CTM-AI: A BLUEPRINT FOR GENERAL AI INSPIRED BY CONSCIOUSNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite remarkable advances, today's AI systems remain narrow in scope, falling short of the flexible, adaptive, and multisensory intelligence that characterizes humans. This gap has fueled longstanding debates about whether AI might one day achieve human-like generality or even consciousness, and whether principles of consciousness can inspire new architectures for general AI. This paper presents an early blueprint for implementing a general AI system based on the Conscious Turing Machine (CTM), a formal machine model of consciousness. The CTM has an enormous number of powerful processors ranging from specialized experts (e.g., vision–language models, search engines, APIs) to unspecialized general-purpose learners poised to develop their own expertise. Crucially, for whatever problem must be dealt with, the system need not know in advance which processors hold the relevant expertise; instead, multimodal machine learning methods enable the system to select, integrate, and fuse information across processors. We extend the CTM into a practical framework, the CTM-AI, and demonstrate its utility on diverse tasks including multimodal perception, tool learning with multiple APIs, and multi-turn web agent tasks. Together, this work offers a principled and testable blueprint for general AI inspired by computational models of consciousness.

1 INTRODUCTION

In recent years, progress toward AI models capable of human-like intelligence has inspired debates regarding whether today's AI and its future counterparts can one day display human-like levels of consciousness. Flipping the debate, we present a concrete blueprint for general AI based on a formal machine model of consciousness, the Conscious Turing Machine (CTM) (Blum & Blum, 2021; 2022; Liang, 2022). The CTM is a simple and formal model of consciousness inspired by Alan Turing's model of computation (Turing, 1936) and Bernard Baars' theater model of consciousness (Baars, 1993). Critically different from other cognitive architectures and modern LLM/agentic workflows, the CTM has no central executive - no conductor, no stage director (Blum & Blum, 2023). Instead, the CTM employs a global workspace and distributed competition to integrate the power of an enormous collection of parallel independent cognitive, sensory, motor, and extended processors. When a problem needs to be solved, it becomes globally broadcast to all processors, eliciting help from those who might have the expertise, interest, and resources to tackle the problem, even though their talents and abilities might be unknown to a central executive.

Key contribution. Despite its potential, the CTM is a concept that remains abstract and theoretical. In this work, we bridge this gap by implementing the formal CTM model as a concrete system called CTM-AI which includes (1) multiple specialized processors operating in parallel, (2) a limited capacity workspace enforcing selective attention via up-tree competition, (3) a global broadcast of information via a down-tree from the workspace to all processors, and (4) the formation of links between relevant processors over time, enabling unconscious communication to integrate their knowledge into higher-order multimodal information. Through continuous interaction feedback, and learning from its external world via sensory inputs, predictions, actuators, and feedback, CTM-AI updates its individual processors, processor links, and multiprocessor integration to improve over time. The CTM-AI model addresses several key limitations of current AI paradigms.

1. **Modular and decomposable**: Existing monolithic foundation models are centrally computed and structurally fixed, which blocks the update of new skills and processors. CTM-AI is more modular, decomposable, and supports the flexible addition or removal of processors and capabilities. CTM-AI can adapt to task-specific features effortlessly without extra training.

2. **Free of a central executive**: CTM-AI does not require an orchestrator akin to modern agentic workflows, but rather uses its dynamics to automatically determine the information flow and learning over multiple processors. Therefore, compared with multi-agent workflows that have a fixed workflow defined or learned for specific tasks, CTM-AI is a more general and flexible framework suitable for different tasks.
3. **Integrated reasoning and agentic flexibility**: Today's agentic frameworks still struggle with reasoning over multiple modalities. CTM-AI can carry out multi-step multimodal reasoning across processors (integrating text, vision, tools, and more). A special case recovers the 'o1-style' single-LLM reasoning when only one processor is active, showing that CTM-AI generalizes LLM reasoning and multimodal multi-agent workflows.

Main results. To evaluate CTM-AI as a general multisensory and multi-action AI, we present quantitative results that showcase its versatility across a broad range of language modeling, multimodal perception, human behavior understanding, and agentic tool use tasks. This wide range of tasks highlights its ability to use external tools, processors, and APIs, integrate and reason over multimodal information, and solve complex multi-step problems. Based on our experiments, we find that CTM-AI can achieve comparable or state-of-the-art performance on multimodal perception tasks (MUStARD for sarcasm detection, URFunny for humor detection, NYCartoon for multimedia analysis), tool learning API-using tasks (StableToolBench), and multi-turn agentic tasks (WebArena). Moreover, our ablation study shows that mechanisms designed inside CTM-AI, including long-term memory, fusion, up-tree competition, and down-tree broadcasting, all contribute to the final improvement. Such experiments prove the value of the architecture and inference mechanism design in CTM-AI.

2 Related Work

Consciousness and AI There have been several directions in building AI systems inspired by human consciousness (Blum & Blum, 2024; Zeng et al., 2024; Zhao et al., 2023); we point the reader to Butlin et al. (2023) for a review. Prior efforts have typically emphasized high-level analogies, such as developing multimodal languages that mirror human multisensory processing (Liang, 2022; Liang et al., 2022b; Lohse et al., 2021; Murray & Wallace, 2011; Nanay, 2018), constructing world models that integrate perception, planning, and action (Nottingham et al., 2023; Singer et al., 2022; Hao et al., 2023; Liu et al., 2024; Prasad et al., 2023), or pursuing reasoning paradigms inspired by human cognition, including robustness (Sun, 1995; Zeng et al., 2023), compositionality (Gupta & Kembhavi, 2023; Wu et al., 2021; Zhou et al., 2022), causality (Halpern, 2000; Liu et al., 2023c; Zang et al., 2023), and transparency (Liang et al., 2020; Mota et al., 2021). CTM-AI differs by directly grounding these inspirations in state-of-the-art reasoning and agentic models, operationalizing them into a concrete, extensible system rather than remaining at the level of abstract analogy.

Large foundation models The recent wave of large pretrained and generative models such as large language models (Achiam et al., 2023; Brown et al., 2020; Radford et al., 2019; Touvron et al., 2023; Zhang et al., 2022), image generation models (Ramesh et al., 2021; Rombach et al., 2022; Saharia et al., 2022), and multimodal foundation models (Liu et al., 2023a;b; Li et al., 2023; Liang et al., 2022a) have shown emergent abilities across a wide range of tasks (Schaeffer et al., 2023; Wei et al., 2022). Their impressive generalization capabilities have inspired debate on whether these models possess human-level intelligence (Baum et al., 2011) and consciousness (Chalmers, 2023). Mixture-of-experts (MoE) has also become a popular design choice for scaling foundation models efficiently. CTM-AI differs fundamentally from large foundation models by moving beyond monolithic scaling with a single centrally trained model and enabling modularity, flexible reasoning, and adaptive agentic behavior. Moreover, CTM-AI differs from multimodal foundation models by designing language-based interaction instead of linear projection for multimodal fusion.

