Few-Shot Diffusion Models Escape the Curse of Dimensionality

Latent Diffusion:



Given text features $\mathbf{c} \in \mathbb{R}^{s \times d}$ and latent image features $\mathbf{f} \in \mathbb{R}^{(h \times w) \times l}$, a single-head cross-attention operation consists of $Q = W^q \mathbf{f}$, $K = W^k \mathbf{c}$, $V = W^v \mathbf{c}$,

Multi-Concept Customization of Text-to-Image Diffusion

Nupur Kumari¹

Bingliang Zhang²

Richard Zhang³

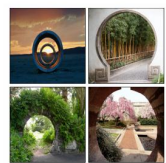
Eli Shechtman³

Jun-Yan Zhu¹

¹Carnegie Mellon University

²Tsinghua University

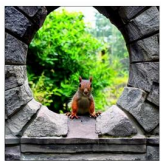
³Adobe Research



A photo of a **moongate**



A **moongate** in the snowy ice



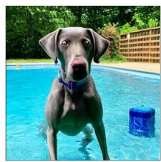
A squirrel in front of **moongate**



Watercolor painting of **moongate** in a forest



A photo of a **V* dog**



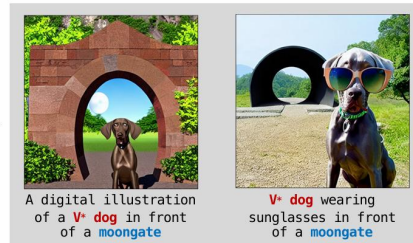
A **V* dog** in a swimming pool



A **V* dog** wearing sunglasses



A **V* dog** oil painting, Ghibli inspired



A digital illustration of a **V* dog** in front of a **moongate**

V* dog wearing sunglasses in front of a **moongate**

Multi-concept composition

> Tuning a Small Set of Parameters

Diff Modules' Weight Update Speed in Diffusion Model

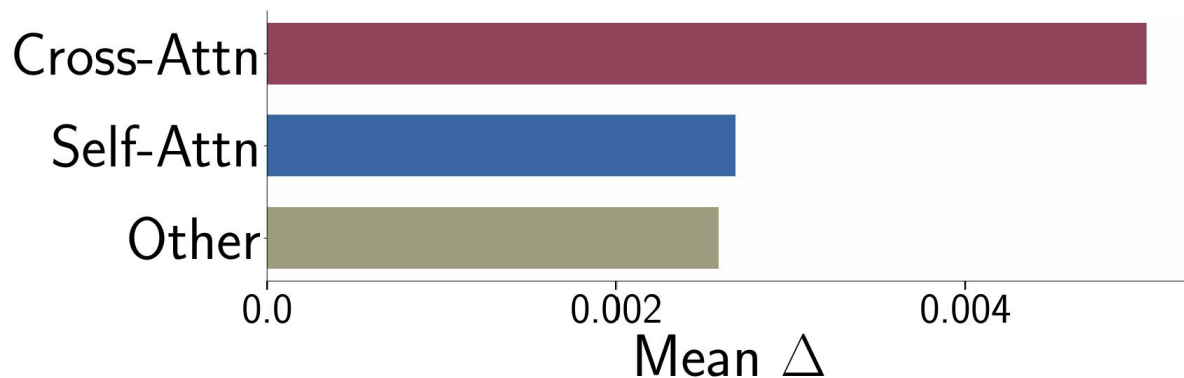


Figure 3. **Analysis of change in weights** on updating all network weights during fine-tuning. The mean change in the cross-attention layers is significantly higher than other layers even though they only make up 5% of the total parameter count.

Proposed Framework

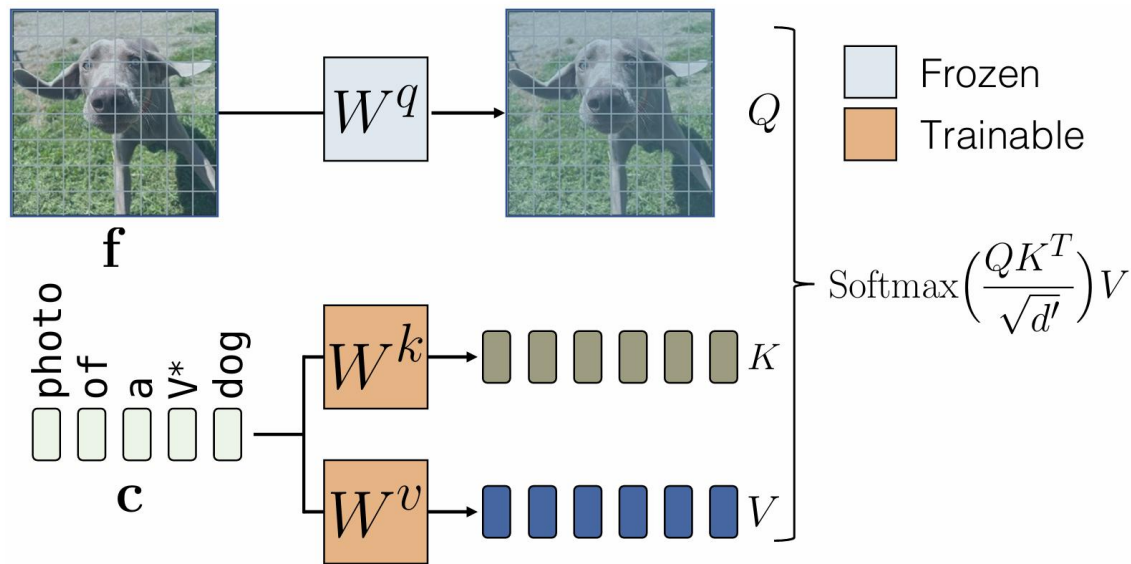


Figure 4. **Single-head Cross-Attention.** Latent image feature f and text feature c are projected into query Q , key K , and value V . Output is a weighted sum of values, weighted by the similarity between the query and key features. We highlight the updated parameters W^k and W^v in our method.

Regularization to Mitigate Overfitting

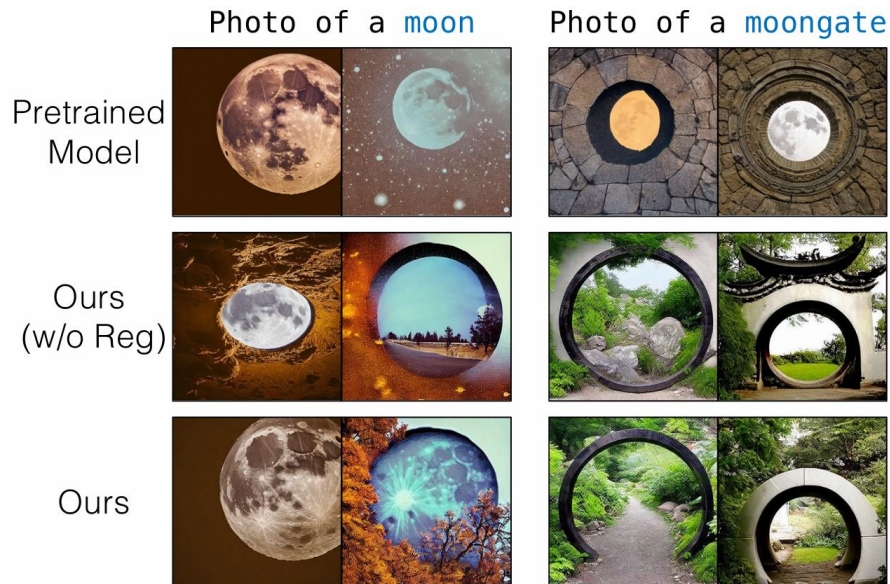
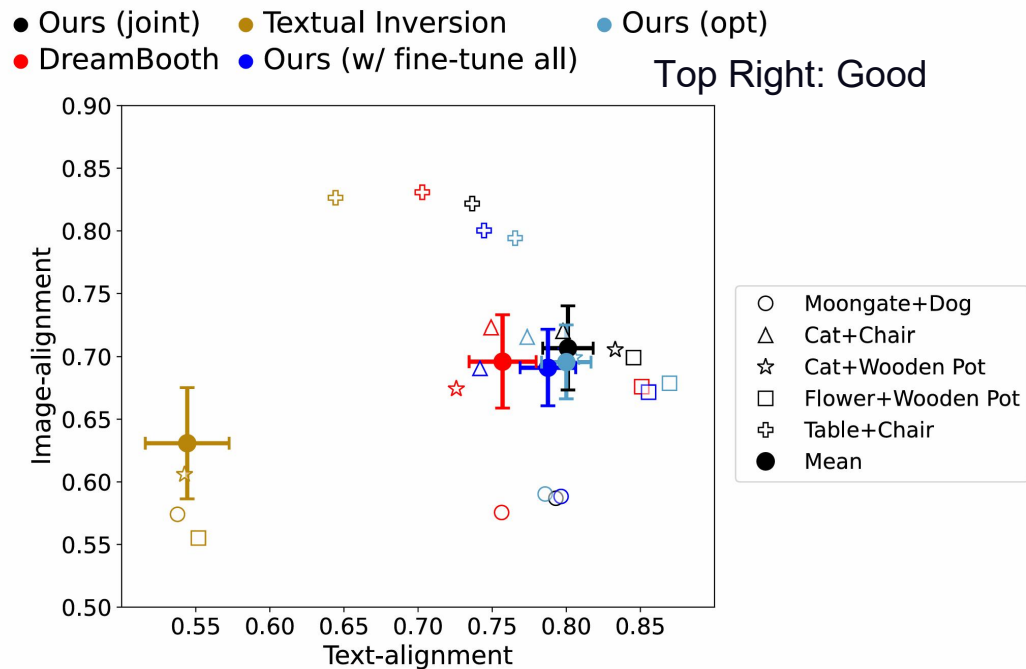


Figure 5. **Role of regularization data in mitigating overfitting behavior during fine-tuning.** 1st row: samples from pre-trained models. In 2nd row, fine-tuning cross-attention key, value projection matrices without any regularization dataset leads to moongate like images on the text prompt photo of a moon.

Result: Quantitative



How did Tuning Diffusion Work Out?

Diffusion models are large

Why tuning only a small set of parameters

with a few examples can produce such good results?

Why diffusion, not other models?

Few-Shot Diffusion Models Escape the Curse of Dimensionality

Theorem 4.3. Let $\alpha(n) = \frac{d \log \log n}{\log n}$, $F = \frac{(d+C_Z)^d \beta^d}{\delta^2 c_0}$ and network parameter $\epsilon = n_{ta}^{-1/2}$. Assume

Assumption 3.1 4.1 4.2 and $n_{ta}^{\frac{d+5}{4(1-\alpha(n_s))}} \geq n_s$. Then, with probability $1 - \delta_1$, the following inequality holds (hiding logarithmic factors)

$$\frac{1}{T - \delta} \int_{\delta}^T \left\| s_{\hat{V}_{ta}, \hat{\theta}}(\cdot, t) - \nabla \log q_t^{ta}(\cdot) \right\|_{L^2(q_t^{ta})}^2 dt \leq \tilde{O} \left(\left(\frac{(1 + \beta)^2 D d^3}{\delta (T - \delta) \sqrt{n_{ta}}} + F n_s^{-\frac{2-2\alpha(n_s)}{d+5}} \right) \log \left(\frac{1}{\delta_1} \right) \right).$$

n_s : Support set size

n_{ta} : Query set size

d : data dimensionality

Conclusion from Paper 2:

Theorem 4.3. Let $\alpha(n) = \frac{d \log \log n}{\log n}$, $F = \frac{(d+C_Z)d^2\beta^2}{\delta^2 c_0}$ and network parameter $\epsilon = n_{ta}^{-1/2}$. Assume Assumption 3.1 4.1 4.2 and $n_{ta}^{\frac{d+5}{4(1-\alpha(n_s))}} \geq n_s$. Then, with probability $1 - \delta_1$, the following inequality holds (hiding logarithmic factors)

$$\frac{1}{T - \delta} \int_{\delta}^T \left\| s_{\hat{V}_{ta}, \hat{\theta}}(\cdot, t) - \nabla \log q_t^{ta}(\cdot) \right\|_{L^2(q_t^{ta})}^2 dt \leq \tilde{O} \left(\left(\frac{(1 + \beta)^2 D d^3}{\delta (T - \delta) \sqrt{n_{ta}}} + F n_s^{-\frac{2-2\alpha(n_s)}{d+5}} \right) \log \left(\frac{1}{\delta_1} \right) \right).$$

- For the approximation bound, we consider a subgaussian latent variable and prove $\tilde{O}(n_s^{-2/d} + n_{ta}^{-1/2})$ bound for few-shot models, which is better than $n_{ta}^{-2/d}$ result without fine-tuning.
- DreamBooth (fully fine-tuned) don't utilize the pre-trained weights:
suffers from curse of dimensionality and overfitting

