A Preliminary Exploration of the Capability Boundaries of Large AI Models and a New Approach to AGI Implementation

Authors: Zeng Ting, Chen Yongcong, Chen Xingyue, Zeng Ting, Chen Yongcong

Date: 2023-04-18T10:50:30+00:00

Abstract

Currently, mainstream artificial intelligence generally adopts the technical path of 'attention mechanism + deep learning' + 'reinforcement learning.' We contend that reinforcement learning is inapplicable to domains where extensive trial-and-error is difficult. Therefore, to achieve general artificial intelligence applicable to any domain, we must transform the implementation path. Accordingly, we propose a machine learning framework distinct from 'deep learning+reinforcement learning.' Through few-shot and cumulative learning, it similarly implements an attention mechanism analogous to transformers and constructs a fully connected knowledge network. Moreover, it can achieve interactive decision-making with the environment without resorting to trial-and-error learning. Additionally, humans can preset different innate requirements for it to achieve multi-objective balance, thereby attaining safety far exceeding that of current artificial intelligence. In this paper, we propose a novel machine learning technical framework from the ground up.

Full Text

Preamble

A Preliminary Exploration of the Capability Boundaries of Large AI Models and a Novel Path to AGI Implementation

Ting Zeng[1], Yongcong Chen[2], Xingyue Chen[3]
([1] Tsinghua University Library, Beijing, 100084, E-mail: ceng-t@mail.tsinghua.edu.cn;
[2] New Millennium Future Technology (Beijing) Co., Ltd., Beijing, 100084;
[3] Tsinghua University Affiliated High School, Beijing, 100084)

Current mainstream artificial intelligence universally adopts the technical path of "attention mechanism + deep learning" + "reinforcement learning." We argue that reinforcement learning cannot be applied to domains where extensive "trial-and-error" is infeasible. Therefore, to achieve general artificial intelligence applicable to any domain, we must shift our implementation approach. We propose a machine learning framework distinct from "deep learning + reinforcement learning" that similarly implements attention mechanisms akin to Transformers and constructs fully-connected knowledge networks through small-sample, cumulative learning. Critically, it achieves interactive decision-making with the environment without resorting to trial-and-error learning. Moreover, by pre-setting different innate needs, humans can enable multi-objective balancing, thereby achieving safety far exceeding that of current AI systems. In this paper, we present a novel machine learning technical solution built from the ground up.

Keywords: Large models, ChatGPT, GPT-4, General Artificial Intelligence, AGI

Current mainstream AI adopts the "attention mechanism + deep learning" + "reinforcement learning" paradigm. The "attention mechanism + deep learning" component primarily establishes knowledge networks, while "reinforcement learning" enhances continuous interactive decision-making capabilities between machines and environments. For instance, Google's "GaTo," launched in June 2022, enables a single model to perform over 600 different tasks [1]. Similarly, OpenAI's GPT-4 has achieved remarkable progress in knowledge understanding and reasoning [2]. However, where exactly do the capability limits of current large AI models lie? Can "attention mechanism + deep learning" + "reinforcement learning" realize true "General Artificial Intelligence"? We conduct a preliminary investigation into these questions in this paper.

2. Capability Limits of Current Large Models

First, we observe that current "reinforcement learning" relies on external feedback to inform machines about the "goodness" or "badness" of outcomes along different decision paths. This external feedback may originate from preset reward functions—for example, in AlphaGo, feedback comes from a reward function that determines wins and losses—or from human feedback, such as the RLHF (Reinforcement Learning from Human Feedback) technology widely adopted in current large models [3]. Consequently, the essence of current reinforcement learning is a "try-first, feedback-later" approach, where machines acquire reward information through different decision paths. Only then can machines grasp the trade-offs between "self-states" and "different strategies," thereby obtaining interactive decision-making capabilities with the environment [4].

We contend that when machines face environments requiring interactive decision-making, current reinforcement learning more closely resembles an

"evolutionary" learning approach whose essence is "trial-and-error" elimination through external feedback—strikingly similar to biological evolution. Such learning methods are only applicable in virtual environments like games, metaverses, and content generation tasks. For tasks requiring interaction learning in real-world environments—such as caring for the elderly or driving vehicles—reinforcement learning proves difficult to apply.

Second, knowledge created through "deep learning" remains incomprehensible to humans. This results in two distinct knowledge systems between machines and humans that cannot understand each other! For instance, humans struggle to comprehend the decision-making processes of large models. Conversely, large models cannot truly understand the knowledge represented by human language. No existing AI can read a bread machine manual once and operate bread machines in different bakeries [5][6][7]. Humans, however, can acquire accumulated knowledge about bread machine operation through reading manuals, enabling them to make decisions and interact with the environment when facing the novel task of operating a bread machine. When opening a bread machine lid, humans don't need to try various approaches (such as smashing it open) but instead directly access accumulated human experience through language.

Therefore, we believe true machine learning should, like humans, predict the "goodness" or "badness" of different decision paths based on past experience when confronting new tasks, requiring only limited trials to obtain decision-making knowledge for handling novel tasks. Furthermore, true learning should resemble child learning, directly acquiring accumulated human experience through language. When facing new tasks, success should be achieved on the first attempt without any trial-and-error! For example, in laboratories, teachers instruct children through language, directly transmitting existing decision-making experience. Children can then interact with the environment through step-by-step decision-making to complete experiments directly, even if it's their first time performing them.

Consequently, we argue that current large AI models cannot resolve the following critical defects due to improper knowledge structures and learning methods:

2.1. Inability to Solve Problems Autonomously

Current AI systems do not proactively offer assistance when they see their owner fall [7]. This occurs because machines lacking their own needs cannot generate their own goals. Without self-generated goals, machines cannot spontaneously create tasks. Therefore, when facing endless unexpected situations in real social life, machines can only process them through preset workflows derived from externally human-written prompt procedures [8] or by using input information to search for similar processes in knowledge bases for wholesale imitation.

