A Method for Creating a Artificial General Intelligence with Self-Awareness

Authors: Yongcongchen, Ting Zeng, Jun Zhang, Yongcong Chen, Ting Zeng

Date: 2023-12-18T00:00:00+00:00

Abstract

In this paper, we propose a new approach to building a artificial general intelligence with self awareness, which includes: (1) a new method to implement attention mechanisms; (2) a way to give machines self-demands; (3) how to form a value evaluation system compatible with the network; (4) a way to create the world models; (5) how to realize a top-down, hierarchical thinking decision-making chain; (6) a way to achieve general decision-making and response capabilities; (7) a way for a machine to directly obtain human experience through language. In the paper, we first analyze some of the shortcomings of current LLMs (Large Language Model) and propose ideas for improvement. Then we analyze why our scheme can solve the above problems and provide detailed steps for implementing our scheme. In chapter 6, we analyze the advantages and disadvantages of our scheme and propose further research directions. Finally, we propose our thoughts on the next step of AI development.

Full Text

Preamble

A Method for Creating Artificial General Intelligence with Self-Awareness

Yongcong Chen
New Millennium Future Technology Co., Ltd., Beijing, China
yongcongchen@sina.com

Ting Zeng
Library, Tsinghua University, Beijing, China
ceng-t@mail.tsinghua.edu.cn

Jun Zhang
Department of Computer Science and Technology, Shenyang Normal University,

Shenyang, China

November 23, 2023

Abstract

In this paper, we propose a novel approach to building artificial general intelligence with self-awareness, comprising seven key components: (1) a new method for implementing attention mechanisms; (2) a framework for endowing machines with self-demands; (3) a mechanism for forming a value evaluation system compatible with the network; (4) a methodology for creating world models; (5) a technique for realizing top-down, hierarchical thinking and decision-making chains; (6) a pathway to achieving general decision-making and response capabilities; and (7) a method enabling machines to directly acquire human experience through language. We begin by analyzing the shortcomings of current large language models (LLMs) and propose corresponding improvements. We then explain why our scheme can address these limitations and provide detailed implementation steps. In Section 6, we analyze the advantages and disadvantages of our approach and suggest directions for further research. Finally, we offer our perspectives on the next stage of AI development.

Keywords: Artificial General Intelligence · AGI · Reinforcement Learning · Large Language Model · Memory and Forgetting Mechanisms · Chain Association Activation · Attention Mechanism · Machine Awareness

Introduction

Current mainstream artificial intelligence models have ignited the spark of artificial general intelligence [?][?][?]. However, whether LLMs can achieve truly "Artificial General Intelligence" remains a subject of debate [?][?]. We contend that existing LLMs suffer from three fundamental flaws that prevent them from achieving genuine AGI.

First, they cannot solve problems independently. For instance, an AI system does not proactively offer help when it sees its owner fall [?]. This stems from the machine's lack of intrinsic needs, which makes it impossible to generate its own goals or create temporary tasks. Consequently, machine decision-making relies primarily on predefined rules—either from programmatic presets or statistical patterns derived from data—rather than on self-generated objectives [?][?][?][?][?]. This limitation prevents the formation of long chains of logical reasoning, making it difficult for machines to handle the endless unexpected situations encountered in real life.

Second, knowledge cannot be updated in real time. Current AI systems require massive amounts of data and are trained as pretrained models in a single batch process. This approach prevents real-time knowledge updates, yet such updates are crucial for machines that interact with their environment. The interaction between machine and environment is precisely the process through which new

knowledge is acquired. If knowledge acquired by the machine cannot be updated in real time, the machine will repeatedly make the same mistakes when facing identical input information [**?**].

Third, it is difficult to apply these models to tasks requiring interaction with the real environment. In domains such as autonomous driving, household chores, and patient care, machines must build knowledge about interactive decision-making between their actions and the external environment. However, these fields do not permit extensive trial and error, making it impossible to use reinforcement learning (RL) to build decision-making knowledge through real-world interaction [**?**][**?**]. In this paper, we propose a new approach to establishing an artificial intelligence machine with autonomy, a thinking chain, the ability to perform general tasks, and real-time knowledge updating capabilities.

2 The Basis of Our Solution

A prevailing viewpoint suggests that different branches of AI technology are essentially pursuing the same goal: the sparsity of the coefficient matrix of information [**?**][**?**][**?**][**?**]. We believe humans also leverage this "sparsity" principle. Through memory and forgetting mechanisms, as well as associative abilities, humans summarize common token combinations and represent them with symbols—language. Humans use linguistic symbols to describe any information, whether in conversation or writing. This process resembles using a suitable set of coordinate basis clusters to facilitate the description of any information (vector) in an information (vector) matrix. We can consider linguistic symbols themselves as a set of coordinate basis clusters, similar to the preferred basis clusters that emerge through gradual survival of the fittest among human groups.

Due to limited information processing capabilities, humans tend to use symbols to represent common token combinations through memory and forgetting mechanisms. "Common" means these token combinations can be repeated frequently. Consequently, repeatable token combinations gradually evolve into "concepts." When using this set of "concepts" to describe common information, the resulting sparse matrix is indeed sparse. Therefore, humans have long pursued the principle of "coefficient matrix sparsity." While such coordinate basis clusters may not efficiently express all information, they can efficiently express common information.

The high-dimensional features created by deep learning constitute a set of coordinate basis clusters in their information matrix. However, deep learning lacks the constraint of "using common token combinations first." Consequently, it achieves overall information sparsity, whereas humans achieve sparsity for common information. Overall sparsity is more efficient when all information needs to be expressed equally, but it differs from human "concepts." This discrepancy makes communication between deep learning systems and humans difficult because they use different conceptual frameworks.