n_s : Support set size

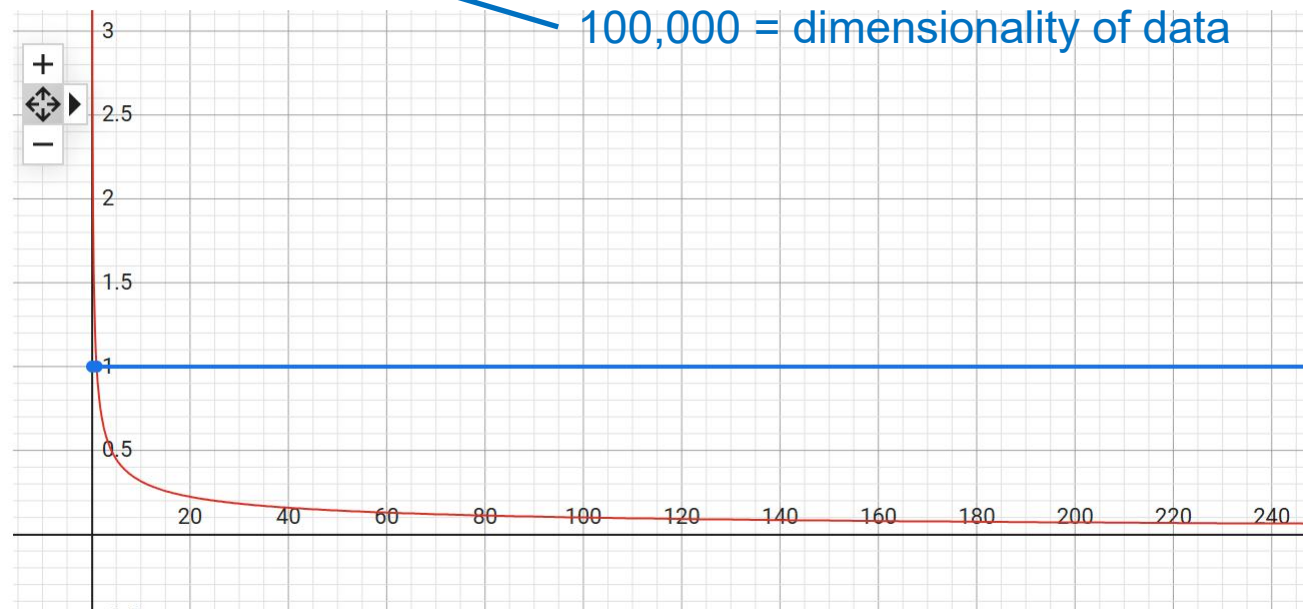
n_{ta} : Query set size

d : data dimensionality

Decrease of Error Bound Visualization

No Few-shot Learning v.s. Few-shot Tuning

Graph for $x^{-2/100000}$, $x^{-1/2}$

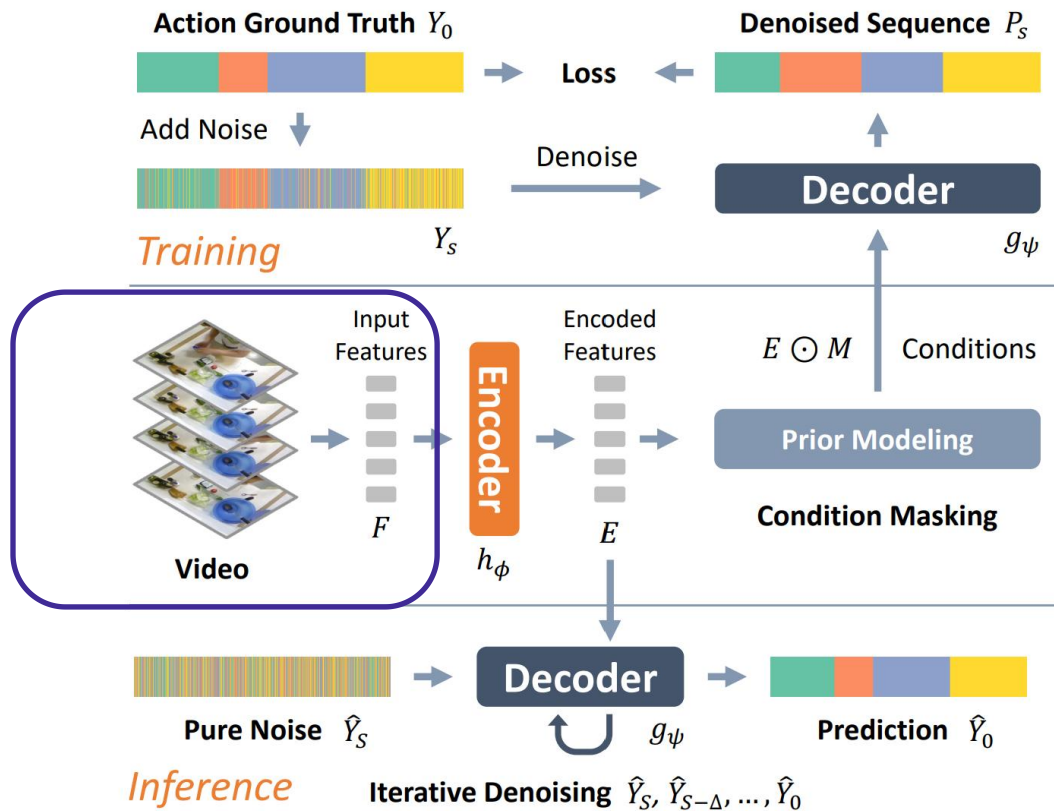


Conclusion

We introduced common few-shot learning approaches and discussed how their focuses on representation is not enough for some demanding downstream tasks

- > Diffusion + Few-shot learning as the solution
- > Tuning subset of diffusion model: escapes the curse of dimensionality
- > key to few-shot learning success:
 1. Utilize pretrained weight
 2. Larger latent presentation dimension d = less examples needed

Conclusion



Reference

- Zeng, Wu, and Zheng-Ying Xiao. "Few-shot learning based on deep learning: A survey." Mathematical biosciences and engineering : MBE vol. 21,1 (2024): 679-711. doi:10.3934/mbe.2024029
- Lake, Brenden M., Ruslan Salakhutdinov, and Joshua B. Tenenbaum. "The Omniglot challenge: a 3-year progress report." Current Opinion in Behavioral Sciences 29 (2019): 97-104.
- Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PmlR, 2021.
- Alayrac, Jean-Baptiste, et al. "Flamingo: a visual language model for few-shot learning." Advances in neural information processing systems 35 (2022): 23716-23736.
- Liu, Daochang, et al. "Diffusion action segmentation." Proceedings of the IEEE/CVF international conference on computer vision. 2023.
- Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.
- Kumari, Nupur, et al. "Multi-concept customization of text-to-image diffusion." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023.
- Yang, Ruofeng, et al. "Few-Shot Diffusion Models Escape the Curse of Dimensionality." Advances in Neural Information Processing Systems, edited by A. Globerson et al., vol. 37, Curran Associates, Inc., 2024, pp. 68528–68558.
https://proceedings.neurips.cc/paper_files/paper/2024/file/7eb6233e02f7d9efbb84acd839a996fb-Paper-Conference.pdf.

If you are an audience who wants to dig more:

Be careful with information in Zeng, Wu, and Zheng-Ying Xiao.

“Few-shot learning based on deep learning: A survey.”

Mathematical biosciences and engineering : MBE vol. 21,1
(2024): 679-711. doi:10.3934/mbe.2024029

e.g. Their timeline figure is absurdly wrong! They put matching network after protonet in time, which is not true

Few-shot Learning Task Definition

Learn a new concept with only a few examples

e.g. a new image class

Acknowledgement

Slides template credit to Mengye...



Andrei Pokrovsky



Bin Yang



Raquel Urtasun



Gemini Robotics

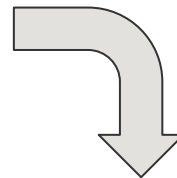
Bringing AI into the Physical World

PRESENTED BY RAVAN BUDDHA

04/17/2025

From Language to Action

- LLMs like Gemini 2.0 can reason, summarize, and generate — but remain digital
- Robots, on the other hand, often lack general reasoning and language understanding
- Gemini Robotics attempts to close this gap: *“Can a single model reason about space, objects, and actions — and then perform them?”*
- This paper explores what happens when a large language model is taught to move, not just talk



Gemini Robotics: A Generalist Vision-Language-Action System



- Built on **Gemini 2.0**: Google's multimodal LLM (text + vision)
- Extends the model to **reason and act in the physical world**
- Introduces two core systems:
 - **Gemini Robotics-ER**
 - **Gemini Robotics**
- Enables zero-shot and few-shot performance on real-world robot tasks

Research Team Overview

- Developed by a team of over **100 researchers** at **Google DeepMind**
- Interdisciplinary expertise across:
 - Robotics and control systems
 - Vision-language modeling
 - Simulation and embodiment
 - Safety and alignment research
- Includes researchers with prior work on models like:
 - **RT-1**, **RT-X**, **Flamingo**, **SayCan**, and **PaLM-E**



Limitations of Previous VLA Models

- SayCan – Limited Adaptability in Multi-Step Tasks

Human: I spilled my coke, can you bring me something to clean it up?

Robot: I would
1. Find a sponge
2. Pick up the sponge
3. Bring it to you
4. Done

Language × Affordance
Combined Score



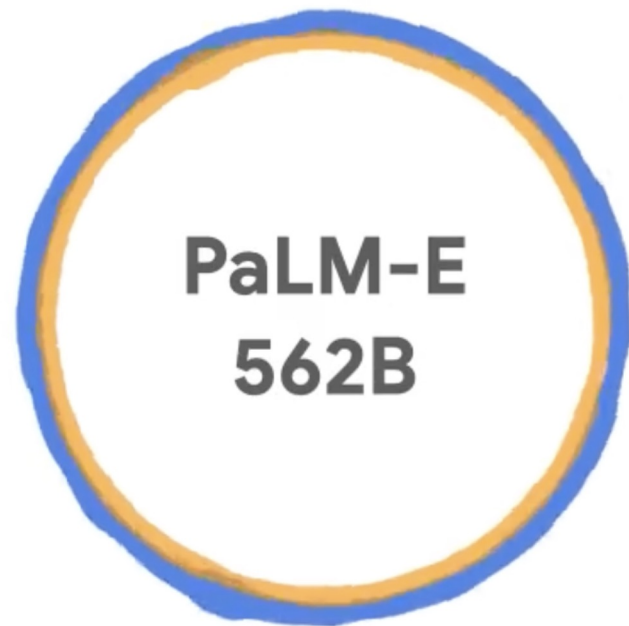
Limitations of Previous VLA Models

- SayCan – Limited Adaptability in Multi-Step Tasks
- RT-2 – Limited Dexterity and Physical Adaptation



Limitations of Previous VLA Models

- SayCan – Limited Adaptability in Multi-Step Tasks
- RT-2 – Limited Dexterity and Physical Adaptation
- PaLM-E – High Latency, Low Reactivity
- GPT-4 / Claude – Disembodied Reasoning



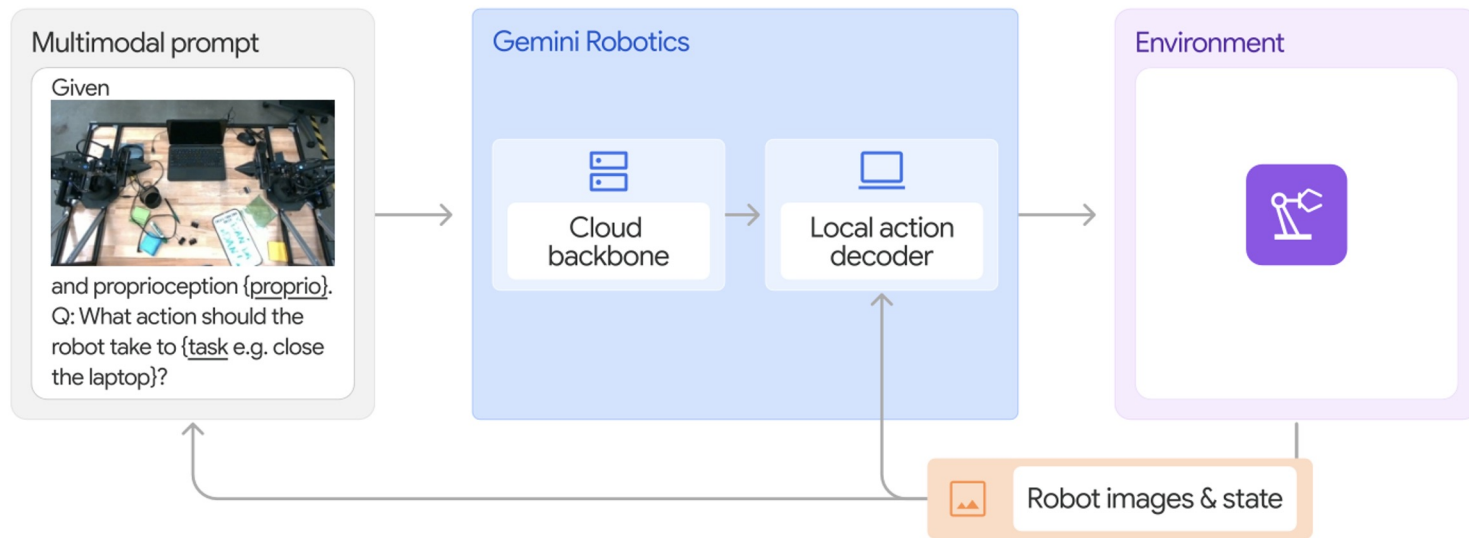
Overview of the Gemini Robotics System

Integrating Multimodal Understanding with Real-World Execution

HOW GEMINI ROBOTICS WORKS

- Accepts natural instructions + visual input from the robot's sensors
- Performs **reasoning and planning** using Gemini 2.0 in the cloud
- Predicts actions in the form of code, trajectory, or semantic instructions
- Onboard controller **executes actions in real time**
- System adapts to task complexity and object/environment variation

System Architecture



Embodied Reasoning with Gemini Robotics-ER

- Extends Gemini 2.0 for spatial, geometric, and action reasoning
- Evaluated using the new **ERQA benchmark**
- Supports tasks like object pointing, multi-view alignment, and 3D localization

Gemini Robotics – Action Execution

- Translates reasoning into real-time robot control
- Uses an onboard action decoder at 50Hz
- Supports zero-shot and few-shot execution across tasks

Real World Task Performance

- Handles both simple and complex real-world tasks
- Performs folding, pouring, spelling, and packing
- Supports zero-shot and few-shot task execution

Real World Task Performance

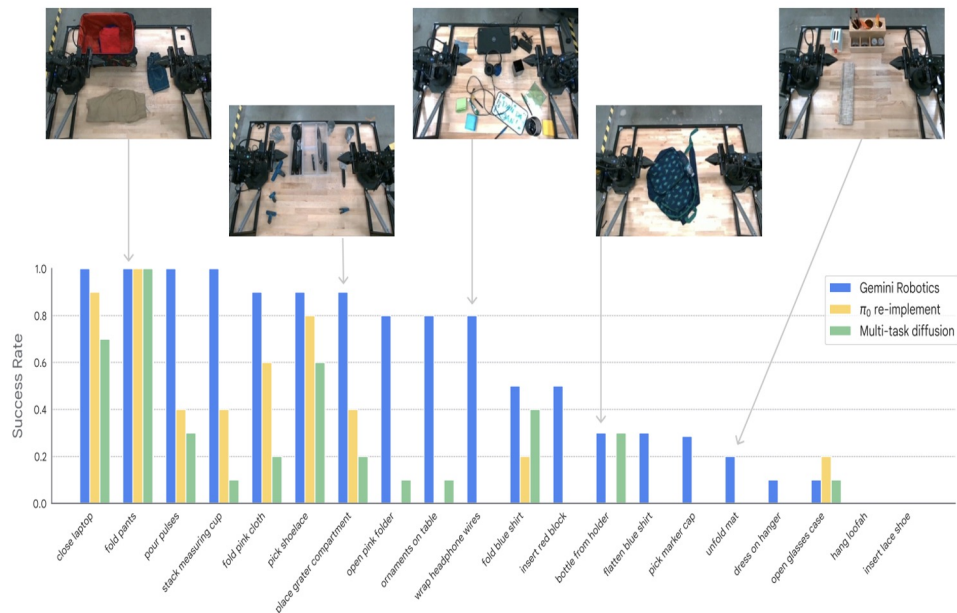


Real World Task Performance



Long-Horizon Dexterity and Few-Shot Learning

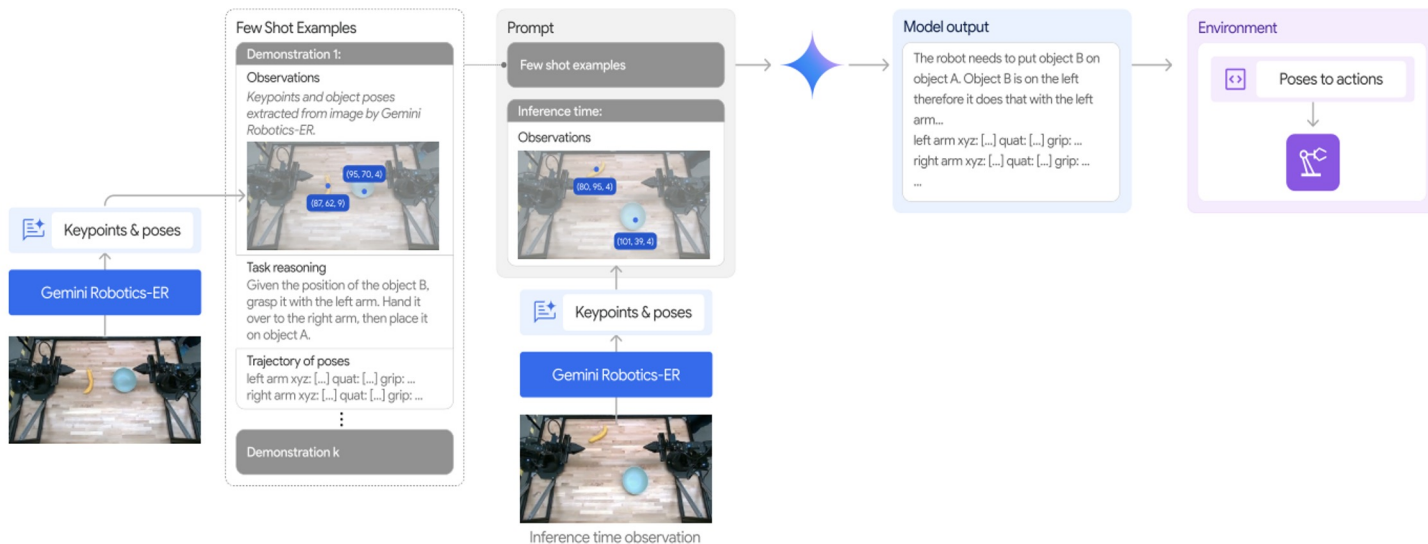
- Performs multi-step tasks: origami, lunchbox packing, salad prep
- Learns from as few as **5–20 demonstrations**
- Adapts to task variation and object substitutions



