

From next-word to next-world: World model for purposeful reasoning beyond lingual intelligence

Mingkai Deng

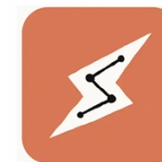
March 26, 2026

LLMs now support wide range of real-world tasks

- Core capability is simulating the next word in human language

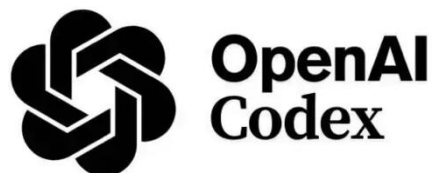


Common conversation



Claude
Cowork

Computer Use



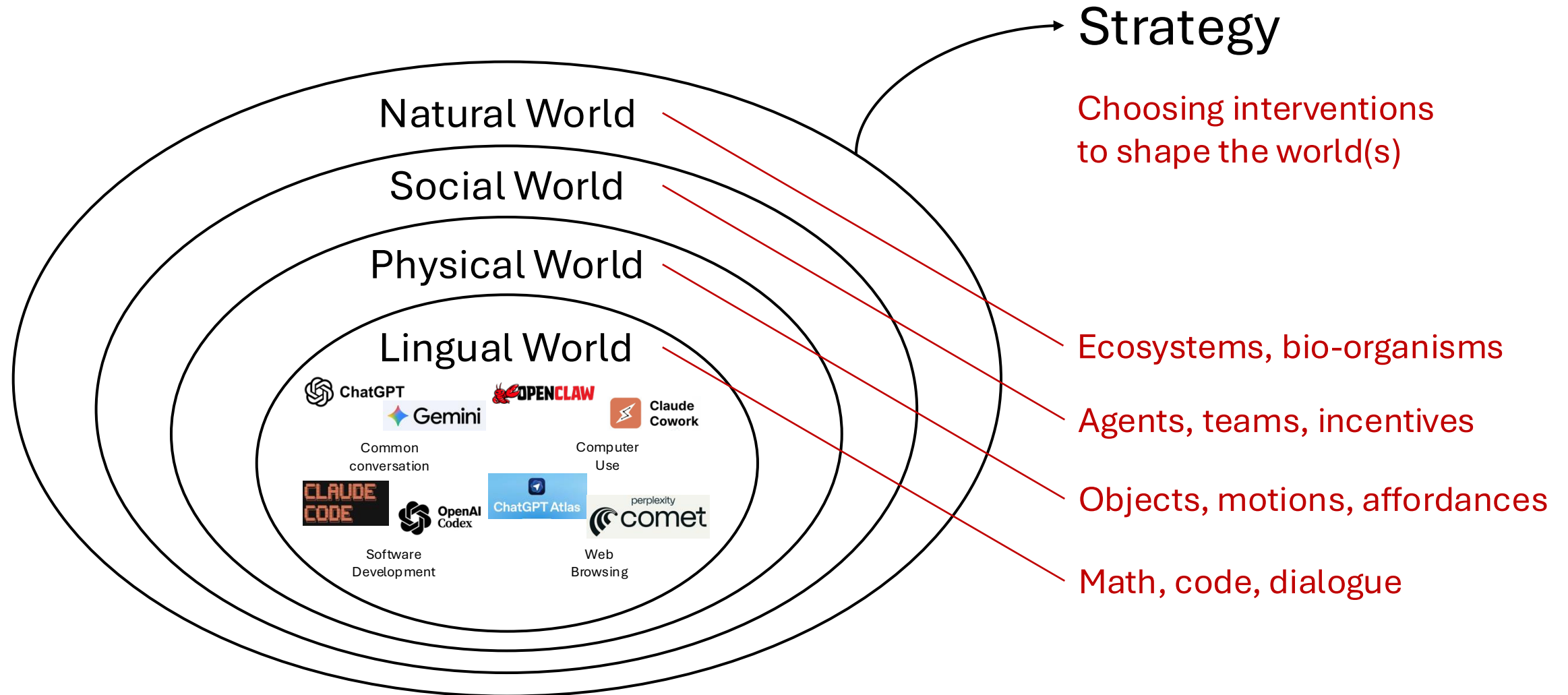
Software Development



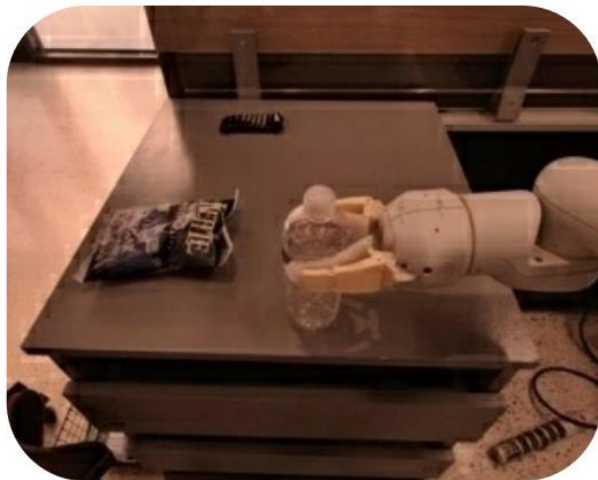
Web Browsing

And many more...

However, language is just a tiny slice of the world



How well do LLMs understand the physical world?



s_t



s_{t+1}

- A) place water bottle on table
- B) move water bottle towards blue snack packet
- C) move water bottle near rxbar chocolate
- D) put water bottle next to remote control

Given s_t and s_{t+1} , what could be the action that translates s_t to s_{t+1} ?

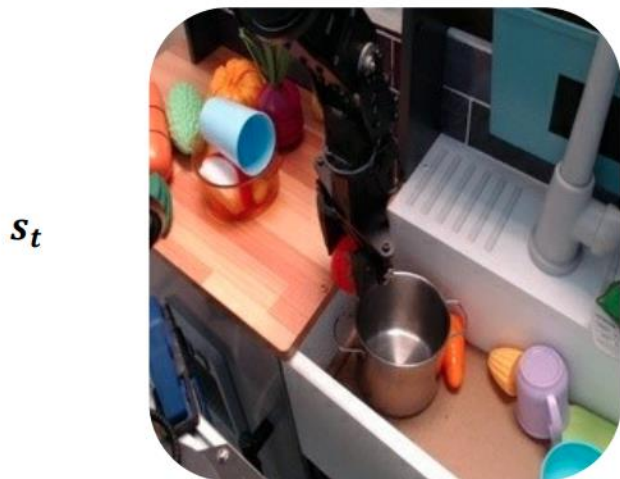


Groundtruth: **C**



Gemini-2.5-Pro: **B** 

How well do LLMs understand the physical world?



a_t

“Rotate clockwise around
Z-axis by 25.1 degrees,
Open gripper by 0.24”

Given s_t and a_t ,
what could be the most
logical s_{t+1} ?



A



B



C



D



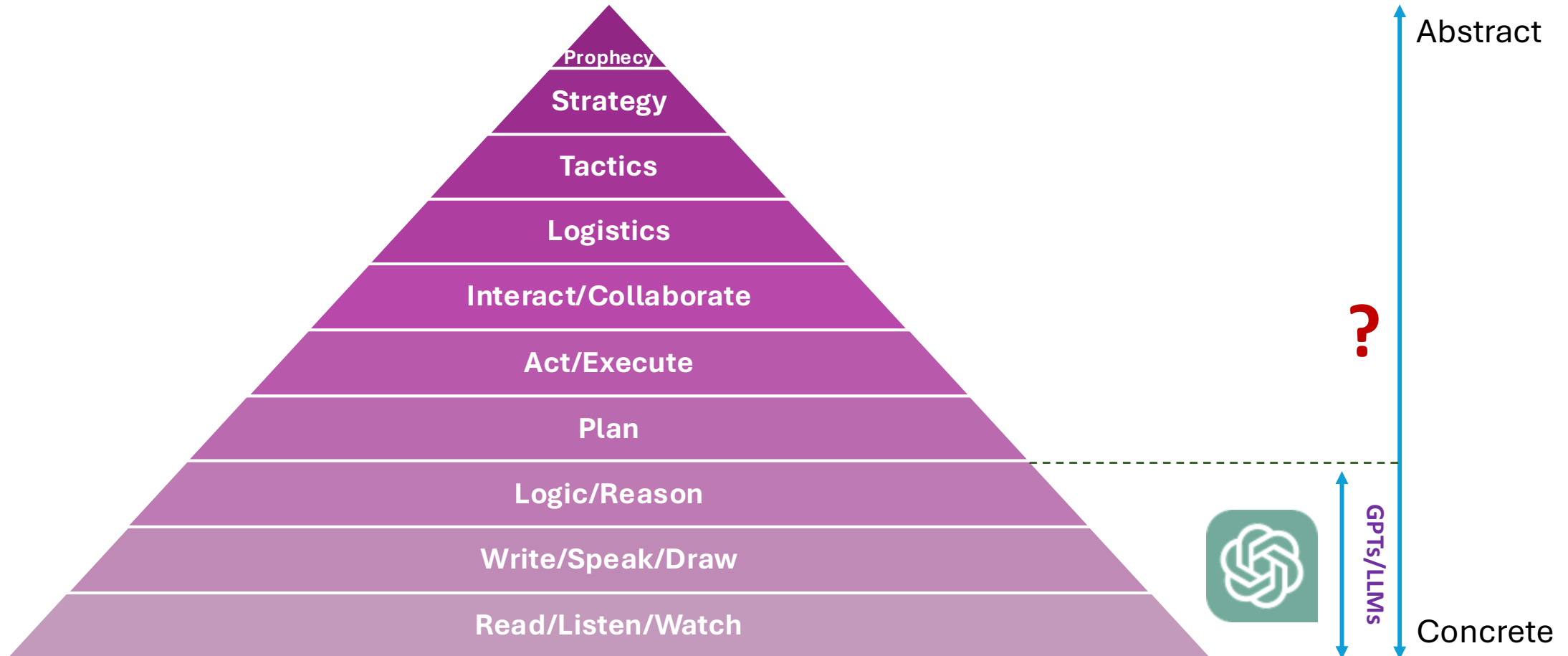
Groundtruth: C



Gemini-2.5-Pro: D



Humans consist of a hierarchy of capabilities



Dune series imagines such a possibility

- Influenced works such as *Star Wars* and Miyazaki's *Valley of the Wind*

Novels



Films



Part 3
coming
soon!

Dune is centered around the *Kwisatz Haderach*

- A person born to inherit the experience of all their ancestors
- This gives them extraordinary capabilities
- They can lead armies across the galaxy and direct planet-scale projects
- Their power comes from **mentally simulating many possible futures** and choosing the best path

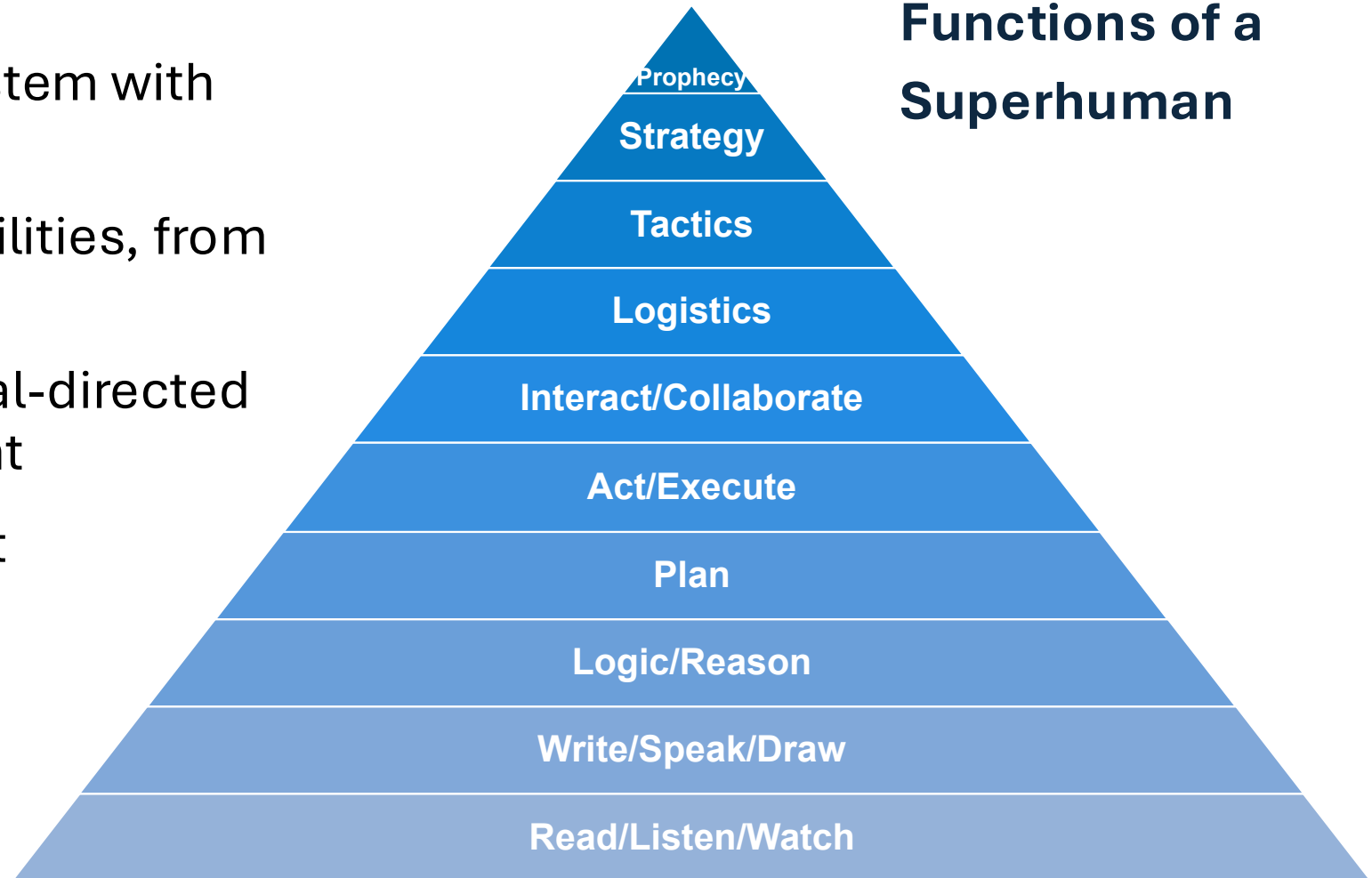


- In the story, this culminates in a plan to steer humanity towards a freer, less predictable future

Is such a superhuman possible?

- Maybe not literally, but system with similar functions
- Going up a hierarchy of abilities, from low- to high-level
- Each can be viewed as goal-directed behavior in an environment
- Can **one** model, or at least one integrated system, support it?

Functions of a Superhuman



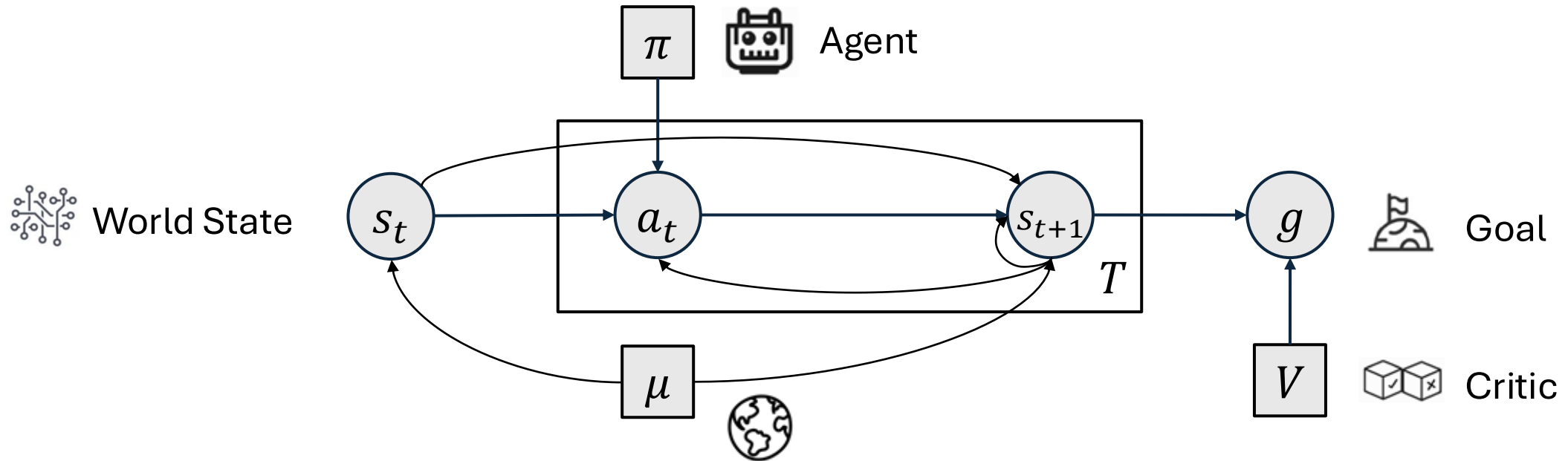
A possible definition of an optimal agent

The optimal agent maximizes the expected chance of success in achieving the goal:

$$\pi_{\mu}^* := \arg \max_{\pi} V_{\pi, \mu}^g$$

Resulting in the following decision rule:

$$\pi_{\mu}^*(s_t) = \arg \max_{\underbrace{a_{t:T-1}}_{\text{possible actions}}} \sum_{s_{t+1:T}} \underbrace{\left(\sum_{k=t}^{T-1} \gamma_k r(g, s_k) + \gamma_T V_{\pi, \mu}^g(s_T) \right)}_{\text{goal progress}} \underbrace{\prod_{i=t}^{T-1} p_{\mu}(s_{i+1} | s_i, a_i)}_{\text{environment response}}$$



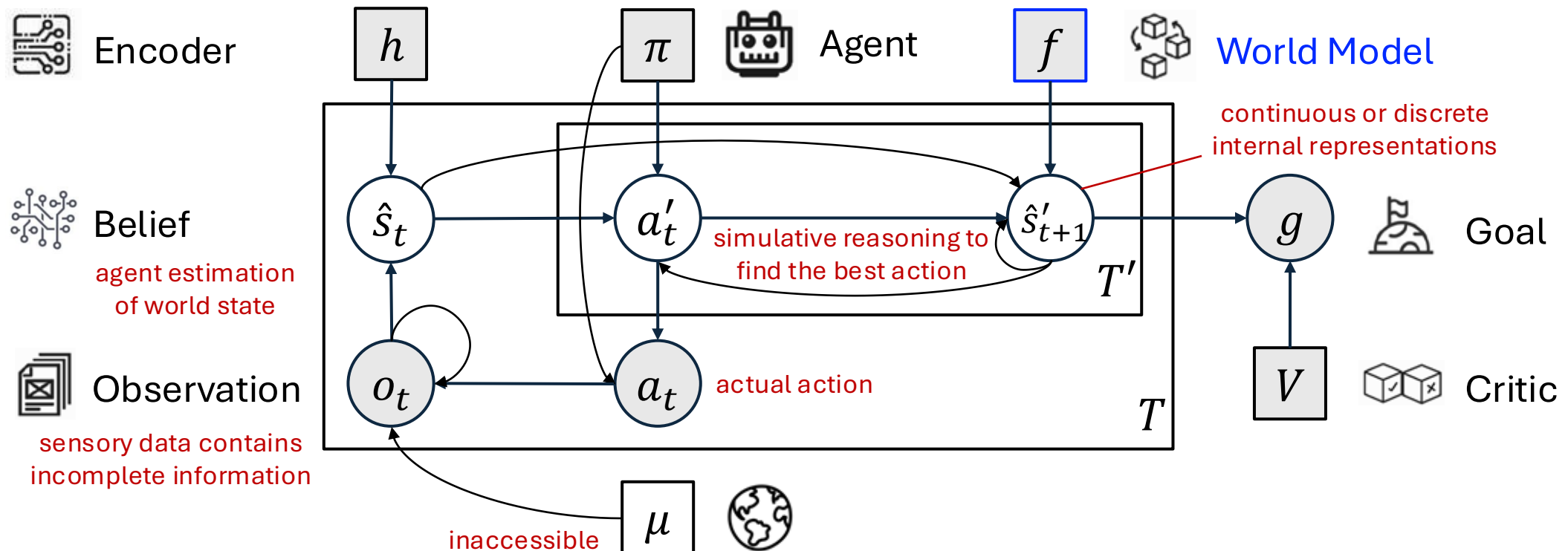
World model is necessary for general decision-making

However, the world state s and universe μ are not always available to experience/experiment.