Multi-agent and tool-augmented frameworks Most multi-agent systems (Qian et al., 2023; Hong et al., 2024; Schmidgall et al., 2025) rely on multi-step prompting pipelines tailored to specific tasks such as coding or reasoning, where each LLM-based agent is assigned a fixed role (e.g., planner, coder, or reviewer). Beyond such task-specific prompting, recent work has focused on enhancing reasoning abilities in LLMs and multimodal models (Li et al., 2025; Dai et al., 2025), reasoning across multiple modalities, extending context and memory (Zhou et al., 2025), and enabling dynamic tool use (Guo et al., 2024; Qin et al., 2023; Yao et al., 2024). While these advances move toward more capable systems, they typically remain tied to fixed role assignments, rigid tool-calling pipelines, or predefined multimodal fusion strategies. In contrast, CTM-AI departs from both directions: unlike

multi-agent systems, it is not a task-specific workflow with fixed roles but a general framework where flexible processors can work together, and unlike tool-augmented LLMs, it does not rely on a single-step inference but enables adaptive and iterative inference over multiple steps.

3 CTM-AI: THE CONSCIOUS TURING MACHINE WITH MODERN AI

In this section, we present background on the Conscious Turing Machine (CTM) in §3.1 and explain how we implement this conceptual model based on modern AI technologies, creating CTM-AI. We discuss CTM-AI's core components in §3.2 and its key learning dynamics in §3.3.

3.1 BACKGROUND ON THE CONSCIOUS TURING MACHINE (CTM)

The CTM is a simple and formal model of consciousness (Blum & Blum, 2021; 2022) inspired by Alan Turing's model of computation (Turing, 1936) and Bernard Baars' theater model of consciousness (Baars, 1993). However, CTM differs from Turing machines and Baars' model. While Baars describes consciousness via the activity of actors performing on a stage directed by a stage director, the CTM has no stage director or central executive. Designing a central executive can be prohibitive since we often do not know how such an executive operates. Consider the typical example of trying to recall the name of a person you've previously met. Although we may recall their name eventually, we do not know which processors are relevant and how to combine processor outputs beforehand. Rather, a federation of processors runs simultaneously, recalling different locations, events, and memories, before deciding which outputs are salient and integrating them to form the final answer. Similarly, the CTM employs a global workspace and distributed competition that determines which information from its vast collection of "unconscious" cognitive, sensory, and motor processors gets admitted to the "conscious" arena. When a problem needs to be solved, it becomes globally broadcast to all processors, eliciting help from those who might have the expertise, interest, and resources to tackle the problem, even though their talents and abilities might be unknown to a central executive. These features set the stage for its capability to be a model for general AI (Blum & Blum, 2023).

3.2 CTM ARCHITECTURE

The formal definition of the CTM is a 7-tuple < STM, LTM, Up-Tree, Down-Tree, Links, Input, Output >. We provide a brief explanation for each of them here:

- CTM is born at time 0 and has a finite lifetime T. Time is measured in discrete clock ticks, $t = 0, 1, 2, ..., T \approx 10^{10}$.
- STM (short-term memory) is a small memory capable of holding a single chunk of information at each time t.
- LTM (long-term memory) is a collection of K powerful processors $p_1, p_2, ..., p_K$, K can be as large as $K > 10^7$.
- Up-Tree is an up-directed binary tree of height h with K leaves, one leaf in each LTM processor, and a (single) root in STM.
- Down-Tree is a simple down-directed tree of height 1 with a single root in STM and K edges directed from that root to the leaves, one leaf in each LTM processor.
- Links are the channels for transmitting information directly between processors.
- Input: $\mathbb{R}^d \rightarrow$ LTM carries information from the external (outer) world via sensors (*e.g.*, eyes, ears) to special LTM processors (*e.g.*, visual and auditory processors). $\mathbb{R}^d$ is CTM's external world where $\mathbb{R}$ represents the real numbers and d is a positive integer. It also includes a user intent like a query about the external world.
- Output: LTM $\rightarrow \mathbb{R}^d$ carries information from special processors (*e.g.*, motor processor) that can be considered as feedback to the external (outer) world.

LTM processors. An LTM processor p_i (with parameters θ_i) operates in a shared space $\mathcal{H} \cong \mathbb{R}^d$ and maintains a private memory state $M_t \in \mathcal{M}$ updated over time. At step t, it receives an observation $o_t \in \mathcal{O}$ and a user query $q_t \in \mathcal{Q}$. We view the LTM processor at time t as a function $\mathrm{LTM}_t(\cdot)$ equipped with three operations: (1) **execute** produces a chunk based on the current observations and previous memory; (2) **read** returns a view of its memory at a specified timestamp; and (3) **write** integrates one or more chunks into its memory. Formally:

$$\textbf{execute:} \quad \mathrm{LTM}_t(o_t, q_t) = \mathrm{chunk}_t \tag{1}$$

$$\textbf{read:} \quad \mathrm{LTM}_t(\cdot) = \mathrm{M}_t \tag{2}$$

$$\textbf{write:} \quad \mathrm{LTM}_t(\mathrm{chunk}_t^i) = \mathrm{M}_t \oplus \mathrm{chunk}_t^i = \mathrm{LTM}_{t+1}(\cdot) \tag{3}$$

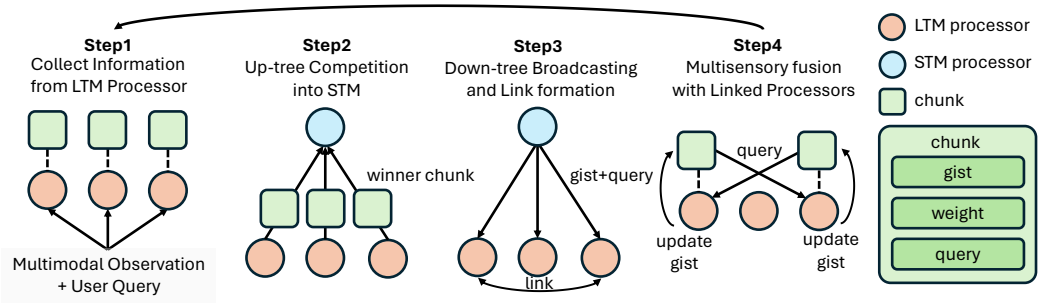

Figure 1: **Key dynamics of CTM-AI:** (1) multiple specialized LTM processors operating in parallel; (2) a limited-capacity STM workspace enforces selective attention via up-tree competition; (3) a global broadcast of information via a down-tree from the workspace to all processors; (4) the formation of links between relevant processors over time, enabling unconscious communication to integrate their knowledge into higher-order multimodal information. CTM-AI continuously interacts with the world through sensing, prediction, action, and feedback, updating its individual processors, processor links, and multiprocessor integration over time.