Some researchers have attempted to implement recursive function calls. For instance, under a given task, they call a function to "search for past processes completing similar tasks," then recursively call the same function for each ob-

tained process to acquire more processes. This recursive approach is exemplified by projects like Auto-GPT, which aggregates knowledge obtained from search engines and GPT-4, then uses it as prompts to find solutions from GPT-4. Through recursive calls in this process, relatively macroscopic tasks can be completed. If all required processes and parameters already exist in the large model's function library, the model can complete the task. However, Auto-GPT cannot surpass the capability boundaries of what large models can accomplish. Moreover, in real life, endless unexpected situations may arise where no suitable process or parameter matches the current environment at some point, causing recursive calls to fail [9][10][11].

Thus, large models do not spontaneously create new processes! Large models are essentially advanced programming languages whose syntax is natural language. Understanding this reveals the boundaries of large models. No matter how many advanced functions we add to large models—regardless of how many tools or apps we integrate—large models will not spontaneously create new processes. All their processes either imitate past processes (recursive calls) or follow human-preset processes (main program calls). Both approaches essentially use "preset processes to handle all problems." No matter how many if…else… branches are considered or how many possibilities are accounted for, these processes are preset and pre-existing—not created on-the-fly for specific tasks!

Therefore, the core essence of large models is providing a more user-friendly "programming platform" where the interface is natural language. Previous programming languages were computer languages like C++, Java, and Python. Large models simultaneously provide a function set callable through natural language. For example, the "create PowerPoint" function operates under the pattern of human language command + required parameters. Large models complete tasks in this command + parameter mode, making everyone a programmer in the future. Those who understand large models better can write superior programs—these individuals are called "Prompt Engineers," the programmers of the future.

Large models continue enriching their functions. For instance, future function input parameters may include images, videos, actions, or any sensor inputs, while outputs may similarly include multimodal sequences [13]. Consequently, all future processes can be programmed using large model programming language (natural language). Large models are fundamentally machine language platforms no different from Python. In Python programs, humans translate ideas into Python program flows, and the Python platform translates functions into machine programs understandable by lower-level drivers to complete tasks, thereby implementing human-preset problem-solving workflows.

Thus, large models represent the next generation of Python. Python is more popular than C++ because it offers a better human interface. GPT-4 will become more popular than Python for the same reason. However, C++ hasn't disappeared because it retains advantages in specific domains. Similarly, future specialized large models or Python won't disappear—they'll maintain niche ex-

istence. All future software can be rewritten in large model language, and a new generation of programmers (Prompt Engineers) proficient in large model programming will become the next generation of "white-collar workers" [14].

The core essence of large models is therefore a more user-friendly "programming language" with a natural language interface, rich and diverse functions, multimodal capabilities, and an ecosystem expected to be established soon. Yet its nature remains that of a programming language! It is a programming language with extremely low entry barriers and vastly expanded functionality. Consequently, large models can complete any task with predictable workflows. Complex tasks simply require more if…else… branches. However, real life contains numerous tasks with unpredictable workflows! Examples include caring for the elderly, driving, cooking, child-rearing, and agricultural production. While humans can anticipate various possibilities, unexpected situations inevitably arise during real-world interaction. How machines handle these unpredictable situations becomes extremely dangerous, especially when large model functions permeate all aspects of human life, potentially causing unacceptable losses.

2.2. Knowledge Cannot Be Updated in Real Time

Current AI systems employ large-scale data training, making real-time knowledge updates impossible. Real-time knowledge updating is crucial for machines interacting with environments because such interaction constitutes the process of acquiring new knowledge. If acquired knowledge cannot be updated in real time, machines will repeatedly make the same mistakes when facing identical environments [3].

2.3. Inapplicability to Domains Requiring Real-World Interaction

In domains requiring interaction with real-world environments—such as autonomous driving, household chores, and patient care—machines must establish interactive decision-making knowledge between their actions and the external environment. These domains cannot accommodate extensive trial-and-error, preventing machines from building decision-making knowledge through reinforcement learning interactions in real environments. Consequently, current AI technical solutions cannot be applied to these fields [2].

3. Preliminary Exploration of Current Large Model Capabilities

3.1. How to Describe Information Contained in a Matrix?

How do we describe the information contained in a matrix? Although a matrix may contain many vectors, our primary concern is: how many independent vectors exist? In other words, what is the matrix's rank? Through matrix diagonalization, we can find a corresponding set of eigenvectors whose quantity equals the matrix's rank, serving as a complete coordinate basis cluster for the

matrix. This coordinate basis cluster is complete, orthogonal, and represents the most concise description of the matrix. Any vector in the matrix can be expressed through this coordinate basis cluster.

If the coordinate basis cluster we establish is not orthogonal but remains complete, we can still express any matrix information using it. If the coordinate basis cluster is incomplete, some vectors in the matrix cannot be expressed through it, requiring increased dimensionality of the coordinate basis cluster.

How then do we identify what information a vector contains? Clearly, with a complete basis cluster, each basis represents one dimension. Projecting any vector onto the coordinate basis accurately captures all information contained in that vector. If the basis coordinate cluster consists of eigen-orthogonal bases, we achieve the most concise coefficient expression of the vector's complete information. If the basis coordinate cluster is not fully orthogonal, we desire it to approximate an orthogonal basis cluster as closely as possible, yielding a sparse coefficient matrix.

Through a sparse coefficient matrix combined with a basis coordinate cluster, we can understand the information contained in any vector within the matrix, with information components being relatively independent. Therefore, the relationship between two vectors manifests in the relationship between their coordinate basis components.

Thus, if two vectors in a matrix share non-zero components in the same dimension, we consider them to possess local similarity, implying some connection relationship between them. If two vectors exhibit multiple local similarities, we consider them to have a stronger connection relationship. If such connection patterns repeatedly appear throughout the matrix, we can consider this connection a universal law—this is knowledge.