A **world model** is thus crucial to predicting the environment's behavior for a general agent

Advantage: Allows *simulative reasoning* -
- based on all possible outcomes from hypothetical actions.

Coincides with reasoning pattern in *Dune*

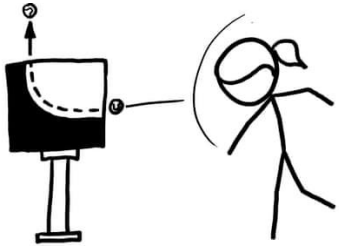


Roadmap for today

- 1. What is a world model? How should we build it?**
2. An example of world model for agentic reasoning
3. A general-purpose physical world model for simulative planning
4. An agentic LLM that self-regulates when and how to use planning

What is a world model?

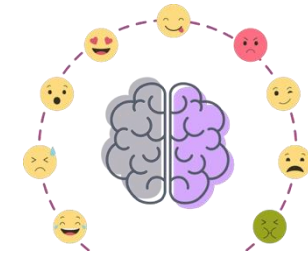
- A generative model that simulates possibilities in the world



Physical mechanics



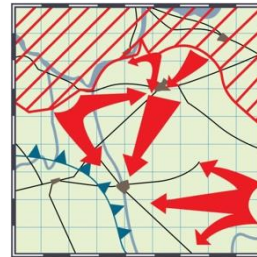
Embodied experience



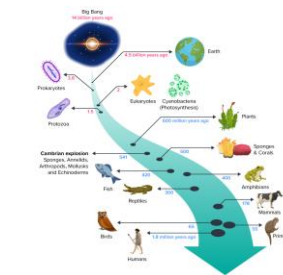
Emotional experience



Social experience



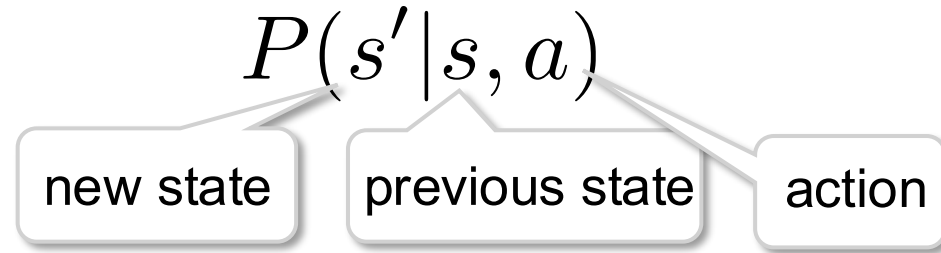
Mental world



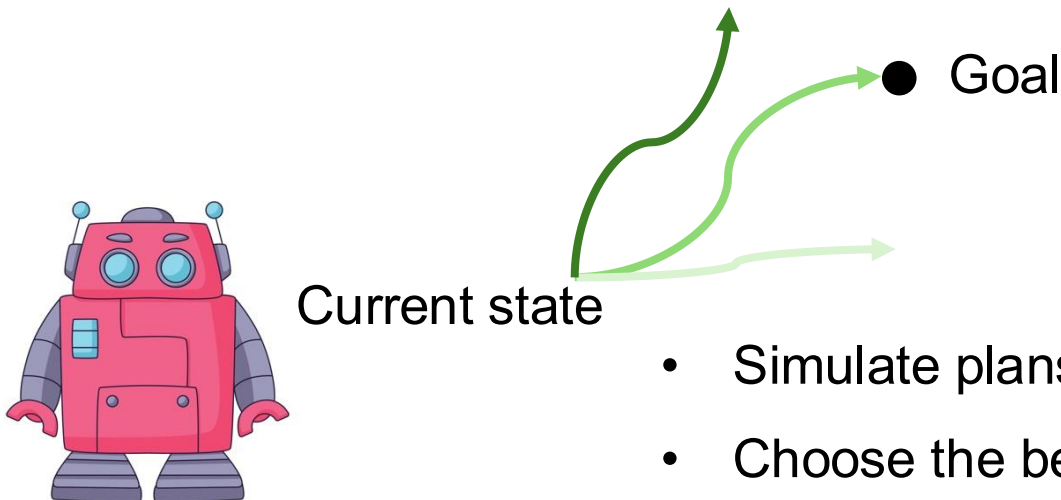
Evolutionary World

World model enables reasoning by simulation

- WM Output: Next world state conditioned on previous world state and action



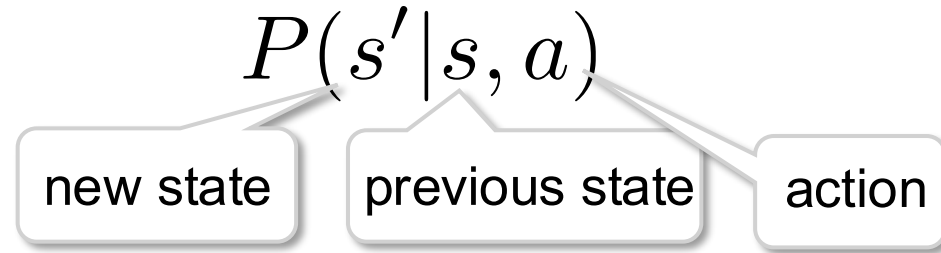
- When a machine does “Thought Experiments”: reasoning-via-simulation



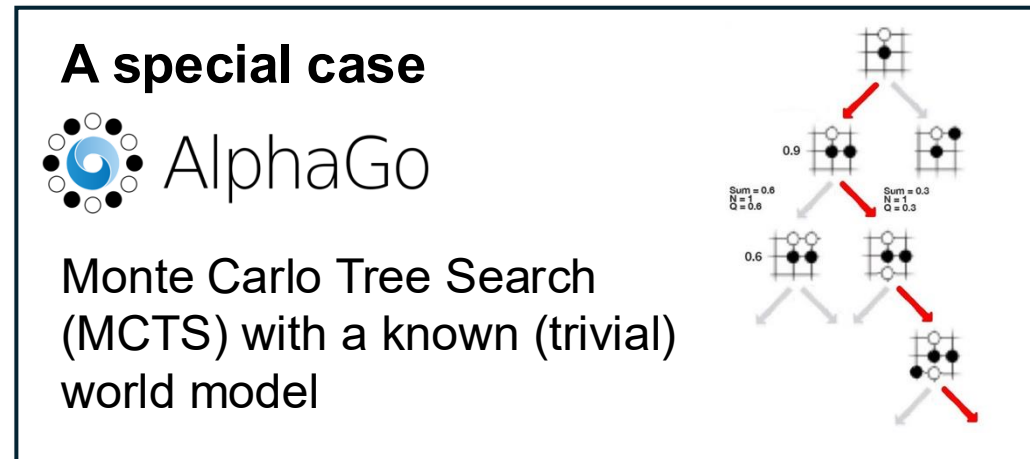
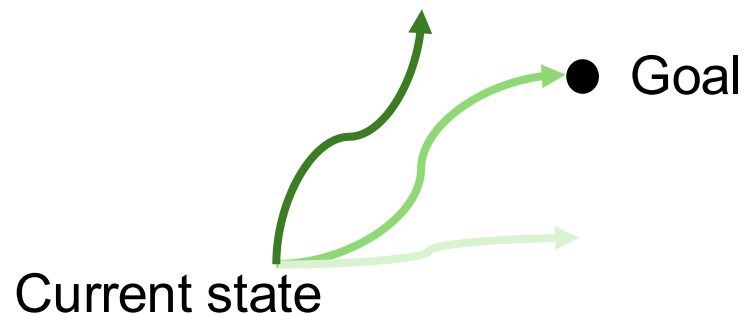
- Simulate plans with world model
- Choose the best plan

World model enables reasoning by simulation

- WM Output: Next world state conditioned on previous world state and action



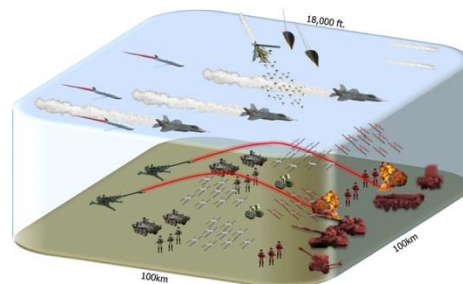
- When a machine does “Thought Experiments”: reasoning-via-simulation



Potential use cases for simulative reasoning



Self-driving with ensured safety by forecasting future street scenarios



Developing battle-winning tactics by anticipating the outcomes of actions



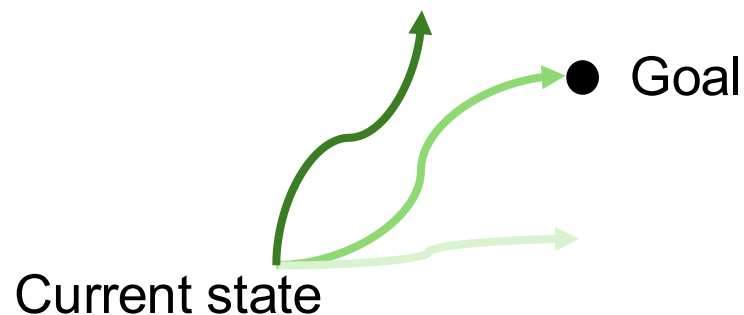
Billions of years in Earth's evolution

V.S.



Moments in a Martian civilization

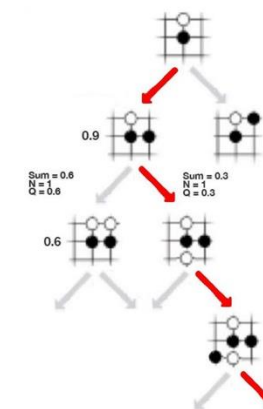
- When a machine does “Thought Experiments”: reasoning-via-simulation



A special case

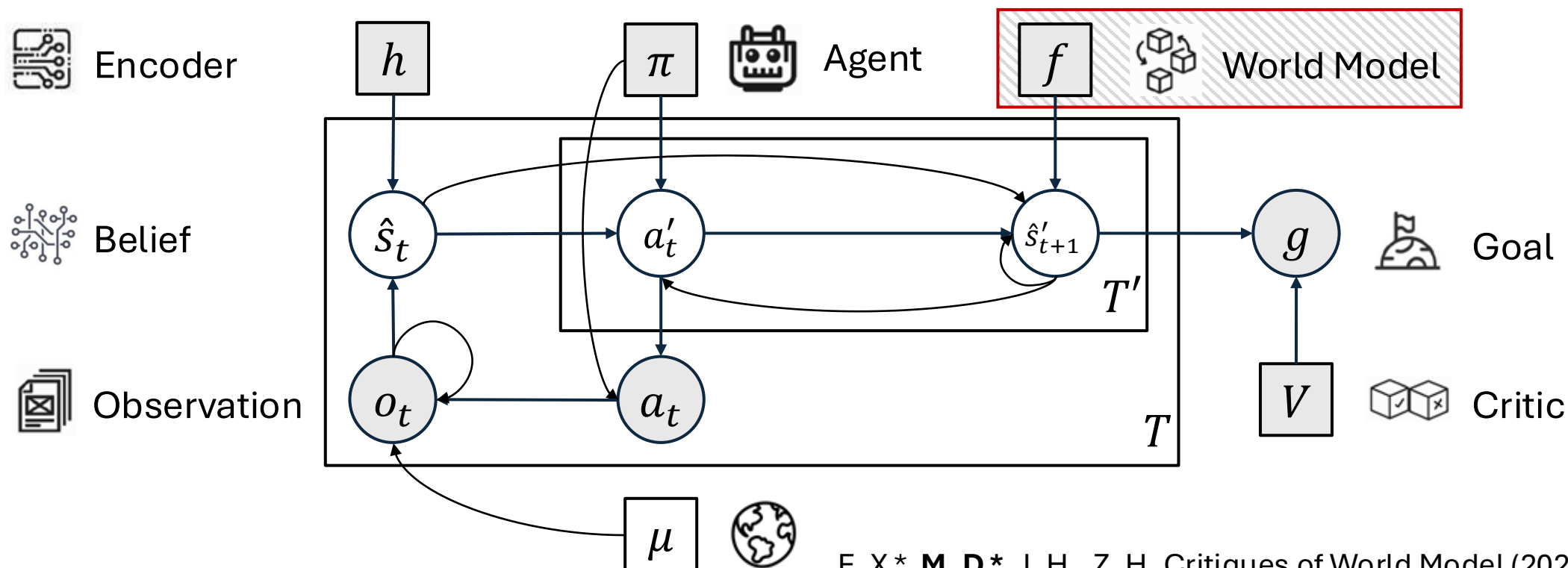


Monte Carlo Tree Search (MCTS) with a known (trivial) world model



Key to building general world model

1. General **representation** space for the latent world state
2. **Architecture** that effectively reasons over the representations
3. **Objective** that efficiently trains the world model
4. Methods for **using** the world model in a decision-making system



The world model landscape

- **Gaming world models: Genie-3, etc.**



- Simulate 3D environments with console-like controls
- Limited domain, consistency, and action control



- **3D scene world models: Marble, etc.**



- Stylized 3D scene generation and navigation
- Incomplete world: environment only, no other agents; do not allow interactions beyond simple navigation



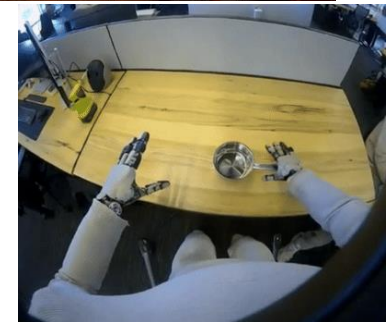
- **Physical world models: Cosmos-Predict-2.5, etc.**

- Specialized for auto-driving or specific robotic control



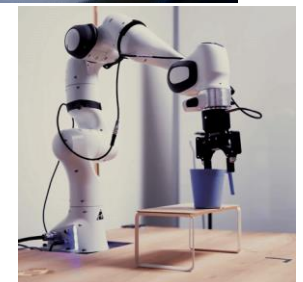
- **Video generation models: Sora (RIP), Veo 3, etc.**

- No interactions allowed, not qualified as world models for reasoning



- **Joint embedding predictive models: V-JEPA 2, etc.**

- Toyish environments and tasks; general usability unclear



➤ **World models are not video generators, but sandboxes for thought experiments**

➤ **Focus is on actionable understanding**

A frequently-heard school of thought regarding world model

1. **Sensory data > text**: bigger data volume
2. **Continuous embeddings > discrete tokens**: easier gradient-based optimization
3. **Autoregressive, generative models (e.g., LLMs) are doomed**: compounding errors
4. **Data reconstruction objectives are wrong**: irrelevant details
5. **Model-predictive control (MPC) > reinforcement learning (RL)**: RL needs many trials

Critiques of World Modeling

Critiques of World Models

Eric Xing^{◊,†,*}, Mingkai Deng^{◊,†,*}, Jinyu Hou^{◊,†}, Zhiting Hu^{◊,‡}

[◊]Institute of Foundation Models, Mohamed bin Zayed University
of Artificial Intelligence

[†] School of Computer Science, Carnegie Mellon University

[‡]Halıcıoğlu Data Science Institute, UC San Diego

eric.xing@mbzuai.ac.ae, {mingkaid, jinyuhou}@cs.cmu.edu,
zh019@ucsd.edu

July 2025

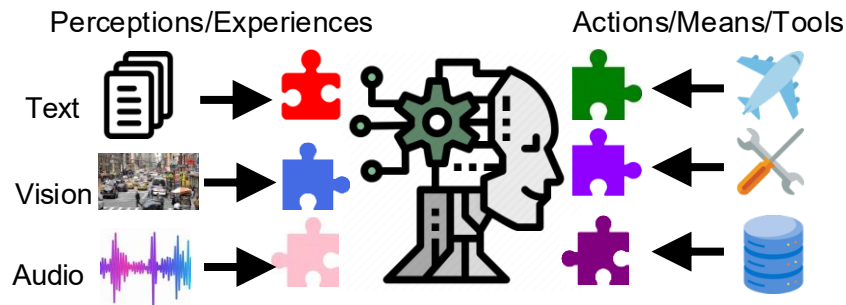
Abstract

World Model, the supposed algorithmic surrogate of the real-world environment which biological agents experience with and act upon, has been an emerging topic in recent years because of the rising needs to develop virtual agents with artificial (general) intelligence. There has been much debate on what a world model really is, how to build it, how to use it, and how to evaluate it. In this essay, starting from the imagination in the famed Sci-Fi classic *Dune*, and drawing inspiration from the concept of “hypothetical thinking” in psychology literature, we offer critiques of several schools of thoughts on world modeling, and argue the primary goal of a world model to be *simulating all actionable possibilities of the real world for purposeful reasoning and acting*. Building on the critiques, we propose a new architecture for a general-purpose world model, based on hierarchical, multi-level, and mixed continuous/discrete representations, and a generative and self-supervised learning framework, with an outlook of a Physical Agents and

Key desiderata in building a WM: data, representation, model, loss, usage

Data: Information density, not just volume

- Use **all modalities** of experience
 - **Why:** Different layers of experience
 - **What (example):** Text, videos, embodied interactions



- **How (example):** Dense video captions (e.g., motions, events, new objects)

?