A chunk c_t^i produced by processor p_i at step t is formally defined as a tuple:

$$\text{chunk}_t^i \;=\; \left\langle \text{addr}(p_i),\, t,\, h_t^i,\, q_t^i,\, w_t^i \right\rangle \tag{4}$$

It stores its unique identifier $\text{addr}(p_i)$, the timestep t, a gist $h_t^i \in \mathcal{H}$ in English language that summarizes information relevant to the user's query (e.g., from audio: "laughter detected; likely humorous"), a follow-up query $q_t^i \in \mathcal{Q}$ that the processor proposes to other processors if answering it could improve the final answer (e.g., ask vision processor for facial expressions), and a weight $w_t^i \in [0, 1]$ indicating the processor's confidence/utility for how useful the gist is to answering the user's query. While all LTM processors share the above input–output interface, they differ in their specialties (*e.g.* input modalities, output tasks, internal memories). Generally, we can group them into five families of LTM processors (Card et al., 1980; Clark & Chalmers, 1998):

- **Sensory processors**, which convert raw perceptual signals such as vision, language, speech, code, music, or more into latent representations.
- **Extended or artificial processors**, which wrap external tools and APIs (e.g., calculators, web search, weather services) so that they can be accessed as internal modules.
- **Cognitive processors**, which handle reasoning and planning over the given query, supporting tasks like commonsense inference or long-horizon problem solving.
- **Motor processors**, which generate outputs by mapping internal intents to external actions, including dialogue utterances, API calls, or embodied movements.
- **Unspecialized "free" processors**, which serve as expandable slots that can acquire new observation, reasoning, or output skills over time through practice and feedback.

STM processors. Besides LTM processors, an STM processor is a stateless LLM (e.g., GPT-4o) that, given the STM at step t that wins the competition among all chunks and the current query q_t, produces a final, text-based answer y_t (i.e., the action a_t) and a quality score $\alpha_t \in [0, 1]$. Unlike LTM processors, it has no long-term memory and therefore only exposes a single **execute** operation; it performs no reads or writes to persistent state and simply grounds its output.

$$\textbf{execute:} \quad (y_t, \alpha_t) \;=\; \text{STM}_t(\text{chunk}_t^*,\, q_t), \qquad \alpha_t \in [0, 1]. \tag{5}$$

3.3 CTM Dynamics

Besides the definition of CTM based on a 7-tuple, there are the following dynamics of multisensory processing, information integration, feedback, and learning defined on top of CTM to support its functionality. The overview design principles behind building learning dynamics between multiple processors in the CTM architecture are as below:

1. Different LTM processors perform distinct functions, *e.g.*, cognitive, sensory, or motor. Some processors may be "off-the-shelf" while others' functionalities are realized over time. While individual processors may have their own internal language, communication within the CTM is in a common multimodal language we call Brainish. All processors start as independent entities.
2. Conscious communication between processors is conducted via an Up-Tree competition that decides whose chunk of information gets into STM.

3. The winning chunk (CTM's conscious content) is immediately globally broadcast to all processors via the Down-Tree, which causes the CTM to pay conscious attention to this information.
4. Links between processors form over time as one processor views another as having relevant information, enabling unconscious communication to integrate their knowledge into higher-order information (*e.g.*, learning to ride a bike requires conscious communication between sight and movement, after a while, links form, enabling unconscious communication).
5. Through continuous interaction, feedback, and learning from its external world via sensory inputs, predictions, actuators, and feedback, the CTM updates its individual processors, processor links, and multiprocessor integration to improve over time.

To fully implement such learning dynamics proposed by CTM with modern AI technologies, we split the overall learning stages of CTM into four parts and provide a more detailed description for each stage of the learning dynamics as below:

LTM processor chunk inference. At time t, all LTM processors $p_1, \ldots, p_K$ run in parallel on the observation o_t and query q_t, each using its private memory M_t^i. The collector applies each processor's $\mathbf{exec}$ to produce chunks (e.g., a VLM on an image, an ALM on audio), yielding:

$$\mathrm{CTM}_{\mathrm{collect}}(o_t, q_t) = \left\{\mathrm{LTM}_t^i(o_t, q_t)\right\}_{i=1}^K = \{\mathrm{chunk}_t^i\}_{i=1}^K \tag{6}$$

Up-tree competition into STM. After collecting all chunks from the LTM processors, only one can be stored in the STM due to its limited capacity. Therefore, an up-tree competition is performed to select the final chunk. In the original CTM design, this competition is hierarchical and local—each group of sibling chunks competes using an additive competition function to ensure the probability of winning is independent of the processor's position in the tree.

However, in our implementation, typically only a few (< 10) LTM processors are active during inference. Under this setting, restricting competition to local sibling groups is unnecessary, and the additive function becomes suboptimal. Instead, we adopt a simplified global competition, where the chunk with the highest score w_t^i (e.g., based on gist quality) is selected as the STM entry:

$$\mathrm{CTM}_{\mathrm{up}}(\{\mathrm{chunk}_t^i\}_{i=1}^K) = \mathrm{chunk}_t^{i^\star}, \quad i^\star \coloneqq \arg\max_{i \in \{1,\ldots,K\}} w_t^i. \tag{7}$$

This approach streamlines selection and works well given the small number of competing chunks in practice. The design can revert to hierarchical selection if large-scale parallelism is introduced.

Down-tree broadcast. Once the up-tree competition selects the winning chunk $\mathrm{chunk}_t^{i^\star}$, it is written into the STM as $\mathrm{STM}_t^{i^\star}$ and immediately broadcast globally to all LTM processors. This process—called down-tree broadcasting—makes the system consciously aware of this information. Operationally, each LTM processor receives the broadcast chunk and applies its own **write** function to update its private memory. This is defined as:

$$\mathrm{CTM}_{\mathrm{down}}(\mathrm{chunk}_t^{i^\star}) = \left\{\mathrm{LTM}_t^i(\mathrm{chunk}_t^{i^\star})\right\}_{i=1}^K = \left\{\mathrm{LTM}_{t+1}^i(\cdot)\right\}_{i=1}^K \tag{8}$$

After this step, the updated memory states are used in the next inference iteration. Conceptually, this mirrors the system "paying attention" to the winning information at the conscious level and committing it across all processors for continued reasoning.