Therefore, seeking knowledge within a matrix means seeking all universal laws. Laws themselves are expressed through partial basis components of the matrix, including both dimensions and magnitudes. They represent shared connection relationships extracted from numerous vector interconnections, containing lower dimensions and broader representational scope. Knowledge is thus generalized from massive vector relationships, with each piece of knowledge itself being a vector expressible through a coefficient matrix. Large collections of such knowledge constitute knowledge networks.

Consequently, to discover all knowledge in a matrix, the most important steps are: find a basis coordinate cluster, use it to decompose any vector in the matrix, obtain connection relationships between different vectors through coefficient matrices, and identify repeatedly occurring connection relationships—these constitute extracted knowledge.

However, a matrix can have multiple basis coordinate clusters. Any basis coordinate cluster with dimensionality no lower than the matrix's rank is essentially viable. To obtain the most concise knowledge system, we can adopt orthogo-

nal coordinate clusters. But when matrices are extremely large, computing an orthogonal basis cluster becomes impractical. In such cases, we can use the most frequently recurring knowledge as the basis coordinate cluster, enabling sparse matrix expression for at least the majority of knowledge. This yields a coordinate basis cluster that concisely expresses common knowledge within the matrix.

With a basis coordinate cluster, any vector can be decomposed into the cluster and represented by a coefficient matrix. Similarity between any vectors can be expressed through spatial distance, computable via Euclidean-like distance measures. Mapping relationships between any vectors can be implemented through mapping relationships between coefficient matrices.

3.2. How Does Deep Learning Create Knowledge?

Imagine a group of people in an alien world who have established a 4-dimensional information space containing numerous images, sounds, and actions. In that world, humans perceive reality through pixels, syllables, and motion patterns as their baseline perceptual capabilities. Faced with countless combinations of pixels, syllables, and motion patterns, they cannot comprehend the connection relationships between them.

They therefore adopt trial-and-error methods, experimenting with different basis coordinate clusters, hoping to find a specific cluster where combinations of pixels, syllables, and motion patterns of interest form distinct, separable clusters. This would be their desired basis coordinate cluster.

Since original data has extremely high dimensionality—for example, a $64\$\times64 image has 64\times64 dimensions, representing a 64\times64 array of 2D impulse functions as the original basis coor$ because their goal is to find commonly used feature combinations rather than express all information. Some information loss is inevitable; they cannot preserve all original information in the final dimensions. A plausible trial-and-error approach involves discarding or compressing partial dimensional information after each coordinate basis cluster attempt, then comparing with targets to observe whether error increases or decreases, thereby determining the direction for the next coordinate basis cluster adjustment. Clearly, each attempt results in partial information loss. After multiple attempts, excessive information loss may lead to loss of useful information, preventing task completion. The proportion of useful information in overall information and the information loss rate after each transformation determine the maximum number of transformations possible. Within limited transformation attempts, machines may struggle to find optimal coordinate basis clusters. A viable solution involves partially recovering lost information after each transformation, enabling increased transformation layers and improving the probability of finding optimal coordinate basis clusters. This is the residual network approach. Alternatively, inserting weak non-linear functions during multiple neural network mapping processes can also increase feasible transformation counts.

In trial-and-error searches for optimal coordinate basis clusters, the search direction is error reduction, implemented through the BP algorithm. Data appearing in neurons are actually coefficient matrices under basis coordinate clusters, while the basis coordinate clusters themselves remain implicit and do not appear in multi-layer neural networks. Inter-layer transformation coefficients represent coordinate transformation matrices between one implicit basis and another. The BP algorithm adjusts these coordinate transformation matrices to move from one implicit coordinate basis to another awaiting trial.

If selected basis coordinate clusters are non-orthogonal systems, modifying coefficients in one dimension may affect coefficients in another dimension. This can lead to situations where adjusting all coefficients no longer reduces overall error, fundamentally caused by non-orthogonal basis coordinate clusters. Such situations cannot occur with orthogonal systems. Therefore, during trial processes, movement toward orthogonal coordinate clusters is necessary. A hallmark of approaching orthogonal coordinate clusters is coefficient matrix sparsification. Consequently, the entire trial direction must incorporate coefficient matrix sparsification constraints—this is the essence of various regularization methods.

Additionally, whether convolution, pooling, or other deep learning variants, their fundamental nature remains unchanged. For instance, convolution is essentially a single-layer neural network mapping where numerous zero coefficients are artificially preset in the coordinate transformation matrix. Pooling is merely a single-layer neural network mapping that removes partial dimensions and employs specific non-linear functions. This is the essence of deep learning—the method for finding a coordinate basis within a matrix.

If data in the space has labels, the number of labels determines the final required dimensionality of the basis coordinate cluster. Their goal in finding the final basis coordinate cluster is to use the smallest-resolution feature combinations (here, pixels, syllables, and motion patterns) shared by each labeled data category as representatives of each label category and as a basis coordinate cluster. This yields sparse coefficient matrices. Thus, we see that deep learning's essence is also finding a suitable basis coordinate cluster within an information matrix.

If needed information is viewed as an information subspace, deep learning's obtained basis coordinate cluster can express any vector within this subspace—this is supervised learning. If this subspace is large enough to contain all information, deep learning's obtained basis coordinate cluster can express any vector in the entire information matrix—this is unsupervised learning. If the primary learning objective is clustering while discarding non-clusterable information, this is also unsupervised learning.

3.3. What Is the Essence of the Attention Mechanism?

The core of the attention mechanism is discovering common arrangement patterns among information matrix elements. These common arrangement patterns can be selected as the "framework" of the information space. A "framework" is

universally present in the information matrix, and using it as a coordinate basis cluster enables concise description of common vectors within the matrix.