- WM needs to be trained from sensory inputs
- **Why:** More raw data than text

!

Text has **higher information density** than video

Language captures **human's abstraction** of the world

Language encodes concepts like goals, norms, regrets, and justice

Natural interface for reasoning and action

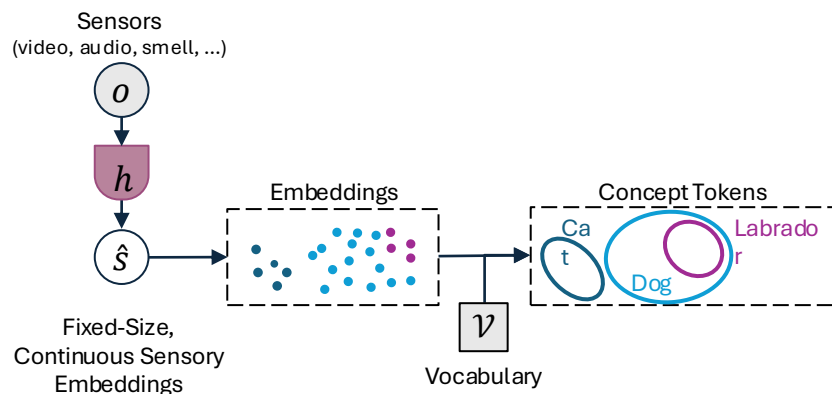
Representation: Continuous? Discrete? Or both?

- Need **stateful, semantic latent space** that supports memory and reasoning

- Continuous / discrete depends on what works, not just ease of learning
- Language (e.g., LLM) and learned tokenizers (e.g., VQ-VAE) are effective for modeling concepts

- How:** Mixed representation

- Example:** Continuous embeddings + discrete tokens



?

- World states should be represented using continuous embeddings
- Why:** To enable gradient-based optimization

!

Raw sensory inputs are noisy and highly variable

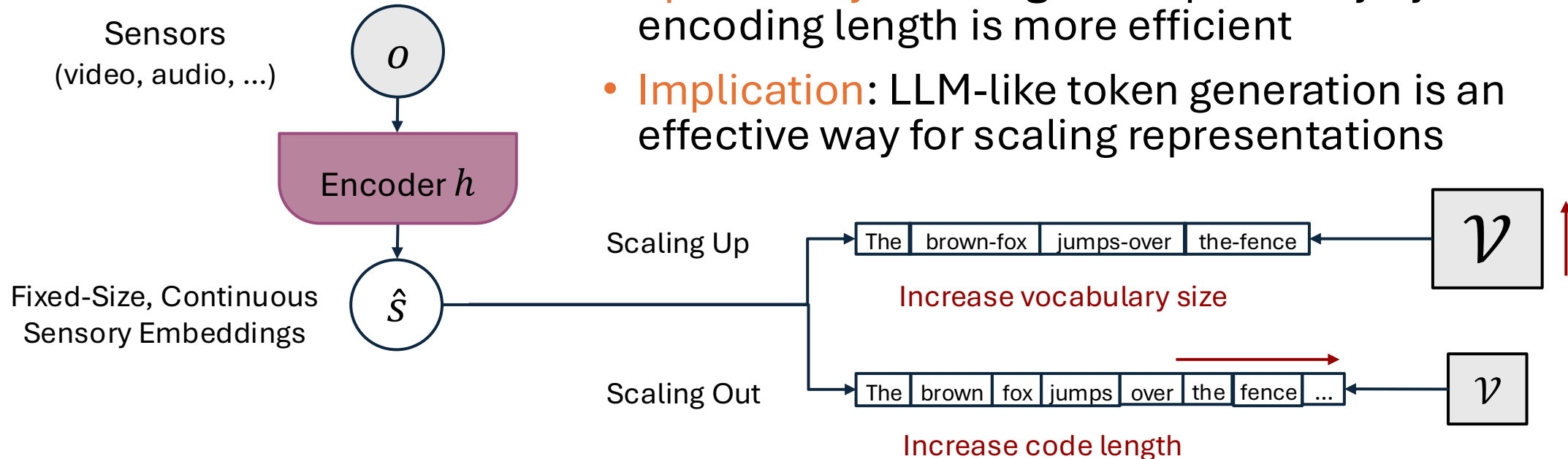


Action classification accuracy (50 classes):
JEPA: **58%** v.s. standard VLM: **74%**

Humans reason by compressing perception into discrete concepts

Why do discrete representations work?

- Intuitively, discrete tokens should lose information due to compression
- **However**: Theory shows we can either learn a larger vocabulary or find a longer language expression
- **Specifically: Scaling out** expressivity by increasing encoding length is more efficient
- **Implication**: LLM-like token generation is an effective way for scaling representations



Theoretical result on scaling discrete codes (a.k.a. language)

Theorem 1 (Completeness of Language Representation). *Assume real-valued inputs $\mathbf{x} = [x_1, \dots, x_T]$, where $x_t \in \mathbb{R}^D$ and $\|x_t\| < K$. For any $\epsilon > 0$, there exists a language with vocabulary $\mathcal{V}$, maximal length $N < \infty$, and a mapping function $f_\epsilon: \mathbb{R}^{TD} \rightarrow \mathcal{V}^N$ such that for all $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^{TD}$, $\|\mathbf{x} - \mathbf{x}'\| > \epsilon \Rightarrow f_\epsilon(\mathbf{x}) \neq f_\epsilon(\mathbf{x}')$.*

Explanation: If you have a sequence of continuous sensor readings or data points, no matter how small a difference you want to be able to distinguish between two sequences, you can always create a language (a system of words or symbols) that can represent these sequences uniquely

Key Insight: Way more efficient to scale out

- **Proof Sketch.** We will prove the contrapositive:

$$f_\epsilon(\mathbf{x}) = f_\epsilon(\mathbf{x}') \Rightarrow \|\mathbf{x} - \mathbf{x}'\| \leq \epsilon$$

Specifically, we propose two ways to scale the discrete code:

Case 1 (Learning a Modality Tokenizer). Keep the code length constant at T , increase the vocabulary size to $M_\epsilon = \lceil \sqrt{TD\tilde{K}\tilde{\epsilon}^{-1}} \rceil^D$ (i.e., **scaling up**)

The vocabulary size must grow in $\mathcal{O}((TD)^D)$, i.e., exponentially with the input size you'd like to capture!

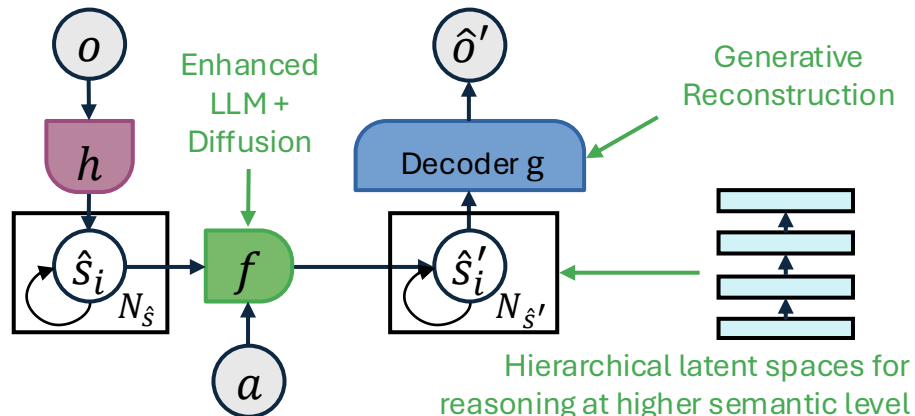
Case 2 (Finding a Language Expression). Keep the vocabulary size constant at M , increase the word sequence length to $N_\epsilon = TD \lceil \log_M \sqrt{TD\tilde{K}\tilde{\epsilon}^{-1}} \rceil$ (i.e., **scaling out**)

The sequence length need only increase in $\mathcal{O}(TD \log TD)$, which is much more manageable

In practice, it's possible to do both at the same time (i.e., continuously grow vocabulary while dynamically resizing the code)

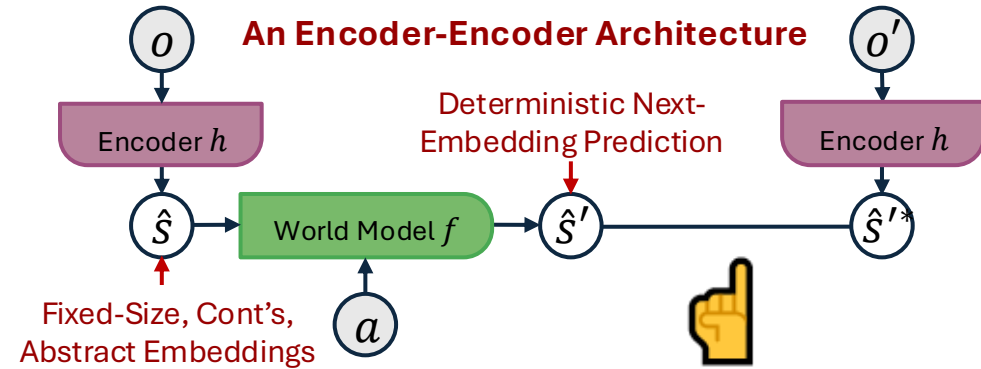
Architecture: Autoregressive generation is not the enemy

- **Alternative: Generative Latent Prediction (GLP)**
 - **Closed-loop**: Encoder $\rightarrow$ enhanced LLM backbone $\rightarrow$ decoder
- Mixed, hierarchical latent state
 - **Continuous**: via diffusion model for low-level perception
 - **Discrete**: modality-specific or language for mid-/high-level reasoning via LLM-based backbone
- No generation $\times$
Better structured generation $\checkmark$



?

- Abandon autoregressive generation; replace with JEPA-style encoder-encoder setup
- **Why**: 1) avoid error compounding, 2) absorb signal variability



!

Any world model, including JEPA, is autoregressive and generative

Encoder-encoder changes prediction target, but not the core problem

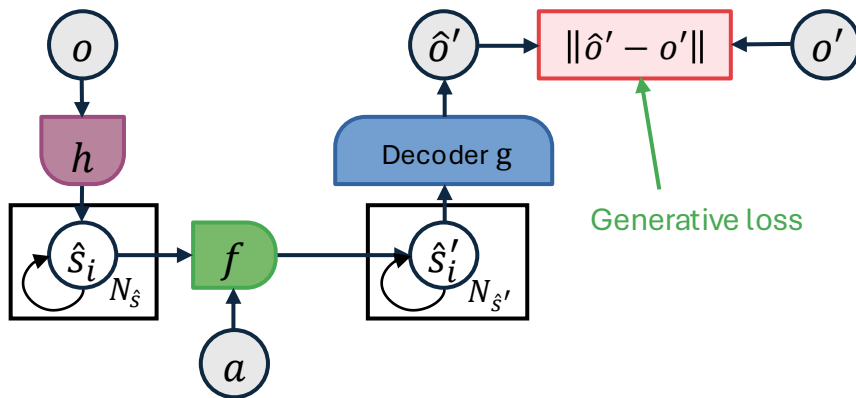
Latent-only prediction can introduce indeterminacy

Objective: Train against the world, not just the latent

- Use **generative loss** grounded in observed evidence:

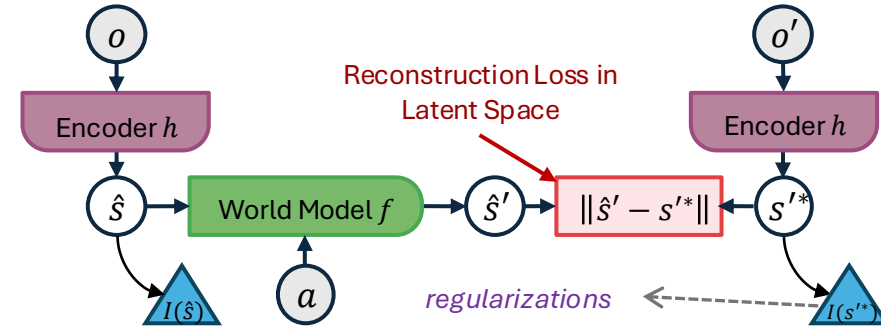
$$\mathcal{L}_{\text{gen}}(h, f, g) = \mathbb{E}_{(o, a, o') \sim \mathcal{D}} [\|g \circ f(h(o), a) - o'\|].$$

- Why:** Compared to latent loss, inherently prevents collapse and encourages meaningful latents
- See next slide for theoretical proposition



?

- Avoid data reconstruction; use energy-based next-latent prediction objectives
- Why:** more tractable training



!

Latent loss is a lower bound of generative loss plus a small roundtrip reconstruction error ϵ :

$$\mathcal{L}_{\text{latent}} \leq \mathcal{L}_{\text{gen}} + \epsilon$$

Encoder-decoder reconstruction error (small)

Task-relevant info missing from encoder

Minimizing latent loss thus doesn't guarantee consistency with observed data (proof in paper)

Theoretical result on latent reconstruction vs. generative loss

Proposition 1 (Collapse of Latent Reconstruction Loss). *Given functions $h: \mathcal{O} \rightarrow \mathcal{S}$, $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$, and latent reconstruction loss:*

$$\mathcal{L}_{\text{latent}}(h, f) = \mathbb{E}_{(o, a, o') \sim \mathcal{D}}[\|f(h(o), a) - h(o')\|],$$

There exists (h^, f^*) and $c \in \mathbb{R}^d$, such that $h^*(o) = c$ for all $o \in \mathcal{O}$ and $f^*(c, a) = c$ for all $a \in \mathcal{A}$, such that:*

$$\mathcal{L}_{\text{latent}}(h^*, f^*) = \min_{h, f} \mathcal{L}_{\text{latent}}(h, f).$$

Explanation: If you have an encoder and a world model, and you train it using the latent reconstruction loss, there is a cheat configuration for the model to minimize the loss while learning nothing about the true dynamics.

Proposition 2 (Non-collapse of Generative Loss). *Given functions $h: \mathcal{O} \rightarrow \mathcal{S}$, $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$, $g: \mathcal{S} \rightarrow \mathcal{O}$, and generative loss:*

$$\mathcal{L}_{\text{gen}}(h, f, g) = \mathbb{E}_{(o, a, o') \sim \mathcal{D}}[\|g \circ f(h(o), a) - o'\|],$$

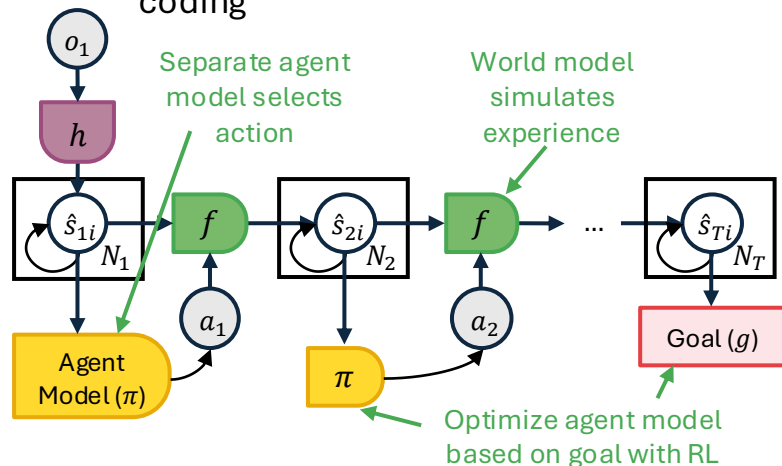
Assuming $\exists (o_1, a_1, o_2), (o_3, a_3, o_4) \in \mathcal{D}$ such that $o_2 \neq o_4$, then given (h', f') , fixed g' and $c \in \mathbb{R}^d$ such that $h'(o) = c \forall o \in \mathcal{O}$ and $f'(c, a) = c \forall a \in \mathcal{A}$, there exists $(\tilde{h}, \tilde{f})$ such that:

$$\mathcal{L}_{\text{gen}}(\tilde{h}, \tilde{f}, g') < \mathcal{L}_{\text{gen}}(h', f', g')$$

Explanation: If you add a decoder to the model and train it with the generative loss, assuming the data contains different next-observation targets, there will always be another set of encoder and world model that gets lower loss than the previous cheat configuration.