Link formation between LTM processors. Beyond conscious attention, we enable unconscious communication by dynamically forming links between LTM processors. An unconscious link is created when one processor identifies another as holding complementary information useful for improving task performance. For instance, in sarcasm detection, the vision, text, and audio processors each detect different cues (e.g., sad face, angry tone, exaggerated speech), and over time, they recognize each other's utility and form links to exchange information. Concretely, after broadcasting the winning chunk $\mathrm{chunk}_t^{i^\star}$ to processor j, if p_j's response yields a high estimated relevance score w_t^j, we update the link matrix by adding a small weight increment δ: $L_{i^\star j} \leftarrow L_{i^\star j} + \delta$ and $L_{ji^\star} \leftarrow L_{ji^\star} + \delta$. This mechanism ensures efficient, dynamic linking for cooperative inference. To prevent interference or propagation of contradictory information, links are not permanent: their linking weights can decay or be reduced by δ, effectively removing weak or unhelpful connections over time.

Multimodal fusion to update LTMs. After down-tree broadcast and link formation, each processor p_i has a record $\mathcal{N}(i)$ of linked processors $\{p_j\}_{j=1}^M$ that are linked useful for further reasoning. These

links support unconscious information exchange. As part of the fusion process, each processor first generates a query q_{t+1}^i based on its current long-term memory M_{t+1}^i, including the newly broadcast chunk. Each processor then consults its neighbors in $\mathcal{N}(i)$ in parallel, posing its query q_{t+1}^i to them. The neighbors respond by running their **execute** function, and the initiating processor uses their responses to update its memory via the **write** function. This overall process is defined as:

$$\mathrm{CTM}_{\mathrm{fuse}}(o_t) = \left\{\mathrm{LTM}_{t+1}^i\left(\left\{\mathrm{LTM}_{t+1}^j(o_t, q_{t+1}^i)\right\}_{j\in\mathcal{N}(i)}\right)\right\}_{i=1}^K = \left\{\mathrm{LTM}_{t+2}^i(\cdot)\right\}_{i=1}^K \tag{9}$$

Multisensory integration enables the discovery of richer, higher-order redundant, unique, or synergistic information from linked processors (Liang et al., 2022b; 2023; Partan & Marler, 1999; 2005).

Overall: prediction, feedback, and learning. The CTM-AI system operates through an iterative cycle of prediction, feedback, and learning.

- **Prediction.** The overall prediction phase is described as $\mathrm{CTM}_{\mathrm{up}}(\mathrm{CTM}_{\mathrm{collect}}(o_t, q_t))$. For each user query q_t, the system collects chunks from LTM processors via $\mathrm{CTM}_{\mathrm{collect}}$, conducts up-tree competition via $\mathrm{CTM}_{\mathrm{up-tree}}$. This winning chunk that fits into the STM is considered the prediction provided by CTM.
- **Feedback.** The STM processor (an LLM) evaluates the prediction and assigns a quality score α_t. If $\alpha_t \geq \tau$, the prediction is accepted as output. Otherwise, negative feedback triggers learning, prompting the system to refine its internal state before reattempting inference.
- **Learning.** Learning is implemented via in-context learning with memory updates. It consists of two parts: (1) Down-tree broadcast: The winning chunk is written into each LTM processor's memory via $\mathrm{CTM}_{\mathrm{down}}$; (2) Multimodal fusion: Each processor generates a follow-up query, consults its linked neighbors $\mathcal{N}(i)$, and fuses the resulting information via $\mathrm{CTM}_{\mathrm{fuse}}$, enriching its LTM. These updates prepare the system for improved reasoning in the next iteration.

This *prediction–feedback–learning* loop generalizes multiple recent AI innovations, including multi-step reasoning, agentic workflows, multimodal interaction, and tool use. Crucially, unlike orchestrator-based systems, CTM-AI does not rely on an external controller. Instead, its intrinsic dynamics—up-tree selection, down-tree broadcasting, and processor linking—autonomously regulate information flow and drive continual learning across iterations.

4 Evaluating the Capabilities of CTM-AI

In this section, we present quantitative results that showcase CTM-AI's versatility across a broad range of tasks, including language modeling, multimodal perception, tool use, and agentic tasks. This wide range of tasks highlights its potential ability to serve as a general AI framework.

4.1 Evaluation Tasks

To assess the generality of CTM-AI, we select tasks that *activate distinct subsets of processors* within each : (i) multimodal grounding and perception (text–audio–image–video); (ii) abstract/social understanding (affect, humor, sarcasm); (iii) temporal reasoning; (iv) tool use and actuation (planning API calls and reading/writing external state); and (v) interactive, long-horizon agency (goal decomposition, feedback handling, recovery from errors). These axes require CTM-AI to *compose perception, cognitive, tool, and agentic processors* iteratively to complete end-to-end tasks.

Multimodal perception. Real data combine and conflict across modalities (e.g., words vs. tone vs. visuals). We use MultiBench (Liang et al., 2021) and HEMM (Liang et al., 2024) for broad modality coverage, plus MuStARD (Castro et al., 2019), UR-Funny (Hasan et al., 2019), and NYCartoon (Hessel et al., 2023) for socially grounded semantics (sarcasm, humor, cultural references). These tasks primarily engage audio/video/text perception processors and cognitive processors for cross-modal reasoning.

Tool learning. General systems must not only perceive but also *act*. StableToolBench (Guo et al., 2025) evaluates planning, argument construction, multi-tool composition, and error recovery. These tasks chiefly engage multiple tool processors (typed API connectors with schema/argument grounding) to accomplish one task.

Agentic tasks. Autonomy requires long-horizon control and robustness to stochastic interfaces. WebArena (Zhou et al., 2023) probes end-to-end web interaction: parsing noisy pages, tracking state,

MUStARD				
Model	**Acc↑**	**P↑**	**R↑**	**F1↑**
LMF	–	70.73	70.90	70.68
LF-DNN-v1	–	71.55	71.52	70.99
ALBEF	54.49	47.08	50.22	48.51
BLIP2	53.75	48.46	**90.13**	62.65
MMoE	70.41	60.64	89.04	71.78
BaseModel	70.42	70.44	70.90	70.26
CTM-AI	**73.88**	**73.96**	74.44	**73.77**

Table 1: **CTM-AI evaluation results on MUStARD**. CTM-AI is able to reach state-of-the-art results on sarcasm detection, beating the base model a lot.

URFunny				
Model	**Acc↑**	**P↑**	**R↑**	**F1↑**
MulT	66.65	–	–	–
FDMER	**71.87**	–	–	–
ALBEF	66.77	64.29	73.74	68.67
BLIP2	70.43	65.14	**86.60**	**74.31**
MMoE	71.88	69.18	78.16	73.29
BaseModel	60.69	60.77	60.73	60.66
CTM-AI	71.64	**71.63**	71.62	71.62

Table 2: **CTM-AI evaluation results on URFunny**. CTM-AI is able to reach comparable results with state-of-the-art models on humor detection.

and replanning. These tasks engage agentic web processors—DOM parser, screenshot/OCR, and AXTree handlers—together with cognitive processors to conduct multi-turn learning and close the perception–planning–action loop.