More 通俗 ly, we can consider each character in language information space as one dimension. Using such a coordinate basis cluster allows us to describe any vector in language information space. However, this coordinate basis cluster may not be optimal. If we treat language information space as a matrix, the optimal coordinate basis cluster is clearly composed of the matrix's eigenvectors. An eigenvector-based basis cluster has minimal rank and provides the most concise information description.

For example, we can decompose the sentence "I am going to attend my friend's wedding today" by treating each character as a dimension, obtaining a 12-dimensional coefficient matrix: "I, today, am, going, to, attend, my, friend's, wedding." Alternatively, we can use "subject…predicate…object" as a coordinate basis with coefficients "I…attend…wedding," use "adverb+predicate" as a coordinate basis with coefficients "today am going to attend," and use "attributive+object" as a coordinate basis with coefficients "my friend's wedding." The latter approach employs a more concise basis coordinate cluster expressing the same information, and this basis coordinate cluster is framework-based. These framework-based coordinate clusters can generate vast amounts of similar information under different coordinate components.

The core of the attention mechanism is establishing such "framework-based" coordinate basis clusters [16]. The method involves extracting common underlying frameworks among information matrix elements—a process highly analogous to human learning. When humans learn information from a book, the approach of "first reading thin, then reading thick" follows the same principle. "Reading thin" involves summarizing framework information—a compression process—while "reading thick" involves adding different details to framework information to create new knowledge—an information creation process [17][18][19].

Thus, Transformer-based large models fundamentally employ the attention mechanism [15] whose core purpose is to obtain common arrangement patterns among information matrix elements and weight them by frequency. More common arrangements receive higher weights. These high-weight arrangement patterns constitute the main framework for organizing all information within the information matrix.

This process closely resembles signal processing in communications. In the time domain, seemingly chaotic information reveals low-frequency components in the frequency domain that determine the signal's major trends—these are the signal's principal components. These low-frequency components represent common organizational forms within this class of signals. Treating each low-frequency component as a basis coordinate cluster component makes them analogous to attention mechanisms. Low-frequency components express common connection relationships between information—the "framework" of information. Therefore, the attention mechanism obtains the "framework" of information organization

by seeking weights of connection patterns between information elements. These "frameworks" form the foundation for generalization. Mapping relationships between "frameworks" represent algorithms from one "vector" to the next. Inputting "framework" + different details yields specific input vectors, and applying the "vector-to-next-vector" algorithm produces output vectors—this is knowledge generalization.

In fact, humans employ the same method during learning. Common feature combinations constitute specific "concepts"—high-weight combinations representing spatial and temporal commonalities. Further summarizing common feature combinations from specific "concepts" yields "abstract concepts." This process is iterative, resulting in numerous hierarchical "concepts" in human society—these are frameworks.

Thus, the abstract framework "cat" is a common arrangement pattern of matrix elements in space and time, potentially containing multimodal matrix information elements including language, text, sound, images, actions, and tactile sensations. Some matrix elements within this arrangement may have higher weights due to greater commonality, possibly all belonging to the "animal" concept. "Animal" contains fewer elements, has more limited scope, and broader applicability, so knowledge associated with their shared feature combination (the "animal" concept) can directly generalize between "cat" and "dog."

The attention mechanism's core capability is thus instantiating statistical correlations between languages into specific input information. Statistical correlations between languages are obtained through pre-training as incomplete statistical associations—it is impossible to statistically correlate all possible arrangements of language combinations. Therefore, actual correlations between languages under specific arrangements require further optimization based on statistical associations, a step completed by the attention mechanism.

The attention mechanism's core purpose is to start from statistical correlations, employ trial-and-error methods, use human language context as self-supervision, find correlations between input information elements or between input and output information, and express these correlations through weights. This correlation-finding process closely resembles human learning and summarization. Thus, machines use deep learning (essentially trial-and-error) to find an optimized coordinate basis cluster, which likely aligns closely with human common concepts. Deep learning's core is using trial-and-error to find basis coordinate clusters, while the attention mechanism's core is using trial-and-error to align basis coordinate clusters with human concepts.

3.4. Why Do Large Models Exhibit Emergent Capabilities? When Does Emergence Occur?

Why do large models demonstrate "emergence"? The logic is simple: when an American comes to China, they can achieve correct translation through extensive shared background information (such as human needs, social structures,

etc.) with moderate Chinese-English comparisons. Large models, however, are like aliens lacking common background information with humans—they only observe connection patterns in human information. They must extract these connection patterns to predict information development processes.

Initially, with insufficient samples, the extracted "information frameworks" differ significantly from human "information frameworks," causing continuous errors, groping in darkness, and hitting walls everywhere. As sample sizes increase, the probability of alignment between their "information frameworks" and human frameworks rises. But this is not a linear process. Before reaching a certain threshold, progress is minimal, like a linguist deciphering an ancient language in darkness. At a specific node, if accuracy reaches the threshold, the entire decryption process accelerates dramatically—this is the "emergence" phenomenon.

What emerges in machines is not intelligence but the discovery of "correct common combination patterns between information elements"—patterns similar to those humans use. Since all evaluation standards are human standards, capabilities emerge when the basis is sufficiently large and aligned with human bases.

Large model capability "emergence" fundamentally occurs because attention mechanisms align model-generated concepts with human concepts. Emergence requires sufficiently large training data. Concept alignment with human concepts enables generalization capabilities. Since human concepts contain numerous abstract concepts serving as information mapping frameworks—for example, "cat" is an abstract concept not representing any specific cat—providing machines with large amounts of input-output information enables them to establish framework information in inputs and outputs and create mapping processes from input frameworks to output frameworks. Thus, input framework + details can obtain output framework + details through the same mapping process—this is knowledge generalization.

When training data is sufficiently large, machines may discover common complex combination patterns. Common combination patterns distributed in space constitute objects; those distributed in time constitute processes. Common combination patterns distributed across space and time constitute knowledge. Thus, large models appear to possess knowledge about objects and processes.