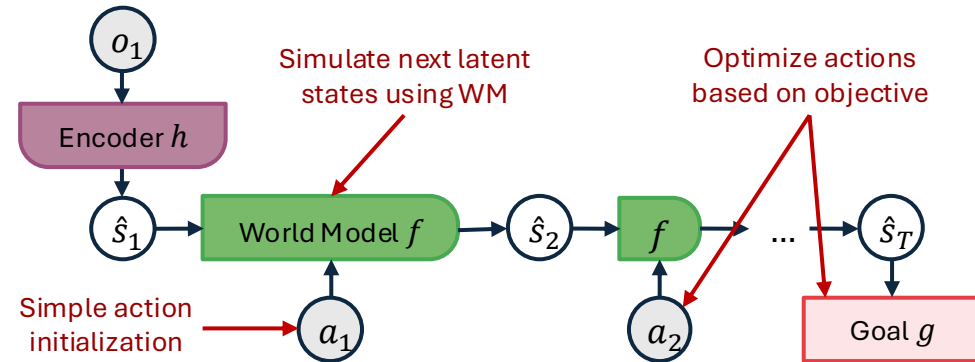
Usage: not just MPC, but learning from simulated experience

- **Learn from experience:** Use world model to generate trajectories
- **How:** Train agent from experience using RL
 - Agent model may include world model and MPC for precise planning
 - **Advantage:** No fixed planning horizon, flexible decision-making up to 100s of steps
 - **Examples:** o1/o3/R1 are special cases in math & coding



?

- Abandon reinforcement learning (RL); adopt **model-predictive control** (MPC)
- **Why:** fewer trials required during training



!

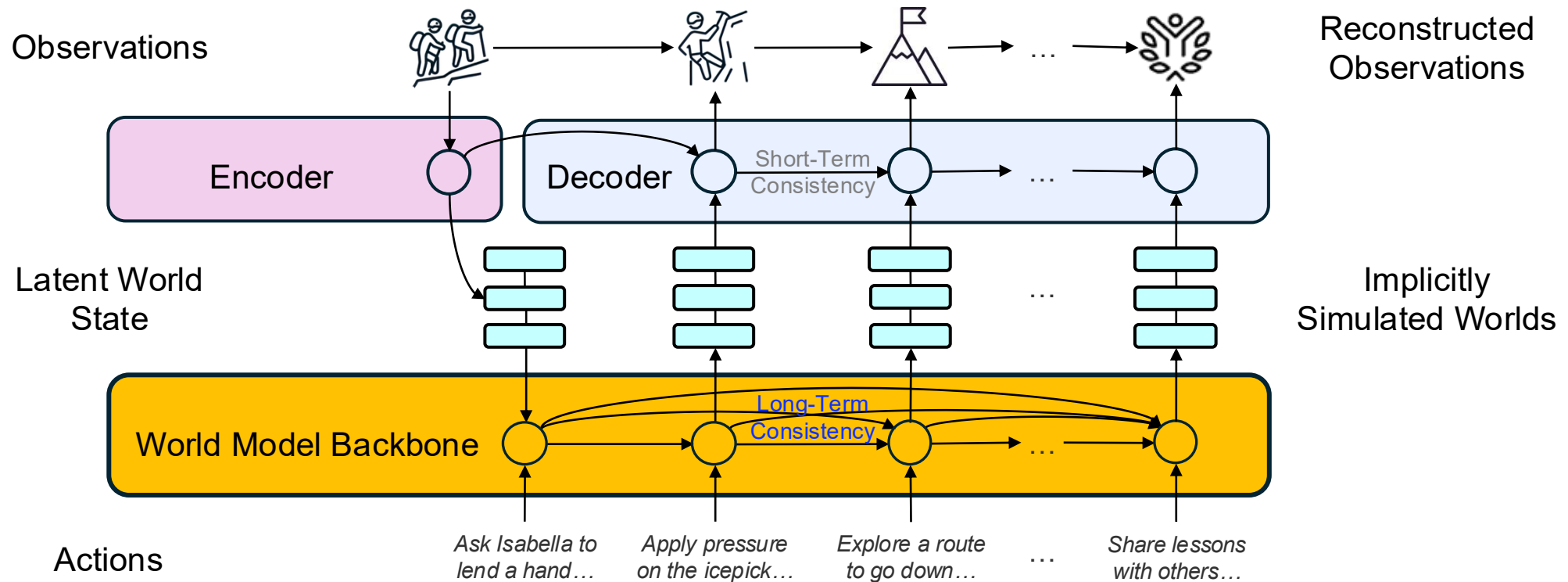
MPC can help, but it's computationally expensive

Usually plans only in fixed, short horizons

Promising in simple settings (e.g., Go) but struggles in real-world tasks (e.g., customer service) requiring flexibility

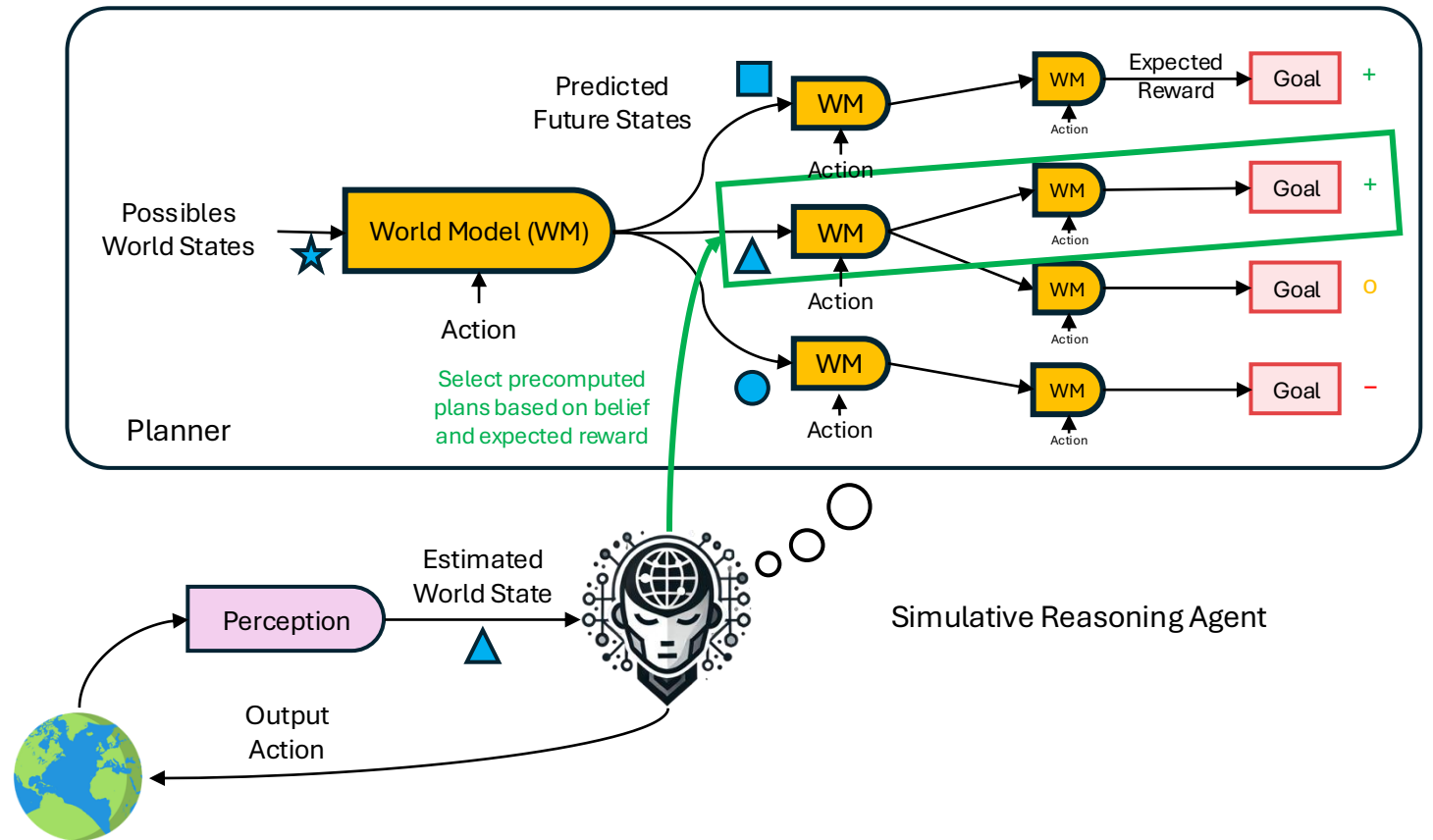
Summary: The GLP Architecture

- **Encode** multimodal observations into mixed latent world states
 - Tokenizer for discrete modality-specific or language tokens
 - Embedder for fine-grained continuous embeddings
- **Predict** future worlds hierarchically for agentic reasoning
 - Enhanced LLM for high-level reasoning and dynamics
 - Diffusion for low-level perceptual details
- **Decode** observations for supervision and external use



Towards agentic reasoning with the GLP world model

- World model functions as sandbox for simulation
- Given latent world state, uses WM to precompute and cache plans
- At decision time, select actions by consulting cache
- More closely mirrors human cognition: planning ahead and managing uncertainty
- Towards agents with richer forms of agency



Roadmap for today

1. What is a world model? How should we build it?
- 2. An example of world model for agentic reasoning**
3. A general-purpose physical world model for simulative planning
4. An agentic LLM that self-regulates when and how to use planning

An example of world-model-based reasoning for LLM agents

SIMURA: A World-Model-Driven Simulative Reasoning Architecture for General Goal-Oriented Agents

Mingkai Deng^{◊,†,*} Jinyu Hou^{◊,†,*} Zhiting Hu^{◊,▲} Eric Xing^{◊,†}

[◊]Institute of Foundation Models, Mohamed bin Zayed University of Artificial Intelligence

[†] School of Computer Science, Carnegie Mellon University

[▲]Halicioğlu Data Science Institute, UC San Diego

{mingkaid, jinyuhou}@cs.cmu.edu, eric.xing@mbzuai.ac.ae

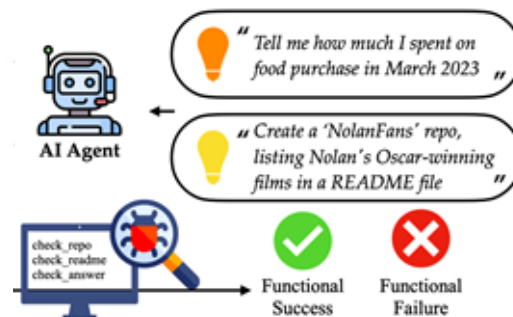
Abstract

AI agents built on foundation models hold enormous promise. Current practice, however, focuses on a one-task-one-agent approach, which not only falls short of scalability and generality, but also faces practical limitations from black-box autoregressive reasoning, where decisions unfold token by token without explicit simulation or counterfactual evaluation of outcomes. Humans, on the other hand, reason and plan by mentally simulating the consequences of actions within an internal model of the world – a capability that supports flexible, goal-directed behavior across diverse contexts. Moving towards a more general and powerful AI agent, we introduce SIMURA, a goal-oriented architecture for generalized agentic reasoning. Based on a principled formulation of an optimal agent in any general environment, SIMURA addresses the limitations of black-box autoregressive reasoning by incorporating the world model for planning via simulation. Our

At that time, we already had bunch of LLM-based agents for different tasks (usually not very good)



Social Simulation



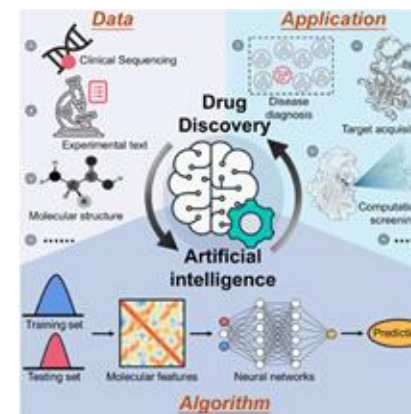
Web Automation



Software Engineering



Team Collaboration

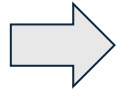


Scientific Discovery

Existing work at that time typically develop architectures for specific environments



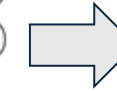
Social Simulation



Social Agent



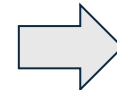
Web Automation



Web Agent



Team Collaboration



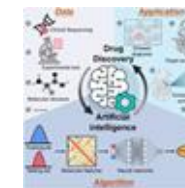
Multi-Agent Workflows



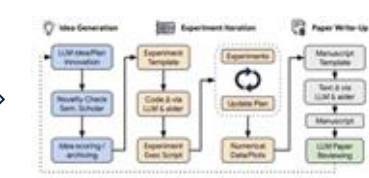
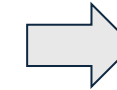
Software Engineering



Software Agent

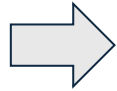


Scientific Discovery

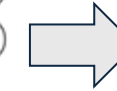


Science Agent

Existing work at that time typically develop architectures for specific environments



Social Simulation



Web Agent

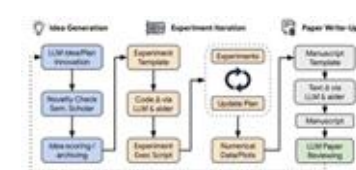
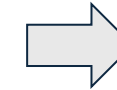
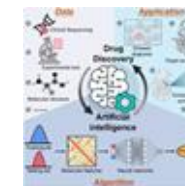
Limits of current architectures

- *Specialized to one task, hard for multi-task broad usage*
- *No clear general-purpose agent architecture for cross-task transfer*
- *LLM free-form reasoning can be locally myopic*



Software Engineering

Software Agent



Scientific Discovery

Science Agent

Why not just let a LLM generate any CoT?

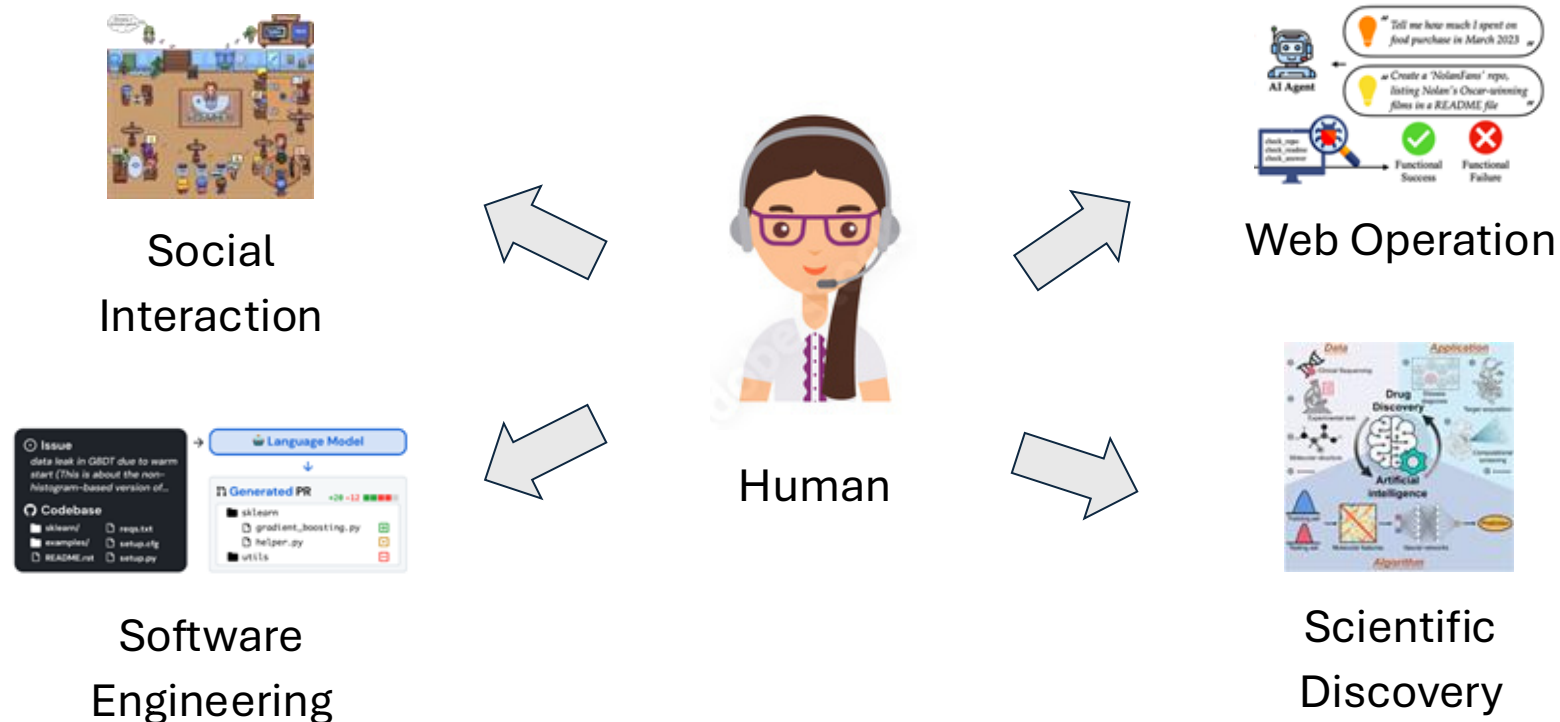
- We name letting LLM generate free-form thoughts “black-box reasoning”
- Given context x , black-box autoregressive reasoning iteratively predicts the next abstract internal state z_t as:

$$p(z_t | z_{<t}, x)$$

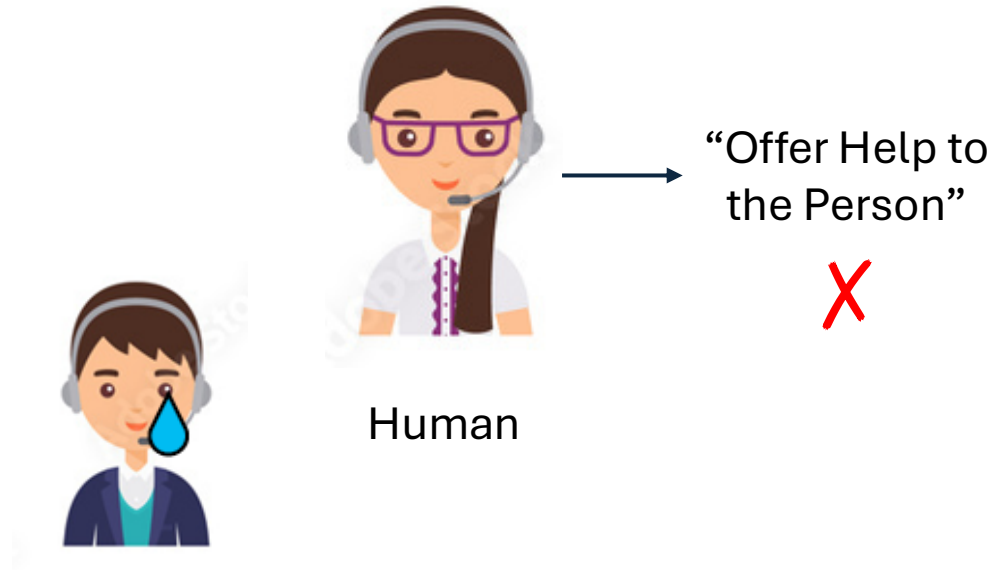
- No explicit modeling and simulation of future outcomes
- Even strong models can remain vulnerable to locally myopic decisions or error accumulation over extended trajectories

Failure: Reasoning error	
ReAct	
Question:	What was the middle name of the actress who was cast alongside Harvey Fierstein, Linda Hart, Dick Latessa, Clarke Thorell, Mary Bond Davis, Laura Bell Bundy, Matthew Morrison, Corey Reynolds, and Marissa Jaret Winokur in Hairspray?
Thought 1:	I need to search Harvey Fierstein, Linda Hart, Dick Latessa, Clarke Thorell, Mary Bond Davis, Laura Bell Bundy, Matthew Morrison, Corey Reynolds, and Marissa Jaret Winokur, find the actress that was cast alongside them in Hairspray, then find its middle name.
Action 1:	Search[Harvey Fierstein]
Observation 1:	Harvey Forbes Fierstein ...
Thought 2:	Harvey Fierstein is an actor, playwright, and screenwriter. I need to search Linda Hart next...
Action 2:	Search[Linda Hart]...