4.2 Baseline Settings

Backbone model. To support evaluation across multimodal perception, tool use, and agentic tasks, we adopt `gemini-2.0-flash-lite` as the base model. It natively accepts text, audio, and vision inputs and supports function calling, allowing CTM-AI to expose these capabilities as *processors* (multimodal models and tool callers) within a unified architecture.

Multimodal perception baselines. For MUSTARD and UR-FUNNY, we compare against strong multimodal baselines including MMoE (Yu et al., 2023), BLIP-2 (Li et al., 2023), and ALBEF (Li et al., 2021), which jointly process text and images and report competitive performance on cross-modal understanding.

Tool-using baselines. To assess tool-use competence on STABLETOOLBENCH, we include GPT-4o (Achiam et al., 2023) and ToolLLaMA v2 (Qin et al., 2023) with standard prompting strategies (Chain-of-Thought and DFS-style planning). These systems exhibit strong function-calling ability, composing multiple tools to complete multi-step tasks.

Agentic baselines. For web-based agentic evaluation, we use GPT-4o as a baseline agent. Both the CTM-AI-based agent and the GPT-4o agent receive identical observations (DOM tree, screenshots, and AXTree) and follow a ReAct-style loop (Yao et al., 2023), ensuring a fair comparison of planning and interaction capabilities.

4.3 Main Results

CTM-AI achieves state-of-the-art or competitive results across multimodal, tool calling, and agentic benchmarks. As shown in Table 1 and Table 4, CTM-AI attains state-of-the-art performance, improving by *3 points* on multimodal sarcasm detection and by *6+ points* on the tool-calling benchmark. On UR-FUNNY, CTM-AI delivers performance comparable to strong baselines. These settings require non-trivial coordination across processors (e.g., audio–video–text fusion for perception; planning and execution for tools), underscoring CTM-AI's ability to function as a general AI framework that composes multiple capabilities. Additionally, on a random sample of 40 ONESTOPSHOP cases from WebArena, CTM-AI surpasses LLM-only baselines (CTM-AI succeeds 8 tasks while baseline models only succeed 6) by leveraging its web-agent processors (DOM parsing, screenshot/OCR, AXTree handling) alongside planning and state tracking.

Performance gains stem from CTM-AI's mechanisms rather than base-model scaling. Our improvements arise from CTM-AI's processor orchestration—not from a stronger underlying base model. Built atop the same base model, CTM-AI introduces structured interaction mechanisms (up-tree / down-tree message passing, cross-modal fusion, and link formation) that route information among processors. In multimodal perception, this enables precise cross-modal alignment; in tool use, it captures sequential dependencies and argument grounding across multi-step calls. The same base model, when equipped with CTM-AI's interaction layer, can thus activate *different subsets of processors* to solve heterogeneous tasks without retraining.

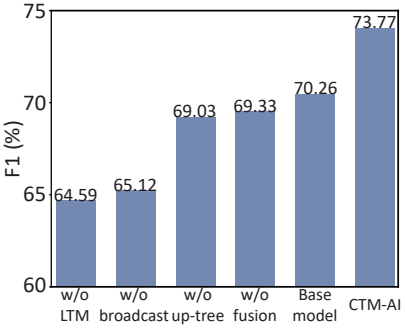

Table 3: **Ablation study on each mechanism of CTM-AI dynamics**. Full results are available in Appendix §D.

StableToolBench			
Method	**I2-Cat.**	**I2-Inst.**	**I3-Inst.**
ToolLLaMA v2 CoT	19.9±1.0	22.3±0.4	19.1±0.8
ToolLLaMA v2 DFS	22.8±1.5	19.2±1.6	18.6±1.5
GPT-4o mini CoT	24.5±1.0	22.3±2.7	20.8±1.5
GPT-4o mini DFS	25.8±1.7	25.8±2.7	20.2±0.8
GPT-4o CoT	32.5±1.7	29.6±1.6	27.9±3.5
GPT-4o DFS	32.8±1.5	28.3±1.3	23.0±1.3
Base Model+CoT	26.3±1.2	37.2±2.1	18.5±0.9
CTM-AI	**39.1±2.0**	**51.5±1.9**	**38.5±1.3**

Table 4: **CTM-AI evaluation results on StableToolBench**. We refer to the MirrorAPI-Cache. Solvable Pass Rate Score evaluated with GPT-4o. CTM-AI improves the performance a lot.

CTM-AI adapts to diverse real-world tasks with minimal adjustment. Because CTM-AI is made up of multiple processors and modular, it can be ported to *new* tasks by changing only light prompts and routing (i.e., selecting the relevant processor subset) rather than retraining. In practice, adapting from multimodal perception to tool-use or web-based agentic tasks amounts to swapping/adding processors (e.g., a search engine or calculator tool, a DOM/OCR/AXTree stack) and updating task instructions for the same processor; the base model and interaction layer remain unchanged. This plug-and-play design lets CTM-AI meet diverse task requirements with minimal overhead while preserving performance and stability.

4.4 Ablation Studies

Ablation on dynamic mechanisms of CTM-AI. Motivated by the cognitive theory behind CTM-AI, we instantiate CTM dynamics with four key mechanisms: (i) *chunk inference*, (ii) *up-tree competition*, (iii) *down-tree broadcast*, (iv) *link formation*, and (v) *multimodal fusion*. To isolate their contributions, we run comprehensive ablations that selectively disable or replace each mechanism and measure the resulting performance deltas across tasks. As shown in Figure 3, each component plays a non-trivial role and contributes to the overall reasoning ability of CTM-AI, with performance consistently degrading when any of them is removed. The existence of long-term memory is the most important part in the CTM dynamics.

Ablation on Processors of CTM-AI				
Method	**Acc↑**	**P↑**	**R↑**	**F1↑**
Base model (Only text input)	56.17	61.85	59.66	55.14
CTM-AI (Only language processor)	69.66	69.57	67.41	67.59
Base model (Only audio input)	64.40	67.98	59.77	57.11
CTM-AI (Only audio processor)	67.89	68.11	65.14	65.06
Base model(Only video modality)	61.79	60.59	60.04	60.07
CTM-AI (Only video processor)	58.43	60.37	51.82	42.35
Base model (All modalities combined)	70.42	70.44	70.90	70.26
CTM-AI	**73.88**	**73.96**	**74.44**	**73.77**

Table 5: **Ablation on single-modality inputs.** When restricted to audio-only or text-only inputs, CTM-AI still outperforms the base model by leveraging broadcasting and unconscious link formation to reason more deeply with limited information.