In an LLM, the attention mechanism essentially predicts the probability of spe-

cific token combinations based on statistical knowledge obtained from pretraining [?][?][?][?], then assigns higher weights to common token combinations, making these combinations implicit coordinate bases. Thus, the basis clusters created by the attention mechanism align more closely with human habits. While overall expression efficiency may not be maximized, it is more efficient for expressing common information, similar to human language. We believe the core capability of LLMs stems from the attention mechanism. We have taken an alternative path to achieve the same goal through different means.

We identify two fundamental problems with current LLMs. First, while the attention mechanism itself is correct, deep learning has drawbacks because it alters the original temporal/spatial organization of tokens. The knowledge generated by large models becomes difficult for humans to understand, preventing us from imitating their organizational form and implanting innate "self-needs" (innate knowledge) into machines. Without "self-demand," machines cannot have "their own ideas" and cannot make "autonomous decisions." Consequently, machines can only passively "make decisions" according to predetermined processes (either presets or statistical patterns) without flexible adjustment—currently a major problem in AI.

Second, the approach of "task-oriented reinforcement learning" is fundamentally flawed. Humans are "universal" because we make decisions based on "seeking benefits and avoiding harm" when facing all tasks. Machines should operate similarly. Tasks are countless, and task-oriented reinforcement learning will never be completed. Moreover, many tasks have high trial-and-error costs. For example, few people are willing to hand over their children to machines for experimental purposes.

Our solutions are as follows: (1) We adopt an alternative approach that achieves the attention mechanism without disrupting the original temporal/spatial organization of tokens. (2) Consequently, the knowledge created by machines can be understood and imitated by humans. (3) We can use special tokens to represent machine needs, then imitate the organizational form of knowledge to pre-install machines with "innate needs" (i.e., innate knowledge and simplest means of communication). (4) As a special type of token, "innate needs" form common combinations with other tokens through attention mechanisms, becoming part of common sense. (5) In common sense, the combination of external tokens constitutes the "world model" (objective common sense), while the combination of external tokens and the machine's own needs forms the "value system" (subjective common sense). This solves the problem of creating common sense. (6) Machines only learn one thing: "how to meet one's own needs" and only handle one thing: "how to meet one's own needs." This constitutes universal decision-making ability. (7) Because machines have their own needs and a value system integrated with knowledge, they may create tasks for themselves and execute them even without human instructions. This solves the problem of initiative, enabling machines to reason based on value and achieve top-down, autonomous, layer-by-layer decomposition and gradual implementation of any

human-assigned task. Thus, machines can complete universal tasks while pursuing their own needs, similar to human behavior.

3 How to Create a Knowledge Network

Here are the ten steps to implement the knowledge network in our solution.

3.1 Step 1: Obtain Tokens from Information

This step is similar to other AI technologies, so we can leverage existing results such as LLM tokens, the first few layers of neural network image processing, speech processing syllables, etc. However, our scenario has special requirements: (1) We only extract the lowest-level features as tokens. According to the holistic first principle, the machine extracts underlying features such as words, overall outlines, textures, topologies, lines, images, horns, ridges, vertices, voice time/frequency domain tones, timbres, etc. (2) More complex tokens are formed through the attention mechanism plus memory and forgetting mechanisms via learning. (3) We use the memory and forgetting mechanism to prioritize and eliminate tokens. If a token has a high repetition rate during pretraining, it may be retained, while tokens with low repetition rates are forgotten and no longer extracted. This applies the "common tokens combination first principle," similar to humans—a gift from evolution, as extracting common token combinations first maximizes energy utility due to their high probability of occurrence. This principle forms the basis of our approach and the source of the attention mechanism's effectiveness. (4) There is no need to identify tokens or combinations; simply storing them suffices. Even with imperfect algorithms, through the attention mechanism plus memory and forgetting mechanisms, a machine will accumulate common token combinations, forming its "world models." Obviously, every machine's world models differ, related to its upbringing, similar to humans. (5) In our scenario, the machine-created model differs from LLMs. We retain the original temporal and spatial organization of tokens in the final pattern, making this world model understandable and imitable by humans. LLMs do not preserve the original temporal/spatial organization, making their final patterns difficult for humans to understand. (6) Temporal relationships are realized by preserving storage time interval information, while spatial relationships are realized through a whole-to-local approach. The purpose of the whole is to establish local spatial relations, with the whole and local connected through temporal proximity relationships. (7) We can use special tokens to represent machine needs (including survival needs and value needs) and emotions (a general communication tool bestowed upon machines by humans). Through the attention mechanism plus memory and forgetting mechanisms, connections between special tokens and ordinary tokens are established, forming the value system integrated with knowledge.

shows the composition of each token data record, which includes four fields: Record Time, Token, Memory Value, and Activation Value.

3.2 Step 2: Save the Tokens

Each token is appended with three additional fields to form a record stored in the memory database, as shown in Table 1. The Memory Value (Field 3) indicates the memory strength of the token. The Activation Value (Field 4) indicates the strength of activation by input tokens. The Record Time (Field 1) indicates when a storage event occurred, with intervals representing the temporal relationship between tokens. Notably, our method emphasizes the time interval relationship between tokens. We believe all relationships between things can essentially be simplified as temporal interval relationships between tokens. This storage method, which preserves token time intervals, is called the "simultaneous storage method."

3.3 Step 3: Perform Similarity Activation

Each input token receives a uniform initial activation value of A_0, where $A_0 = f(V)$. Here, A_0 is the initial activation value assigned to input tokens, and V is the statistical result of activation values from reward and punishment symbols during the previous activation process. A simple statistical method sums the activation values of reward symbols minus those of punishment symbols. The initial activation value A_0 affects the range of the chain association activation process. When A_0 is high, the spread of chain association activation is larger because our scheme's activation value propagation coefficient is less than 1, and chain propagation ends when a token's activation value falls below a preset threshold. Thus, when A_0 is high, the machine activates more tokens in memory to gather more information for decision-making, similar to human behavior when previous inputs suggest high potential rewards or penalties. For example, words from a boss trigger more associative information before decision-making.