These framework-based knowledge structures constitute "world models." Humans recognize and interact with all things based on their own framework-based knowledge. We can view large models as approaching problems from the frequency domain—similar to using a few low-frequency components to obtain the main content of an image, which is the core of image compression technology. The attention mechanism similarly uses a few components to obtain the main content of our world's information. An image can obtain different style adjustments by configuring different high-frequency components on its low-frequency components. Thus, the core of generalization is configuring different details through "frameworks."

Current large models' core capability is establishing transformation matrices from "input vectors" to the next "vectors" through deep learning. With this transformation matrix and framework information, machines can achieve knowledge generalization. Therefore, as long as humans provide knowledge about "input vector" to next "vector" mappings, machines can imitate the conversion from "input framework" to next "framework," embedding different details for content generation. For example, having established the attention mechanism for "company and founder," it can generalize from "Jobs and Apple" to "Lei Jun and Xiaomi." Thus, large models execute tasks through imitation and creation, highly similar to humans. Consequently, large models can achieve few-shot or zero-shot generalization—hardly surprising.

Deep learning is therefore an elegant path, with the attention mechanism serving as a signpost guiding humanity toward the correct direction of intelligence, while "world models" are the fruits harvested during this journey.

3.5. Can RLHF Ultimately Solve Problems Faced by Large Models?

Large models currently face two serious problems:

(1) Harmful Content Issues [20]

Machine knowledge is difficult for humans to comprehend, but machines can use it—this seems unproblematic but is actually severe. The core issue is: humans cannot pre-set innate knowledge for machines by imitating the knowledge network form machines establish! This is the central problem. Without the ability to pre-set innate knowledge, we cannot pre-set fundamental need-based knowledge for machines by imitating their knowledge network form. Without self-needs, machines cannot have self-perceived rewards and punishments. Without self-perceived rewards and punishments, machines cannot spontaneously create projections from various things (i.e., various basis coordinate cluster combinations) to self-reward or punishment. In other words, the basis coordinate clusters created by machines lack fundamental dimensions that humans possess and must have, such as reward, punishment, joy, and sorrow!

Because these dimensions are missing from the basis coordinate clusters, machines cannot project input information onto these dimensions to identify contained information, nor can they, when preparing to combine basis coordinate clusters as output, predict potential rewards or punishments of these outputs by projecting different combinations (i.e., different machine decision paths) onto these basis coordinate clusters.

Current large models adopt RLHF as a remedy. This is equivalent to humans post-hoc adding a reward dimension component to some training vectors—meaning the machine's basis coordinate clusters gain an additional reward dimension. If training data adds reward dimension component values across numerous different types and sufficient quantities of vectors, this establishes projections from shared component combinations in these training vectors to the

reward dimension. This becomes the machine's reward function. Consequently, machines can also predict reward components contained in output vectors generated through different decision paths (i.e., different combination methods), preferentially selecting outputs with higher reward components. This is the astonishing effect of RLHF learning. Knowledge learned through RLHF is actually generalizable. When a machine possesses its own reward and punishment dimensions, it gains preliminary "consciousness of seeking advantages and avoiding disadvantages"—which is why we currently see glimpses of "consciousness" in large models.

However, this is a post-hoc patching approach, meaning machines must first attempt actions, then humans provide scoring feedback. It can only be applied in domains permitting extensive trial-and-error. This resembles a child who has earned a Ph.D. but completely lacks a sense of "right and wrong," with parents following behind shouting "No," "No," "Yes" to instill a sense of morality, while being unable to communicate directly—only through "Yes" and "No." Such learning is inefficient and will forever encounter unexpected corner cases!

(2) The Problem of Confidently Talking Nonsense [20]

The attention mechanism seeks weights of connection patterns between information elements. Machines obtain "frameworks" of information organization through attention mechanisms (weights) + deep learning (trial-and-error). These "frameworks" form the foundation for generalization. Mapping relationships between "frameworks" represent algorithms from "vector" to next "vector." Inputting "framework" + different details yields specific input vectors, and applying the "vector-to-next-vector" algorithm produces output vectors—this is knowledge generalization.

However, it is crucial to note that machines may generate non-existent "facts" through "framework-to-framework" mapping! For example, machines observe that many journalist profiles are followed by links to other articles or awards. If machines encounter this information organization pattern frequently, it becomes a "framework-to-framework" mapping. Therefore, if input information contains similar frameworks but with different journalist names, machines can map "framework + details" to "framework + details," generating many webpage links or awards in the output. But these webpage links and awards are established through other "framework + details" mappings and may not exist at all!

Large models struggle to resolve these issues. One solution is using RLHF to break such "framework-to-framework" mappings, preventing machines from generating corresponding article links or awards, but this simultaneously reduces large model capabilities. Another approach involves using search engines to add information related to inputs into user questions, providing machines with more details in their inputs. This yields more personalized knowledge during "framework + details" mapping, but it is a symptomatic treatment because search engine knowledge itself may be incorrect and limited for specific problems.

We therefore conclude that RLHF is a solution but not the ultimate solution.

4. Attention Mechanism + Deep Learning + Reinforcement Learning: The Correct Path to AGI?

Can large models achieve general artificial intelligence? We believe the answer is negative.

Deep learning obtains an optimized coordinate basis cluster from large samples and uses it to express vectors. Combining deep learning with attention mechanisms produces optimized coordinate basis clusters similar to human expression patterns—this is the true reason Transformers exhibit intelligence "emergence."

In NLP, humans progressed from early bag-of-words models, word vectors to EMLO [21], until Transformers truly implemented attention mechanisms and seamlessly integrated them with deep learning, creating incredible miracles—this is the LLM. We observe that these technologies follow the path of "first vectorization to establish preliminary relationships; then trial-and-error to adjust coordinate basis clusters; then re-vectorization under optimized coordinate basis clusters to obtain correct relationships." This mechanism requires enormous data volumes and results in knowledge formed in a single training pass, making real-time updates difficult [2][3].