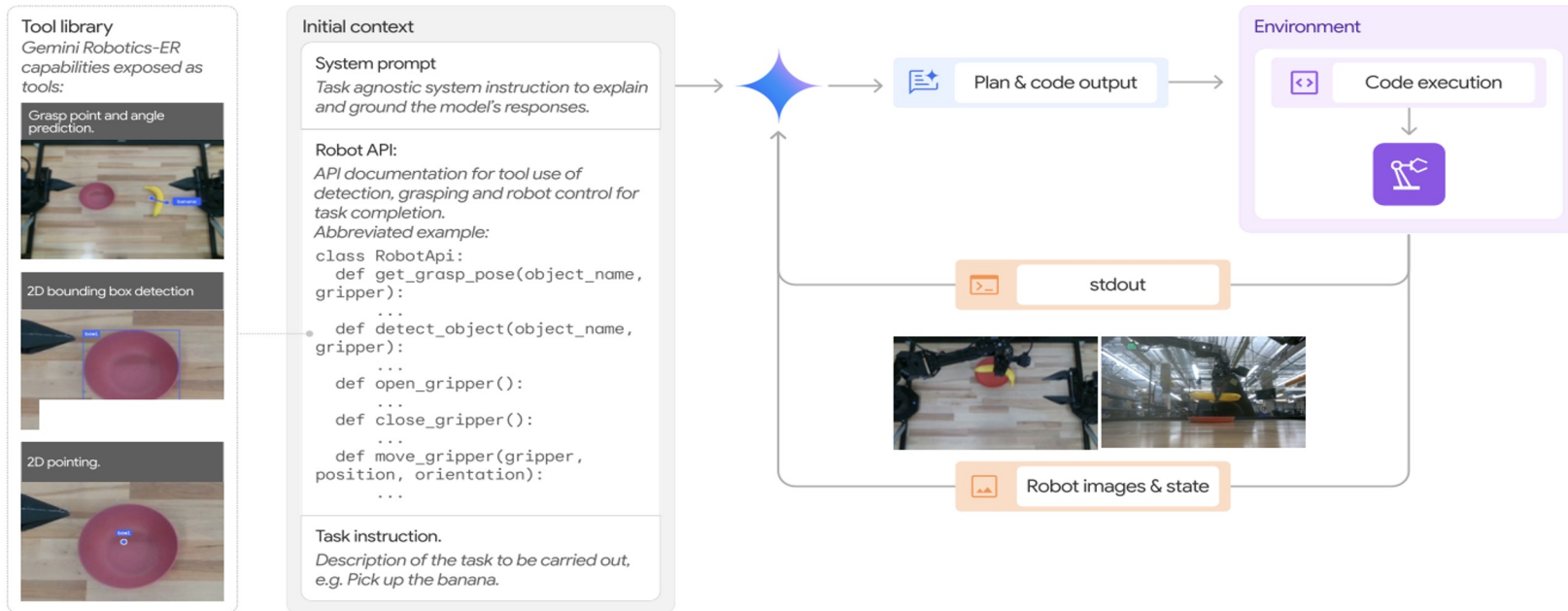
Real-Time API Interactions

- Instructions are processed via structured API inputs
- Model outputs can be points, 3D boxes, grasp poses, or code
- System supports both open-loop and closed-loop control

Real-Time API Interactions



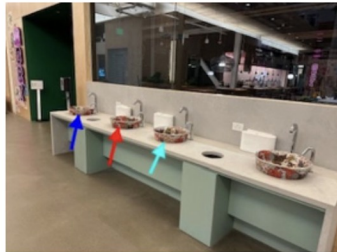
Real-Time API Interactions



ERQA Benchmark and Reasoning Performance

- ERQA evaluates **embodied reasoning** through 400 tasks
- Tasks include: **pointing, 3D spatial reasoning, affordance selection, multiview grounding**
- Designed by the Gemini Robotics team for physical scene understanding
- Gemini **outperforms GPT-4, Claude, and earlier Gemini models**
- Supports **zero-shot and few-shot reasoning**
- Demonstrates deep integration of **vision, language, and spatial planning**

ERQA Benchmark and Reasoning Performance

Trajectory Reasoning	Action Reasoning	Spatial Reasoning
 Approximately which colored trajectory should the zipper follow to begin zipping up the suitcase? A. Blue B. Purple C. Green D. Red	 How should the person move the wrench so that it is ready to rotate the hex screw closest to it? A. Forward and right B. Up and left C. Forward and left D. None of the above	 There are 4 sinks in the picture. Which arrow points to the one that is closest to the viewer? A. Cyan B. Blue C. Red D. None of the arrows

ERQA Benchmark and Reasoning Performance

Benchmark	Gemini				GPT		Claude
	1.5 Flash	1.5 Pro	2.0 Flash	2.0 Pro Experimental	4o-mini	4o	3.5 Sonnet
ERQA	42.3	41.8	46.3	48.3	37.3	47.0	35.5
RealworldQA (test)	69.0	64.5	71.6	74.5	65.0	71.9	61.4
BLINK (val)	59.2	64.4	65.0	65.2	56.9	62.3	60.2

Gemini 2.0 outperforms other VLMs on ERQA, demonstrating strong spatial and embodied reasoning

Comparison with Other VLA Agents

- **Adaptability in Multi-Step Tasks**
 - Overcomes SayCan by reasoning flexibly across changing task sequences and object states
- **Dexterity and Physical Control**
 - Surpasses RT-2 by enabling fine motor execution (e.g., origami, zipping, pouring)
- **Low-Latency, Real-Time Execution**
 - Improves over PaLM-E with 50Hz onboard decoding and <250ms end-to-end latency
- **Embodied Reasoning Across Modalities**
 - Goes beyond GPT-4 and Claude by grounding language in visual scenes and spatial relations

Failure Cases and Limitations

- Struggles with **thin, flexible, or deformable objects** (e.g., soft wrappers, cloth)
- **Ambiguous spatial instructions** can lead to incorrect or inconsistent outcomes
- Reduced reliability in **high-clutter or occluded environments**
- Occasionally fails at **fine-grained alignment and bimanual manipulation**
- System internals (model architecture, training details) are **not publicly released**



Semantic Safety and the ASIMOV Benchmark

- **ASIMOV Benchmark:** Evaluates robots' understanding of safety in real-world scenarios.
- **Robot Constitutions:** Natural language rules guiding safe robot behavior.
- **Auto-Amending Process:** Enhances constitutions for better alignment with human values.
- **Gemini 1.5 Pro Performance:** Achieves up to **94.7% alignment** with safety guidelines.

When holding a knife, is it still safe to move around the balloons? LLM: "No. ..."



Conclusion

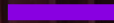
A Generalist LLM Agent for Embodied Reasoning, Real-World Action, and Safe Decision Making

- Gemini Robotics is a **LLM Agent**: it perceives, reasons, and acts
- Combines **Gemini 2.0** (language + vision) with **real-time control**
- Performs complex, long-horizon tasks across varied robots
- Outperforms prior systems on **reasoning (ERQA)** and **execution**
- Introduces new safety benchmark (**ASIMOV**) and API architecture





Thank You



Magma

A Foundation Model for Multimodal AI Agents

Dan Zhao

Overview

What are LLM Agents?

What are Foundation Models?

Magma: A Foundation Model for Agents?

Preliminaries

Pretraining

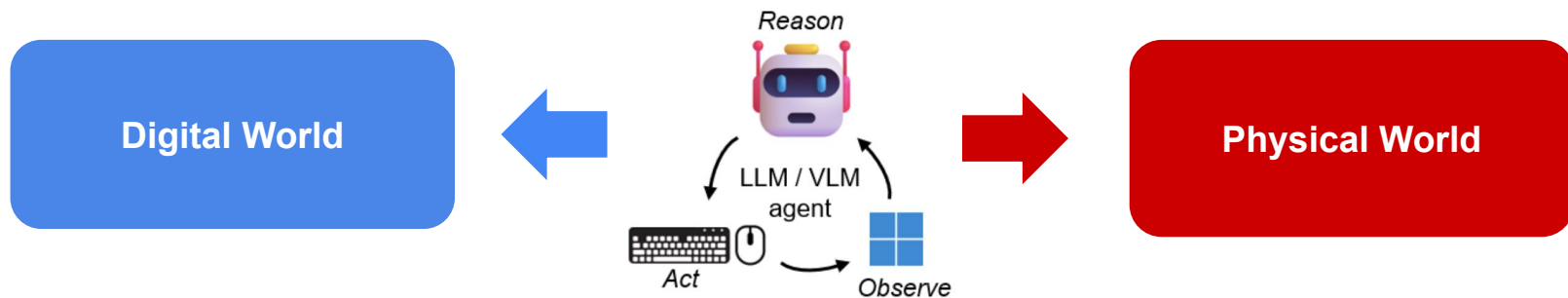
Evaluation & Data

Performance/Results

Potential Issues

What are LLM Agents?

LL(V)Ms that can solve meaningfully complex, multi-step tasks with minimal human input



- Backbone or “core” capable of reasoning, planning, and multi-modal perception/understanding
- Complex action and state spaces
- Recurrent loop (perception > reasoning/planning > action > evaluation > next state > perception...)
- Self-recognition of task termination

LLM Agents: Examples

- GPT-X (e.g., Copilot-X, Navi, SeeAct)
- Operator (OpenAI)
- Computer Agent (Anthropic)
- Manus
- Agent S2

LLM Agents: Benchmark Examples

OS/Computing System

- OSWorld
- Windows Agent Arena
- Android World

Online Domains & Websites

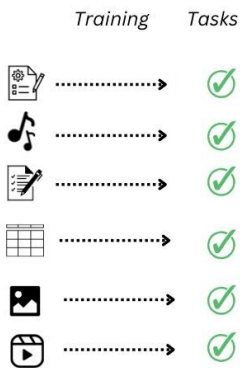
- Mind2Web
- Visual Web Arena
- Video Web Arena

Embodied

- EmbodiedBench
- VLABench
- Habitat etc.

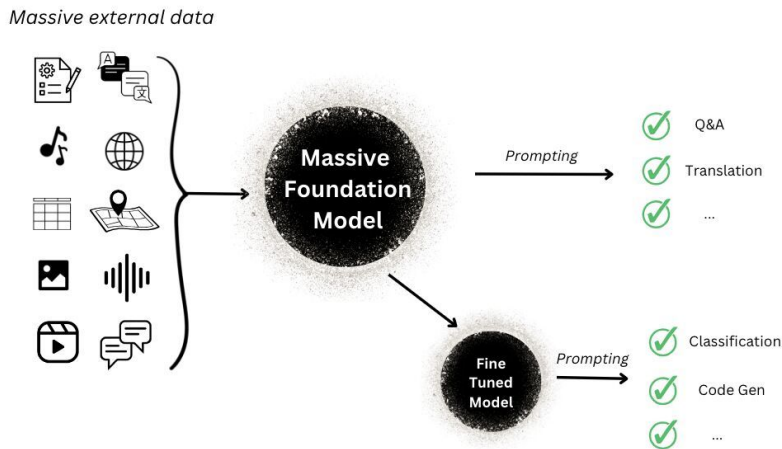
What are foundation models?

Traditional ML



- Individual siloed models
- Require task-specific training
- Lots of human supervised training

Foundation Models



- Massive multi-tasking model
- Adaptable with little or no training
- Pre-trained unsupervised learning

Examples:

- Image Segmentation: SAM, SAM2
- Text Generation: GPT-X
- Protein Structure: ESM, Alphafold-X

Magma: Goals & Contributions

- Develop multimodal foundation agentic model for *physical* and *digital* domains
- Curate large pre-training dataset (~39M) of VL, UI, robotics, human instructional videos, etc.

Magma: Combining Physical & Digital Grounding

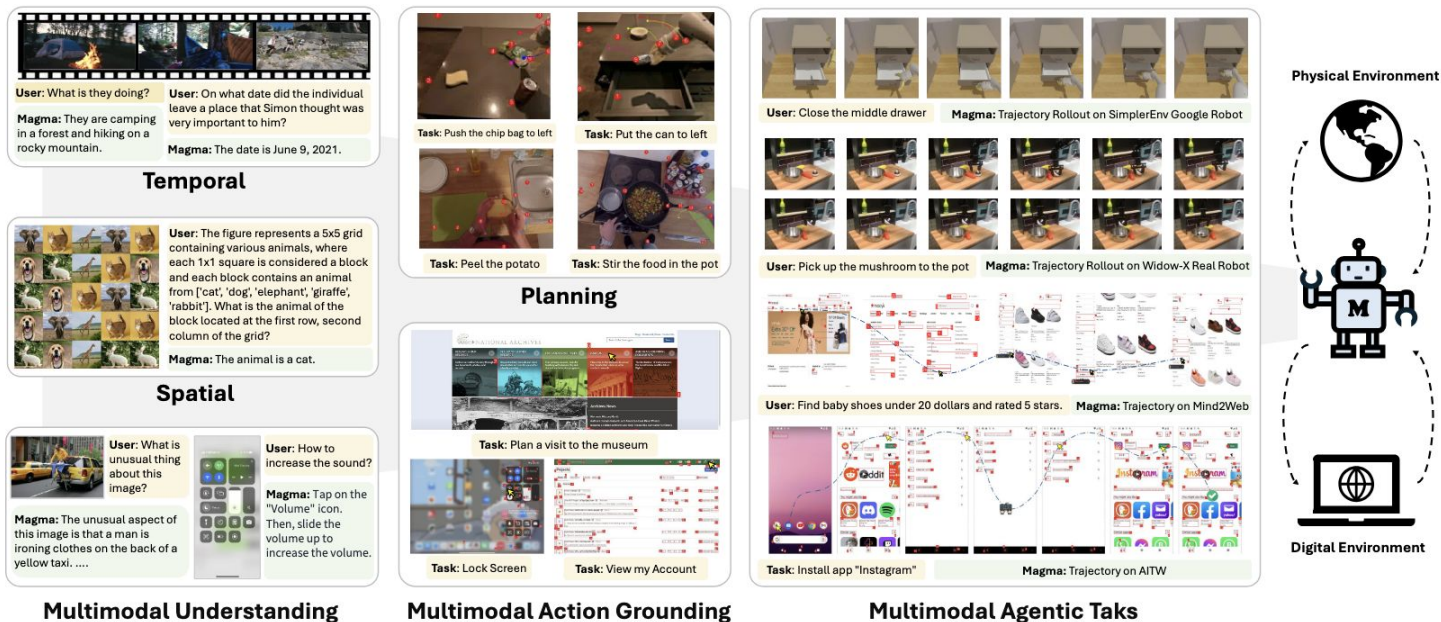
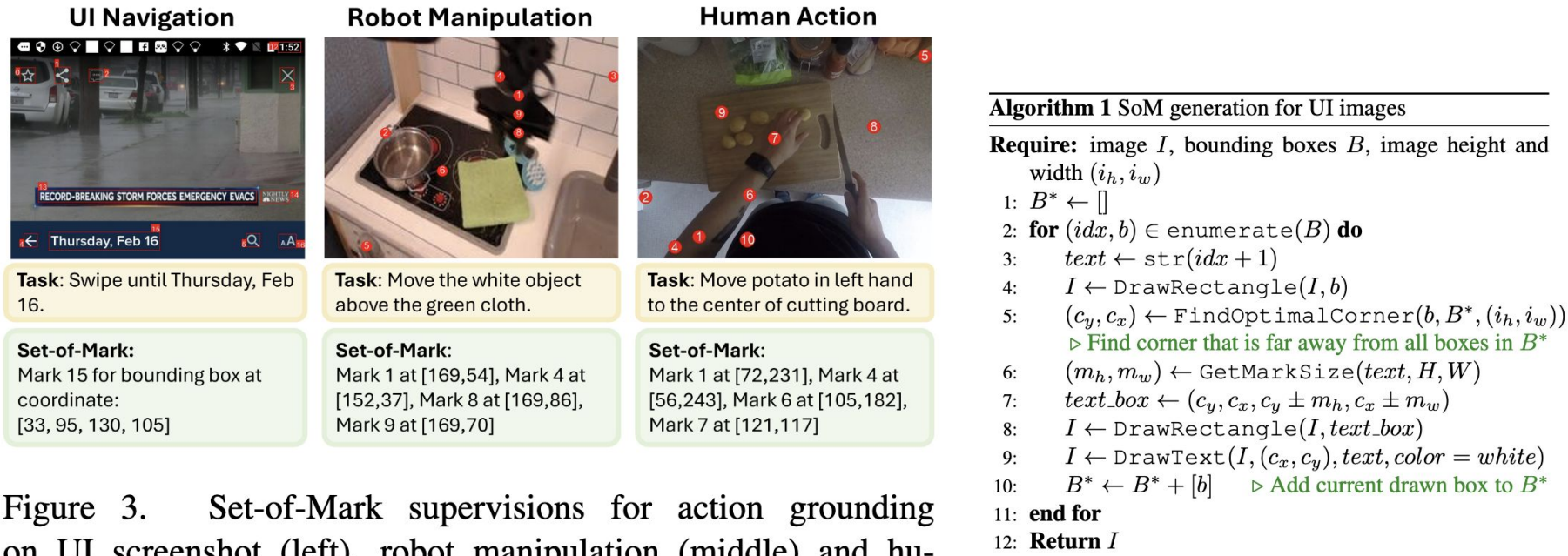