Each input token propagates its activation value to similar tokens in the memory database. The propagation coefficient is positively correlated with similarity and with the memory value of the propagated token. Through similarity activation, the machine establishes connections between information (token combinations) via token similarity. By positively correlating the activation value propagation coefficient with memory value, we achieve the principle that common token combinations receive higher weights.

3.4 Step 4: Perform Proximity Activation

We believe temporal relationships between tokens represent implicit correlations, where closer occurrence times indicate stronger potential relationships. If a temporal combination repeats, it becomes a common combination deserving higher weight. Each activated token propagates activation value to adjacent tokens; the closer the position in the memory database (representing smaller time intervals), the greater the transmission coefficient. Higher memory values also yield greater transmission coefficients. If tokens in the memory database have close relationships, they once formed a combination pattern. High memory

values for tokens in the pattern indicate a common token combination.

The combination of proximity activation and similarity activation aims to increase the weight of common token combinations. Each input token may activate numerous similar tokens through similarity activation, with each activated token residing in specific token combinations in the memory database. If multiple input tokens propagate activation values to a particular combination, that combination obtains high weight (high cumulative activation value). Essentially, proximity activation performs reasoning similar to the attention mechanism, inferring that tokens adjacent to input tokens may have higher weights. Its function resembles one layer of attention, while the iterative activation process (chain activation) resembles multiple attention layers.

3.5 Step 5: Chain Activation Process

An input token propagates activation value through similarity activation. Each activated token in the memory database can continue propagating if its activation value exceeds the preset stop threshold. Tokens activated through proximity relationships then launch their own similarity and proximity activation processes. This process iterates until all activation stops, forming a chained process. Consequently, the machine activates not only memory information similar to the input but also the "antecedents" and "consequences" of similar information in the memory database—temporal front and back information. Multiple "antecedents" and "consequences" with different reward or punishment correlations may be activated, providing the basis for machine decision-making.

3.6 Step 6: Accumulate Activation Values

If an activation value propagation path exists between a memory token and multiple input tokens, the activation values from each input token accumulate, yielding higher cumulative activation values. If a common token combination appears in the input, each of its tokens activates corresponding combinations in memory. Therefore, combinations with similarities to memory and input are activated, with higher similarity producing higher cumulative activation values. This gives common combinations greater weight in the machine's decision-making process, analogous to the attention mechanism in transformers: finding highly correlated token combinations and assigning them higher weights. In other words, common combinations see their activation values rise from zero when the chain activation process stops. These tokens constitute one or more "world models."

3.7 Step 7: Activation Values Decrease Over Time

All activation values constantly decrease over time. By accumulating activation values while simultaneously decreasing them, we establish connections between sequentially entered information. The activation value generated by previous information has not completely disappeared when the following information's

activation value is superimposed, making tokens with activation values in the entire memory database related to both prior and subsequent input. The decay rate represents the length of the association time window. High decay rates produce short windows, while low rates produce long windows. The machine's decisions are based on all activated tokens, so information within a specific time window is considered. This window length depends on both the decay rate and the initial activation value A_0. Larger activation values produce longer windows. Since A_0 relates to statistical reward or punishment values, the machine selects the time span of information based on these statistics, adjusting A_0 accordingly. This method surpasses the transformer's attention mechanism by being more human-like.

3.8 Step 8: Obtain Pretraining Weight Matrix

Our implementation forms a network where each token consists of four fields. Numerous tokens are stored chronologically while retaining time interval information. Through attention and memory-forgetting mechanisms, a fully connected knowledge network emerges. The memory value represents the pretraining weight, while the activation value represents the weight calculation result under the attention mechanism. If a token repeatedly appears in different token combinations, it obtains high memory values and establishes connections between these combinations. If a token combination pattern repeats, all tokens in this pattern frequently appear together, resulting in each token receiving additional higher memory values due to this pattern, forming a common combination.

The final result is a memory database where activation value propagation relationships represent "relationships between things." This network has tokens as nodes and activation value propagation relationships as connections. Each basic token combination forms through temporal proximity. Through memory and forgetting mechanisms, representative tokens are retained while rarely repeated ones are eliminated. Thus, different token combinations in memory, including their memory values, represent the statistical weights of these combinations. In our solution, the pretraining statistical process is not simply counting token repeatability but counting the repeatability of various token combinations, using their internal token memory values to represent overall weight.

When input tokens receive initial activation values and undergo chain association activation, common token combinations have their activation values recalculated based on memory value weights and connection relationships between input and memory combinations. This constitutes the inference process in the attention mechanism. Whether in LLMs or our solution, the essence of the attention mechanism is using incomplete statistical information (from training data) to infer the conditional probability of specific combinations—a Bayesian inference process. Our method represents another implementation of the attention mechanism.

3.9 Step 9: Preset Minimum Machine Requirements

We realized the attention mechanism without disrupting the original temporal and spatial organization of tokens. Consequently, the knowledge network formed by our solution is understandable by humans. We can therefore imitate the organizational form of tokens in the memory database to build the initial minimum innate knowledge for the machine, including minimal needs and communication means between machine and human. This resembles creating a baby machine (presetting what a baby is born with).

The method uses special tokens to represent each "demand," "reward and punishment," and "emotion." In daily life, these tokens, together with external tokens that trigger them, establish connectivity through the attention mechanism, forming the machine's self-constructed value system. However, before training, we must preset minimal connectivity among tokens related to "needs," "reward and punishment," and "emotion." This innate knowledge is necessary for machines to initiate learning like babies. These innate connection capabilities resemble a baby's instinctive reactions after birth. A baby must provide feedback on its status and have basic communication ability with the outside world; robotic babies need these capabilities as well.