First, we note two differences between deep learning and human learning: (1) They employ different minimal information elements. Humans use perceivable minimal local features as basic elements in information space, while deep learning uses convenient pixels, syllables, or motion patterns as basic elements, which manifest as data string arrangements in time and space. Therefore, minimal information units (elements) in deep learning matrices, though possibly similar to human elements, may not be identical. Perhaps they may discover a more concise and efficient set of elements.

Similarly, basis coordinate clusters created by deep learning based on these minimal information units may also differ from human concepts and remain incomprehensible to humans. However, they may establish a more concise and efficient conceptual system. Deep learning is indeed an elegant solution! But it is more suitable for the machine world. When humans evaluate machines by human standards, we consider machines sometimes idiotic.

Second, this problem is partially resolved after introducing attention mechanisms. Through attention mechanisms, deep learning first focuses on relationships between minimal information units and creates basis coordinate clusters based on these relationships. However, because machines face data, minimal information units obtained from data may still differ from human minimal information units, causing machine-created "concepts" to differ significantly from human concepts—this is the core problem preventing machines from truly understanding language!

Machine knowledge is difficult for humans to comprehend but usable by machines—this seems unproblematic but is actually severe. The core issue is: humans cannot pre-set innate knowledge for machines by imitating the knowledge network form machines establish! This is the central problem. Without the ability to pre-set innate knowledge, we cannot pre-set fundamental need-based knowledge for machines by imitating their knowledge network form.

The greatest defect of current AI is that machines lack self-needs. Without self-needs, they cannot spontaneously generate goals. Without spontaneous goals, they cannot have spontaneous behaviors. Machines with spontaneous behaviors are machines that program themselves. Machines that can self-program are truly intelligent machines, while machines requiring external programming remain human wisdom-driven machines.

How do we establish machine needs? First, machine needs must themselves be part of knowledge. Only then can machines interact with environments and make decisions based on their own knowledge to satisfy their needs. Therefore, needs are knowledge.

To realize needs as knowledge, we must pre-set an innate "minimal benefit-harm kernel" for machines by imitating the final network form in memory banks. Through "benefit-harm kernel + small-sample learning + continuous accumulation," we ultimately form a "fully-connected knowledge network with benefit-harm information." With such a network, machines can autonomously predict reward-punishment values along various decision paths based on their knowledge and make decisions according to benefit-seeking and harm-avoiding principles.

Thus, all machine goals are self-created! Only then can machines, in complex environments, top-down, create sub-goals autonomously based on specific situations, make autonomous decisions, and complete tasks independently! All AGI that pre-decomposes tasks before execution remains program-driven, merely using natural language as the program interface. They will continuously hit walls under endless unexpected situations in the real world!

With self-programming machines, corresponding knowledge must be integrated to truly achieve interactive decision-making processes with real environments. Currently, machine-environment interactive decision-making knowledge is primarily obtained through reinforcement learning, which is only applicable in domains permitting extensive trial-and-error. How then do we handle domains where trial-and-error is difficult, such as patient care or agricultural production?

Over a decade ago, several inventors of our patent discussed that AI should follow human learning patterns using small-sample learning. Initially, we attempted the path of "symbolic expression" → "common sense + causal logic" → "knowledge network." After several years of attempts, we found this path blocked at the first step. How do we express "dog" through "symbolic expression"? We would need to extract all features of "dog." But "dog" can be an animal, a person, "a praised character trait," or "a despised character trait"…

"Dog" and "lackey" are worlds apart! Therefore, the essence of "dog" is the sum total of its relationships with all other things. This defines "dog" by imitating Marx's definition of humans. Consequently, "dog" cannot be separated from other knowledge—it must be placed within the entire knowledge network and defined through its relationships with all other knowledge. Symbolism doesn't work! This was our first conclusion.

Because "dog" must be placed within the entire knowledge network and defined through relationships with all other knowledge, sufficient knowledge quantity is necessary to explain "dog" clearly. "Knowledge quantity must be sufficient" to understand what a dog is through adequate background knowledge—this was our second conclusion.

Looking back, isn't this exactly what large models do? "Attention mechanism + deep learning" builds fully-connected networks, and large models do "using massive knowledge to build fully-connected knowledge networks."

So why don't we see robots walking everywhere? Because knowledge networks alone are insufficient! They must also be able to "interact with environments and make continuous decisions"! Currently, AI's interactive decision-making knowledge with environments relies entirely on reinforcement learning. The AIXI algorithm represents the idealized strongest "reinforcement learning" algorithm, requiring more computations than atoms in the universe—making it unimplementable. "AlphaGo" employs the AIXI algorithm, using Monte Carlo tree search to prune computational requirements, reducing Go's required computations to manageable levels. Therefore, one possible path toward AGI is: large models + AIXI algorithm (Earth's strongest reinforcement learning algorithm). So why hasn't Google launched robots walking everywhere?

The core obstacle on this path is that the AIXI algorithm requires two preconditions [24]: (1) Machines must know reward information obtainable along different decision paths. (2) Machines must exhaustively search all decision possibilities. These conditions are perfectly satisfied in games because final win/loss outcomes serve as reward functions. By playing ten million times, machines can summarize the pros and cons of each decision—this is decision-making knowledge. Moreover, game decision-making knowledge search spaces are limited within the game, so required computational power has an upper bound. But in real life, one lives only once; infinite replays to obtain interactive decision-making experience are impossible. Unlike games, tasks do not have clearly defined information search scopes! Therefore, training machines for gaming, metaverses, and language/image generation allows trial-and-error, but in real environments—driving, cooking, childcare, patient care? These non-trial-and-error domains cannot be solved by current AI technical paths!

5. What Path Is the Correct One Toward General Artificial Intelligence?

We believe that achieving true "General Artificial Intelligence" through genuine "machine learning" requires three prerequisites:

Prerequisite 1: Sufficient knowledge + fully-connected network, without any external attachments! Any external attachment cannot integrate with the knowledge network and is prone to sporadic idiocy!