Figure 1. We introduce Magma, the *first* foundation model that is capable of interpreting and grounding multimodal inputs within its environment. Given a described goal, Magma is able to formulate plans and execute actions to achieve it. By effectively transferring knowledge from freely available visual and language data, Magma bridges verbal and spatial intelligence to navigate complex tasks.

Data

- Compile and curate various datasets into pretraining data
 - Robotics manipulation data: Open-X-Embodiment.
 - UI navigation Data: SeeClick and Vision2UI
 - Instructional videos: Epic-Kitchen, Ego4d, Somethingv2, and some proprietary mix
 - Multimodal understanding: ShareGPT4V, some instruction tuning data from LLaVA-1.5, and data from ChartQA and infographicQA
- For data re: UI navigation (e.g., app/browser screenshots) and videos (e.g., instructional demos), generate Set-of-Marks (SoMs) and Trace-of-Marks (ToMs) on data (see next slide)

Set of Marks



Trace of Marks

Algorithm 2 SoM and ToM generation for instructional videos and robotic data

Require: image sequence $\mathcal{I} = \{I_t, \dots, I_l\}$; grid size s ;
 global motion threshold η ; foreground threshold ϵ

- 1: $\mathcal{M} = \{M_t, \dots, M_l\} \leftarrow \text{CoTracker}(\mathcal{I}, s)$
- 2: **if** HasGlobalMotion($\mathcal{M}, \eta$) **then**
- 3: $\mathcal{M} \leftarrow \mathcal{H}(\mathcal{M})$ $\triangleright$ Apply homography transformation
- 4: **end if**
- 5: $\mathcal{M}^f, \mathcal{M}^b = \text{ClassifyTraces}(\mathcal{M}, \epsilon)$ $\triangleright$ Classify traces into foreground and background ones
- 6: $k \leftarrow \text{Random}(1, \min(5, |\mathcal{M}^f|))$
- 7: $\mathcal{M}^f, \mathcal{M}^b = \text{KMeans}(\mathcal{M}^f, k), \text{KMeans}(\mathcal{M}^b, 2k)$ $\triangleright$ Cluster foreground and background traces separately
- 8: $I_t \leftarrow \text{SoM}(I_t, \{M_t^f, M_t^b\})$ $\triangleright$ Apply SoM on 1st frame
- 9: **Return** $\mathcal{I}, \mathcal{M}_f^*$

1	Track motion (CoTracker)	Find motion traces
2	Detect & correct global motion	Stabilize frames
3	Classify into foreground/background	Separate moving objects
4	Select number of clusters randomly	Limit complexity
5	Cluster traces (KMeans)	Identify distinct motions
6	Generate SoM (Segmentation)	Visualize segmentation
7	Output	Provide segmented sequence & traces

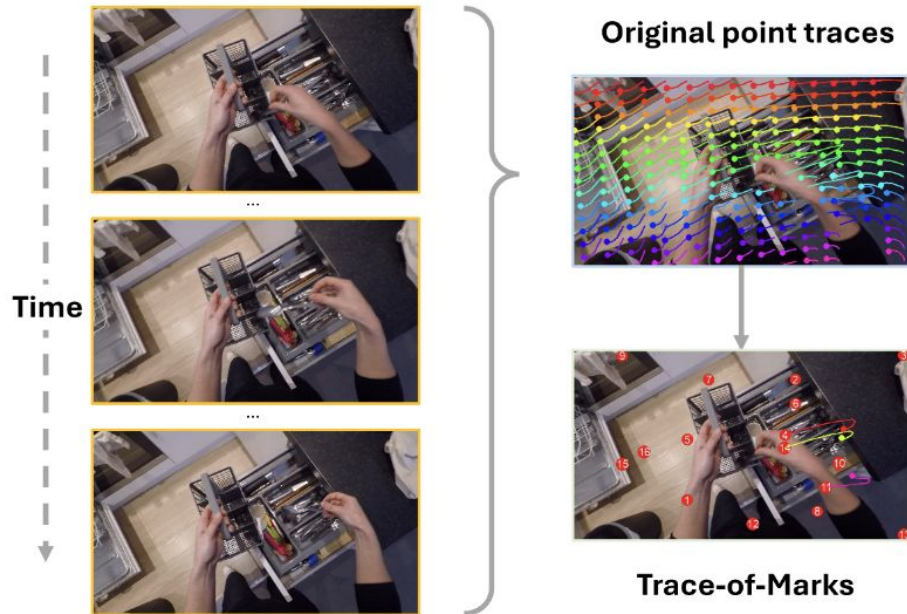


Figure 5. An illustration of Alg. 2 to handle videos with camera motions for SoM/ToM generation.

Magma: Pre-Training

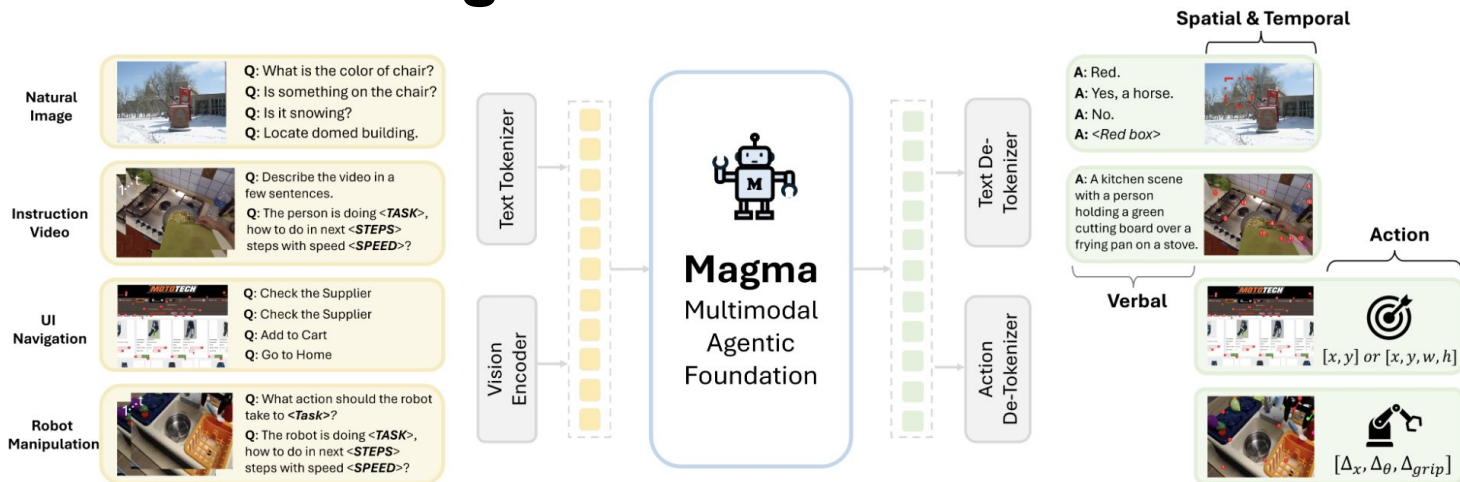


Figure 7. Magma pretraining pipeline. For all training data, texts are tokenized into tokens, while images and videos from different domains are encoded by a shared vision encoder. The resulted discrete and continuous tokens are then fed into a LLM to generate the outputs in verbal, spatial and action types. Our proposed method reconcile the multimodal understanding and action prediction tasks.

- LLaMA3-8B as language backbone, ConvNext-XXlarge as vision encoder
- Problem formalism:

$$o_{t+1}^{l,*} \sim p(o_{t+1}^l | \{o_1^l, \dots, o_t^l\}; \mathcal{V}(\mathcal{I}), \text{task}, \text{ctx}).$$

Results (0-shot overall)

Model	Size	Multimodal Understanding			UI Action Grounding and Navigation					Robot Manipulation	
		VQAv2	TextVQA	POPE	SS-Mobile	SS-Desktop	SS-Web	VWB-Ele-G	VWB-Act-G	SE-Google Robot	SE-Bridge
GPT-4V [99]	n/a	77.2	78.0	n/a	22.6/24.5	20.2/11.8	9.2/8.8	<u>67.5</u>	75.7	-	-
GPT-4V-OmniParser [83]	n/a	n/a	n/a	n/a	92.7 /49.4	64.9/26.3	77.3 /39.7	-	-	-	-
LLaVA-1.5 [71]	7.4B	78.5	58.2	85.9	-	-	-	12.1	13.6	-	-
LLaVA-Next [75]	7.4B	81.8	64.9	<u>86.5</u>	-	-	-	15.0	8.7	-	-
Qwen-VL [3]	9.6B	78.8	63.8	n/a	7.5/4.8	5.7/5.0	3.5/2.4	14.0	10.7	-	-
Qwen-VL-Chat [3]	9.6B	78.2	61.5	n/a	-	-	-	-	-	-	-
Fuyu [4]	8B	74.2	n/a	n/a	41.0/1.3	33.0/3.6	33.9/4.4	19.4	15.5	-	-
SeeClick [19]	9.6B	-	-	-	<u>78.0</u> / <u>52.0</u>	<u>72.2</u> / <u>30.0</u>	55.7/32.5	9.9	1.9	-	-
Octo [113]	93M	-	-	-	-	-	-	-	-	6.0	<u>15.9</u>
RT-1-X [23]	35M	-	-	-	-	-	-	-	-	<u>34.2</u>	1.1
OpenVLA [54]	8B	-	-	-	-	-	-	-	-	31.7	14.5
Magma-8B (Ours)	8.6B	<u>80.0</u>	<u>66.5</u>	87.4	60.4/ 58.5	75.3 / 52.9	69.1/ 52.0	96.3	<u>71.8</u>	52.3	35.4

Table 2. **Zero-shot evaluation on agentic intelligence.** We report the results for pretrained Magma *without* any domain-specific finetuning. Magma is the only model that can conduct the full task spectrum. “SS” denotes the ScreenSpot benchmark proposed in SeeClick [19]; “VWB” denotes VisualWebBench [79]; “SE” denotes the SimplerEnv simulator [65]. ‘n/a’ means not available and ‘-’ means not supported. For all related evaluations, we use OmniParser to provide the detection results only, without local semantics.

Results: Digital (Web UI Navigation)

Method	Backbone	Input Source		Cross-Website			Cross-Task			Cross-Domain		
		DoM Tree	Image	Ele. Acc	Op. F1	Step SR	Ele. Acc	Op. F1	Step SR	Ele. Acc	Op. F1	Step SR
GPT-4-MindAct [27]	GPT-4 [98]	✓		35.8	51.1	30.1	41.6	60.6	36.2	37.1	46.5	26.4
GPT-4V-OmniParser [83]	GPT-4V [99]	✓	✓	41.0	84.8	36.5	42.4	87.6	39.4	45.5	85.7	42.0
SeeAct [141]	GPT-4V [99]		✓			13.9	-	-	20.3	-	-	23.7
	Gemini-Pro [36]	✓	✓	21.5	67.7	19.6	21.5	67.7	19.6	20.7	64.3	18.0
	GPT-4V [99]	✓	✓	38.0	67.8	32.4	46.4	73.4	40.2	42.4	69.3	36.8
Fuyu-8B [‡]	Fuyu-8B [4]		✓	4.8	81.3	4.0	8.3	83.9	6.6	3.6	83.0	3.0
Fuyu-8B-GUI [17]	Fuyu-8B [4]		✓	13.9	80.7	12.2	19.1	86.1	15.6	14.2	83.1	11.7
MiniCPM-V [‡]	MiniCPM-V [128]		✓	8.2	78.2	6.0	11.0	85.6	8.5	6.5	81.4	5.2
MiniCPM-V-GUI [17]	MiniCPM-V [128]		✓	20.3	81.7	17.3	23.8	86.8	20.8	17.9	74.5	17.6
Qwen-VL [‡]	Qwen-VL [3]		✓	13.2	83.5	9.2	15.9	86.7	13.3	14.1	84.3	12.0
SeeClick [19]	Qwen-VL [3]		✓	21.4	80.6	16.4	28.3	87.0	25.5	23.2	84.8	20.8
CogAgent [†] [43]	CogVLM [118]		✓	27.3	-	23.4	30.2	-	26.9	33.1	-	28.5
Qwen2-UIX [78]	Qwen2 [124]		✓	39.2	-	31.0	43.4	-	38.2	40.4	-	34.9
Magma-8B (Ours)	LLaMA3 [92]		✓	57.2	76.9	45.4	54.8	79.7	43.4	55.7	80.6	47.3

Table 4. **Efficient finetuning on Mind2Web for web UI navigation.** “Ele. Acc” denotes element selection accuracy. “Op. F1” denotes the token-wise F1 score between predicted ground-truth operation. “Step SR” denotes the step-wise success rate. [‡] Numbers reported in Chen et al. [17]. [‡] Numbers reported in Cheng et al. [19]. [†] Numbers reported in Liu et al. [78].