Ablation on single modality inputs. In Figure 3, we present the ablation results when only a single modality is provided as input, comparing the Base Model (Gemini-2.0-flash-lite) with CTM-AI. The results show that when restricted to audio-only or text-only inputs, CTM-AIconsistently outperforms the base model. We argue that this improvement is cased by the broadcasting mechanism and unconscious link formation within CTM-AI, even with limited information, processors can generate follow-up questions that the original modality-specific processor may not have considered. This allows the system to continue reasoning iteratively and explore the input from different perspectives, leading to deeper inference. However, when only the video modality is available,

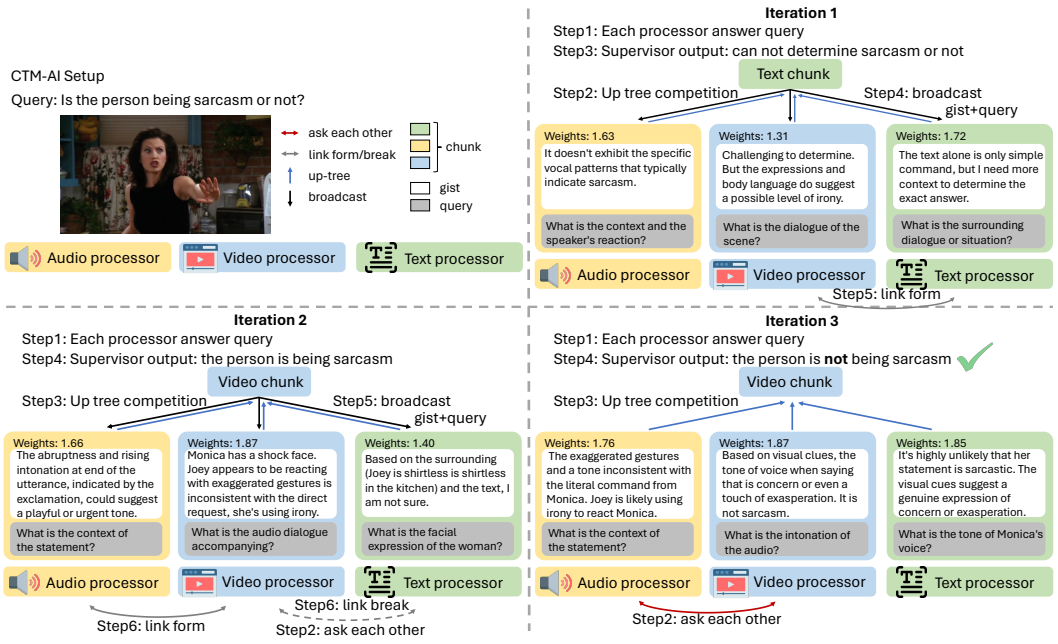

Figure 2: **Case study of CTM-AI dynamics.** We show three iterations of CTM-AI for sarcasm detection.Through multiple rounds of structured interaction, the system progressively integrates multimodal cues and convergence on the correct interpretation.

CTM-AI performs worse than the base model. We hypothesize that visual information alone can sometimes be misleading, causing the system to propagate incorrect cues through broadcasting without information from other modalities to correct them. Overall, these findings underscore the importance of multi-processor collaboration, when all modalities are jointly available, the system benefits from richer cross-modal interactions, and the performance improves significantly compared to any single-modality setting.

5 Case Study

Based on Figure 2, we analyze a multimodal perception case for identifying *sarcasm*. In the **first** iteration, all three processors are initially *uncertain* about their judgments. The text processor wins the competition and broadcasts its partial understanding of the dialogue to the other processors and explicitly asking for more surrounding context. This broadcast enables the video processor to respond with relevant visual cues, forming a link of shared information. In the **second** iteration, the video and text processor did unconscious communication with each other. The video processor integrates the contextual cues from the text and its own vision frames, and infers that the speaker is likely being sarcastic, but it still asks for the accompanying audio for a more comprehensive answer. In the **third** iteration, the video processor further queries the audio processor, receiving prosodic and tonal cues. With this enriched multimodal evidence, it refines the judgment and concludes that the speaker is not sarcastic, but instead expressing genuine concern with a shocked and somewhat exaggerated facial expression. Through repeated broadcasting and mutual asking, the processors progressively link their evidence, fuse perspectives, and converge on the correct answer.

6 Conclusion

Our work bridges the Conscious Turing Machine (CTM) theory with practical AI by implementing a system that integrates a large number of distributed processors and operates through an iterative prediction–feedback–learning loop. Experiments demonstrate that CTM-AIachieves strong and versatile performance across multimodal perception, tool use, and agentic tasks. Moreover, the architecture adapts to new tasks with minimal adjustment and without retraining. We present CTM-AIas a prototype that connects consciousness theory with general AI, offering a promising foundation for future development.

Reproducibility Statement

The datasets used in our experiments, MUStARD, URFunny, NYCartoon, StableToolBench and WebArena, are publicly available. Details of the test dataset are provided in Section 4 and the implementation of CTM-AI are provided in Appendix F.

Ethics Statement

This work builds upon publicly available datasets, no private or sensitive user data were collected or used in this research, and all experiments were conducted in controlled research settings.

Our research provides a concrete implementation that bridges the theoretical framework of CTM with practical AI technologies. The goal is to enhance LLMs' capabilities in affective learning, decision-making, multi-step reasoning, and tool use, thereby contributing to the development of more reliable and trustworthy general AI systems. Importantly, our intention is not to replicate human identity or create systems indistinguishable from humans, thereby avoiding potential ethical risks associated with anthropomorphization (Deshpande et al., 2023).

We also recognize the inherent risks of applying large language models and AI agents. These risks include biases that may arise from cultural or social factors. We are committed to ongoing analysis aimed at detecting, understanding, and mitigating such biases. Addressing these challenges remains central to our ethical research framework.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Bernard J Baars. *A cognitive theory of consciousness*. Cambridge University Press, 1993.

Seth D Baum, Ben Goertzel, and Ted G Goertzel. How long until human-level ai? results from an expert assessment. *Technological Forecasting and Social Change*, 78(1):185–195, 2011.

Lenore Blum and Manuel Blum. A theory of consciousness from a theoretical computer science perspective: Insights from the conscious turing machine. *Proceedings of the National Academy of Sciences*, 119(21):e2115934119, 2022.

Lenore Blum and Manuel Blum. A theoretical computer science perspective on consciousness and artificial general intelligence. *Engineering*, 2023.

Lenore Blum and Manuel Blum. Ai consciousness is inevitable: a theoretical computer science perspective. *arXiv preprint arXiv:2403.17101*, 2024.