A baby has initial communication means such as "crying," understanding "smiling" and "angry," enabling humans to communicate with and cultivate them into college students while aligning their values with human values. A robot baby establishes its values similarly. This is the purpose of establishing innate knowledge for robot babies—similar to human babies' innate capacity to know when to "cry" and recognize basic human feedback.

The common combination of objective world tokens forms "objective common sense," while combinations including needs/emotions tokens form "subjective common sense." Together, they constitute "common sense," which represents the "world models" containing the "Framework" of human cognition of the world and the relationship between world and self. "Framework" means these patterns can repeat in our world, discovered through attention mechanisms plus memory and forgetting mechanisms.

Notably, tokens are not only static features but also include simple dynamic features (such as rotation, swing, etc.). World models are also not static or fixed. In our scheme, the "world model" built by the machine directly relates to its training data and "life experience," so different machines have different "world models," similar to humans.

The activation value propagation path from input tokens to reward and punishment tokens forms a value logical chain compatible with the neural network. It is explicit, understandable, and imitable, making machine decisions visible. This represents a solution combining neural networks with causal reasoning—a key step proposed by Professor Yoshua Bengio for implementing AGI [**?**]. Step 9 is essentially the first step in creating a machine baby, but we can train on test

data through pretraining to understand the organizational forms of machine-created knowledge, then imitate these forms to implement Step 9.

[Figure 1: see original paper] illustrates the method for presetting innate knowledge. For example, we preset a symbol representing "hungry" in innate memory, placing a "punishment" symbol and an emotional symbol representing "hungry mood" next to the "hungry" symbol, giving them appropriate memory values. When power is insufficient, the vital state monitoring program directly assigns an initial activation value to the "hungry" symbol in innate memory, which propagates through the memory database like input tokens. The adjacent "hungry mood" and "punishment" symbols are activated, giving the machine a "hungry mood" and "punishment value." To avoid the penalty, the machine will use its experience to actively find a charging plug.

Presetting "higher-order needs" (such as caring for life, respecting others, showing self-respect and self-love) requires presetting the simplest means of human communication. Through human feedback, connections between these higher-order special tokens and other tokens are established. For example, specific symbols represent "respect for others," with related behaviors filled in during later training. Like a baby, we must educate the machine about "values" from an early age and throughout daily life, enabling seamless integration of values and knowledge. Our solution uses small-sample, cumulative, real-time learning, allowing seamless connection between values and knowledge. In contrast, LLMs form knowledge at once through large samples, then fine-tune through RLHF (Reinforcement Learning with Human Feedback), making seamless value-knowledge integration difficult.

How do we preset the most basic communication between machines and humans? For example, preset basic head-nodding features (assuming X tokens) and head-shaking features (assuming Y features). Place a "respected" symbol and a "reward" symbol next to nodding tokens, giving these symbols higher memory values to form long-term memory. When partial nod tokens appear in input, the machine obtains "reward value" through chain association activation. In pursuit of reward, the machine may plan various decisions to obtain human nods. A simpler approach is having the machine recognize "yes" or "no" in speech, text, or any symbols representing agreement or opposition.

Similar to a child, starting from the simplest communication methods, the machine gradually acquires complex learning abilities, including various communication methods. The "reward function" logical chain progresses from "milk," "pacifier," "bottle," "milk powder can" to "academic performance," "house, car," "social status," to "self-fulfillment." After training, numerous reward and punishment-related token combinations with causal relationships exist in the memory database, representing the value system—similar to human development.

3.10 Step 10: Form a Fully Connected Knowledge Network

Our solution creates a network encompassing knowledge, value systems, and machine needs. The network's nodes are tokens, and its connections are activation value propagation relationships. Because the network contains machine "needs" and related "logical chains" (activation value propagation chains), the machine will proactively learn and iterate itself. For example, it may spontaneously recharge batteries, go to the library to read books, or search Google—behaviors not assigned by humans. In our solution, knowledge and decisions develop around "demand," the core reason our machine can achieve "universality." A machine faces only one task: "needs," rather than countless "external tasks." Thus, our solution represents "active wisdom," while other solutions represent "passive wisdom."

Our solution enables small-sample learning, real-time knowledge updates, and integrated training and usage processes, making the machine a lifelong learner capable of self-iteration. Because the machine's knowledge exists as a memory database stored chronologically, preserving relative time interval information rather than absolute time, different memory databases can be directly stitched together to form larger databases. By combining a chef's memory database with a doctor's memory database, a robot can possess both skills instantly without retraining on massive combined data. Current AI technology cannot achieve this. In LLMs, mastering both skills requires training on large amounts of both doctor and chef data, making it impractical for machines to possess diverse abilities such as doctor, teacher, nanny, and president simultaneously. shows a simple schematic of the memory database structure (text tokens only).

4 Why the Knowledge Network Works

Deep learning pioneer and Turing Award winner Professor Yann LeCun believes the right direction for AGI is the "world model" through "humanoid AI," proposing three conditions [**?**]: (1) AI must learn to represent the world; (2) AI must learn to think and plan in ways compatible with gradient-based learning; (3) AI must learn hierarchical representations of action plans. While they proposed these ideas, no complete technical solution exists. Our scheme can achieve all three conditions.

4.1 The World Model

Input tokens activate token combinations in memory, and high-activation-value combinations become "activated world models." World models are speculations by the machine based on input tokens. Some tokens may already be in the input, while others may not. According to the "seeking benefits and avoiding harm" rule, the machine decides whether to seek more input information for better decisions, depending on whether its current value statistics have converged. Further recognition incurs energy and time costs (negative value), so the machine must weigh whether additional value from new information exceeds the

cost. This forms the basis of its decision-making. If further recognition occurs, the machine adds details to make abstract world models more concrete; otherwise, it initiates the response process. Thus, machine recognition of external information is proactive and on-demand, unlike other AI systems.