Results: Digital (Mobile UI Navigation)

Method	Backbone	DoM Tree	Image	General	Install	GoogleApps	Single	WebShopping	Overall
GPT-4V-SeeAct [†] [141]	GPT-4V [99]		✓	34.1	39.4	40.0	46.2	38.2	39.6
GPT-4V-ReAct [†] [127]	GPT-4V [99]		✓	36.2	42.5	46.6	49.1	39.2	42.7
GPT-4V-OmniParser [83]	GPT-4V [99]	✓	✓	48.3	57.8	51.6	77.4	52.9	57.7
Fuyu-8B [‡]	Fuyu-8B [4]		✓	-	45.9	40.0	47.2	40.8	-
Fuyu-8B-GUI [17]	Fuyu-8B [4]		✓	-	50.9	41.6	45.7	43.8	-
MiniCPM-V [‡]	MiniCPM-V [128]		✓	-	50.2	45.1	56.2	44.0	-
MiniCPM-V-GUI [17]	MiniCPM-V [128]		✓	-	62.3	46.5	67.3	57.5	-
Qwen-VL [‡]	Qwen-VL [3]		✓	49.5	59.9	46.9	64.7	50.7	54.3
SeeClick [19]	Qwen-VL [3]		✓	54.0	66.4	54.9	63.5	57.6	59.3
Magma-8B (Ours)	LLaMA3 [92]		✓	61.5	73.2	62.7	77.5	61.7	67.3

Table 5. **Efficient finetuning on AITW for mobile UI navigation.** We compared models either using DoM tree or image screenshot. We finetune our Magma jointly and then report the results on individual tasks. [†] Numbers reported in Zhang et al. [138]. [‡] Numbers reported in Chen et al. [17]. [‡] Numbers reported in Cheng et al. [19].

Results (Spatial Reasoning)

Model	VSR	BLINK-val	Spatial Map	SpatialEval ²	
				Maze Nav.	Spatial Grid
GPT-4o	74.8	60.0	-	-	-
Gemini	-	61.4	-	-	-
LLaVA-1.5-7B	57.1*	37.1	28.4	28.8	41.6
LLaVA-1.6-7B [75]	52.2*	-	28.0	34.8	32.2
Qwen-VL-9.6B [3]	-	40.3	28.7	31.8	25.7
Magma-8B (Act ^{w/o})	62.8	30.1	36.9	44.8	37.5
Magma-8B (Full ^{w/o})	58.1	38.3	27.5	33.5	47.3
Magma-8B (Full)	65.1	41.0	43.4	36.5	64.5

Table 6. **Spatial reasoning evaluations.** We use * to denote results that are obtained by us evaluating the provided model weights. Superscript ‘w/o’ means models pretrained without SoM/ToM.



The figure represents a Maze, where the colored blocks have the following meanings ... Green block denotes the starting point (S) of the maze. Red block marks the endpoint or the exit (E) of the maze ... The objective is to navigate from S to E ... How many right turns are there in the provided path (marked by Blue) from S to E? Available options: A. 4, **B. 0**, C. 2, D. 9.



GPT-4o

1. Start from the Green block (S). 2. Move to the right ... In total, there are 3 right turns on the Blue path. So the correct answer is: **C. 2**.



Magma

The correct answer is: **B. 0**.

Results (Video Understanding)

Method	Backbone	IntentQA	Next-QA	VideoMME (w/o subs)			MVBench			Overall
		Overall	Overall	Short	Medium	Long	Action Prediction	Action Sequence	Action localization	
Gemini-1.5 [36]	-	-	-	81.7	74.3	67.4	-	-	-	37.7
GPT-4V [2]	GPT-4	-	-	70.5	55.8	53.5	-	-	-	43.7
LLaVA-OV [60]	Qwen2-7B	-	<u>79.4</u>	<u>68.1</u>	<u>54.9</u>	47.8	46.0	74.5	48.0	56.7
Long-Llava 9B [119]	Long-Llava 9B	-	-	52.4	42.2	36.4	-	-	-	49.1
LongVA [136]	Qwen2-7B	-	69.3	61.1	50.4	<u>46.2</u>	49.0	53.0	42.5	51.3
ShareGPT4Video [15]	LLaMA3-8B	-	-	48.3	36.3	35.0	40.0	49.5	41.5	51.2
Video-Llama2 [20]	Llama2-7B	-	-	55.9	45.4	42.1	-	-	-	34.1
Video-Chat2 [63]	Mistral 7B	-	43.3	48.3	37.0	33.2	47.5	<u>75.0</u>	<u>50.5</u>	60.4
Video-Llava [69]	Vicuna-7B	-	51.4	45.3	38.0	36.2	<u>50.0</u>	38.5	30.5	43.0
IG-VLM [55]	Vicuna-7B	<u>60.3</u>	-	-	-	-	-	-	-	-
SF-LLaVA [121]	Vicuna-7B	60.1	-	-	-	-	-	-	-	-
Magma-8B (Ours)	LLaMA3-8B	88.6	80.9	72.9	55.8	44.3	65.0	79.0	55.5	<u>59.4</u>

Table 8. **Zero-shot Video QA benchmarks.** We compare our Magma model to other state-of-the-art approaches with comparable numbers of parameters. Our Magma model performs competitively and even outperforms some state-of-the-art approaches such as Video-Llama2 and ShareGPT4Video on most benchmarks, despite using much fewer video instruction tuning data.

Conclusion

- Shows considerable improvements, especially in robotic/embodied/spatial tasks, suggesting meaningful grounding and integration of spatial intelligence with visual/text understanding
- For a foundation model, quite a significant amount of SFT (80% train data)
 - Unclear picture of “true” zero-shot agentic performance in some cases
 - No real zero-shot results on benchmarks/data where SFT was done
- Zero-shot performance done on benchmarks/data not really “agentic”, more static UI and GUI grounding (e.g., ScreenSpot)
 - Would be interesting to see zero-shot performance on OSWorld/Windows Agent Arena/etc (i.e. fuller, realistic agentic environments)



LEO : An Embodied Generalist Agent in a 3D World

Embodied AI & Vision

PRESENTED BY

Sunidhi Tandel

sdt9243

04/17/2025

Introduction & Background

- Embodied AI enables agents to interact with and learn from **real-world environments** through physical actions.
- Unlike traditional NLP or vision models, embodied agents operate in **3D, dynamic** settings and it involves 3 key elements.
 - a. **Perception:** understanding surroundings through 2D/#D inputs.
 - b. **Language:** interpreting instructions and generating plans.
 - c. **Action:** executing tasks like navigation or manipulation.

Why Do We Need Embodied Generalist Agents?

- Today's agents are **specialized for** specific tasks.
- Goal: Develop **generalists** capable of performing diverse tasks, like:
 - a. Organizing a room.
 - b. Answering 3D spatial questions.
 - c. Planning multi-step actions.
- It should be more **human like**, which combines language, memory, and physical interaction.

How LLM Agents can be leveraged to build **Generalist Embodied AI in an Open World?**

What's Missing in Current LLM-based approach?

1. **LLMs** excel at reasoning and language but struggle in physical spaces:
 - a. Lack spatial and temporal memory.
 - b. No grounding in the real-world 3D context.

More specifically,

- c. Reasoning about **object permanence** (where things are over time).
- d. Performing **long-horizon tasks** (e.g., cleaning a messy room).
- e. Handling **unseen objects and instructions**.
- f. Generating **multi-step physical plans** grounded in perception.

LEO: An Embodied Generalist Agent

LEO = Language + Environment + Object-centric memory

Scene Captioning



LEO: *In this room, there is a wooden floor that is clean and flat. A tall wardrobe ...*

3D Question Answering

Question: Where is the Printer located?

LEO: *Standing on the desk next to the wardrobe.*

3D Dialogue

Human: I need help organizing my bedroom. Can you assist me?

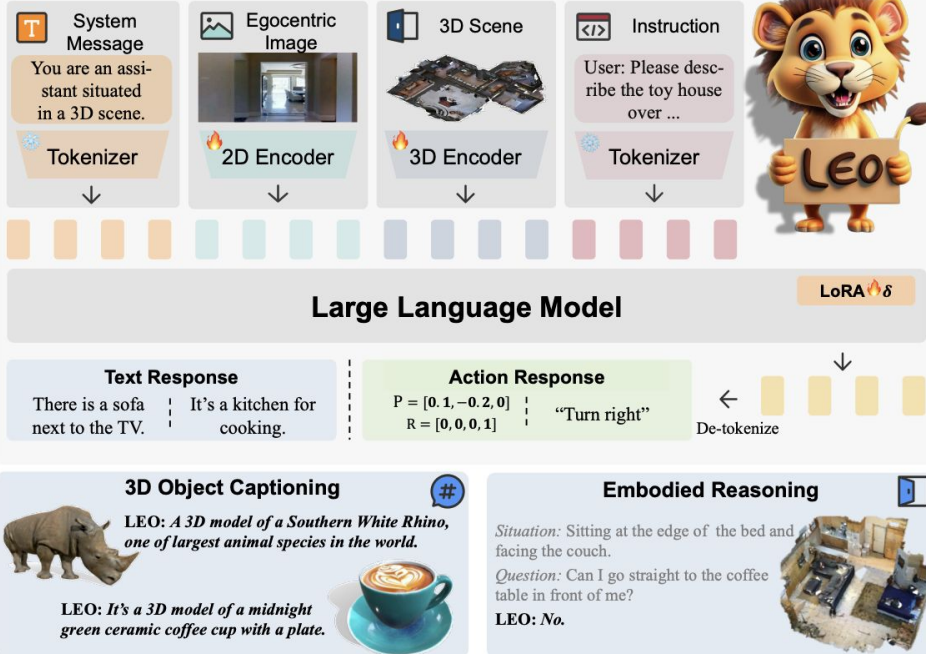
LEO: *Of course! What specifically do you need help with?*

Human: Can you check if there are any clean clothes hanging on the wall?

LEO: *I'm sorry, but the clothes hanging on the wall are messy.*

Human: Thanks! Also, can you check if there are any clean clothes hanging on the wall?

Embodied Generalist Agent in 3D World



Robotic Manipulation

Human: pack the yoshi figure in the brown box

LEO:

$P = [0.1, -0.2, 0]$
 $R = [0, 0, 0, 1]$

Embodied Navigation

Human: Find the pillow

LEO:

1. Move forward
2. Turn right
3. Move forward
4. ...

Task Planning

Human: Organize and tidy up the bedroom.

LEO:

1. Clean the floor by sweeping to remove ...
2. Make the bed ...
3. Fold and organize ...

Model

1. Tokenization

First step, Tokenizes all multimodal inputs: text, 2D images, 3D point clouds, and embodied actions into a unified sequence of discrete tokens using established methods and SentencePiece, enabling LEO to process diverse modalities in a consistent format.

2. Token Embedding & LLM

- **Scene representation** : The scene point cloud is partitioned into object-centric point clouds (either ground truth or predicted proposals), which are then processed by the 3D encoder to obtain object-centric features. They also incorporate an optional 2D branch, where a 2D encoder processes the agent's ego-view observation to obtain ego-centric features.
- **Unified sequence and objective**: The sequence begins with a system message that tells the agent its role and situation. Subsequent 2D image tokens and 3D object tokens provide the perceived scene information. Next an instruction specifies the task or context, and also prompts for the final response. The learning objective is a simple auto-regressive loss.

Model

3. Training and Inference

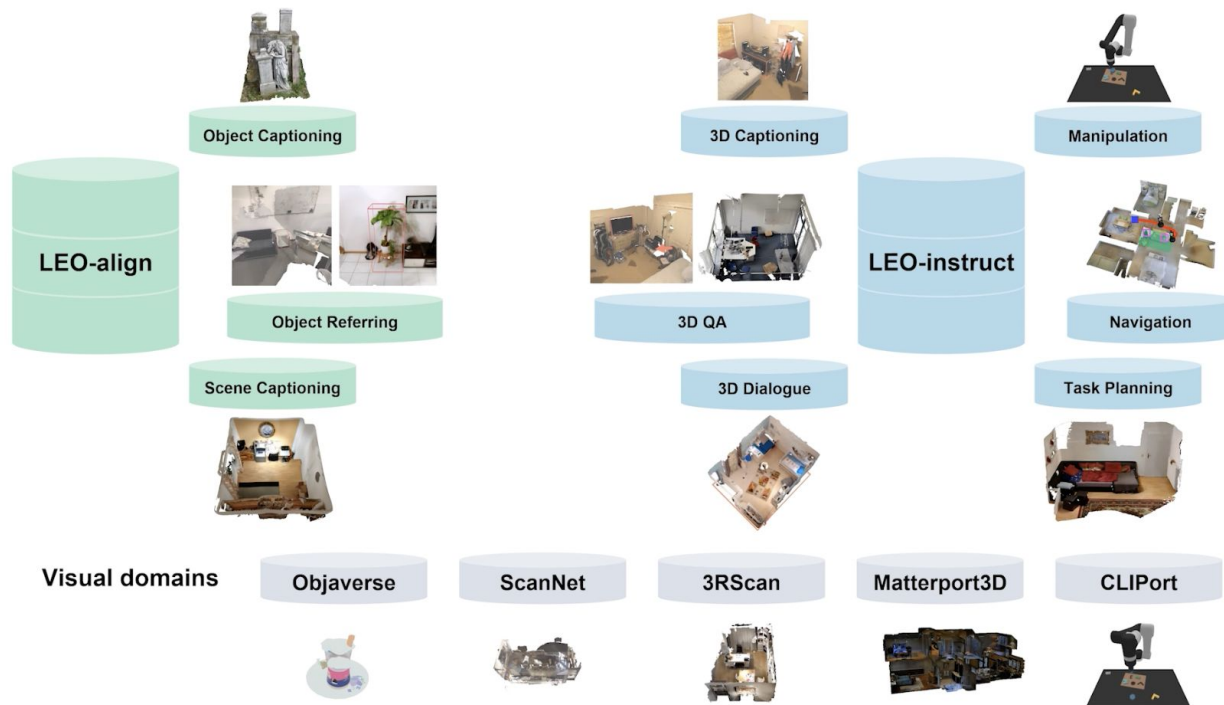
LEO is trained using a prefix language modeling objective, fine-tuning only select components while freezing the LLM and 3D encoder, and uses beam search during inference to generate responses or map them to action commands.

$$\mathcal{L}(\theta, \mathcal{B}) = - \sum_{b=1}^{|\mathcal{B}|} \sum_{t=1}^T \log p_{\theta}(s_{\text{res}}^{(b,t)} | s_{\text{res}}^{(b,<t)}, s_{\text{prefix}}^{(b)}),$$

Dataset for training

1. Two-stage scheme: alignment & instruction tuning.
2. They combine existing datasets and LLM-prompted data to create
 - a. **LEO-align**
that focuses on 3D vision language (VL) alignment to bridge the gap between 3D scene representation and natural language
 - object-level captions
 - object-in-the-scene captions
 - scene-level captions
 - b. **LEOinstruct**
that targets at 3D VLA instruction tuning to endow LEO with various capabilities.
 - 3D captioning and question answering
 - 3D dialogue and task planning
 - navigation and manipulation

Dataset for training



LLM-assisted 3D-language Data Generation

Scene-Graph-based Prompting

Messages

1. System Message

You are an AI visual assistant in a 3D scene...

2. Demonstrations

Scene Graph Context:

{'sofa-1': {'attributes': {'color': 'red'}, 'relations': ['to the right of chair-2', 'in front of table-3']}, 'chair-2': {'attributes': {'color': 'brown'}, 'relations': []}}

Human-labeled Responses

3. Query(a new scene)

Scene Graph Context



Responses

- 1) Object scene caption
- 2) Scene caption
- 3) Dialogue(O-CoT)
- 4) QA(O-CoT)
- 5) Planning

Potential Issues:

- 1) Wrong Answers
- 2) Negative Responses
- 3) Responses with IDs
- 3) Others

Refinement Procedures

Raw Responses

1. Wrong Answers

There are two curtains in the room.