Manuel Blum and Lenore Blum. A theoretical computer science perspective on consciousness. *Journal of Artificial Intelligence and Consciousness*, 8(01):1–42, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Patrick Butlin, Robert Long, Eric Elmoznino, Yoshua Bengio, Jonathan Birch, Axel Constant, George Deane, Stephen M Fleming, Chris Frith, Xu Ji, et al. Consciousness in artificial intelligence: insights from the science of consciousness. *arXiv preprint arXiv:2308.08708*, 2023.

Stuart K Card, Thomas P Moran, and Allen Newell. The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410, 1980.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. Towards multimodal sarcasm detection (an _obviously_ perfect paper). In *ACL*, pp. 4619–4629, 2019.

David J. Chalmers. Could a large language model be conscious?, 2023.

Andy Clark and David Chalmers. The extended mind. *analysis*, 58(1):7–19, 1998.

Wei Dai, Peilin Chen, Chanakya Ekbote, and Paul Pu Liang. Qoq-med: Building multimodal clinical foundation models with domain-aware grpo training. *arXiv preprint arXiv:2506.00711*, 2025.

Ameet Deshpande, Tanmay Rajpurohit, Karthik Narasimhan, and Ashwin Kalyan. Anthropomorphization of ai: opportunities and risks. *arXiv preprint arXiv:2305.14784*, 2023.

Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. Stabletoolbench: Towards stable large-scale benchmarking on tool learning of large language models. *arXiv preprint arXiv:2403.07714*, 2024.

Zhicheng Guo, Sijie Cheng, Hao Wang, Shihao Liang, Yujia Qin, Peng Li, Zhiyuan Liu, Maosong Sun, and Yang Liu. Stabletoolbench: Towards stable large-scale benchmarking on tool learning of large language models, 2025. URL https://arxiv.org/abs/2403.07714.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14953–14962, 2023.

Joseph Y Halpern. Axiomatizing causal reasoning. *Journal of Artificial Intelligence Research*, 12: 317–337, 2000.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

Md Kamrul Hasan, Wasifur Rahman, AmirAli Bagher Zadeh, Jianyuan Zhong, Md Iftekhar Tanveer, Louis-Philippe Morency, and Mohammed Ehsan Hoque. Ur-funny: A multimodal language dataset for understanding humor. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2046–2056, 2019.

Jack Hessel, Ana Marasović, Jena D. Hwang, Lillian Lee, Jeff Da, Rowan Zellers, Robert Mankoff, and Yejin Choi. Do androids laugh at electric sheep? humor "understanding" benchmarks from the new yorker caption contest, 2023. URL https://arxiv.org/abs/2209.06293.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. International Conference on Learning Representations, ICLR, 2024.

Hengzhi Li, Brendon Jiang, Alexander Naehu, Regan Song, Justin Zhang, Megan Tjandrasuwita, Chanakya Ekbote, Steven-Shine Chen, Adithya Balachandran, Wei Dai, et al. Puzzleworld: A benchmark for multimodal, open-ended reasoning in puzzlehunts. *arXiv preprint arXiv:2506.06211*, 2025.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Paul Pu Liang. Brainish: Formalizing a multimodal language for intelligence and consciousness. *Annual Meeting of the Association for the Scientific Study of Consciousness*, 2022.

Paul Pu Liang, Yiwei Lyu, Xiang Fan, Zetian Wu, Yun Cheng, Jason Wu, Leslie Yufan Chen, Peter Wu, Michelle A Lee, Yuke Zhu, et al. Multibench: Multiscale benchmarks for multimodal representation learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Paul Pu Liang, Yiwei Lyu, Xiang Fan, Jeffrey Tsaw, Yudong Liu, Shentong Mo, Dani Yogatama, Louis-Philippe Morency, and Russ Salakhutdinov. High-modality multimodal transformer: Quantifying modality & interaction heterogeneity for high-modality representation learning. *Transactions on Machine Learning Research*, 2022a.

Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Foundations and recent trends in multimodal machine learning: Principles, challenges, and open questions. *arXiv preprint arXiv:2209.03430*, 2022b.

Paul Pu Liang, Yun Cheng, Xiang Fan, Chun Kai Ling, Suzanne Nie, Richard J Chen, Zihao Deng, Nicholas Allen, Randy Auerbach, Faisal Mahmood, et al. Quantifying & modeling multimodal interactions: An information decomposition framework. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Paul Pu Liang, Akshay Goindani, Talha Chafekar, Leena Mathur, Haofei Yu, Ruslan Salakhutdinov, and Louis-Philippe Morency. Hemm: Holistic evaluation of multimodal foundation models. *Advances in Neural Information Processing Systems*, 37:42899–42940, 2024.

Weixin Liang, Feiyang Niu, Aishwarya Reganti, Govind Thattai, and Gokhan Tur. Lrta: A transparent neural-symbolic reasoning framework with modular supervision for visual question answering. *arXiv preprint arXiv:2011.10731*, 2020.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023b.

Yang Liu, Guanbin Li, and Liang Lin. Cross-modal causal relational reasoning for event-level visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023c.

Yuchen Liu, Luigi Palmieri, Sebastian Koch, Ilche Georgievski, and Marco Aiello. Delta: Decomposed efficient long-term robot task planning using large language models. *arXiv preprint arXiv:2404.03275*, 2024.

Michael Lohse, Johannes C Dahmen, Victoria M Bajo, and Andrew J King. Subcortical circuits mediate communication between primary sensory cortical areas in mice. *Nature Communications*, 12(1):1–14, 2021.

Tiago Mota, Mohan Sridharan, and Aleš Leonardis. Integrated commonsense reasoning and deep learning for transparent decision making in robotics. *SN Computer Science*, 2(4):242, 2021.

Micah M Murray and Mark T Wallace. *The neural bases of multisensory processes*. CRC Press, 2011.

Bence Nanay. Multimodal mental imagery. *Cortex*, 105:125–134, 2018.

Kolby Nottingham, Prithviraj Ammanabrolu, Alane Suhr, Yejin Choi, Hannaneh Hajishirzi, Sameer Singh, and Roy Fox. Do embodied agents dream of pixelated sheep: Embodied decision making using language guided world modelling. In *International Conference on Machine Learning*, pp. 26311–26325. PMLR, 2023.

Sarah Partan and Peter Marler. Communication goes multimodal. *Science*, 283(5406):1272–1273, 1999.

Sarah R Partan and Peter Marler. Issues in the classification of multimodal communication signals. *The American Naturalist*, 166(2):231–245, 2005.

Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. Adapt: As-needed decomposition and planning with language models. *arXiv preprint arXiv:2311.05772*, 2023.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 55565–55581. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/adc98a266f45005c403b8311ca7e8bd7-Paper-Conference.pdf.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*, 2025.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Ron Sun. Robust reasoning: integrating rule-based and similarity-based reasoning. *Artificial Intelligence*, 75(2):241–295, 1995.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Alan Turing. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58(345-363):5, 1936.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022.