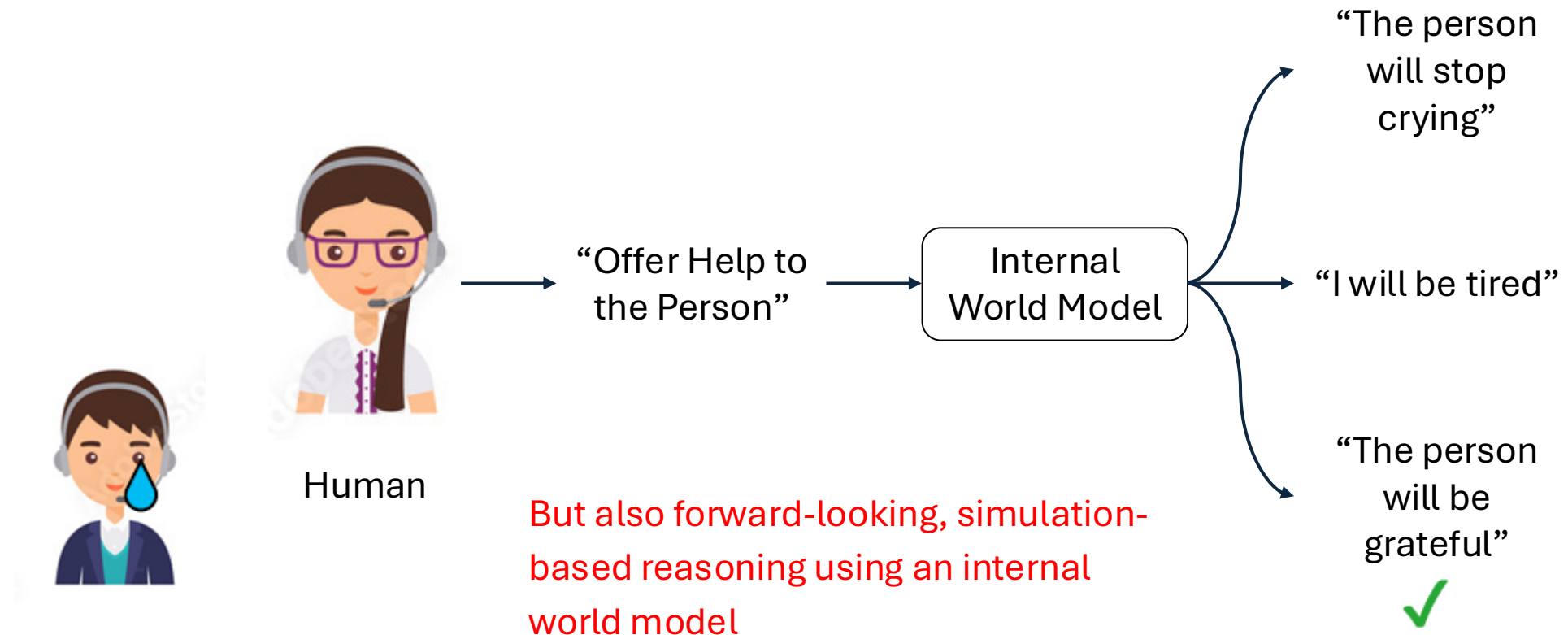
Humans are generalists with a single cognitive architecture, but can achieve goals in diverse environments



Human reasoning is not just linear, autoregressive reasoning

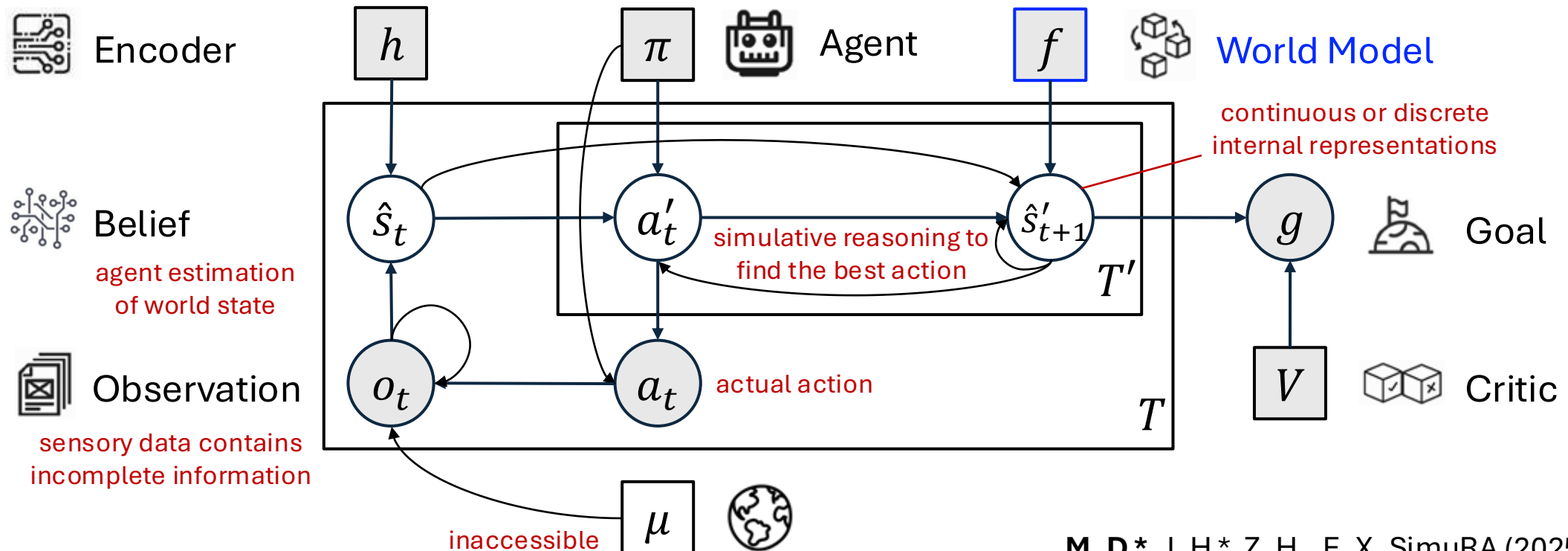


Human reasoning is not just linear, autoregressive reasoning



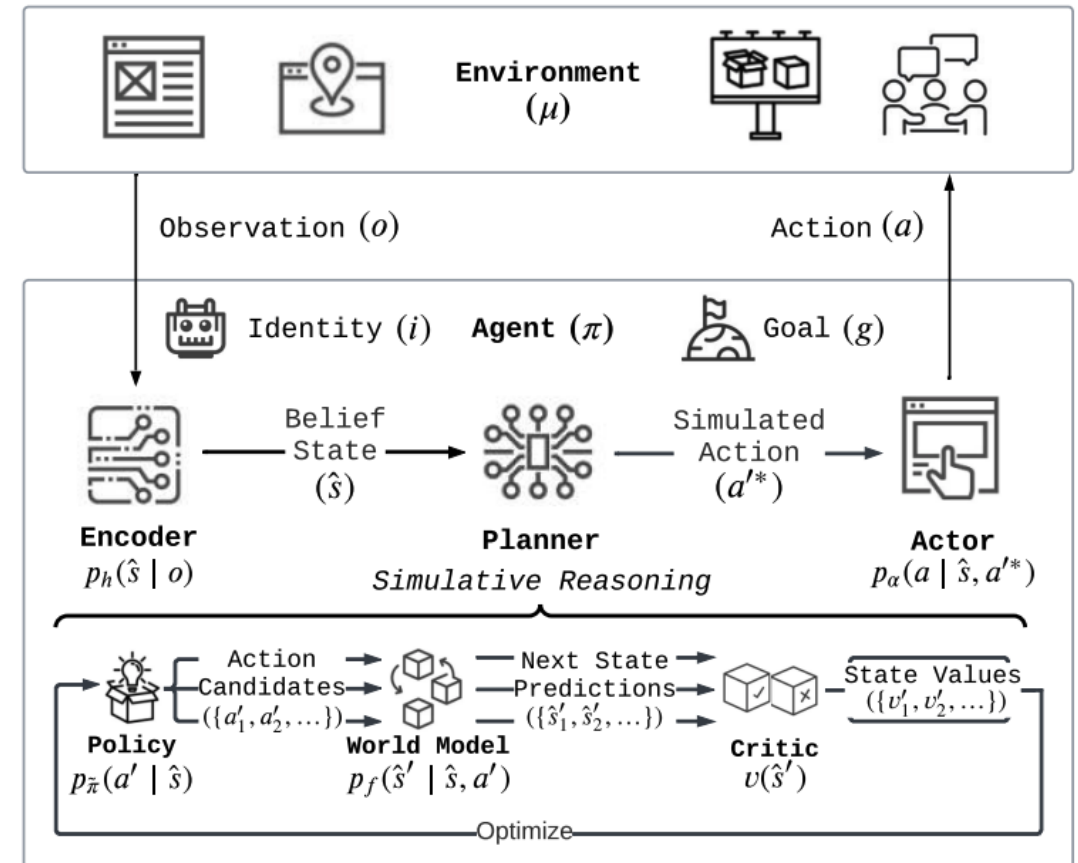
We build on our previous formulation of world model for simulative reasoning

$$\pi_f^*(\hat{s}_t) = \underbrace{\arg \max_{a_{t:T-1}}}_{\text{possible actions}} \sum_{\hat{s}_{t+1:T}} \underbrace{\left(\sum_{k=t}^{T-1} \gamma_k r(g, \hat{s}_k) + \gamma_T V_{\pi, f}^g(\hat{s}_T) \right)}_{\text{goal progress}} \prod_{i=t}^{T-1} \underbrace{p_f(\hat{s}_{i+1} | \hat{s}_i, a_i)}_{\text{simulation with world model}}$$



And instantiate it into SimuRA, a more concrete architecture

- **Encoder:** Discrete, semantic state representation using natural language
 - Inferred using LLM to summarize observation
- **Planner:** Hierarchical planning via abstract actions
 - **Policy:** Propose actions in natural language (e.g., click('id') -> “click on button”)
 - **World model:** Predicts next state using natural language as latent state space
 - **Critic:** Evaluates goal progress
- **Actor:** Grounds abstract actions in environment action space



And instantiate it into SimuRA, a more concrete architecture

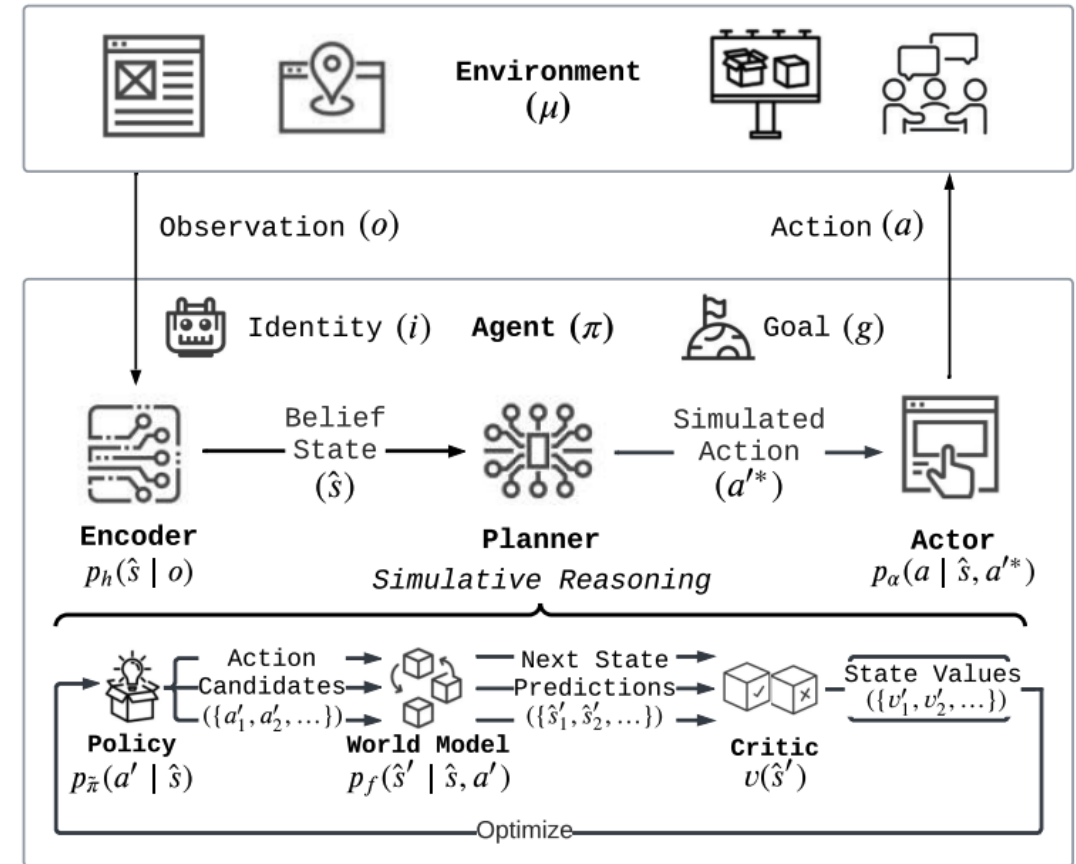
- Inference can be seen as solving multi-level optimization problem:

$$\hat{s}_t = \arg \max_{\hat{s}} \underbrace{p_h(\hat{s} | o_t)}_{\text{encoder}} \quad (\text{Perception})$$

$$a'_{t:T'-1}^* = \arg \max_{\underbrace{a'_{t:T'-1}}_{\text{sampled from policy } \pi}} \sum_{\hat{s}_{t+1:T'}} \underbrace{v(\hat{s}_{T'})}_{\text{critic}} \prod_{k=t}^{T'-1} \underbrace{p_f(\hat{s}_{k+1} | \hat{s}_k, a'_k)}_{\text{world model}} \quad (\text{Planning})$$

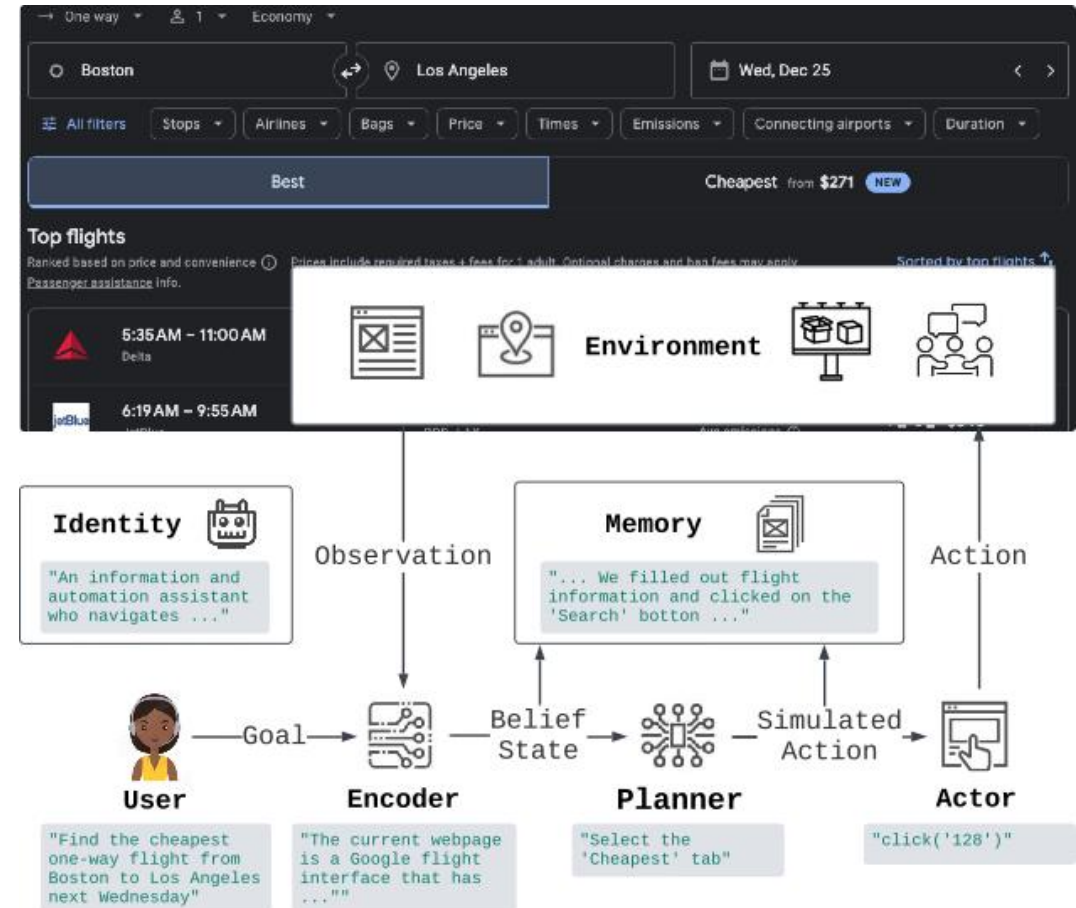
$$a_t = \arg \max_a \underbrace{p_\alpha(a | \hat{s}_t, a_t'^*)}_{\text{actor}} \quad (\text{Acting})$$