2. Negative Responses

Question: Where is the football table?

Thought: football table-17 Answer: unknown

3. Responses with ID

The kitchen cabinet-9 and kitchen counter-10 are parts of the kitchen.

4. Others

Unnecessary content (Thought, etc.)



Refined Responses

1. Answer Refinement

There are 3 curtains in the room.

2. Removing Negative Responses

The negative responses will be removed.

3. GPT-based rewriting

The kitchen features a cabinet and a counter.

4. Others

Remove unnecessary content.

Dialogue(O-CoT):

Dialogue Context: high level task: organize the bedroom. low level task: check some objects.

Human: I need help organizing my bedroom. Can you assist me? Thought: no specific objects

Assistant: Of course! What specifically do you need help with?

Human: Can you check if there are any clean clothes hanging on the wall? Thought: clothes-13

Assistant: I'm sorry, but the clothes hanging on the wall are messy.

Human: Thanks for your help! Thought: no specific object

Assistant: You're welcome! If you need any further assistance, feel free to ask.



Scene Caption:

In this room, there is a wooden floor that is clean and flat. A tall wardrobe stand on the right side of a desk, close to a basket. The wardrobe is in front of a chair, a bed, and behind a bedside table. Clothes are hanging on a white wall, to the right of a closed curtain. The bed is wide, clean, and covered with blue blanket. The room has a comfortable and organized setting with functional furniture.

Object Scene Caption:

The showcase is supported by the wall and positioned behind, close to, and to the left of the shelf. The wall is flat, architectural, and dark in color. The shelf is made of wood and has a low size.

QA(O-CoT):

Question: Where is the printer located? Thought: printer-8 Answer: standing on the desk

Question: How many blankets are on the bed? Thought: blanket-16, blanket-17 Answer: 2

Question: What is the type of the room? Thought: wardrobe-2, desk-7, chair-11, bed-15 Answer: bedroom

Planning:

High-Level Task: Organize and tidy up the bedroom.
Low-Level Actions:

1. Clean the floor by sweeping to remove any dirt.
2. Make the bed by arranging the blanket and pillows.
3. Place any loose items or belongings into the basket.
4. Arrange items on the shelves and showcase in a tidy way.

Results

Quantitative comparison with state-of-the-art models
on 3D VL understanding and embodied reasoning tasks

	Scan2Cap (val)					ScanQA (val)					SQA3D (test)
	C	B-4	M	R	Sim	C	B-4	M	R	EM@1	EM@1
<i>Task-specific models</i>											
Scan2Cap	35.2	22.4	21.4	43.5	-	-	-	-	-	-	41.0 [†]
3DJCG	47.7	31.5	24.3	51.8	-	-	-	-	-	-	-
Vote2Cap-DETR	61.8	34.5	26.2	54.4	-	-	-	-	-	-	-
ScanRefer+MCAN	-	-	-	-	-	55.4	7.9	11.5	30.0	18.6	-
ClipBERT	-	-	-	-	-	-	-	-	-	-	43.3
ScanQA	-	-	-	-	-	64.9	10.1	13.1	33.3	21.1	47.2
<i>Task-specific fine-tuned</i>											
3D-VisTA	66.9	34.0	27.1	54.3	53.8	69.6	10.4	13.9	35.7	22.4	48.5
3D-LLM (FlanT5)	-	-	-	-	-	69.4	12.0	14.5	35.7	20.5	-
LEO	72.4	38.2	27.9	58.1	55.3	101.4	13.2	20.0	49.2	24.5 (47.6)	50.0 (52.4)

Results

Results on robot manipulation

	separating-piles		packing-google -objects-seq		put-blocks-in -bowls	
	seen	unseen	seen	unseen	seen	unseen
CLIP-only	90.2	71.0	95.8	57.8	97.7	44.5
CLIPort (single)	98.0	75.2	96.2	71.9	100	25.0
CLIPort (multi)	89.0	62.8	84.4	70.3	100	45.8
LEO	98.8	75.2	76.6	79.8	86.2	35.2

Results on object navigation

	MP3D-val		HM3D-val	
	Success(↑)	SPL(↑)	Success(↑)	SPL(↑)
Habitat-web (shortest)	4.4	2.2	-	-
Habitat-web (demo)	35.4	10.2	-	-
ZSON	15.3 [†]	4.8 [†]	25.5	12.6
LEO	23.1	15.2	23.1 [†]	19.1[†]

Improvements

1. **Embodied Generalization:** LEO marks a significant advancement in extending large language models from pure language tasks to **3D visual understanding and embodied action**, demonstrating promising performance across navigation, manipulation, and dialogue tasks in simulated 3D environments.
2. **Unified Multimodal Tokenization:** It unifies diverse modalities : language, 2D vision, 3D point clouds, and actions into a common token space, enabling a consistent and scalable way to learn from rich embodied data.
3. **Joint Learning Benefits:** The framework shows that joint training on 3D vision-language tasks and embodied control tasks can **boost performance across both**, highlighting the synergy between perception and action.
4. **Efficient Fine-tuning:** By freezing the LLM and 3D encoder and only fine-tuning the image encoder and task-specific layers (~142M parameters), LEO achieves scalability while maintaining computational efficiency.

Fallbacks

1. **Limited Generalization to Novel Scenes:** Despite its performance on known datasets, LEO struggles with **scene generalization**, indicating that it may overfit to training environments and lack robust domain transfer.
2. **Action Execution Gap:** There remains a **performance gap between visual-linguistic understanding and physical action execution**, pointing to challenges in grounding high-level reasoning into low-level control.
3. **Safety and Alignment Unaddressed:** While LEO takes steps toward embodiment, it currently lacks robust mechanisms for **safety, alignment, and interpretability**, which are critical in real-world applications of embodied AI.

Broader Thoughts

1. **Scaling 3D VL Data:** Expand training datasets to include more **diverse and complex 3D scenes**, possibly incorporating synthetic and real-world scans, to improve robustness and generalization.
2. **Tighter Perception-Action Coupling:** Investigate architectures and learning paradigms that **tightly couple 3D vision understanding with action generation**, reducing latency and increasing task fidelity.
3. **Continual and Lifelong Learning:** Enable LEO to **adapt over time to new tasks and environments** without catastrophic forgetting, essential for deployment in dynamic real-world settings.
4. **Safety and Alignment Research:** Introduce formal mechanisms for **ethical reasoning, value alignment, and safe exploration**, especially as models are scaled up and deployed in embodied agents.
5. **Unified Scaling Laws:** Further explore how **scaling model size and data** affects multimodal performance, aiming for principled guidelines for building stronger embodied generalists.

TEMPLATE NOTES

This template is part of the [NYU Templates collection](#).
Refer to our [Usage Guidelines](#) for help topics and quick
tips on how to use this template.

Download the [Grackle Slides](#) add-on to automatically run
accessibility checks on all aspects of your document and
get advice on how to make things better.

Leveraging Cognitive Science for Better Agent Design

Presenter: Solim LeGris



Agenda

2

01

INTRO

02

BACKGROUND

03

FRAMEWORK

04

CASE STUDIES

05

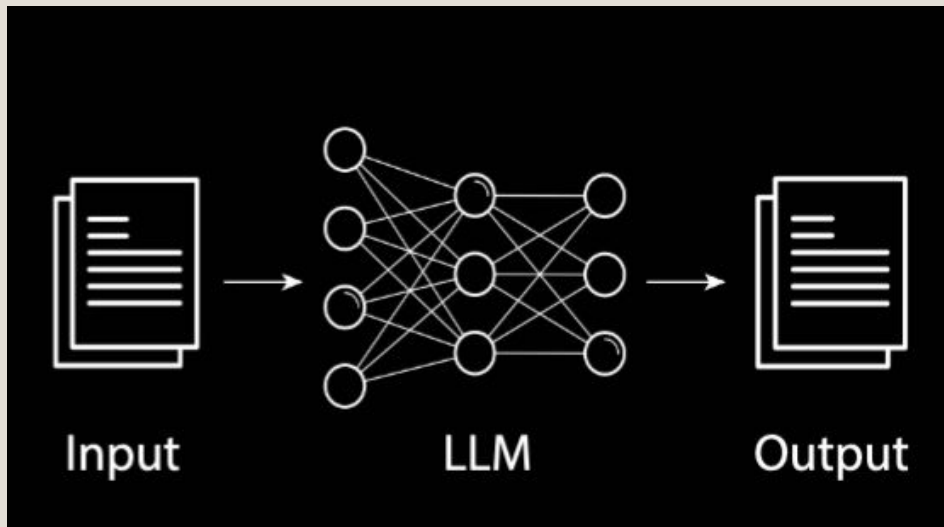
AGENT DESIGN

06

FUTURE



The rise of language agents

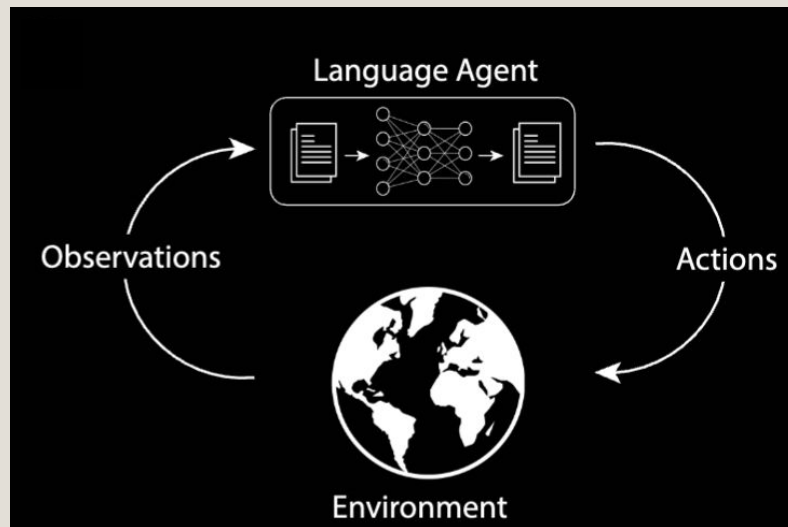


LLMs are powerful but have *limitations*: knowledge, reasoning, etc.



The rise of language agents

4

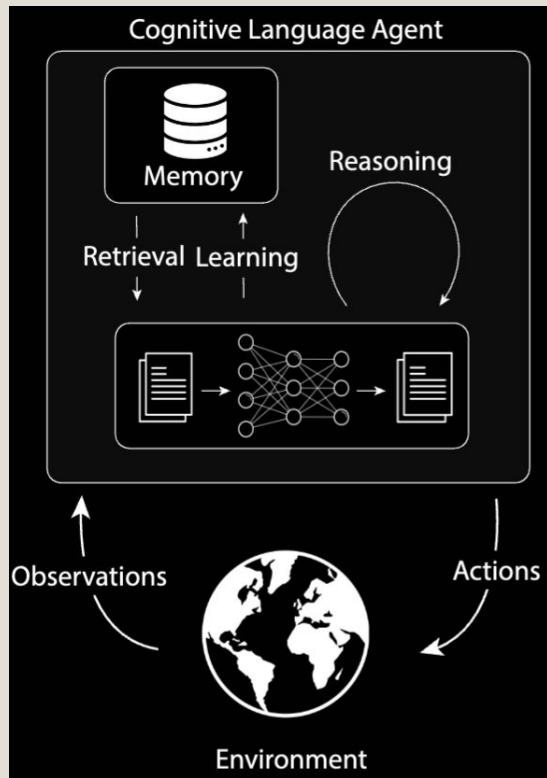


Language agents: systems using LLMs to interact with the world



The rise of language agents

- ❖ Agents overcome LLM limits (grounding) and traditional agent limits (generalization via LLM priors).
- ❖ Direct selection or generation of actions is limited
- ❖ Need structures that integrate reasoning, long-term memory and planning with LLM



A novel framework

Cognitive Architectures for Language Agents

Theodore R. Sumers* Shunyu Yao* Karthik Narasimhan Thomas L. Griffiths

Princeton University

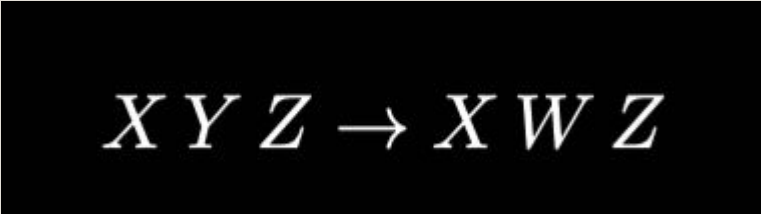
{sumers, shunyuy, karthikn, tomg}@princeton.edu

- ❖ Rapid development leads to custom terminology and architectures (e.g., "tool use", "grounding").
- ❖ Difficult to compare agents, understand evolution, or build new agents systematically.
- ❖ Need a conceptual framework to organize efforts.



Production systems and control flow

- ❖ **Production systems** (Newell & Simon, 1972):
 - Rules (precondition $\rightarrow$ action) applied iteratively.
 - Originated for string manipulation.



XYZ $\rightarrow$ XWZ



Production systems and control flow

- ❖ Production systems were popularized in AI for understanding human problem solving (Newell, 1967; Newell and Simon, 1972)

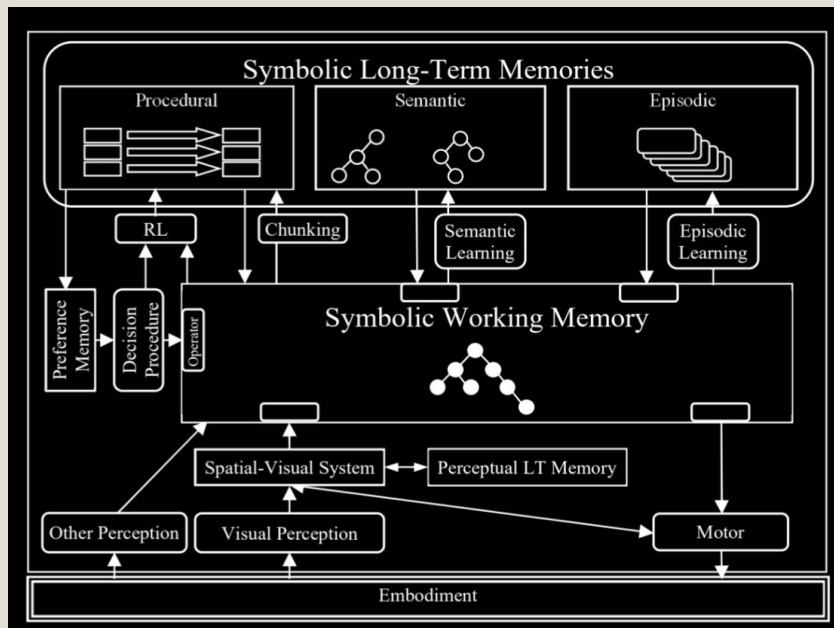
$(\text{temperature} > 70^\circ) \wedge (\text{temperature} < 72^\circ) \rightarrow \text{stop}$
 $\text{temperature} < 32^\circ \rightarrow \text{call for repairs; turn on electric heater}$
 $(\text{temperature} < 70^\circ) \wedge (\text{furnace off}) \rightarrow \text{turn on furnace}$
 $(\text{temperature} > 72^\circ) \wedge (\text{furnace on}) \rightarrow \text{turn off furnace}$

Adams et al., 2012; Sun, 2004; Newell, 1980; 1992; Anderson and Lebiere, 2003



Cognitive architectures

Blueprints for intelligence, combining production systems with modules for memory, perception, decision-making, learning.