The identification process determines which large category input belongs to through common features, then identifies specific subcategories through unique features—different from deep learning's goal of finding unique characteristics. "World models" contain two aspects: (1) the machine iteratively identifies the world through "pattern recognition"; (2) the machine finds "the relationship between the world and itself." Therefore, world models concern not only how the world operates but, more importantly, the relationship between the world and the agent. Human knowledge creation centers around value, which our solution adopts. LLMs focus mainly on discovering objective connections, adding value only through final patching—an incorrect knowledge creation method.

Why are high-activation-value token combinations "world models"? First, the machine selects a small number of highest-activation-value tokens as expected model elements. Due to high repeatability, these high-memory-value tokens are usually common features of large object classes, with high generality and representativeness—they are abstract concepts. Abstract concepts limit token numbers while maintaining high representativeness to cover large ranges. High-activation-value token combinations precisely match these abstract concept characteristics. The machine then enters decision-making based on abstract world models. If it decides to further recognize information, it concretizes abstract models by adding new tokens, increasing token numbers, reducing representation scope, and making models more specific. Thus, world models are created layer-by-layer from abstract to concrete, crucial for empirical generalization. Experience always comes from specific past events, yet differences exist between past experiences and current tasks. However, if some attributes belong to a certain abstract concept, that concept becomes the bridge for empirical generalization. Therefore, forming hierarchical world model structures is crucial for empirical generalization and a key AGI step.

A world model does not create independent models according to "words"; it contains tokens spread throughout the memory database, with patterns temporarily created by activation processes. It is temporarily connected by numerous tokens and activation value propagation processes related to input information. Thus, a "world model" is not static but distributed, temporarily composed of high-activation-value tokens motivated by input. The world model's meaning is the common combination sequence of tokens—an abstract summary of objective things including their advantages and disadvantages for the intelligent agent. It is temporarily created in memory through chain association activation, with no separate world model existing in the memory database. This implements "world models."

4.2 Logical Reasoning Capabilities Compatible with Neural Networks

All "reward and punishment" tokens motivated by input tokens have activation values representing value predictions. The propagation paths from input tokens to activated reward and penalty tokens represent reasoning ability fully compatible with connectionism. Activation value propagation is essentially an inference process.

The attention mechanism in LLMs has partially implemented logical reasoning compatible with neural networks through RLHF, but with a defect: the value system is achieved through post-hoc fine-tuning after knowledge system completion. Although LLMs have a value dimension, they lack information for projecting knowledge into the value dimension. Turing Award winner Professor Yoshua Bengio, a deep learning pioneer, believes the most important AGI step is combining neural networks with causal reasoning. Our scheme achieves this: the memory network is a fully connected neural network; from input tokens to activated "world model" is causal reasoning of objective world organization; from input tokens to activated "subjective token combinations" (representing demands, emotions, and reward/punishment tokens) is causal reasoning between the objective world and machine needs.

RLHF is indeed a method to build value systems in LLMs, but is it too late for a child to start learning right and wrong concepts from parents' "yes" or "no" after earning a doctoral degree?

4.3 Hierarchical General Decision-Making Capability

The machine performs reinforcement learning on only one task: "How to meet my self's needs?" It handles only one task: "how to meet my self's needs." Thus, the machine makes decisions by "facing its own needs," while other AI solutions face various "external tasks." After information input, the machine obtains various activation value propagation paths from input to reward and penalty tokens. It reduces the probability of paths bringing "punishment" and increases those bringing "reward." This constitutes "general decision-making" in our solution, similar to human decision-making and therefore universal.

4.3.1 How to Achieve True "Machine Learning" Current AI typically uses RL, a "try first, then eliminate" approach that should be called "machine evolution" rather than "machine learning." Real machine learning should, like humans facing new tasks, predict "good or bad" outcomes across different decision paths based on past experience. Through limited trials, the machine can find good solutions for new tasks based on experience.

Furthermore, real learning should resemble children's learning style—directly acquiring accumulated human experience through language. For example, when teachers teach children experiments in laboratories, they pass existing human decision-making experience directly through language instruction. After obtaining this knowledge, children can use the experience gained from language and

interact with the environment to complete experiments, even in different laboratories for the first time.

Real tasks and scenes vary greatly, making it impossible to conduct extensive RL for each task type. Our solution transforms all tasks into a single task: "How to meet myself's needs?" All machine training focuses on this single task, providing abundant "state" and "policy knowledge" for predicting potential "pros and cons." For example, if "acquire human approval" is a basic machine need, machines incorporate "completing human tasks" into overall pros/cons statistics, similar to humans finishing boss-assigned tasks while weighing personal benefits.

In our solution, a machine counts values obtained across different propagation paths, then seeks maximum benefit. Each propagation path represents a decision path, with activation values of reward and punishment symbols representing potential reward or penalty values. The machine balances long-term and short-term benefits, planning behavior according to maximum benefit—forming a human-like thought chain.

Current AI's core obstacle is RL, which requires two prerequisites [**?**]: (1) Because reward information is scarce and delayed, current solutions require extensive trial-and-error training. (2) The machine must search all possible decisions. These conditions are perfectly satisfied in games, where constant trials are possible and decision search spaces have boundaries. However, real-life tasks often lack trial-and-error data (e.g., childcare—no one wants extensive trials) and usually have no clear task boundaries. This explains why many AIs can play complex strategy games but cannot launch basic "home nanny robots." Our solution's decision-making process is also essentially RL, but only on how to seek advantages and avoid disadvantages. We use the chain association activation process to automatically limit the search range to activated information. Moreover, we use the logical chain of "tokens" and "reward and punishment symbols" to automatically predict reward and punishment information rather than obtaining it only through post-hoc feedback, solving the aforementioned problems.

4.3.2 Implementation Flow of "Hierarchical Decision" Input information includes all sensor data, making environmental information at any moment part of the input. Machine-environment interactive decision-making includes two aspects: (1) optimal decision-making; (2) decision execution. These steps are intertwined and processed in parallel.