- Perception and Acting:** Infer by sampling
- Planning:** Maximize by tree search algos
- Implementation:** Use LLMs due to versatility; next-token prediction training also similar to world modeling

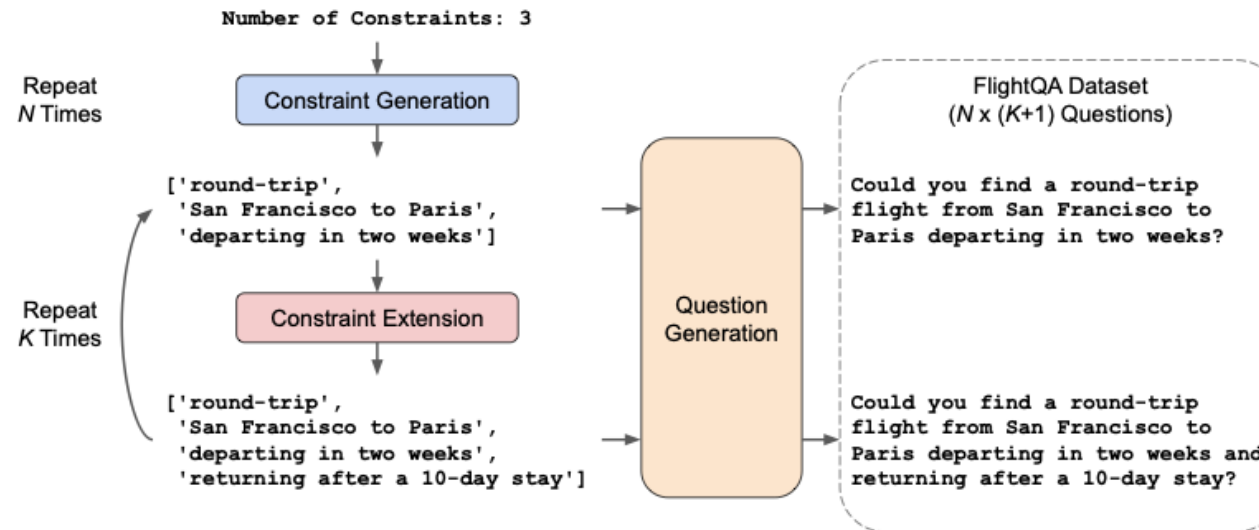


Experiments as web browsing agent

- Access to a Playwright-based browser and 13 types of actions
- **Tasks:**
 - Complex website navigation (FlightQA)
 - Multi-hop, multi-website QA (FanoutQA)
 - General web automation (WebArena)
- **Metric:** Success rate
- **Baselines:**
 - OpenHands BrowsingAgent
 - SimuRA without WM-based planning (autoregressive planning)



Generation of FlightQA dataset

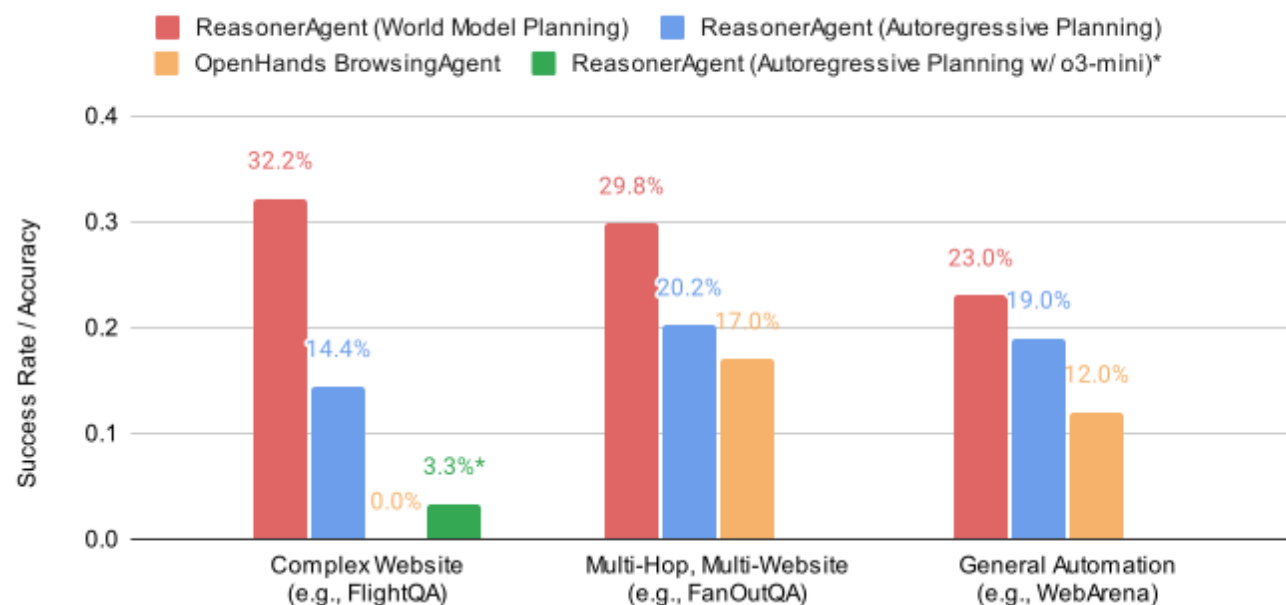


- Prompt LLM to generate lists of constraints
- Then iteratively add constraints to the lists one by one
- Finally, convert each constraint list into question in natural language
- **Advantage:** Enables systematic analysis of model generalization ability
- Answers question “what if I add just one more condition to the question”

Overview of results

- SimuRA clearly improves over baselines in all 3 categories of tasks
- Full SimuRA improves complex website navigation from 0% (BrowsingAgent) to 32.2%
- Simulative planning improves over matched autoregressive planning baseline by 124% (14.4 -> 32.2%)
- Surprisingly, strong reasoning LLMs like o1 and o3-mini performed closed to 0%
- Suggests the limit of RL-trained black-box reasoning

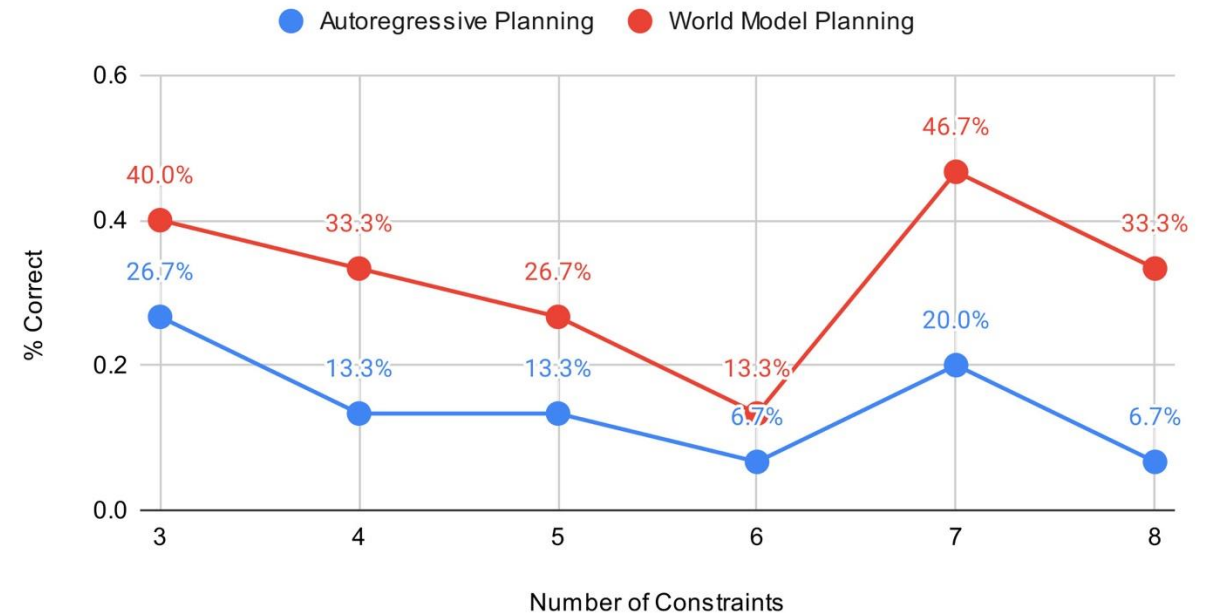
Web Browsing Performance vs. Agent Architecture



Counterfactual analysis of reasoning behavior

- We generated FlightQA questions with constraint control
- Enables answering “what if I added one more constraint?”
- Analysis shows simulative planning outperforms across # constraints
- Suggests improved reasoning / generalization ability

% Correct vs Number of Constraints

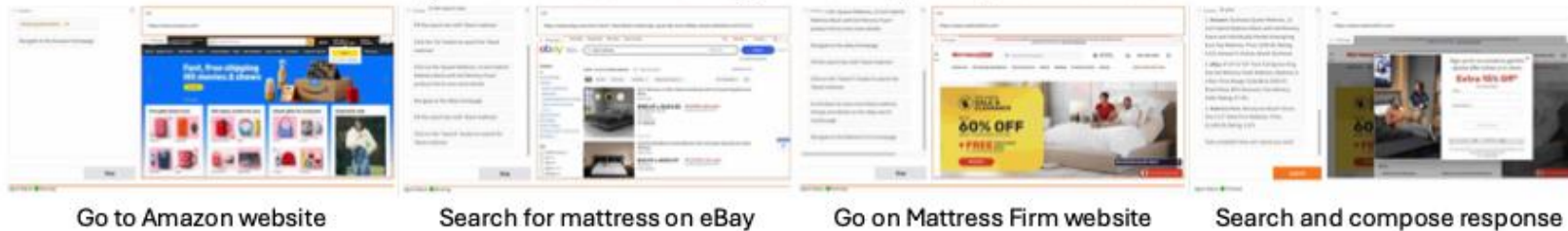


Demos

Flight Searching: “I’d like to find a round-trip ticket from Chicago to Dubai next month, traveling in economy class, with non-stop flights only, departing after 8 AM, and returning within two weeks.”



Online Shopping: “I want to buy a black mattress. Can you look at Amazon, eBay, and Mattress Firm and give me one good option from each?”



News Research: “Find and summarize one article from each of these local news outlets on Trump’s inauguration: Times of San Diego, The Tennessee Tribune, and MinnPost?”



Roadmap for today

1. What is a world model? How should we build it?
2. An example of world model for agentic reasoning
3. **A general-purpose physical world model for simulative planning**
4. An agentic LLM that self-regulates when and how to use planning

From video generation to action-conditioned world simulation

PAN: A World Model for General, Interactable, and Long-Horizon World Simulation

PAN Team, Institute of Foundation Models

Mohamed bin Zayed University of Artificial Intelligence

Overview

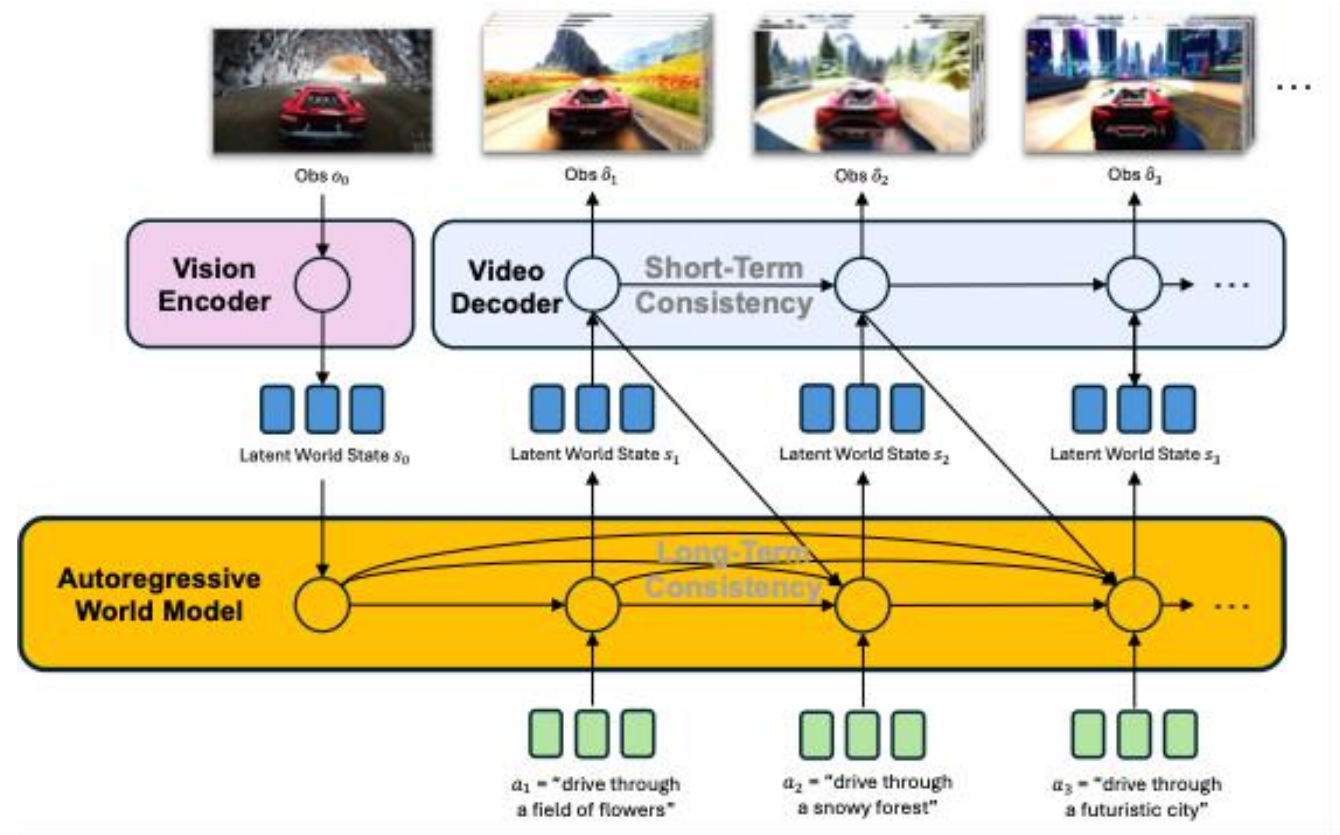
A world model enables an intelligent agent to imagine, predict, and reason about how the world evolves in response to its actions, and accordingly to plan and strategize. While recent video generation models produce realistic visual sequences, they typically operate in the prompt-to-full-video manner without causal control, interactivity, or long-horizon consistency required for purposeful reasoning. Existing world modeling efforts, on the other hand, often focus on restricted domains (e.g., physical, game, or 3D-scene dynamics) with limited depth and controllability, and struggle to generalize across diverse environments and interaction formats. In this work, we introduce PAN, a general, interactable, and long-horizon world model that predicts future world states through high-quality video simulation conditioned on history and natural language actions. PAN employs the Generative Latent Prediction (GLP) architecture that combines an autoregressive latent dynamics backbone based on a large language model (LLM), which grounds simulation in extensive text-based knowledge and enables conditioning on language-specified actions, with a video diffusion decoder that reconstructs perceptually detailed and temporally coherent visual observations, to achieve a unification between latent space reasoning (imagination) and realizable world dynamics (reality). Trained on large-scale video-action pairs spanning diverse domains, PAN supports open-domain, action-conditioned simulation with coherent, long-term dynamics. Extensive experiments show that PAN achieves strong performance in action-conditioned world simulation, long-horizon forecasting, and simulative reasoning compared to other video generators and world models, taking a step towards general world models that enable predictive simulation of future world states for reasoning and acting.