Zhengxuan Wu, Elisa Kreiss, Desmond C Ong, and Christopher Potts. Reascan: Compositional reasoning in language grounding. *arXiv preprint arXiv:2109.08994*, 2021.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. tau-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.

Haofei Yu, Zhengyang Qi, Lawrence Jang, Ruslan Salakhutdinov, Louis-Philippe Morency, and Paul Pu Liang. Mmoe: Enhancing multimodal models with mixtures of multimodal interaction experts. *arXiv preprint arXiv:2311.09580*, 2023.

Chuanqi Zang, Hanqing Wang, Mingtao Pei, and Wei Liang. Discovering the real association: Multimodal causal reasoning in video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19027–19036, 2023.

Gangyan Zeng, Yuan Zhang, Yu Zhou, Xiaomeng Yang, Ning Jiang, Guoqing Zhao, Weiping Wang, and Xu-Cheng Yin. Beyond ocr+ vqa: Towards end-to-end reading and reasoning for robust and accurate textvqa. *Pattern Recognition*, 138:109337, 2023.

Yi Zeng, Feifei Zhao, Yuxuan Zhao, Dongcheng Zhao, Enmeng Lu, Qian Zhang, Yuwei Wang, Hui Feng, Zhuoya Zhao, Jihang Wang, et al. Brain-inspired and self-based artificial intelligence. *arXiv preprint arXiv:2402.18784*, 2024.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Lin Zhao, Lu Zhang, Zihao Wu, Yuzhong Chen, Haixing Dai, Xiaowei Yu, Zhengliang Liu, Tuo Zhang, Xintao Hu, Xi Jiang, et al. When brain-inspired ai meets agi. *Meta-Radiology*, pp. 100005, 2023.

Honglu Zhou, Asim Kadav, Aviv Shamsian, Shijie Geng, Farley Lai, Long Zhao, Ting Liu, Mubbasir Kapadia, and Hans Peter Graf. Composer: Compositional reasoning of group activity in videos with keypoint-only modality. In *European Conference on Computer Vision*, pp. 249–266. Springer, 2022.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*, 2025.

A THE USE OF LARGE LANGUAGE MODELS (LLMS)

We used ChatGPT as a writing assistant to help us write part of the paper. Additionally, we utilize the power of CodePilot to help us code faster. However, all the AI-generated writing and coding components are manually checked and modified. There is no full AI-generated content in the paper.

B ARTIFACT DETAILS

B.1 MODEL LICENSE

GPT-4o License: Proprietary (OpenAI)
gemini-2.0-flash-lite License: Apache 2.0

B.2 SOFTWARE VERSIONS

For web-agent evaluation, we adopt BrowserGym v0.14.2 `https://github.com/ServiceNow/BrowserGym`. To access large language models, we employ LiteLLM 1.74.3 `https://litellm.ai` as the serving interface.

C EXPERIMENTAL DETAILS

When querying the Gemini API, we adopt a deterministic decoding configuration with temperature fixed at 0.0, top_n set to 1, and a maximum token limit of 4096.

D FULL RESULTS OF ABLATION STUDY

Ablation on Components of CTM-AI				
Method	**Acc↑**	**P↑**	**R↑**	**F1↑**
Base model (Gemini-2.0-flash-lite)	70.42	70.44	70.90	70.26
CTM-AI w/o up-tree competition	69.94	69.25	68.91	69.03
CTM-AI w/o broadcast	66.01	65.20	65.06	65.12
CTM-AI w/o fusion	69.38	69.91	70.27	69.33
CTM-AI w/o LTM	65.73	64.85	64.48	64.59
CTM-AI	**73.88**	**73.96**	**74.44**	**73.77**

Table 6: Ablation on MUStARD on each components of CTM-AI. The results show that all the up-tree competition, broadcast, fusion and LTM part paly an role in more accurate reasoning. Full results of Figure 3

E ANALYSIS OF FAILED CASE

We present a failure case of CTM-AI, where its performance did not surpass the Base Model. We hypothesize that this is partly due to the query being in a multiple-choice format, which undermined CTM-AI's relevance estimation and confidence calibration. Moreover, as shown by existing baselines, the From Descriptions (FD) setting performs exceptionally well, suggesting that in this case the image modality may have introduced misleading signals rather than helpful cues.

New Yorker Caption Contest		
Model	**Matching**	**Ranking**
From Pixels (FP)		
CLIP	<u>62.3</u>	<u>61.5</u>
From Descriptions (FD)		
GPT-3.5 (5-shot)	63.8	55.2
GPT-4 CoT	**81.9**	64.3
Base Model	59.7	65.3
Base Model+CoT	57.3	62.2
CTM-AI	54.7	56.8

Table 7: **CTM-AI evaluation results on NYCartoon**.

F IMPLEMENTATION OF CTM-AI

We present the pseudocode of CTM-AI in Algorithm 1.

Algorithm 1 Iterative inference of CTM-AI.

Require: processors $\mathcal{P} = \{p_i\}_{i=1}^{K}$, each with long-term memory $\{m_{p_i}\}_{i=1}^{K}$,
processor-link matrix $L \in \{0,1\}^{K\times K}$,
multisensory observation stream $(x_t)_{t\geq 1}$, user query Q, horizon H
Ensure: conscious action a

$t \leftarrow 0$
while TRUE **do**
  $t \leftarrow t + H + 2$; read current observation x_t
  parallel for $i \leftarrow 1$ **to** K
    $\mathcal{N}_i \leftarrow \{j \mid L_{i,j} = 1\}$ ▷ find linked processors
    $m_{p_i} \leftarrow m_{p_i} \cup \textsc{Fuse}(p_i, \{p_j\}_{j\in\mathcal{N}_i}, x_t, Q)$
    $(c_{i,t+1},\, q_{i,t+1}) \leftarrow p_i(x_t, Q)$ ▷ ask processors
  end parallel for
  $(c_{k,t+H+1},\, q_{k,t+H+1}) \leftarrow \textsc{UpTree}(\{c_{i,t+1}\}_{i=1}^{K})$
  if $\textsc{IsReady}(c_{k,t+H+1})$ **then**
    $a_{t+H+1} \leftarrow \textsc{Act}(c_{k,t+H+1})$
    return a_{t+H+1}
  end if
  parallel for $j \leftarrow 1$ **to** K
    $m_{p_j} \leftarrow m_{p_j} \cup c_{k,t+H+1}$ ▷ broadcast to all memories
    $(c_{j,t+H+2}, q_{j,t+H+2}) \leftarrow p_j(x_t, q_{k,t+H+1})$ ▷ answer winner's question
    if $\textsc{IsHelpful}(c_{j,t+H+2})$ **then**
      $L_{k,j} \leftarrow 1$ ▷ form link to helpful processor
    end if
  end parallel for
end while