Step 1: What is the purpose? When information (tokens from outside or machine monitoring programs) inputs, some reward and punishment symbols activate. Each propagation path from input represents a potential logical value link. If realized, the reward or punishment is realized. Therefore, the machine's response to any input is consistent: increase reward logic chain occurrence probability and reduce punishment logic chain probability.

Step 2: How to plan with a purpose?

1. How to increase reward links and reduce punishment links? The method is increasing or decreasing realization probability of high-activation-value tokens on a path. High-activation-value token combinations on a path are high-weight tokens of this logic value chain. When they are true, activation spreads along the chain, making final reward or punishment true.

2. How to operate specifically? From the activation value propagation path of input tokens to reward and penalty symbols, select the N tokens with highest activation values. Their combination forms one or more abstract world models because these tokens are either highly representative or closely related to input. Machine decision-making is hierarchical: first compose decision paths with abstract concepts (top-level thinking), then gradually add tokens to form more concrete concept combinations. This top-down, gradual decision-making and execution process is called "segmented imitation."

An example of segmented imitation: Consider input token combination A and response token combination B. The machine finds the activation path from A to B, identifying high-activation-value tokens on the path as bridges connecting A and B. Call these bridges "C," an abstract solution. The machine then searches for high-activation-value token combinations between A and C, finding a more specific solution "AC." Similarly, it finds "CB" between C and B. Thus, abstract solution C decomposes into more specific AC + CB—a top-down decomposition. This process iterates, enabling layer-by-layer decisions.

Computer implementation may follow this method: (1) All activated reward and penalty tokens are level-1 targets. (2) Find the N highest-activation-value tokens on propagation paths from input to each target. These tokens form abstract concepts—top-level thought chains. (3) Use each abstract concept as level-2 targets, assign initial activation values, and initiate chain association activation again. Tokens with highest activation values are those associated with previous input and level-2 targets. Adding these tokens expands top-level thinking chains with more details into more specific chains. Repeat to obtain level-3 targets. (4) Iterate this process. (5) Each decision expansion activates new reward/punishment symbols and may change original symbols' activation values. Following benefit-seeking and harm-avoidance principles, the machine chooses sub-paths bringing reward value and avoids those bringing penalty, increasing cumulative reward value. When total reward and penalty values converge (cannot be further improved), benefit is maximized and the machine stops expanding, entering execution. This is the hierarchical "general decision ability" proposed by Professor Yann LeCun and the logical reasoning ability compatible with neural networks proposed by Professor Bengio.

This process resembles LLM's next-vector search: using preliminary output as input to find the next target. The difference is LLM searches chronologically, while our solution searches top-down, from frame to detail. Essentially, our

solution is also an LLM, but integrates a value system during knowledge building rather than patching it at the end.

4.3.3 Implementation of Empirical Generalization Why select only a few high-activation-value tokens for each path expansion? Past experience cannot fully match current tasks. Selecting only highest-activation-value tokens means the "model" is either abstract (widely applicable) or closely related to input (good match), achieving empirical generalization. Both past experiences and present tasks are represented by tokens, their combinations, and token weights. Present tasks and past experiences can be decomposed into different properties through local token combinations. Through chain association activation, we find common tokens in both current tasks and past experiences, representing shared properties—the part of past experience suitable for current tasks and generalizable. Thus, our solution automatically implements empirical generalization.

The experience generalization process first abstracts past experience (selects local high-activation-value tokens), generalizes to current tasks, then iteratively finds past experience again, abstracts again, and generalizes to specific implementation steps. This hierarchical iteration process searches top-down for optimal experience step-by-step. It does not imitate a single past experience period but combines many fragments, which we call "segmented imitation." It adopts "overview first," then adds details layer-by-layer. Without appropriate experience, it divides into more steps to find useful experience fragments.

With new information input at any time, chain association activation updates activation value distribution in the memory database. The machine must recount reward and punishment information and re-find optimal decision paths. Past experiences can come from the machine itself or linguistic symbol combinations. Images can be considered a universal language. When language symbol sequences (including images and videos) activate, relevant tokens represented by language symbols also activate through chain association activation. Behind language symbol sequences is the temporal and spatial information flow of various token types, similar to any real token information flow. Thus, experience comes not only from the machine itself but also from "others' experiences" gained through linguistic symbols. Our machine can learn "experience" through language symbols and imitate "experience" through token information flows activated by language sequences. Therefore, our robot can start working on its first day by reading operating manuals without extensive pretraining. Machines in our solution can directly access all experiences accumulated in human civilization history.

5 A Workflow Diagram

[Figure 2: see original paper] shows the workflow diagram. Boxes 1 and 2 represent innate knowledge establishment, which must be completed first. Box 31 shows token extraction from input information. Box 33A indicates that machine

demand, reward/punishment, and emotional tokens may also be activated by the machine's status monitoring program, making these activated tokens part of input information. Another input source in box 33A is the machine's decision-making process. During decision analysis, machines may need supplementary information for intermediate targets, such as in top-down hierarchical decision-making. The machine then uses tokens forming these intermediate targets as input, assigns them initial activation values, and searches for relevant experiences again through chain association activation.

Box 32 shows tokens saved in order into the memory database, retaining time interval information. Box 33B completes the chain association activation process. Box 34 shows the machine selecting a few high-activation-value tokens to form abstract paths from input to reward and punishment tokens, as concepts composed of these tokens are usually abstract.

Boxes 51-56 represent decision-making and execution—only one step in top-down decision-making. The generated decision path contains intermediate goals requiring further iteration and refinement until executable driving command sequences emerge. Arrows return from box 6 to boxes 33A and 31: returning to 33A means the machine activates its own needs, rewards/punishments, emotional tokens, or requires further analysis of intermediate goals; returning to 31 means the machine needs to extract more external information. This is driven by the machine's attention mechanism, which actively selects information extraction forms, focuses on information ranges, and uses appropriate token resolution based on expected size. This is an active recognition process for external information.