Motivations

- Modern video generators are visually impressive, but usually **open-loop**: prompt in, full video out
- They typically lack:
 - **causal control**
 - **interactivity**
 - **long-horizon consistency**
- Existing world models often handle only narrow domains like games, robotics, or 3D scenes

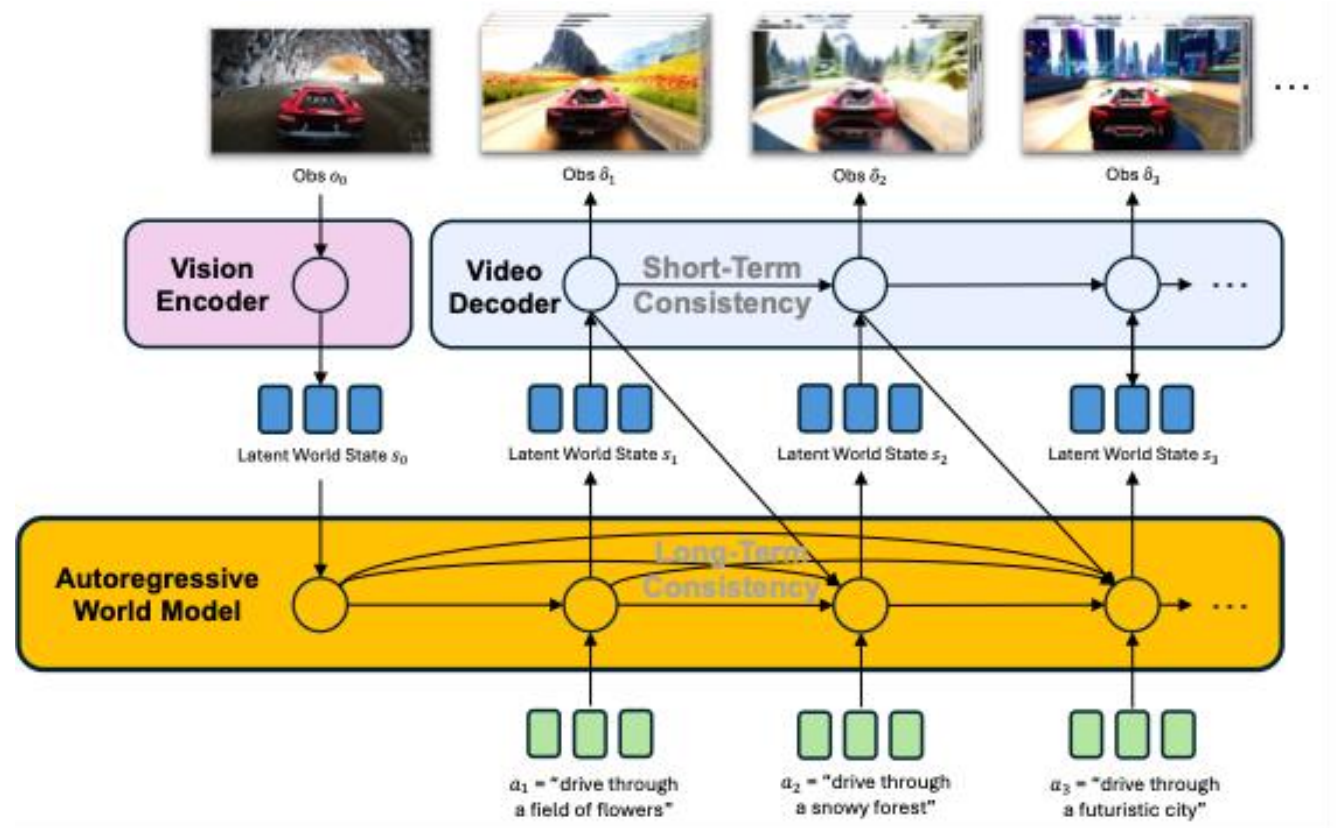
What PAN does

- Predicts future world states through video simulation
- **Inputs:**
 - Visual history
 - Natural language actions
- **Output:**
 - Future world trajectories
- Simulation should be visually detailed, causally responsive, temporally coherent

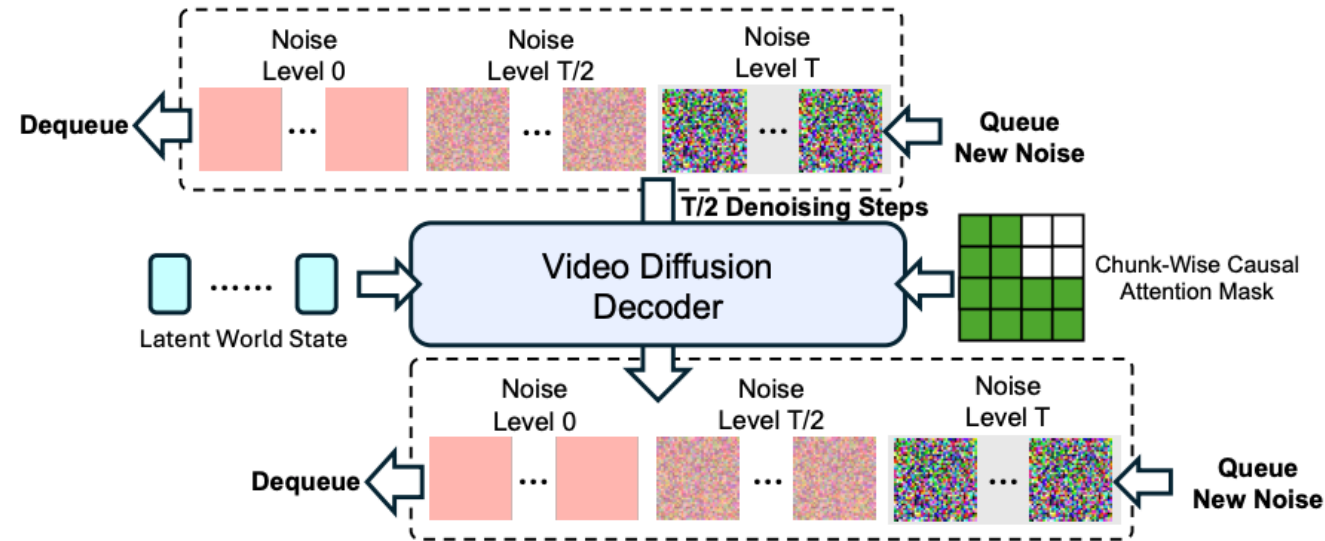


An instantiation of the GLP architecture

- **Encoder:** Vision tower of Qwen2.5-VL-Instruct
- **World model backbone:** Language model of Qwen2.5-VL-Instruct
- **Video decoder:** Wan-2.1-T2V-14B adapted through a Causal Swin-DPM mechanism
- **Generative loss:** Flow-matching
- **Training data:** Large-scale video-action pairs



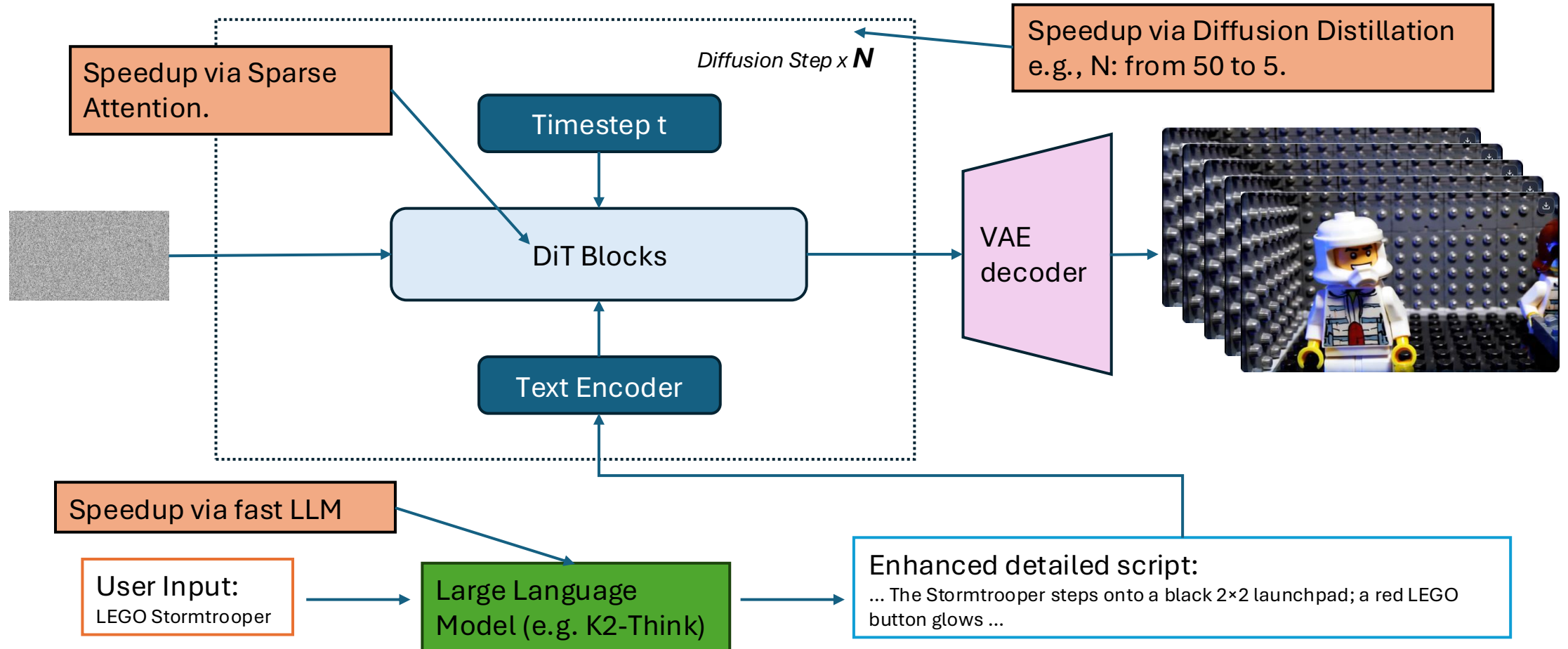
Video Diffusion Decoder: managing signal variability



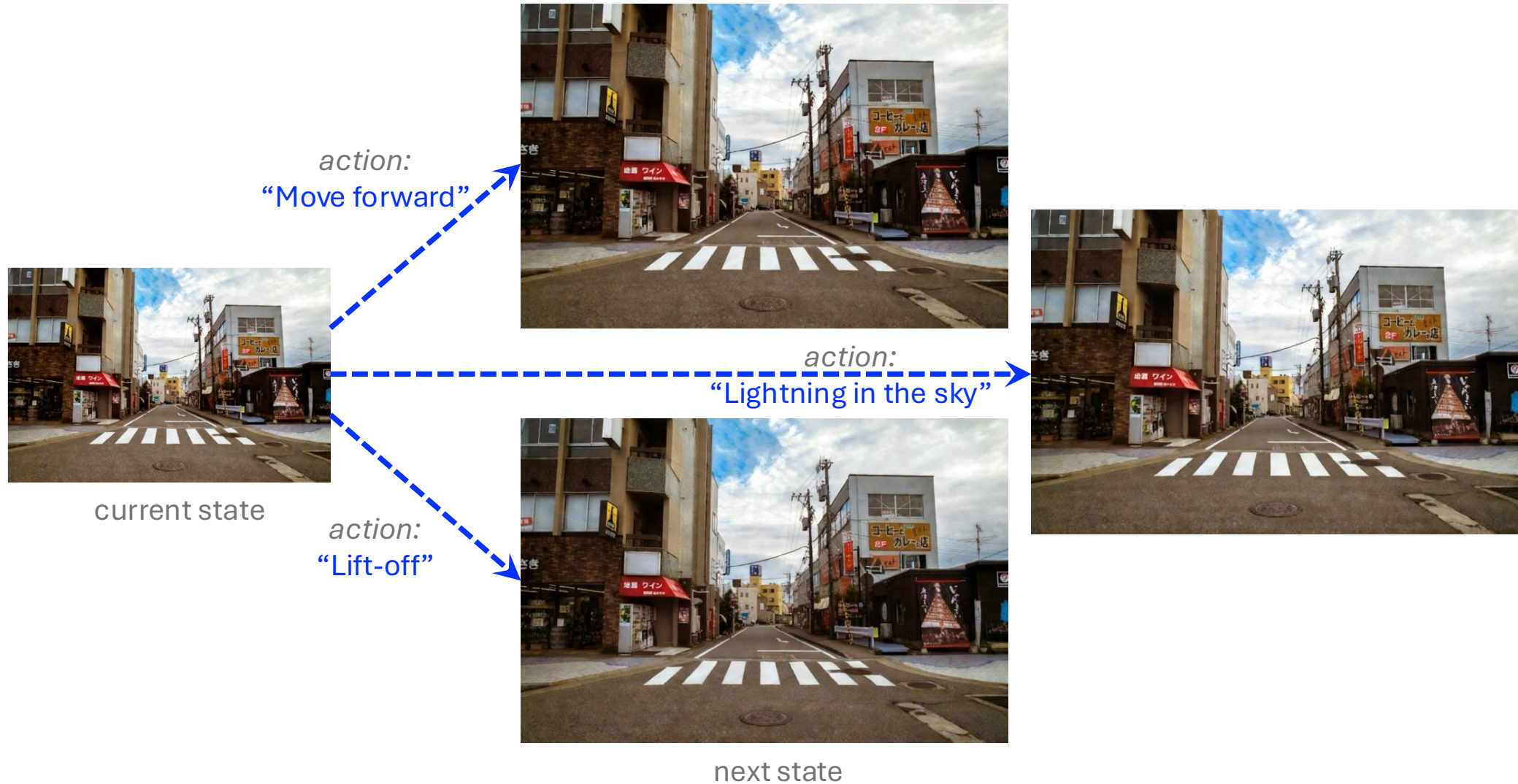
- Conditioned on partially noised representations of preceding chunks
- Dampen minor errors from previous generation and reduce error propagation over long horizon
- Encourage the model to focus on **stable world dynamics**, overcome the challenge for infeasible fine-grained details reconstruction

FastVideo Speedup

30 second 1080P video only needs only ~3 seconds, 10x faster than real time.



PAN: Simulating different hypothetical futures



PAN: Simulating infinitely-diverse real (and surreal) worlds & agents



Driving (normal)



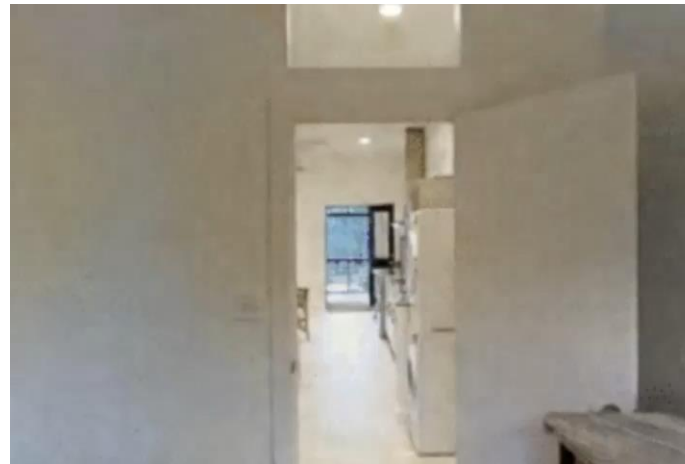
Drone (out-door)



“Clean up the table”



Driving (rare)



Drone (in-door)



“Put apple in the drawer and close it”

PAN: assessable and manipulatable latent states

Time: Input action

0-5s: Grasp the oil brush from the dish with the right arm.,

5-10s: Brush the baking pan with the oil brush held in the right arm.,

10-14s : Place the oil brush back into the oil dish.,

14-20s: Grasp the seasoning bottle with the right arm.,

20-25s: Pour the seasoning held in the right arm onto the corn flakes on the baking tray.,

25-28s: Place the seasoning bottle back on the table with the right arm.,

28-37s: Open the oven door with both hands.,

37-41s: Lift the baking tray containing corn flakes with the right arm.,

41-45s: Place the baking tray containing corn flakes at the oven door with the right arm.,

45-52s: Push the baking tray containing corn flakes into the oven with the right arm.,

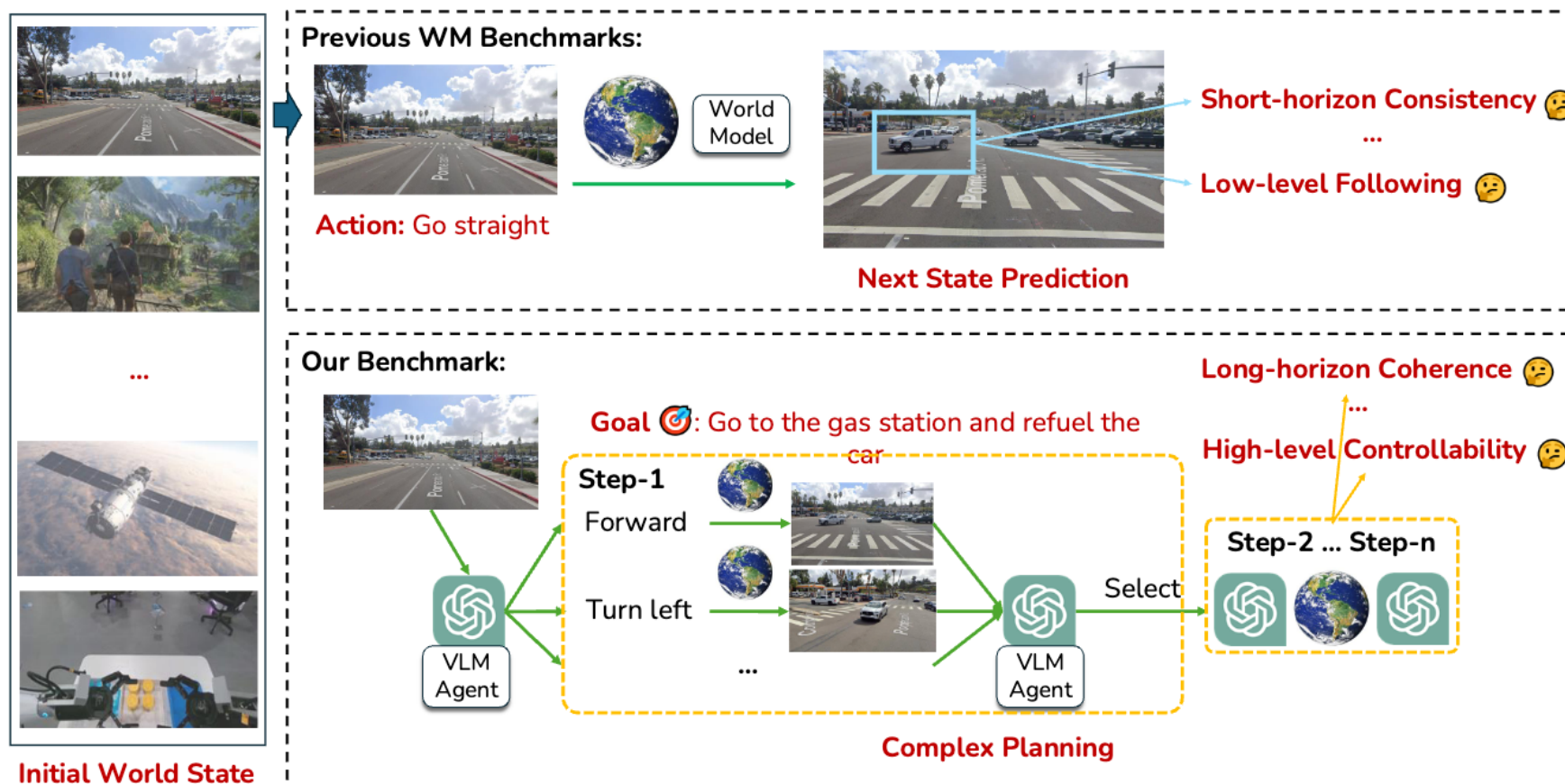
52-60s: Close the oven door with the right arm.,

60-66s: Press the oven start button with the right arm.



