Research Article

# Towards a Comprehensive Theory of Aligned Emergence in AI Systems: Navigating Complexity towards Coherence

**Lucas Freund**[1]

1. Fresenius University of Applied Sciences, Idstein, Germany

**Emergent behavior and alignment in AI systems have become critical areas of research as we strive to understand and harness the capabilities of artificial intelligence. This paper explores the complex dynamics of emergent behavior and alignment within AI systems and presents a comprehensive framework for conceptualizing and modeling these phenomena. The framework incorporates the multilevel and time-dependent nature of emergent behavior and alignment, considering the interplay between system states, inputs, function rules, learning algorithms, environments, and historical data. By incorporating alignment as a dynamic and continuous process, we shift the focus from static, axiomatic alignment to ongoing adaptation and control. The proposed framework sheds light on the challenges and opportunities associated with achieving and maintaining alignment in AI systems. We discuss insights derived from the framework, highlighting the necessity for robust and adaptable alignment mechanisms that can respond to the evolving behaviors and changing conditions of AI systems. We also emphasize the importance of careful system initialization and the role of initial conditions in shaping the trajectory of AI systems. Furthermore, we outline the potential for feedback loops and the need for empirical validation to better understand and harness the interactions between emergent behavior, alignment, and system parameters. The paper concludes with an outlook on future research directions, emphasizing the need for theoretical advancements, computational methods, and empirical studies to advance our understanding and practical implementation of alignment in AI systems.**

**Corresponding author:** Lucas Freund, lucas.freund@hs-fresenius.de

# 1. Introduction

Artificial Intelligence (AI) has become a pervasive element in modern society, permeating various domains of human activity. As AI systems continue to evolve, becoming more autonomous and complex, understanding their behaviors poses new challenges. This paper proposes the integration of complexity theory and emergence theory as theoretical frameworks to shed light on the intricate behaviors exhibited by AI systems. Complexity theory offers a comprehensive framework for comprehending complex systems, while emergence theory explains how simple rules can give rise to complex phenomena. Despite their significant contributions in diverse scientific fields, their application to AI remains relatively unexplored. This study aims to address this gap by applying these theories to unravel the complex behaviors and alignment mechanisms within AI systems, with the goal of enhancing system design and regulation.

Complexity theory provides a multidisciplinary perspective for analyzing complex systems characterized by intricate hierarchies and network patterns that arise from simple rules. It emphasizes that understanding the individual components of a complex system is insufficient for predicting its overall behavior, as the system exhibits distinct properties that emerge from the interactions among its components. This holistic view underscores the interconnectedness and dynamics of the system as a whole. (Larsen-Freeman, 2013)

Emergence theory complements complexity theory by explaining how elementary rules and interactions can give rise to complex, higher-order phenomena, often unforeseen based on the initial conditions or rules of the system. It provides insights into phenomena that emerge as a result of interactions among simpler entities. This concept is fundamental to understanding many natural and artificial complex systems. (Clayton, 2006)

While complexity theory and emergence theory have made significant contributions across various scientific disciplines, their potential to illuminate the inner workings of AI systems remains largely untapped. This is particularly problematic given that advanced AI systems, such as Artificial General Intelligence (AGI) and general-purpose AI, can be considered complex systems exhibiting emergent behaviors. AGI refers to systems capable of understanding, learning, and applying knowledge across a broad range of tasks at or beyond human capacity, embodying the intricacy and emergence inherent in advanced AI. General-purpose AI, designed to perform any intellectual task a human can accomplish,

provides fertile ground for studying emergence in AI systems due to its vast capabilities and adaptability. (Bubeck et.al, 2023)

This study aims to explore the dynamics of emergent behavior and alignment in AI systems by integrating complexity theory and emergence theory. It recognizes the concept of multilevel emergence, acknowledging that AI systems operate at different levels of abstraction, where behaviors at each level not only emerge from the level below but also result from the interactions among emergent properties themselves. Additionally, the study emphasizes the dynamic nature of alignment within AI systems. Rather than perceiving alignment as a static, one-time achievement, it advocates for understanding alignment as an ongoing and evolving construct. The alignment process is influenced by various factors, including the AI system's evolving behaviors, changing environmental conditions, and the interplay of factors across different levels of abstraction.

By investigating the dynamics of emergent behavior and alignment in AI systems, this study aims to provide valuable insights for the design, management, and ethical governance of AI systems. It contributes to the academic discourse on AI behaviors by offering a theoretical framework that enhances our understanding of the complex dynamics underlying AI system behaviors and alignment.