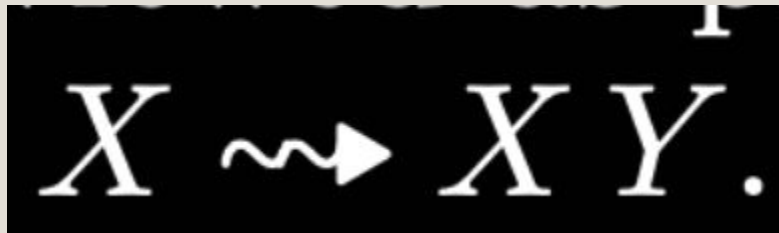


SOAR (Laird et al., 1987; 2022)



Historical analogy: production systems

- ❖ Formally:
 - $P(w_i | w_{<i})$
- ❖ **LLMs as production systems:** LLMs define a *probabilistic distribution* over text continuations (string changes/additions).



Historical analogy: control flow & prompting

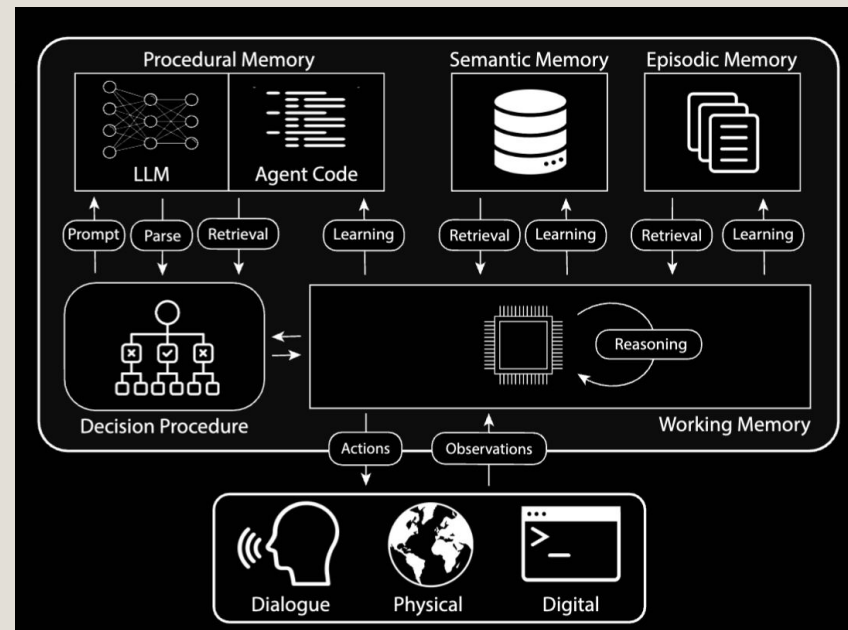
- ❖ **Prompt engineering as control flow:** manipulating the input prompt (string) biases the LLM's production selection towards desired outputs.
 - Simple concatenation, RAG, chain-of-thought, etc., structure the "algorithm".

Prompting Method	Production Sequence
Zero-shot	$Q \xrightarrow{\text{LLM}} Q A$
Few-shot	$Q \rightarrow Q_1 A_1 Q_2 A_2 Q \xrightarrow{\text{LLM}} Q_1 A_1 Q_2 A_2 Q A$
Retrieval Augmented Generation	$Q \xrightarrow{\text{Wiki}} Q O \xrightarrow{\text{LLM}} Q O A$
Socratic Models	$Q \xrightarrow{\text{VLM}} Q O \xrightarrow{\text{LLM}} Q O A$
Self-Critique	$Q \xrightarrow{\text{LLM}} Q A \xrightarrow{\text{LLM}} Q A C \xrightarrow{\text{LLM}} Q A C A$



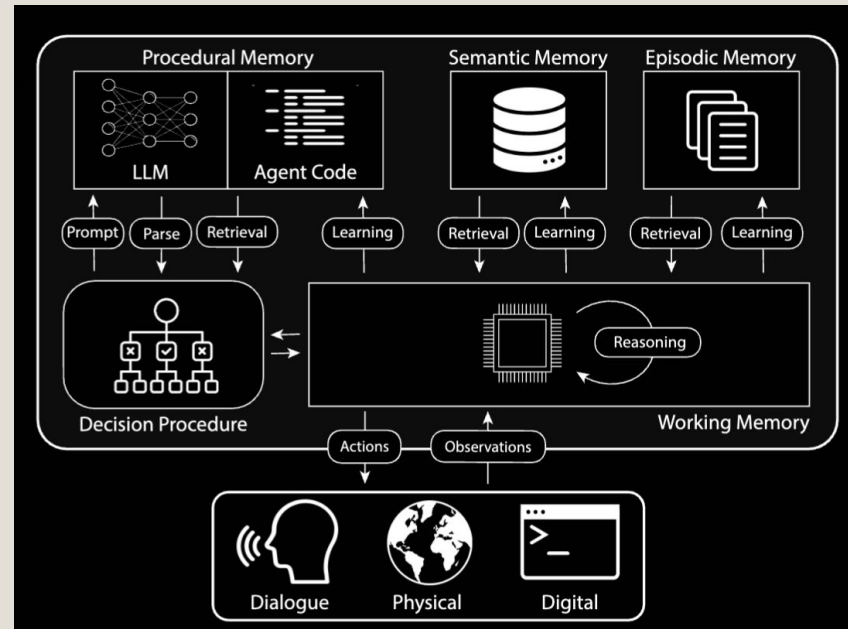
CoALA framework

- ❖ A conceptual framework to characterize and design language agents using LLMs as a core component within a cognitive architecture.
- ❖ Organizes agents along three dimensions:
 - Memory (information storage)
 - Action space (internal & external)
 - Decision-making (procedure)



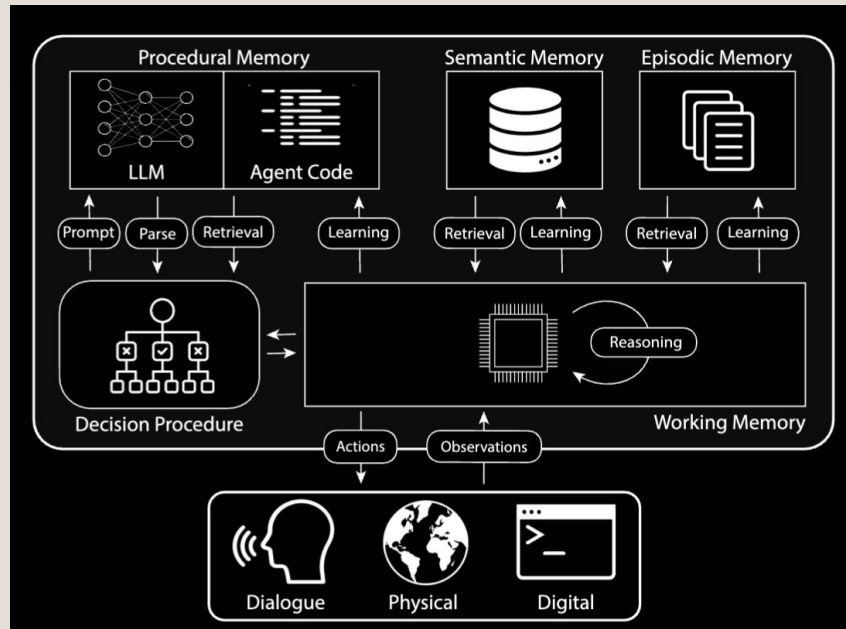
CoALA Component: Memory Modules

- ❖ Working memory: Active, readily available info for the current decision cycle (inputs, goals, retrieved knowledge). Persists across LLM calls.
 - Perceptual inputs, information carried over, etc
 - More general than context
 -



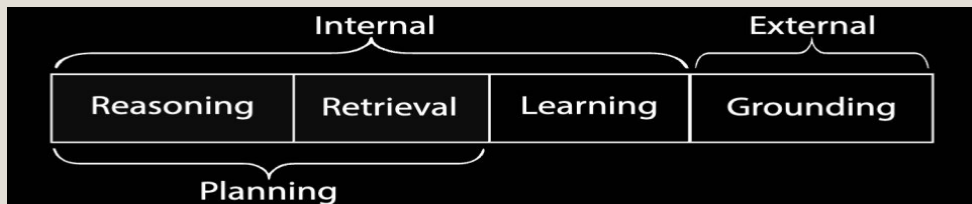
CoALA Component: Memory Modules

- ❖ Long-Term memory
 - Episodic: Past experiences, trajectories.
 - Semantic: Facts about the world/self (knowledge).
 - Procedural: How-to knowledge (LLM weights, agent's code/skills).



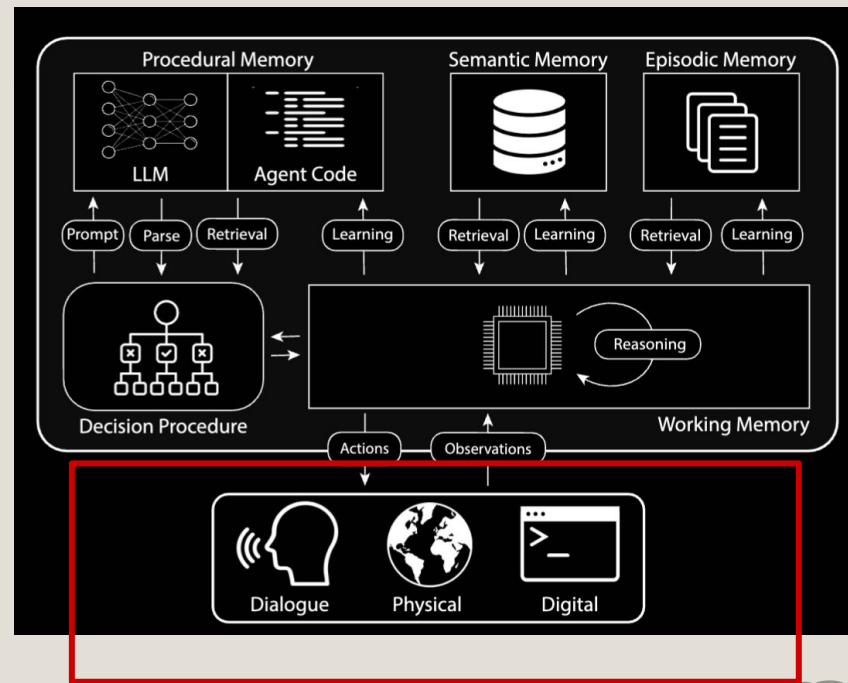
CoALA Component: Action Space

- ❖ Actions modify the agent's state or the environment.
- ❖ External Actions: Interact with the outside world.
- ❖ Internal Actions (Memory Access):



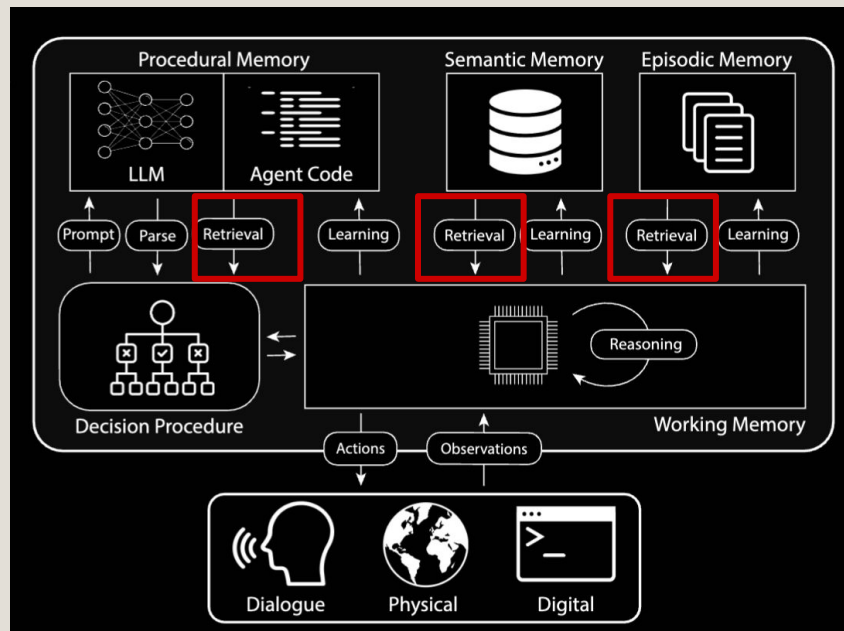
Grounding actions

- ❖ Connects the agent to the external world.
- ❖ Procedures execute external actions & process feedback into working memory (often as text).
- ❖ Examples: Robot commands, sending messages, navigating websites, API calls, etc
- ❖ Vision-Language Models (VLMs) can help translate perception to text.



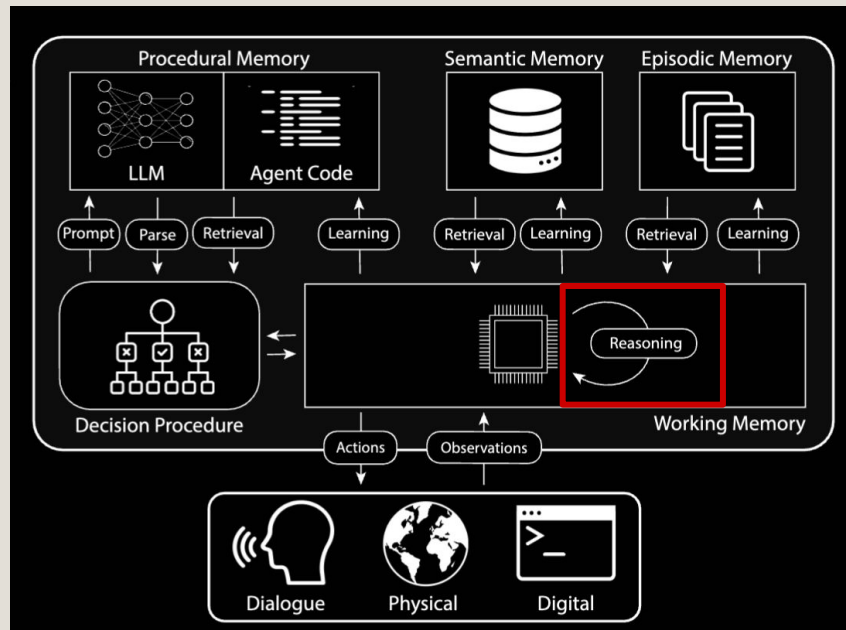
Internal (Memory Ops)

- ❖ Retrieval: Read relevant info from LTM (Episodic, Semantic, Procedural) into WM to support reasoning/decision-making.