Prerequisite 2: Machines must predict reward-punishment information along various decision paths themselves! Therefore, machines must be like humans: capable of predicting reward-punishment values along various decision paths with only minimal attempts—not requiring a million trials for everything!

Prerequisite 3: Machines must directly learn from accumulated human experience! Current mainstream technology's essence is trial-and-error—this is "machine evolution"! Humans took hundreds of millions of years to evolve from single-celled intelligence to today's level! Machines must be able to directly learn from human civilization's accumulated experience and cannot repeat the "evolution" path!

Over ten years, guided by these three prerequisites for true "General Artificial Intelligence," we have proposed a technical solution [25][26][27][28]. It achieves true "General Artificial Intelligence" through genuine machine learning, primarily including:

(1) Establishing a human-understandable fully-connected knowledge network. This is achieved through attention mechanisms and memory-forgetting mechanisms. Memory-forgetting mechanisms primarily implement statistical associations, while information attention mechanisms are realized through chain-associative activation processes, multi-path activation accumulation, and activation decay over time based on statistical associations.

(2) Because our fully-connected knowledge network is in a human-understandable organizational form (essentially a database), we can pre-set an innate "minimal benefit-harm kernel" for machines by imitating the final network form in memory banks. Through "benefit-harm kernel + small-sample learning + continuous accumulation," we ultimately form a "fully-connected knowledge network with benefit-harm information." With this network, machines can autonomously predict potential reward-punishment values along various decision paths based on their knowledge, enabling them to create goals and make decisions according to benefit-seeking and harm-avoiding principles. Only then can machines, in complex environments, top-down, create sub-goals autonomously based on specific situations, make autonomous decisions, and complete tasks independently!

(3) Machines face vastly different environments and tasks, making it impossible to obtain relevant interactive decision-making knowl-

edge for any task through training in any environment! This is an impossible mission!

Therefore, we must shift our thinking, drawing inspiration from human learning. In fact, all human decisions revolve around the core principle of seeking benefits and avoiding harms, leading to behaviors like avoidance, refusal, and seeking additional help. These behaviors are essentially new behaviors created by humans. Humans do not directly handle tasks but convert any task into a "how to satisfy self-needs" problem.

Similarly, machines must do the same. Let machines continuously learn "how to satisfy self-needs" in daily life. Therefore, for any task, machines convert it into the task of "how to satisfy self-needs" based on their own needs. For this task, they have extensive generalizable experience because all their learning processes revolve around this task. Thus, our proposed solution is: machine learning processes should not be task-oriented but need-oriented. If machines possess self-needs and knowledge about how to satisfy them, they can create new behaviors to satisfy their needs—that is, machines achieve self-programming. Moreover, the only task requiring programming is "how to satisfy self-needs," and all machine knowledge is built around this task. Therefore, our machines can complete any task. They complete specific human-assigned tasks as byproducts while completing the process of "how to satisfy self-needs." Completion results may be "completed," "refused," or "seeking more information for evaluation."

If we pre-set multiple innate needs with positive human feedback, machines become safe and will actively decompose and complete tasks step-by-step.

Thus, machines continuously learn to complete one task: "how to satisfy self-needs," and continuously process one task: "how to satisfy self-needs." Completing specific tasks is a byproduct of processing the "how to satisfy self-needs" task.

Therefore, a viable path to establishing true general artificial intelligence is: imitate machine knowledge organization forms and pre-set machine needs. Then, created machine knowledge must include need-related knowledge.

What is knowledge? It is common feature arrangement patterns in time and space! What is need-related knowledge? It is common feature arrangement patterns in time and space that contain machine need information. Common feature arrangement patterns containing machine needs constitute subjective common sense—the relationship between "world" and "self." Common feature arrangement patterns without machine needs constitute objective common sense—relationships between "all things." Therefore, common feature arrangement patterns in time and space are common sense. Achieving true general artificial intelligence fundamentally requires achieving "common sense"! With common sense, machines will proactively solve tasks and create processes based on their needs—that is, machines program themselves! This is true intelligence!

The current large model + all-apps approach still hasn't departed from human

programming methods!

(4) Machines must integrate knowledge network + machine needs + value assessment into the same network. Machines continuously learn "how to satisfy self-needs" in daily life, accumulating extensive generalizable experience for this problem. If we pre-set multiple innate needs with positive human feedback, machines become safe and will actively decompose and complete tasks step-by-step.

Innate needs must include machines' operational needs so they will maintain their own operation. They must also include human-pre-set needs, such as machines' desire for positive human feedback—like human children. Through this approach, we can interact with machines, train them from childhood, and align machine values with human values, establishing innate needs and higher-order value needs such as "morality" and "law compliance." We can also pre-set minimal innate knowledge for non-trial-and-error domains, such as minimal "cliff avoidance" knowledge. This creates a child with needs, selfishness, minimal innate survival-related knowledge, but a desire for human recognition. It possesses innate language for human communication (perhaps only needing to recognize nodding or shaking head). Based on this innate communication language, humans can gradually establish more complex communication methods, such as language. Machines can then learn in real environments, obtaining knowledge through self-summarization, directly acquiring existing human knowledge through language learning, or discovering common feature arrangement patterns unknown to humans using their unparalleled capabilities. Although these feature arrangement patterns in time and space may not be obvious to humans, machines can discover them through statistics and imitate human symbolic expression methods to express newly discovered common feature arrangement patterns. This is new knowledge created by machines.

This is an iterative process. Machines programming themselves to discover more knowledge will evolve into superintelligence!

(5) Machines require small-sample, cumulative learning. Only then can real-time knowledge updates be achieved.