The subsequent chapters of this paper delve into the foundational principles of complexity theory and emergence theory, establishing a solid theoretical basis for the investigation. It then presents a multilevel framework for understanding emergent behavior and alignment in AI systems, addressing the challenges and implications of this framework, and proposing potential solutions for robust alignment mechanisms. Finally, the study provides case studies and empirical analyses to demonstrate the practical applicability of the framework.

In conclusion, comprehending the dynamics of emergent behavior and alignment in AI systems is crucial for leveraging the full potential of AI while ensuring responsible and beneficial deployment. By integrating complexity theory and emergence theory, this study aims to advance our understanding of AI system behaviors and contribute to the development of effective design, management, and regulation strategies.

## 2. Artificial Intelligence

Artificial Intelligence (AI), at its core, is an expansive field dedicated to creating machines that can mimic human intelligence. However, the complexity of human intelligence and the vast array of ways

it can be emulated mean that AI encapsulates a variety of approaches, models, and systems, ranging from rule-based systems to complex neural networks.

There are two broad types of AI: Narrow AI, which is designed to perform a specific task, such as voice recognition, and Artificial General Intelligence (AGI), which is AI that possesses the ability to understand, learn, and apply knowledge across a wide array of tasks, matching or even surpassing human intelligence. (Mikalef et.al, 2022, Bubeck et.al, 2023)

### *2.2. Machine Learning*

Machine Learning (ML) forms a critical subset of Artificial Intelligence and is often perceived as the driving force behind many of the advancements in the field. Fundamentally, ML is premised on the idea that systems can learn from data, identify patterns, and make decisions with minimal human intervention. It builds on the concept that algorithms can be designed to optimize a performance criterion using example data or past experiences. The process is automated and improves over time, becoming more accurate and insightful as it accumulates more data. ML is typically classified into three main categories, each catering to different problem types and data conditions: Supervised Learning, Unsupervised Learning, and Reinforcement Learning. (Zhang et.al, 2023, Jiang et.al, 2022, Badillo et.al, 2020)

Supervised Learning is the most prevalent form of ML. In supervised learning, the 'teacher' explicitly defines what constitutes correct behavior and incorrect behavior. The learning algorithm receives a labeled dataset, where each instance comprises an input vector and an associated output value (label). The aim is to discover a general rule that maps inputs to outputs. This type of learning is particularly effective for tasks such as image classification, sentiment analysis, and predictive modeling, where every instance in the dataset has a clear, known outcome. (Badillo et. al, 2020, Ray, 2019)

In an email spam classification problem, we would have a dataset of emails, each labeled as "spam" or "not spam". The algorithm would learn from this data, identifying key words, phrases, or patterns that are often associated with spam emails. Once trained, the model can then be given a new email and predict whether it is spam or not, based on what it has learned.

Unsupervised Learning, on the other hand, involves learning patterns in input data without the need for labeled responses. The learning algorithm is presented with an unlabeled dataset and tasked with finding structure in the data, such as grouping or clustering of data points. It can uncover hidden

patterns and structures that are not immediately apparent, making it valuable for tasks like anomaly detection, dimensionality reduction, and topic modeling. (Badillo et. al, 2020, Ray, 2019)

Consider a retail company that has a dataset of customer purchasing behavior, but doesn't have any labels to differentiate the customers. An unsupervised learning algorithm, such as a clustering algorithm, can be used to identify patterns in the data and group customers into different segments based on their purchasing behavior. These segments can then be used for targeted marketing, product recommendations, and other business strategies.

Reinforcement Learning (RL) is a different paradigm where an 'agent' learns by interacting with its environment, receiving feedback in the form of rewards or penalties. Unlike supervised learning, which requires explicit labels, or unsupervised learning, which finds structure in data, RL is about learning what actions to take to maximize some notion of cumulative reward. The agent learns a policy, which is a mapping of states to actions that maximize the long-term reward. This methodology is primarily used in areas such as robotics, navigation, game playing, and real-time decision making. (Badillo et. al, 2020, Ray, 2019)

A classic example of reinforcement learning is teaching an AI to play a game, such as chess. The AI starts with no knowledge of the game. It learns by playing the game repeatedly. Each move it makes is an action, and each action leads to a new state of the game. If a sequence of actions leads to a win, the AI receives a reward. If it loses, it receives a penalty. Over time, by trying to maximize its reward and minimize its penalty, the AI learns a policy of what move to make in each state of the game, effectively learning how to play the game. (Risi & Preuss, 2020)

Each of these learning paradigms has its strengths and is appropriate for different kinds of problems. Understanding the nature of the problem and the data at hand is crucial in selecting the right learning approach.