Internal (Memory Ops)

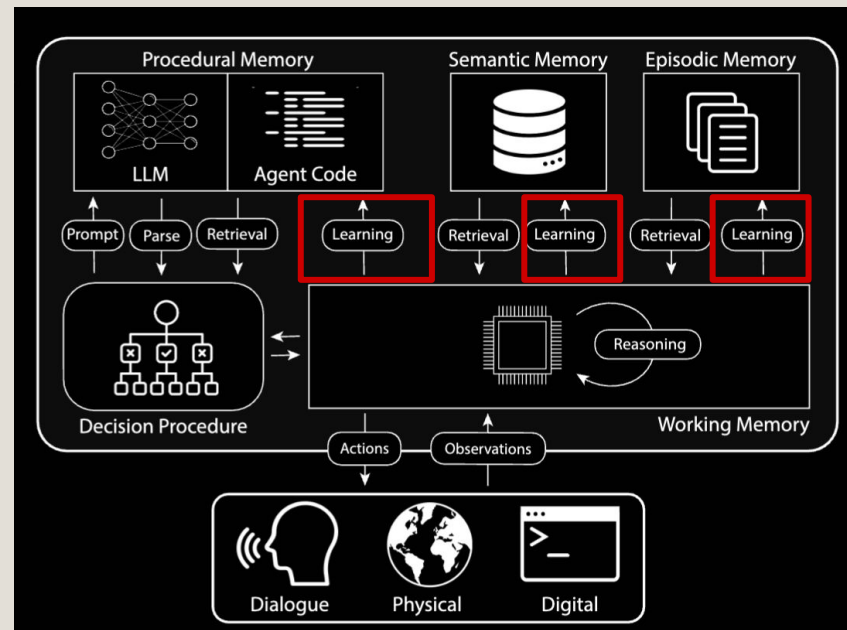
- ❖ Reasoning: Process working memory contents (e.g., summarize, reflect, plan steps) using the LLM. Writes back to working memory.



Internal (Memory Ops)

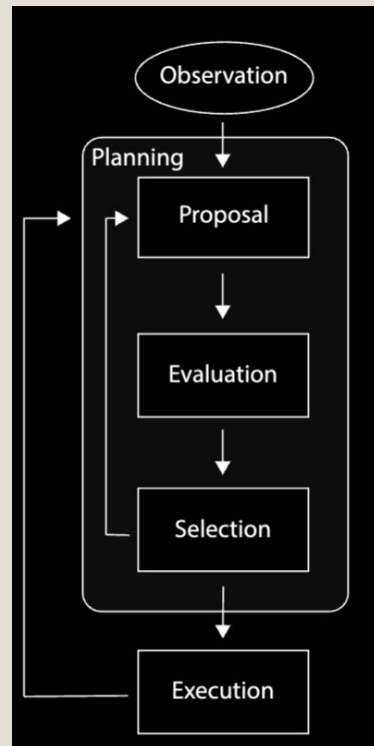
❖ **Learning actions:** Write information to LTM (new experiences, facts, skills/code).

- Episode storage
- Facts storage
- Direct weight updates
- Procedure / agent code updating



Decision Making

- ❖ The "main loop" selecting which action (Grounding or Learning) to execute.
- ❖ Decision Cycle:
 - a. Planning Stage: Use reasoning & retrieval actions to propose, evaluate, and select candidate actions. Can be simple (single LLM call) or complex (tree search, simulation).
 - b. Execution Stage: Execute the chosen Grounding/Learning action procedure.
 - c. Observe feedback, update working memory, loop.



Applying CoALA: Case Studies

- ❖ CoALA can characterize diverse existing agents.
- ❖ Helps understand their mechanisms, similarities, and differences structurally.

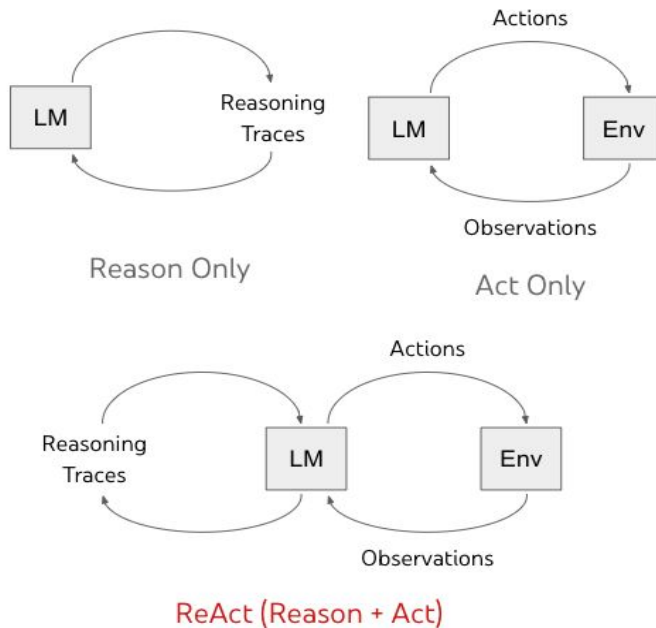
	Long-term Memory [¶]	External Grounding	Internal Actions	Decision Making
SayCan (Ahn et al., 2022)	-	physical	-	evaluate
ReAct (Yao et al., 2022b)	-	digital	reason	propose
Voyager (Wang et al., 2023a)	procedural	digital	reason/retrieve/learn	propose
Generative Agents (Park et al., 2023)	episodic/semantic	digital/agent	reason/retrieve/learn	propose
Tree of Thoughts (Yao et al., 2023)	-	digital	reason	propose, evaluate, select

Table 2: Some recent language agents cast into the CoALA framework.



Case Study: ReAct

(Yao et al., 2022)



Case Study: ReAct

(Yao et al., 2022)

- ❖ **Memory:** WM only (implicit via LLM context).
- ❖ **Grounding:** Digital (Web APIs, Text Games).
- ❖ **Internal Actions:** Reasoning (to analyze, plan). No Retrieval/Learning.
- ❖ **Decision cycle:** Reason (Thought) → Propose Action → Execute Action → Observe.
- ❖ **Key Idea:** Integrating reasoning and acting within a loop.

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: `Search[Apple Remote]`

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the `Front Row media center` program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search `Front Row` next and find what other device can control it.

Act 2: `Search[Front Row]`

Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', '`Front Row (software)`', ...]

Thought 3: `Front Row` is not found. I need to search `Front Row (software)` .

Act 3: `Search[Front Row (software)]`

Obs 3: `Front Row` is a discontinued media center software ...

Thought 4: `Front Row (software)` is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

Act 4: `Finish[keyboard function keys]`



Case Study: Voyager

(Wang et al., 2023)

INTRO

BACKGROUND

FRAMEWORK

CASE
STUDIESAGENT
DESIGN

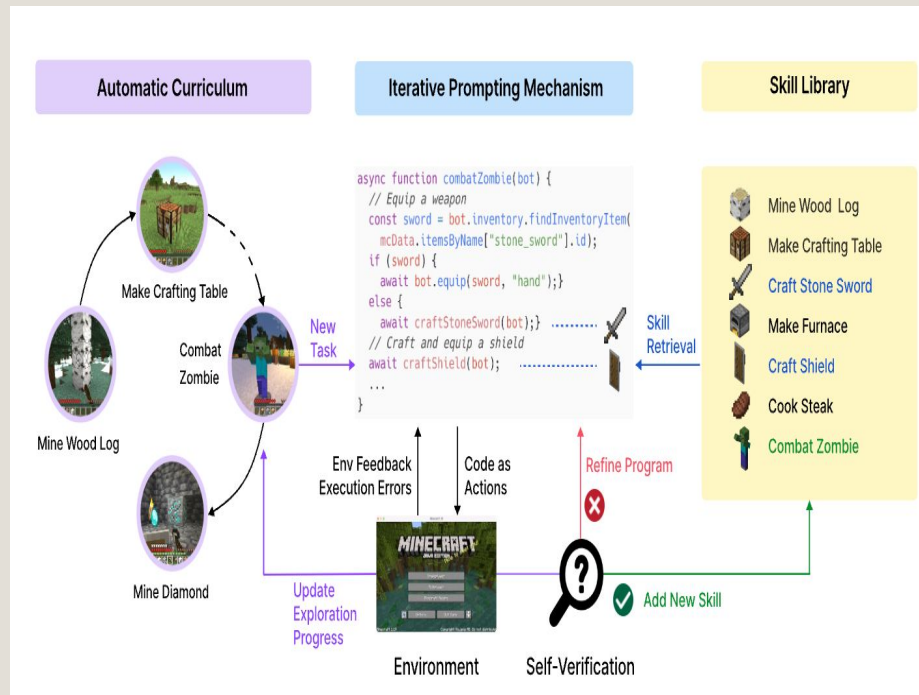
FUTURE



Case Study: Voyager

(Wang et al., 2023)

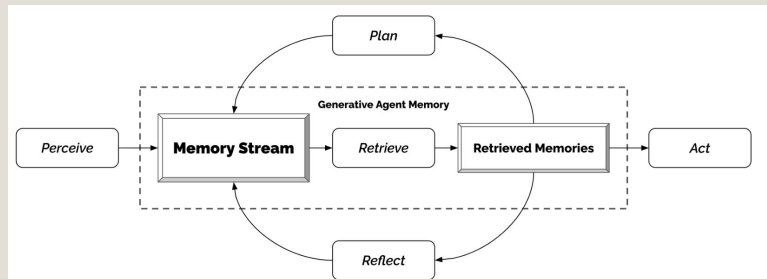
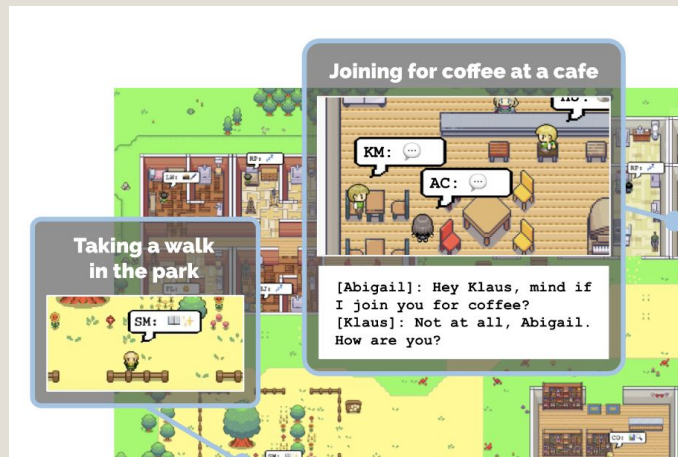
- ❖ **Memory:** Working + Procedural LTM (Skill library).
- ❖ **Grounding:** Digital (Minecraft API).
- ❖ **Internal Actions:** Reasoning, Retrieval (skills), Learning (add new skills).
- ❖ **Decision-cycle:** Propose task → Retrieve/Generate code (skill) → Execute → Reason on feedback → Learn (if successful) or Refine.
- ❖ **Key Idea:** Lifelong learning by building a procedural memory (code library).



Case Study: Generative Agents

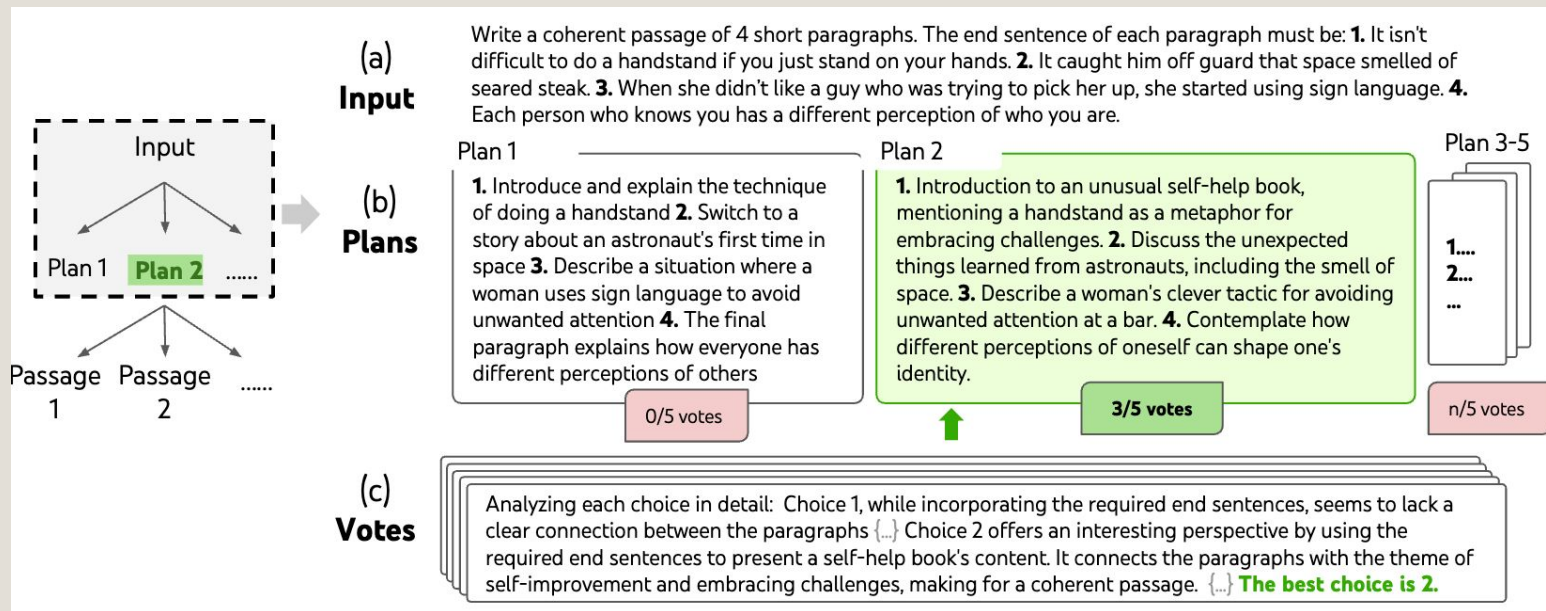
(Park et al., 2023)

- ❖ **Memory:** Working + Episodic LTM (event stream) + Semantic LTM (reflections).
- ❖ **Grounding:** Digital (Sandbox Game) + Agent Dialogue.
- ❖ **Internal Actions:** Reasoning (reflect, plan), retrieval (events, reflections), learning (store events, reflections).
- ❖ **Decision-Making:** Retrieve → Reason (Plan daily schedule) → Execute actions → Reason (update plan based on observations).
- ❖ **Key Idea:** Simulating believable human behavior via rich memory & reflection.



Case Study: Tree of thoughts

(Yao et al., 2023)



Modular Agent Design

- ❖ Think beyond monoliths:
 - Structure agents using modules (Memory, Actions, Decision cycle).
- ❖ Benefits:
 - Conceptual clarity, easier comparison
 - Code reuse, testing, maintenance
 - Standardized abstractions (like MDPs and Gym envs for RL).
- ❖ LLMs vs Code: Use code for stable algorithms (e.g., search) complementing LLM flexibility.



Beyond Simple Reasoning/Planning

- ❖ **Structured reasoning:** Move beyond basic prompt engineering to structured updates of working memory (e.g., LangChain, Guidance).
- ❖ **LTM:** Leverage *writable* LTM for lifelong learning (not just retrieval augmentation). Combine human knowledge + agent experience.
- ❖ **Learning:** Explore diverse learning actions beyond fine-tuning (storing experiences/facts, generating code/skills, meta-learning).
- ❖ **Decision Making:** Move towards more deliberate planning (proposal, evaluation, selection) using LLMs.



Future Directions & Open Questions

- ❖ **LLMs vs VLMs:** Language-only vs integrated multimodal reasoning?
- ❖ **Learning vs Acting:** How to autonomously decide *when* and *what* to learn?
- ❖ **Advanced decision making:** Scaling deliberative planning, metareasoning (value of computation), calibration, alignment.
- ❖ **Co-evolution:** How will agent design and LLM capabilities influence each other?
 - Reasoning models are already a proof of agent design influencing LLM capabilities



Conclusion

- ❖ Language agents are evolving rapidly, incorporating complex internal processes.
- ❖ CoALA provides a framework inspired by cognitive science to:
 - Organize existing work.
 - Guide the design of new, more capable agents.
 - Structure thinking around Memory, Actions, and decision making.
 - Offers actionable insights for building modular, structured, and more intelligent agents.



Q&A

Thank You!