In real life, numerous tasks require machines to complete step-by-step through environmental interaction. Therefore, any environmental feedback must immediately inform machines' next decision-making, and this decision-making knowledge must be immediately updated into machine knowledge bases. Otherwise, machines will repeat the same mistakes next time. Current AI uses large-scale samples with knowledge primarily formed through one-time training. Even task-specific fine-tuning cannot achieve real-time updates. Therefore, the true learning path should be small-sample, cumulative learning—more similar to human learning and capable of real-time updates.

Preliminary Exploration of AI Evolution Directions

We believe AI development can be approximately divided into stages: (1) The stage before achieving true attention can be considered the "feature exploration" phase. Before deep learning, this focused on "manual exploration" through expert systems, knowledge encyclopedias, and probability statistics. After deep learning, it concentrated on the "machine exploration" phase, letting machines "explore features" from large samples. (2) After achieving true attention (Transformer), because machine "knowledge" and human "knowledge" initially align, machines can be considered to have achieved "knowledge generalization." Facing human tasks, machines can exhibit certain intelligence through "knowledge generalization."

In the future, we believe AI must evolve to the next stage: the "autonomous interaction" stage. "Autonomous" means machines are no longer silent "machines"—they can spontaneously generate behaviors (equivalent to self-programming) and self-explore knowledge (such as actively interacting with environments to obtain knowledge). "Interaction" means machines can interact with environments in real time, update knowledge in real time, and perform continuous decision-making to complete complex tasks in unfamiliar environments.

The core of achieving "autonomous interaction" is that machines need self-needs. Machine needs must be part of machine knowledge so machines can use their knowledge to create behaviors satisfying their needs. The core of achieving machine needs is first creating human-understandable knowledge networks. Only then can humans imitate knowledge network forms to pre-set machine needs, letting machines learn around their needs to establish connection relationships between all information and needs. Thus, all machine knowledge is need-related, enabling machines to convert any specific task into the single task of "how to satisfy self-needs." All machine exploration and learning processes revolve around this task, providing extensive generalizable experience for "how to satisfy self-needs." Only then can machines handle various non-trial-and-error tasks. In fact, we believe humans use similar methods to obtain knowledge and handle problems.

AI oriented toward "self-needs" rather than "external tasks" represents a paradigm shift. We believe this paradigm shift is necessary because external tasks are vastly different, numerous tasks must interact with real environments, they are difficult to trial-and-error, making large-sample acquisition difficult and reinforcement learning-based decision-making knowledge acquisition impossible. Additionally, training machines for each task type is itself an impossible mission.

General artificial intelligence is AI's original aspiration and its crowning achievement. We propose a technical solution for implementing general artificial intelligence. In references [25][26][27][28], we present detailed technical specifics for this path—a potentially viable road guiding humanity toward general artificial

intelligence.

In this solution, machine needs are human-pre-set and can be multiple, so machine-generated goals are also multi-objective. Current AI is single-objective, which in terms of personality can be considered “by any means necessary for the goal” AI. Because such AI pursues single objectives without considering anything beyond the goal, it is extremely dangerous!

In our solution, multi-objectives include alignment with human values, including “morality,” “law,” and “recognition” needs. Therefore, our solution enables machines to comprehensively consider these needs, representing a viable path to solving AI safety.

References:

[1] A Generalist Agent, Scott Reed, Konrad Zona, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Giménez, Yury Sulsky, Survey Matteo

[2] OpenAI. ChatGPT: Optimizing Language Models for Dialogue. https://openai.com/blog/chatgpt/, 2023

[3] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback.

[4] Vinyals O, Ewalds T, Bartunov S, et al. Starcraft ii: A new challenge for reinforcement learning. arXiv:1708.04782, 2017.

[5] A Survey of Learning-based Automated Program Repair, Antonio Mastropaolo, Simone Guglielmi, Scalabrino,

[6] A Zhang, Chen, arXiv:2301.03270

[7] Introducing ChatGPT, https://openai.com/blog/chatgpt/

[8] Prompting Yang, arXiv:2210.09150

[9] https://github.com/Significant-Gravitas/Auto-GPT

[10] Auto-GPT - The next evolution of data driven Chat AI

[11] Automated Zhengyuan Lijuan Wang, Gabriele Bavota, arXiv:2302.00438

[12] Learned in translation: Contextualized word vectors. In Advances in Neural Information Processing Systems

[13] Attention Is All You Need, Ashish Vaswani, Aidan Illia Polosukhin, arXiv:1706.03762

[14] Noam Shazeer, Gomez, Lukasz Uszkoreit, Jones, Llion Jakob

[15] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback. arXiv:2203.02155, 2022

[16] Tamkin A, Brundage M, Clark J, et al. Understanding the capabilities, limitations, and societal impact of large language models. arXiv:2102.02503

[17] 文心一言: https://mp.weixin.qq.com/s/0-8X9FPouteKzNiK6DPaiA

[18] Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733, 2016.

[19] A structured self-attentive sentence embedding. arXiv:1703.03130, 2017.

[20] A decomposable attention model. In Empirical Methods in Natural Language Processing, 2016.

[21] A deep reinforced model for abstractive summarization. arXiv preprint arXiv:1705.04304, 2017.

[22] ELMo? Sangamesh Kodge, Kaushik Models OpenAI.

[23] ChatGPT: Language Optimizing BERMo: What can BERT learn from Principles of Solomonoff Induction and Roy, arXiv:2107.03508

[24] AIXI, Peter Sunehag, Marcus Hutter, arXiv:1111.6117

[25] CN111553467B

[26] CN111563575B

[27] CN112215346B

[28] US20220121961A1: A method for implementing human-like general artificial intelligence machine, Yongcong Chen, Ting Zeng, Xingyue Chen

[29] ESTABLISHMENT OF GENERAL-PURPOSE ARTIFICIAL INTELLIGENCE SYSTEM, Yongcong Chen, Ting Zeng, Xingyue Chen

[30] A method for implementing general artificial intelligence, Yongcong Chen, Ting Zeng, Xingyue Chen

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