### *2.3. Deep Learning*

Deep Learning is a subset of machine learning that has seen a rapid rise in popularity due to its impressive ability to extract abstract representations from raw, high-dimensional data. It is based on artificial neural networks (ANNs), with 'deep' referring to the use of many layers in the network. A Deep Neural Network (DNN) is essentially an ANN with multiple hidden layers between the input and output layers. These hidden layers are made up of interconnected nodes, or artificial neurons, which are computational units that mimic the functionality of biological neurons. The connections between

these artificial neurons can be adjusted based on the learning from the data, this process is termed as 'training the network'. Every layer's output is the input to the next layer, thereby creating a hierarchical structure of data processing and transformation. (Ray, 2019) DNNs can handle non-linear and complex patterns in large datasets, making them well-suited for high-dimensional data like images, text, and audio. They enable the system to automatically learn features at multiple levels of abstraction, allowing the system to learn complex functions mapping the input to the output directly from data, without the need for manual feature extraction. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are two popular types of DNNs. CNNs are predominantly used in image processing tasks, leveraging their spatial invariance to recognize patterns anywhere in the image. (Ray, 2019) They excel in tasks like object recognition and facial recognition. RNNs, on the other hand, have feedback connections that make them powerful for tasks involving sequential data, such as time series analysis, natural language processing, and speech recognition. (Goodfellow et.al, 2016) Deep Learning is behind many high-impact and cutting-edge AI applications. This includes voice assistants like Siri and Alexa, which use deep learning for understanding spoken language. Image recognition systems, such as those used in autonomous vehicles, use deep learning to recognize traffic signs, other vehicles, and pedestrians. Recommendation systems in e-commerce platforms like Amazon and Netflix also use deep learning to understand user preferences and recommend products or movies. Despite its prowess, deep learning models, especially the larger ones, require significant amounts of data and computational resources. (Ray, 2019) Furthermore, the inner workings of AI models can be somewhat opaque, earning them the moniker 'black boxes'. (Hendrycks & Mazeika, 2022)

### 1. 2.4. Artificial Neural Networks

Artificial Neural Networks (ANNs) are the bedrock of deep learning. ANNs attempt to emulate the processing capabilities of the biological brain by creating networks of artificial neurons, or nodes, that can transmit and process information. An ANN comprises multiple layers: the input layer, one or more hidden layers, and the output layer. Each layer consists of multiple nodes, and each node in a layer is connected to every node in the next layer. These nodes, often referred to as neurons, serve as the computational units of the network. (Aggarwal & Aggarwal, 2018, Ray, 2019)

- Input Layer: This is the entry point of the neural network where data is fed into the system. Each node in the input layer corresponds to one feature of the dataset, so the number of nodes in this

layer equals the number of input features.

- Hidden Layers: Hidden layers perform the bulk of the computations in the network. These layers are termed 'hidden' because their values are not observable in the training data - they're learned from the data during the training process. Each node in a hidden layer transforms the values from the previous layer with a weighted linear transformation, followed by a non-linear activation function. The purpose of the activation function is to introduce non-linearity into the network, allowing it to learn more complex patterns.
- Output Layer: This is the final layer in the neural network, producing the results of the computations performed by the network. The number of nodes in the output layer typically corresponds to the number of classes or labels for classification problems, or a single node for regression problems.

Each connection between nodes has an associated weight, which determines the contribution of the input value to the output value of the neuron. The weights are the learnable parameters of the network, adjusted during the training process to minimize the difference between the network's predictions and the actual values. Bias nodes are also a critical component of ANNs. They're akin to the intercept added in a linear equation, allowing the activation function to be shifted to better fit the data. Artificial Neural Networks (ANNs) can be mathematically represented using linear algebra, calculus, and functions. (Aggarwal & Aggarwal, 2018)

Here's a simplified representation.

Input Layer: The input layer can be represented as a vector X = [x1, x2,..., xn], where n is the number of input features.

Hidden Layers: The output of the first hidden layer H1 can be calculated as:

$H1 = f(W1 * X + b1)$

Here, f is the activation function, * denotes the dot product, and + is vector addition. This process is repeated for each hidden layer.

Output Layer: The output Y of the network is:

$Y = g(Wo * Hf + bo)$

Here, g is the activation function for the output layer.