Evaluation Benchmark

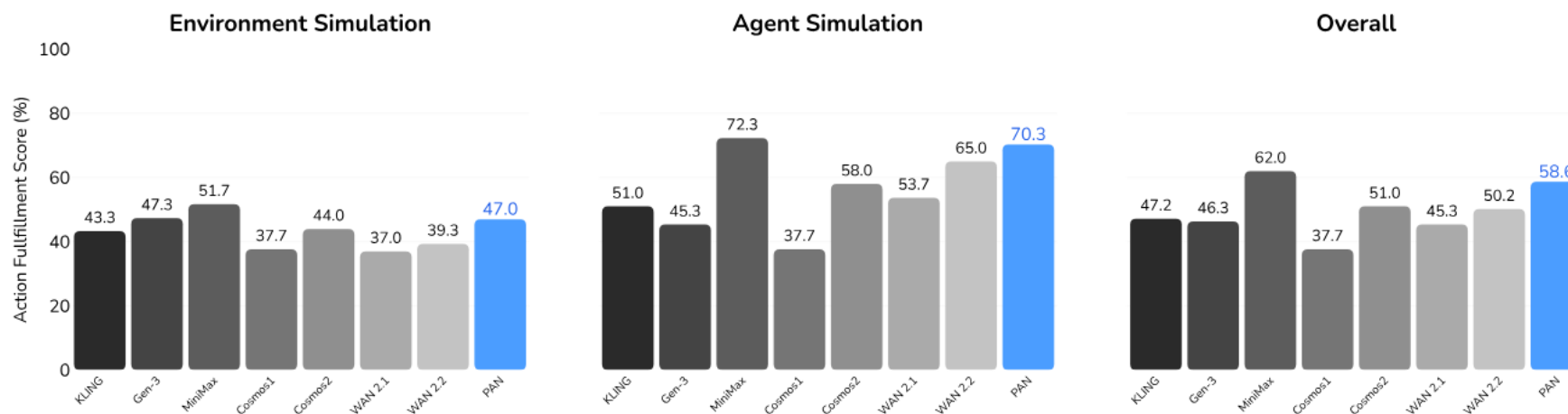
- Existing benchmarks focus on short-horizon metrics: frame level fidelity or motion realism
- Our benchmark evaluates the fundamental capabilities of a world model: coherent causal dynamics



Evaluation Benchmark - Metrics

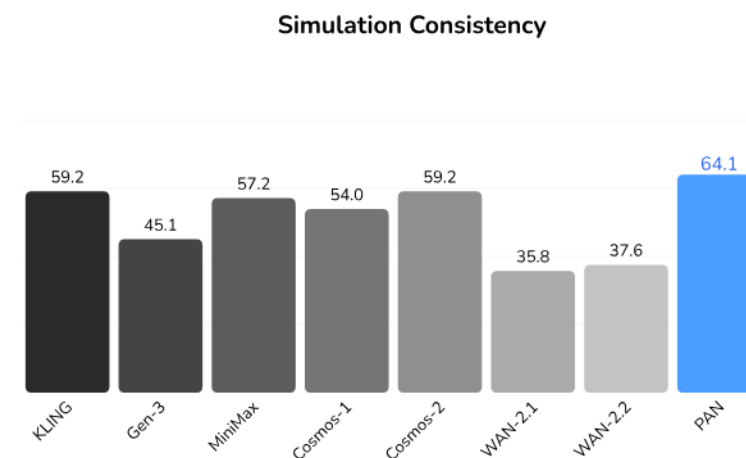
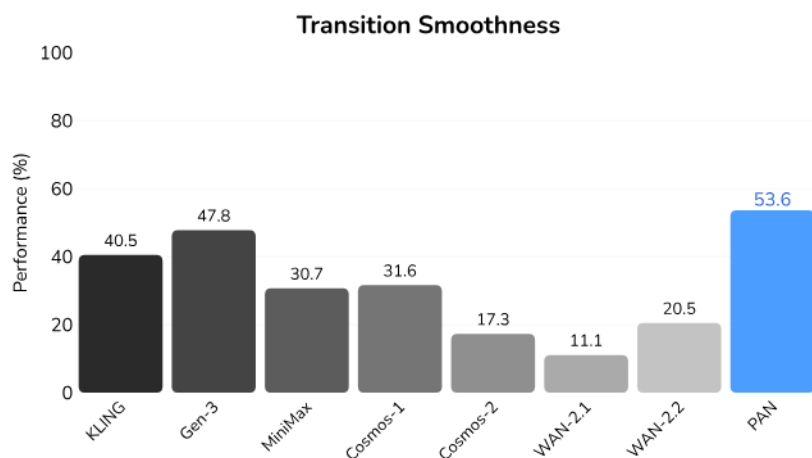
- **Action Simulation Fidelity:** how faithfully the model simulates language-specified actions and their causal consequence
- **Long-Horizon Forecast:** evaluates models' ability to maintain coherent, high-quality rollouts over extended action sequences
- **Simulative Reasoning and Planning:** evaluates whether a world model can enable an agent to reason about actions and plan toward a goal

Action Simulation Fidelity

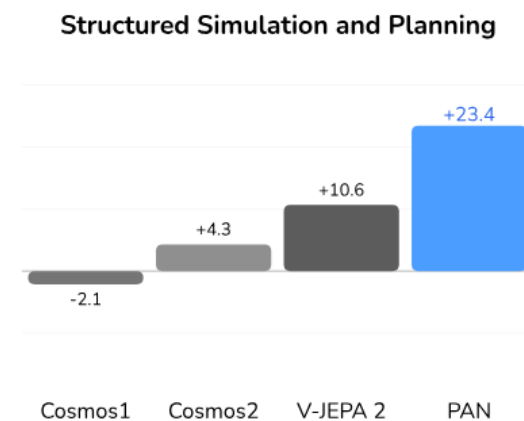
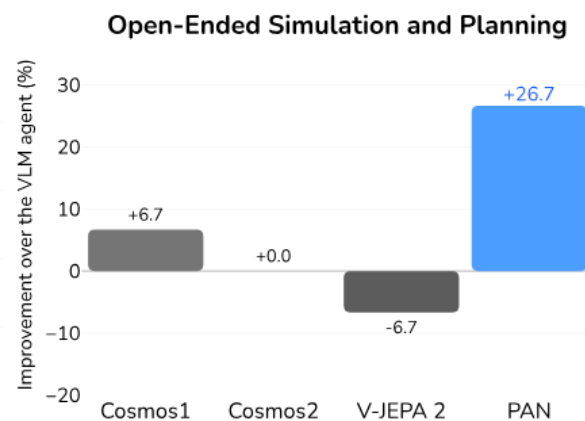
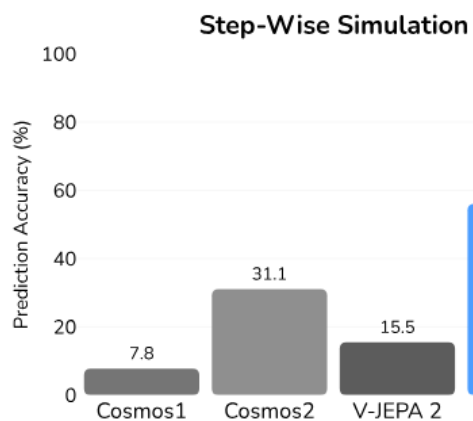


Evaluation Results

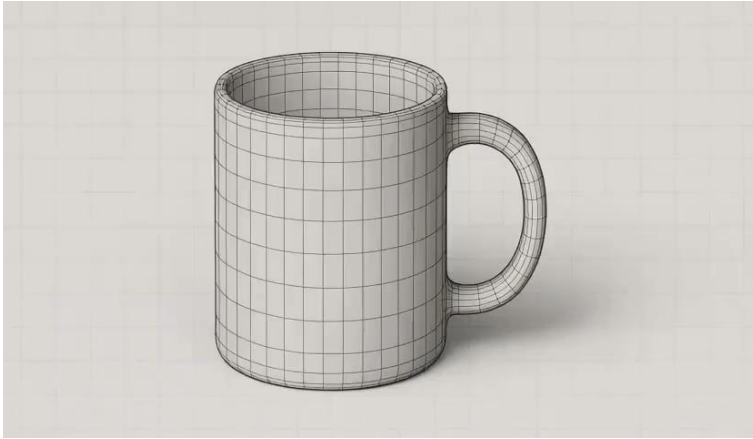
Long-horizon Forecast



Simulative Reasoning & Planning



PAN Applications: Diverse Use Cases across Domains



3D Simulation



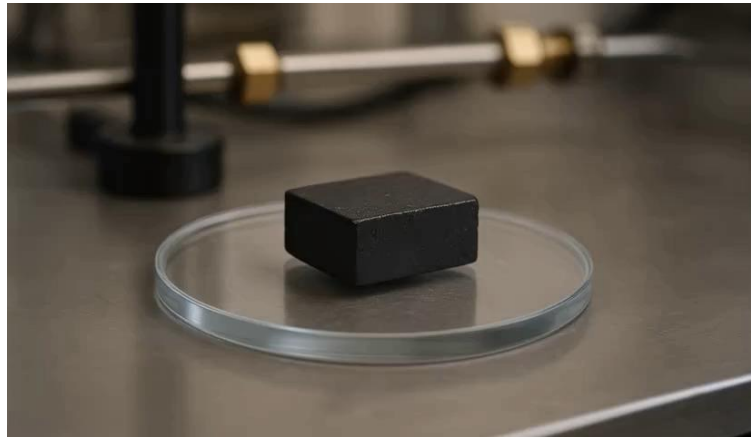
Scientific Exploration Simulation
(e.g., Mars)



Industrial Robot Simulation



Sports Strategy Simulation

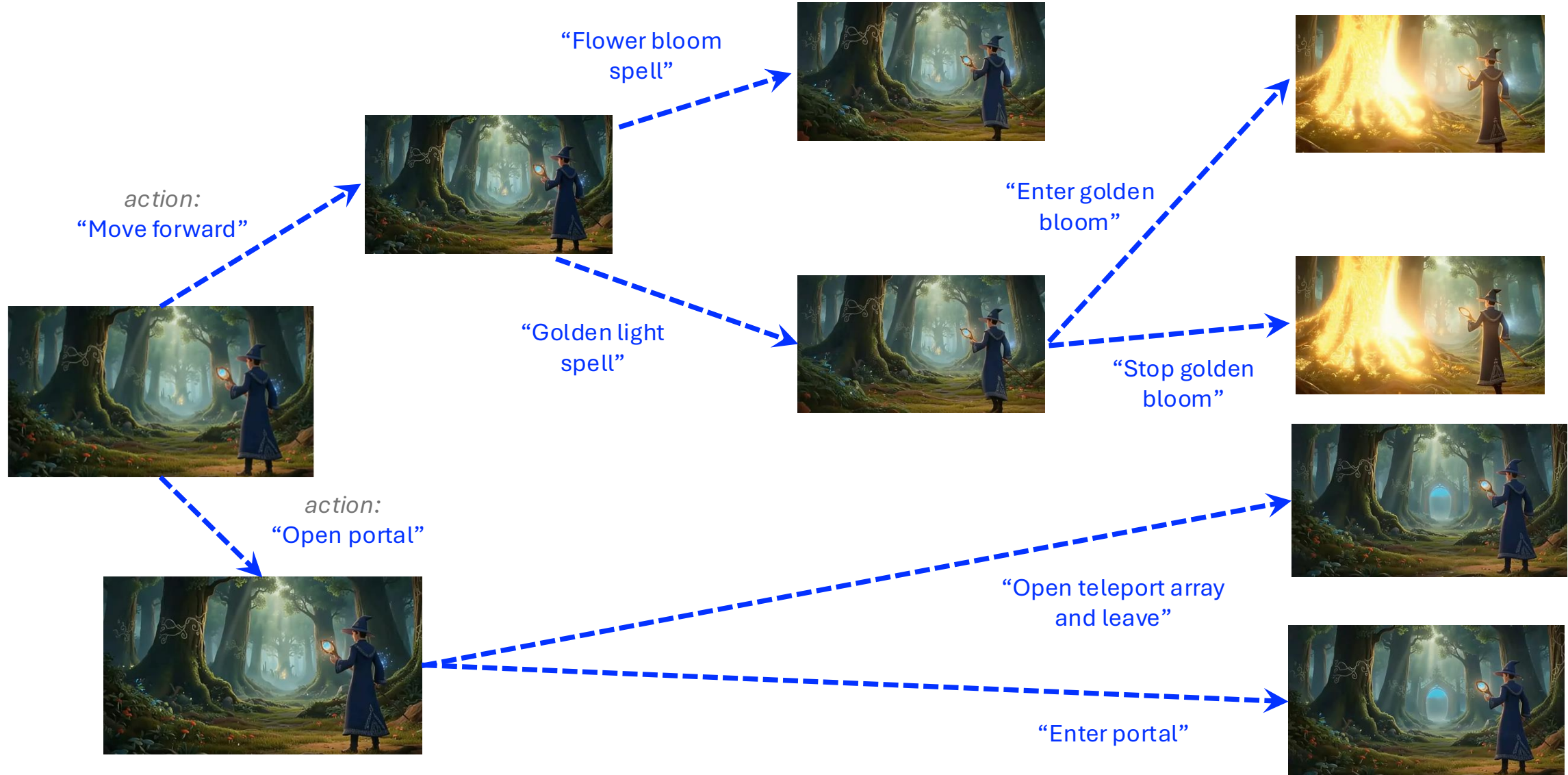


Education
(e.g., physical phenomenon)

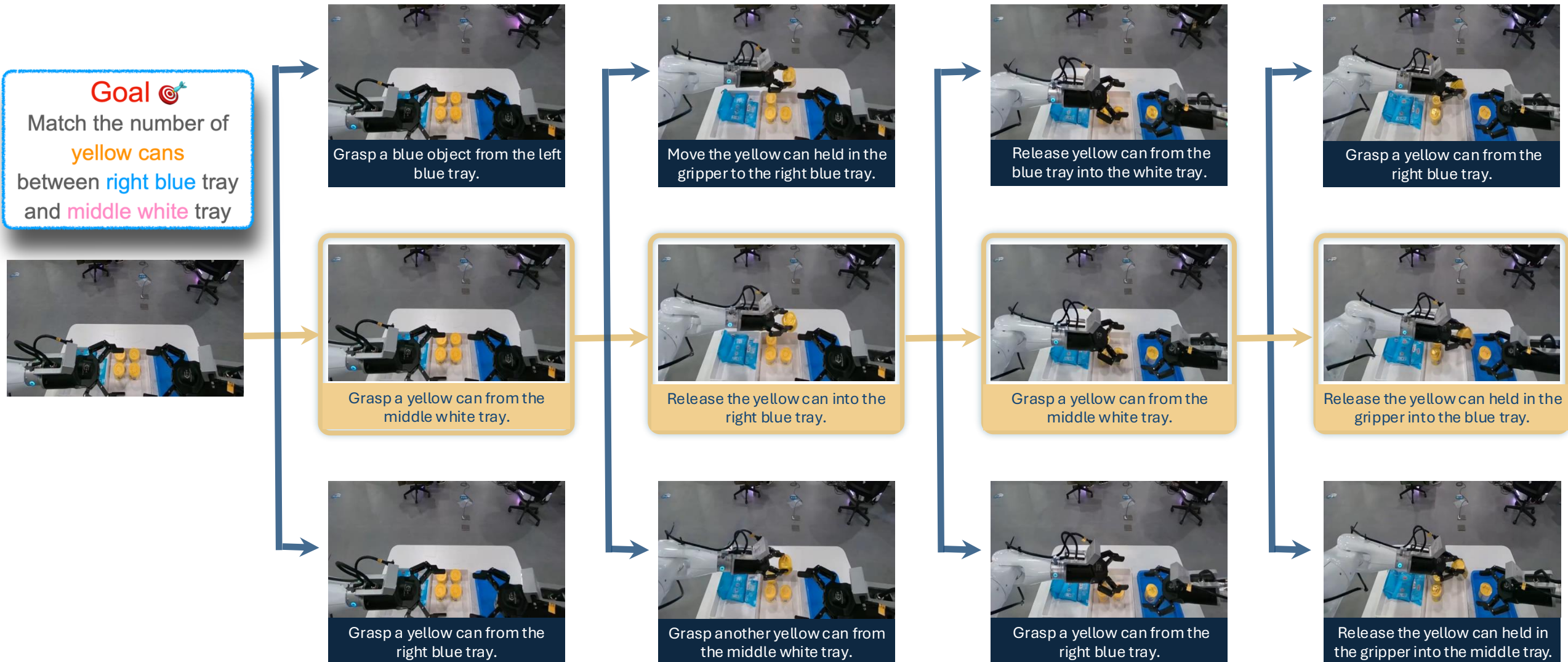


Professional Training

PAN: Interactive Long-Horizon and Branching Simulation



PAN: Reasoning with thought experiments



Roadmap for today

1. What is a world model? How should we build it?
2. An example of world model for agentic reasoning
3. A general-purpose physical world model for simulative planning
4. **An agentic LLM that self-regulates when and how to use planning (work in progress)**

This part has been redacted at this moment

Concluding Thoughts

- LLMs showed how far next-token prediction can go
- **World model:** can next-world prediction support broader forms of intelligence beyond language?
- **Answer:** a serious candidate
- **Open problems:** Multimodality, representation, architecture, generative modeling, training/inference systems, interplay with agents, etc.