The machine iterates through the process shown in [Figure 3: see original paper] to establish and execute decisions top-down.

6 Problems Solved by Our Solution

6.1 How to Build Common Sense

In our scheme, "knowledge" is essentially the permutation relationship of tokens in time and space, which naturally represents causal relationships. Through chain association activation, tokens with large temporal/spatial spans can form close relationships similar to local permutations. If knowledge includes tokens related to "needs," "emotions," and "pros and cons" that can predict potential outcomes, these token arrangements represent "subjective knowledge," while knowledge including only external tokens is "objective knowledge." Combined subjective and objective knowledge constitutes common sense.

6.2 Whether Machines Should Have Consciousness

We have solved how to establish "self-needs" for machines. Therefore, machines can make independent decisions, self-evolve, have emotions, and pursue "self-needs," making our machines "conscious." We believe machine consciousness is

a way to resolve AI's generality problem.

6.3 How Machines Accomplish Assigned Tasks

Facing any task, machines make decisions according to "seeking good and avoiding harm." Human-assigned tasks are by-products of machines' pursuit of "self-needs," similar to humans completing boss-assigned tasks while pursuing personal needs. If conflicts arise, flexible decisions are made according to benefit-harm analysis, testing the boss's true intentions and bottom line, making decisions very flexible.

6.4 Understanding Language

Because we didn't break the original temporal and spatial relationships of tokens, the token temporal and spatial sequences activated by language can be understood and imitated. Thus, machines can learn various skills directly through language, like humans reading an oven manual and starting to toast [**?**][**?**][**?**].

6.5 Handling Difficult-to-Trial-and-Error Tasks

Real life contains many tasks difficult to trial-and-error extensively, such as autonomous driving, home nanny services, elderly care, and child companionship. Because our solution creates "humanoid" AI, it can make general decisions and learn skills through language like humans, obtaining relevant work experience directly through language learning rather than personally traversing various work situations. LLMs cannot handle these tasks without task-specific RL because they can only imitate language itself, not the information sequences behind language, which deep learning has disrupted.

6.6 The Hallucination Problem

LLMs store information as coordinate bases. Their powerful capabilities create coordinate bases far surpassing humans'. Usually, any common information can become a coordinate base regardless of length. However, LLMs do not preserve factual memories. They synthesize content by combining basic information, making information creation their essential job. When required facts are not coordinate bases, LLMs create information—this is the source of hallucination problems, which are difficult to solve.

Some expect to solve hallucination through external fact databases or online queries. This is like making an ordinary person using a dictionary or Google become a professional translator—if difficult to achieve, then solving hallucination through external databases or Google is also difficult.

Our approach has facts first, then extracts frameworks from factual memory to form knowledge. Thus, the closest match to a factual question is the corresponding factual memory. Additionally, our more complete value system adopts

different strategies for different value forecasts. If statistical rewards and punishments do not fall into decision intervals, making decisions difficult, the machine's strategy may be to not respond immediately but create a higher-priority task: finding more relevant information to help decide. It may ask the questioner back, actively search for information, simply state it doesn't know, or adopt avoidance strategies instead of answering or refusing like an LLM.

6.7 Safety

Current AI is single-goal AI—"one-track mind" AI focused on achieving a single goal. Such AI doesn't consider anything outside its goal, and decisions remain black boxes. Consider how dangerous "stuffy pot" + "one-track mind" people controlling your life would be. If such AI fully controlled human life, it could bring incalculable disasters due to wrong understanding.

In our solution, machine "demand types" can be preset, values can be trained, and human values can be aligned. At any time, the machine considers various goals without "extreme" behavior. Moreover, decisions are visible, modifiable, and "white box," making our solution safer.

7 Conclusion

"Build a baby machine, then learn lifelong, and grow up"—this idea has existed for years. Now we propose detailed solution steps.

AI technology needs to move to the next stage: the "autonomous interaction" stage. "Autonomy" means the machine is no longer silent but spontaneously produces behavior (equivalent to self-programming). "Interaction" means the machine can interact with the environment in real time, update knowledge continuously, and make ongoing decisions to complete complex tasks in unfamiliar environments [**?**].

We must make machines preserve and understand temporal relationships between tokens, as temporal relationships represent causal logic necessary for understanding rather than imitating the world. Human cognition of the world began by summarizing temporal relationships between tokens. Babies perceive implicit connections between temporally proximate things through observation, forming the foundation for establishing all relationships between things.

Many scholars have proposed views on achieving true AGI: Professor LeCun proposed the "world model"; Yoshua Bengio believes the key step is combining neural networks with causal reasoning; Professor Song-Chun Zhu proposed four key AGI characteristics: (1) can perform unlimited tasks; (2) can independently generate new tasks; (3) value system-driven; (4) has a world model reflecting the real world. Our solution responds to these ideas from Professors LeCun, Bengio, and Zhu.

However, we found AI results are sensitive to parameters such as memory forgetting rate, activation value transfer coefficient size, and convergence standards

for pros/cons value statistics, requiring repeated trial-and-error adjustment with slow convergence. Required computing resources increase rapidly with memory database size, though theoretically our computational complexity should match LLM inference. It still far exceeds available resources.

In future research, we will consider introducing self-supervised learning and accelerating convergence through gradient optimization. In chain association activation, activation value transfer can be linear or nonlinear functions requiring trial-and-error optimization. Two natural nonlinear processes exist: (1) forgetting mechanism; (2) chain association activation stops when activation value falls below preset threshold. Thus, even with linear transfer functions, the overall optimization process remains nonlinear. Many details require further research.

AGI is AI's original intention and crown. We present technical solutions for implementing general AI with step-by-step implementation details. References [**?**][**?**][**?**][**?**][**?**][**?**][**?**] reveal more technical steps. This may be one possible path to AGI. We are deeply convinced that AGI technology belongs to all mankind, and the great wealth it brings also belongs to all mankind.