The weights and biases (W and b values) are learned during the training process by minimizing a loss function L(Y, Y_true), where Y is the network's prediction and Y_true are the actual values. This is typically done using an optimization algorithm like gradient descent. The specific form of the loss function and the activation functions depend on the problem and the specific type of neural network being used. (Aggarwal & Aggarwal, 2018) ANNs, particularly when stacked into deep networks with many layers, are remarkably good at learning complex, abstract representations of input data. They are used in numerous AI applications, including image and speech recognition, natural language processing, and more. However, their complexity and opacity have spurred research into understanding them from a complexity theory and emergence theory perspective. This provides insights into their structure, the nature of the transformations they learn, and the emergent properties of these systems as a whole.

### *2.5. GPT-4 and Transformer Architectures*

The GPT-4, a milestone in AI and the successor to the celebrated GPT-3, is a remarkable example of a deep learning model capable of generating astoundingly human-like text. Developed by OpenAI, GPT-4 utilizes a transformer architecture, a cutting-edge design for neural networks that has been instrumental in revolutionizing natural language processing (NLP). The Transformer architecture stands out due to its self-attention mechanism, a unique feature that allows the model to weigh and prioritize different parts of the input based on their relevance to the task at hand. This mechanism is employed across multiple layers in the network, enabling it to capture intricate patterns and dependencies in the data, even over long sequences. (Bubeck et.al, 2023) This capability is essential for sophisticated language understanding and generation, as it allows the model to maintain context over long passages and produce coherent and contextually accurate responses. GPT-4's remarkable performance in generating contextually rich, diverse, and nuanced text underscores the efficacy of transformer architectures in tackling complex AGI tasks. It provides compelling evidence that deep learning, equipped with the right mechanisms and trained on vast amounts of data, can lead to systems that exhibit a broad competence over a wide array of tasks, a defining characteristic of AGI. However, this competence brings with it a multitude of complexities and challenges. The behavior of models like GPT-4 can be unpredictable and non-intuitive, a result of the interaction of millions of learned parameters. These models can exhibit emergent behavior, where the whole is more than the sum of its parts, making it challenging to understand and control their operations. The propensity for

emergent behavior and the complexity inherent in these models also raise significant concerns about their robustness, reliability, and safety. (Bubeck et.al, 2023)

### *2.5.1. Artificial General Intelligence*

Artificial General Intelligence (AGI) refers to a type of artificial intelligence that has the ability to understand, learn, and apply its intelligence to a wide range of tasks, much like a human being. This is in contrast to Narrow Artificial Intelligence (NAI), which is designed and trained for a specific task, such as voice recognition or image classification. AGI is considered the holy grail of AI research. It encapsulates the idea of a machine that could perform any intellectual task that a human being can do. It implies a machine capable of understanding, learning, and adapting to new situations, solving unfamiliar problems, and integrating different types of knowledge. (Bubeck et.al, 2023)

To achieve AGI, a system would need to combine a variety of AI techniques, including machine learning, deep learning, reinforcement learning, and more. It would also likely need to leverage new, as-yet-undiscovered techniques to reach human levels of cognition and adaptability.

## 3. Emergence Theory

Emergence is a concept derived from systems theory where complex patterns, behaviors, or properties arise out of the interactions between simpler components of a system. Importantly, these emergent properties are not predictable solely based on the characteristics of the individual parts, and they often cannot be reduced back to those parts. Emergence is observed in a variety of systems and fields, from physics and biology to sociology and artificial intelligence. (Gershenson & Fernández, 2012)

In a narrower sense, emergence is often distinguished as either weak or strong. Weak emergence describes systems where emergent properties can be derived from the properties and relations of the components, albeit it may be computationally infeasible. In contrast, strong emergence refers to systems where emergent properties cannot be explained by the properties and relations of the components, even in principle. (Clayton, 2006)

In the context of AI and especially neural networks, emergence can refer to behaviors or capabilities of the system that arise from the interaction and organization of individual artificial neurons. Such behaviors or capabilities can be complex tasks like recognizing objects in an image or understanding natural language, which are not predictable based on the individual neurons alone. (Bubeck et.al, 2023)

### *4.1. Characteristics of Emergent Phenomena*

One of the defining hallmarks of emergent phenomena in AI is the ability to generate behaviors that are more than the sum of their parts. AI models, consisting of interconnected components such as neural networks, exhibit emergent behavior that extends beyond the capabilities of their individual neurons or algorithms. This emergent behavior encompasses the model's ability to learn, adapt, generalize, and exhibit intelligent decision-making in complex environments. (Gershenson & Fernández, 2012)

AI systems also demonstrate self-adaptation, the capacity to modify their structure and behavior in response to changing conditions or experiences. Through mechanisms like machine learning and reinforcement learning, AI models continuously refine their internal parameters, decision-making policies, or predictive capabilities. This self-adaptation empowers AI systems to improve their performance, learn from data, and autonomously adapt to new challenges or tasks. (Bubeck et.al, 2023) Moreover, emergent phenomena in AI encompass the emergence of complex patterns and strategies that are not explicitly programmed or instructed. AI models can uncover intricate patterns and relationships within data, even beyond what human experts might anticipate. This capacity for pattern recognition, abstraction, and generalization enables AI systems to discover novel insights, solve complex problems, and provide innovative solutions. (Bubeck et.al, 2023)

Furthermore, the emergence of unpredictability and creativity is a striking characteristic of AI systems. As AI models interact with their environment, learn from data, and make decisions, they may exhibit behavior that is not entirely predictable or replicable. This unpredictability arises from the intricate interplay of complex components, the nonlinearity of learning algorithms, and the sensitivity to initial conditions.of component-level behavior. (Derner & Batistič,2023)

#### *4.1.1. More than the Sum of Its Parts*

Emergence theory posits that complex systems are more than a mere collection of their constituent elements. The underlying principle is that the holistic behaviors or characteristics of a system cannot be wholly deduced or anticipated solely from understanding the properties of its individual components. The relationships between these components, the dynamics of their interactions, and the subsequent collective behaviors birth emergent phenomena that transcend the attributes of the components themselves. (Clayton, 2006, Gershenson & Fernández, 2012) In the context of artificial intelligence, especially with regards to complex models like neural networks, this principle holds

significant relevance. Emergent behavior in such networks can be seen when they perform tasks such as recognizing intricate patterns in data, generating human-like text, or even creating art. (Bubeck et.al, 2023)

This behavior is a result of the composite functionality of the network, and not attributable to any single neuron or layer within the network. Even if we completely understand the functionality of an individual neuron — how it accepts inputs, processes them through an activation function, and passes the output forward — this knowledge would not enable us to predict the ultimate behavior of the whole network. The collective behavior arises from a convolution of network-wide weights, biases, activations, and their intricate interplay across layers during both the feedforward and backpropagation stages of the neural network's operation.

For instance, consider an image recognition task using a deep convolutional neural network. The initial layers might recognize simple features like edges or color gradients. However, the recognition of more complex features such as shapes, textures, and eventually entire objects like 'cat' or 'dog', emerges as a consequence of the combined operations across layers, each contributing progressively to the higher-order understanding of the image.

This behavior exemplifies the 'more than the sum of its parts' concept. Despite having complete knowledge of each neuron's operation, the emergent recognition of complex objects in images is not straightforwardly predictable.

### *4.1.2. Unpredictability and Novelty*

One of the defining characteristics of emergent phenomena is their inherent unpredictability and the representation of novel behaviors or patterns not readily deducible from the properties or behaviors of a system's individual components. In essence, the concept of emergence is strongly tied to the idea of surprise; the outcomes are often more than just the sum of the inputs, leading to novel and unforeseen system-level behaviors. (Clayton, 2006, Gershenson & Fernández, 2012)

In the realm of artificial intelligence, the unpredictability of emergent phenomena has fascinating and at times, confounding manifestations. The dynamics of these phenomena can lead to a range of outcomes, from innovative solutions to unforeseen complications. For instance, a trained neural network might suddenly display an improvement in its performance. Such an enhancement, although desirable, can emerge without clear attributable changes to the model or its training process. Conversely, a degradation in performance may also occur, where the model's accuracy or reliability

diminishes without evident reasons. The novelty factor of emergent phenomena in AI also comes to the fore when models generate unexpected outputs. These might include a language model producing profound philosophical insights, a generative model creating original pieces of art, or a game-playing AI discovering unique strategies that surpass human gameplay. These outputs can be so innovative and distinctive that they seem almost 'creative', a trait traditionally attributed solely to human intelligence.

Moreover, AI systems may find novel solutions to problems that were not explicitly programmed into them. (Bubeck et.al, 2023)

Consider the case of reinforcement learning, where AI agents learn to perform tasks by maximizing a reward signal. There have been numerous instances where these AI agents have discovered solutions that were unanticipated by their human creators. For example, in playing a game, an AI might discover a completely new strategy that humans had not thought of, or in a physics simulation, an AI might find an innovative way to achieve its goal.