References

[1] Reed S, Zolna K, Parisotto E, et al. A Generalist Agent[J]. 2022. [2] Optimizing Language Models for Dialogue. https://openai.casa/blog/chatgpt/ [3] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback[J]. arXiv e-prints, 2022. [4] Vinyals O, Ewalds T, Bartunov S, et al. StarCraft II: A New Challenge for Reinforcement Learning[J]. 2017. [5] Quanjun Zhang, Chunrong Fang, Yuxiang Ma, Weisong Sun, Zhenyu Chen. A Survey of Learning-based Automated Program Repair. arXiv:2301.03270, 2023 [6] Antonio Mastropaolo, Luca Pascarella, Emanuela Guglielmi, Matteo Ciniselli, Simone Scalabrino, Rocco Oliveto, Gabriele Bavota. On the Robustness of Code Generation Techniques: An Empirical Study on GitHub Copilot. arXiv:2302.00438, 2023 [7] Introducing ChatGPT. https://openai.com/blog/chatgpt/ [8] Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, Lijuan Wang. arXiv:2210.09150, 2023 [9] An experimental open-source attempt to make GPT-4 fully autonomous. https://GitHub.com/Significant-Gravitas/Auto-GPT [10] Zhuosheng Zhang, Aston Zhang, Mu Li, Alex Smola. Automatic Chain of Thought Prompting in Large Language Models. arXiv:2210.03493, 2023 [11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022), 2022a. https://arxiv.org/abs/2201.11903 [12] AUTO GPT, https://auto-gpt.ai/ [13] ERNIE Bot, https://mp.weixin.qq.com/s/0-8X9FPouteKzNiK6DPaiA [14] Mccann B, Bradbury J, Xiong C, et al. Learned in Translation: Contextualized Word Vectors[J]. 2017. [15] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[J]. arXiv, 2017.

[16] Cheng J, Dong L, Lapata M. Long Short-Term Memory-Networks for Machine Reading[J]. 2016. [17] Lin Z, Feng M, Santos C, et al. A Structured Self-attentive Sentence Embedding[J]. 2017. [18] Parikh A, Tckstrm O, Das D, et al. A Decomposable Attention Model for Natural Language Inference[J]. 2016. [19] Paulus R, Xiong C, Socher R. A Deep Reinforced Model for Abstractive Summarization[J]. 2017. [20] Tamkin A, Brundage M, Clark J, et al. Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models[J]. 2021. [21] Kodge S, Roy K. BERMo: What can BERT learn from ELMo?[J]. 2021. [22] Lee H, Xu G. Optimizing Policy via Deep Reinforcement Learning for Dialogue Management[C]// 2018 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE Computer Society, 2018. [23] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback[J]. arXiv e-prints, 2022. [24] Sunehag P, Hutter M. Principles of Solomonoff Induction and AIXI[C]// Ray Solomonoff memorial conference. Research School of Computer Science, Australian National University, Canberra, ACT, 0200, Australia; Research School of Computer Science, Australian National University, Canberra, ACT, 0200, Australia, Department of Computer Science, ETH Zurich, Switzerland, 2011. [25] Yongcong Chen, Ting Zeng, Xingyue Chen. Method for implementing universal artificial intelligence. CN111553467A[P]. 2020. [26] Yongcong Chen, Ting Zeng, Xingyue Chen. A Machine Intelligence Implementation Method Imitating Human Intelligence, CN111582457A[P]. 2020. [27] Yongcong Chen, Ting Zeng, Xingyue Chen. Establishment of artificial general intelligence system, CN112016664A[P]. 2020. [28] Chen Y, Zeng T, Chen X. ESTABLISHMENT OF GENERAL-PURPOSE ARTIFICIAL INTELLIGENCE SYSTEM:, US2022121961A1[P]. 2022. [29] Yongcong Chen, Ting Zeng, Xiaoyi Qian. A Preliminary Study on the Capability Boundary of LLM and a New Implementation Approach for AGI. ChinaXiv:202305.00046 [30] Yongcong Chen, Ting Zeng, Jun zhang. A New Solution for Realizing True Artificial General Intelligence. ChinaXiv:202308.00026 [31] Yongcong Chen, Ting Zeng, Jun zhang. A New Solution for Realizing True Artificial General Intelligence. https://arxiv.org/abs/2308.09721 [32] Zhao WX, Zhou K, Li J, et al. A survey of large language models. (2023-03-31)/[2023-08-17]. https://arxiv.org/abs/2303.18223. [33] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners. Advances in Neural Information Processing Systems, 2020: 1877-1901 [34] Feng C, Zhang X, Fei Z. Knowledge solver: teaching llms to search for domain knowledge from knowledge graphs. (2023-09-06)/[2023-09-17]. https://arxiv.org/abs/2309.03118 [35] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, Yi Zhang, Sparks of Artificial General Intelligence: Early experiments with GPT-4. https://arxiv.org/abs/2303.12712v1 [36] Yoshua Bengio, From System 1 to System 2: Deep Learning for Cognition and Causality, https://static.aminer.cn/misc/pdf/NeurIPS-11dec2019.pdf [37] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, Nicolas Ballas,

Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture, https://arxiv.org/abs/2301.08243 [38] Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Benjamin D. Haeffele, Yi Ma, White-Box Transformers via Sparse Rate Reduction, https://arxiv.org/abs/2306.01129 [39] Stephen Wolfram, What Is ChatGPT Doing…and Why Does It Work, Wolfram Research, Inc. 2023/03/09, ISBN: [40] Maximilian Schreiner, Meta's AI chief: Three major challenges of artificial intelligence, https://the-decoder.com/metas-ai-chief-three-major-challenges-of-artificial-intelligence/

Note: Figure translations are in progress. See original paper for figures.