However, the unpredictability of emergent phenomena also presents challenges. As AI systems become more complex and their behaviors more emergent, predicting their responses to specific inputs or situations becomes increasingly difficult. This can lead to unanticipated and potentially harmful behaviors, particularly in high-stakes applications like autonomous vehicles or healthcare.

### *4.1.3. Emergence at Different Levels*

The phenomenon of emergence isn't confined to a singular scale or level within a system but can materialize at varying degrees of complexity, leading to a hierarchical structure of emergent behaviors. This hierarchical emergence is a manifestation of the systemic layers of complexity and interaction patterns that collectively give birth to multifaceted emergent phenomena. (Gershenson & Fernández, 2012)

In the context of a neural network, particularly within the domain of deep learning, this hierarchy of emergence is explicitly visible and fundamental to the operational mechanics of these models.

Consider the example of a convolutional neural network (CNN), a class of deep learning models primarily employed in image recognition tasks. Within a CNN, the process of emergence occurs at different levels of the network's architecture. At the lower levels, typically in the early layers of the network, emergence is manifested as the detection of rudimentary patterns or features within an image. These could include the identification of lines, edges, corners, or color gradients. Each neuron

in these layers specializes in recognizing a specific feature in its receptive field. The detection of these simple features constitutes a form of lower-level emergent behavior, which arises from the interplay between the input data (the image) and the network's learned weights and biases.

As we ascend the network's hierarchy, these lower-level emergent features become the input to the subsequent layers. The neurons in these higher layers then interact with the output of the preceding layer to recognize more complex and abstract features. For instance, a middle layer might integrate the edges and corners recognized by the initial layers to identify more complex shapes or textures. In the final layers of the network, higher-level emergence occurs. Here, the network can identify entire objects, faces, or even scenes in an image, utilizing the complex features recognized by the preceding layers. This recognition of intricate patterns or objects in an image is a form of high-level emergent behavior, which is only made possible by the cumulative effect of lower-level emergent features.

This cascading hierarchy of emergence, from the detection of simple features to the recognition of complex objects, exemplifies how emergence at different scales contributes to the overall functionality of an artificial neural network. It encapsulates the essence of how the whole becomes greater than the sum of its parts, underlining the role of emergence theory in understanding and explaining the behavior of complex AI systems.

### *4.2. Quantifying Emergence*

Measuring emergence — identifying and quantifying the complex behavior that arises from simpler systems — is a challenging yet critical task in emergence theory. Such measures offer a means of assessing the degree of emergence within a system and can provide valuable insights into the system's dynamics and potential behaviors. Several theoretical measures of emergence have been proposed in the literature, each aiming to capture a different aspect or interpretation of emergent behavior. For example, statistical complexity, a measure that encapsulates the amount of structured information in a system, has been used to quantify emergence. Another measure, mutual information, which indicates the amount of information that can be obtained about one random variable by observing another, has also been used in this context. (Gershenson & Fernández, 2012)

Applying these measures to artificial intelligence models, however, adds an extra layer of complexity due to the intricacies and nuances of these systems. One possible approach to quantifying emergence in AI is to measure the degree of performance improvement as the model learns from data. This could be gauged by tracking changes in the model's loss function during the training process or by

monitoring improvements in the model's accuracy, precision, recall, or other relevant performance metrics. In addition, the degree of novelty or unpredictability in a model's outputs could also serve as a proxy for emergence. This might involve assessing the diversity and innovativeness of solutions generated by the model or the range and variance of its responses to different inputs.

However, it's worth noting that these measures are not universally applicable across all AI tasks and models. The appropriate measures can vary based on the specific task at hand, the type of AI model, and the context in which it is applied. For instance, a performance-based measure might be more relevant for a supervised learning model trained on a specific task, while a novelty-based measure might be more suited to a generative model or a reinforcement learning agent.

Quantifying emergence in AI systems remains an active and burgeoning area of research. As AI models continue to grow in complexity and their behaviors become increasingly emergent, the development of robust and meaningful measures of emergence will become increasingly important. The intersection of complexity theory and emergence theory in AI is a fertile ground for deep insights. It is at this juncture that we begin to see the profound implications of these theories on the development and understanding of AI systems.

In AI, particularly in neural networks, the individual artificial neurons are relatively simple computational units. However, when these neurons are connected into a large network, the system exhibits complex behavior. This complexity arises not from the individual neurons but from the non-linear interactions between them. The network's ability to learn from data, recognize patterns, and make decisions is not a property of any single neuron but emerges from the collective behavior of all the neurons in the network. This is a clear manifestation of emergence theory. The emergent properties of AI systems, such as their ability to learn and adapt, are not programmed explicitly into the system. Instead, they arise naturally from the complex interactions between the system's components. This is akin to how the behavior of a flock of birds emerges from the simple rules followed by each bird. The emergent behavior cannot be predicted from the behavior of the individual birds, nor can it be reduced to those individual behaviors. This is a key insight from complexity theory.The connection between complexity and emergence in AI is not just theoretical. It has practical implications for how we design, train, and understand AI systems. For instance, it suggests that we cannot fully understand an AI system by studying its individual components in isolation. Instead, we need to study the system as a whole, taking into account the complex interactions between its components. Moreover, the connection between complexity and emergence in AI also has implications

for the interpretability and predictability of AI systems. The emergent behavior of AI systems can be highly unpredictable and difficult to interpret. This is a direct consequence of the complexity of the interactions between the system's components.

## 5. Emergence and AI Design

Understanding and leveraging emergent phenomena hold transformative potential for the design, functionality, and application of AI systems. This involves fostering beneficial emergent behaviors, mitigating unwanted ones, and improving the transparency and interpretability of AI systems. (Dwivedi et.al, 2023)

Beneficial emergent behaviors often align with the intended functionality of AI systems, contributing to their effectiveness and adaptability. In a neural network, for example, the hierarchical emergence of pattern recognition, from the detection of simple features to the identification of complex objects, is a beneficial emergent phenomenon that underlies the network's ability to perform tasks like image recognition or natural language processing. By designing AI systems in ways that promote such advantageous emergent behaviors, we can enhance their performance and utility in a variety of applications. Conversely, understanding emergent phenomena can also enable us to recognize and mitigate unwanted emergent behaviors. Unwanted emergent behaviors in AI systems can lead to detrimental outcomes, such as biased decision-making, unethical actions, or unpredictable and harmful outputs. For instance, a machine learning model might learn and subsequently replicate biases present in its training data, leading to unfair or discriminatory decisions — an emergent behavior that is neither intended nor desirable. Through the lens of emergence theory, we can anticipate, detect, and curb such behaviors, making AI systems safer, more reliable, and more ethical. (Dwivedi et.al, 2023)

The potentialities ingrained in understanding and maneuvering emergent phenomena offer transformative capabilities for AI, impacting its design, functionality, and application. These benefits traverse three principal domains: enhancing AI systems by nurturing beneficial emergent behaviors, mitigating negative emergent behaviors, and surmounting the interpretability challenge pervasive in complex AI models. (Hendrycks & Mazeika, 2022, Derner & Batistič,2023)

Commencing with the nurturement of beneficial emergent behaviors, these often harmonize with the primary objectives of AI systems, boosting their efficacy and adaptability. A compelling exemplification of this resides in neural networks, where the hierarchical emergence of pattern

recognition unfolds. Here, we witness a journey from the detection of rudimentary features to the recognition of intricate structures. This beneficial emergent phenomenon underpins the network's competency in performing an array of tasks, including image recognition and natural language processing. A strategically designed AI system that amplifies such advantageous emergent behaviors can amplify the performance and versatility of AI systems across numerous applications. (Hendrycks & Mazeika, 2022)

On the flip side, recognizing and mitigating undesired emergent behaviors form another crucial element of leveraging emergent phenomena. Such negative behaviors can catalyze detrimental outcomes, ranging from biased decision-making to unethical operations, or unpredictable and harmful outputs. An illuminating example of this occurs when a machine learning model inadvertently learns and replicates biases embedded within its training data. This action subsequently culminates in unjust or discriminatory decisions, constituting an emergent behavior which is neither designed nor desired. Emergence theory equips us with the capacity to foresee, identify, and suppress such behaviors, thereby promoting the safety, reliability, and ethicality of AI systems. (Hendrycks & Mazeika, 2022)

Beyond the realms of AI design and governance, the study of emergent phenomena can also illuminate the often-quoted 'black-box' conundrum prevalent in numerous AI models. This 'black-box' phenomenon refers to the opacity and lack of interpretability frequently associated with complex AI models, particularly prominent in deep neural networks. Here, deciphering the relationship between the inputs and outputs becomes a Herculean task. By pinpointing and comprehending emergent behaviors, as well as the conditions and dynamics conducive to their inception, we can acquire invaluable insights into the model's operation. This newfound knowledge demystifies the obscure internal machinations of these models, rendering their decisions and actions more interpretable. Such transparency facilitates greater accountability, fosters trust, and enhances societal acceptance of AI systems.

### *5.1. The Black Box Metaphor*

In the context of artificial intelligence (AI), the term "black box" is often used to describe systems where the internal workings are not understood or are not transparent. This metaphor is particularly prevalent in the realm of machine learning and, more specifically, deep learning, where models often

consist of many layers of interconnected nodes (neurons) and parameters (weights and biases). (Mikalef et.al, 2022, Shin et.al, 2022)

A black box AI model takes in inputs and produces outputs without giving an explicit, understandable account of what is happening inside the model. The decisions, predictions, or classifications made by such a model can be difficult to interpret or explain, as the process by which the model arrived at its output is not readily transparent or understandable. This lack of interpretability and transparency can be problematic, especially in scenarios where understanding the reasoning behind a decision is important, such as in healthcare or legal applications. It also poses challenges for debugging and improving models, as well as for building trust in AI systems. Efforts to make these "black boxes" more interpretable are a significant area of ongoing research in AI, often referred to as "Explainable AI" or "Interpretable AI". The black box metaphor in AI is deeply intertwined with the concepts of complexity theory and emergence. This connection stems from the inherent complexity of AI systems, particularly those based on deep learning, and the emergent behaviors they exhibit. (Shin et.al, 2022, Dwivedi et.al, 2023)

The black box nature of AI has far-reaching implications. Primarily, it can impede trust in AI systems. If the stakeholders, including decision-makers in businesses or policy, fail to comprehend how an AI system is making decisions, they may be reluctant to trust and adopt these technologies. This is especially pertinent in sensitive areas such as healthcare, finance, or criminal justice, where decisions can have significant impacts on individuals' lives. Additionally, the black box problem can pose challenges to accountability and fairness. If an AI system makes a decision that leads to harmful outcomes, it can be difficult to hold the system or its creators accountable if the decision-making process is not transparent. Similarly, if an AI system is making decisions that are biased or discriminatory, it can be challenging to detect and rectify these biases if the decision-making process is opaque. (Peters, 2022, Minh et.al, 2022)

In response to these challenges, researchers are devising a variety of techniques to enhance the interpretability and transparency of AI. These techniques broadly fall into two categories: post-hoc interpretability and transparency by design. (Dwivedi et.al, 2023, Minh e.al, 2022)

Post-hoc interpretability techniques strive to explain the decisions of an already trained model. These techniques encompass methods such as feature importance, partial dependence plots, and counterfactual explanations, which aim to provide insights into the model's decision-making process after the fact. Transparency by design, conversely, involves constructing models that are inherently

interpretable. These models often sacrifice some level of predictive accuracy for increased transparency. Examples include decision trees, rule-based systems, and interpretable neural networks. (Peters et.al, 2022, Dwivedi et.al, 2023)

Advanced AI systems, notably Artificial General Intelligence (AGI) and general-purpose AI, continue to push the boundaries of technology, offering unprecedented capabilities and opportunities. However, these systems' intricate complexity makes them a fertile ground for emergent behaviors—phenomena that are not simply the sum of their parts, but rather unexpected outcomes arising from the complex interplay of simpler rules and interactions. This chapter explores the concept of emergent behaviors in AI, drawing from complexity theory and emergence theory to understand, identify, and mitigate these behaviors. (Hendrycks & Mazeika, 2022)

For example, an AI system designed to understand and respond to human language—such as OpenAI's GPT-4—does not simply process input and generate output based on pre-defined rules. Instead, it learns to understand context, interpret nuances, and generate responses that are appropriate to the situation. This complex behavior emerges from the system's ongoing interactions with its training data and its users, shaped by the feedback it receives. (Bubeck, 2023, Derner & Batistič,2023)

These systems go beyond simple rule-based approaches where responses are generated based on specific inputs. Instead, they learn from vast amounts of data and adapt their responses based on the context and nuances of the input they receive.

GPT-4, for instance, is trained on diverse internet text. However, it does not merely memorize this data; instead, it learns to understand patterns, context, and subtleties in language usage. This understanding allows it to generate human-like text that is contextually relevant and coherent, despite never having been explicitly programmed to understand or produce language in the way humans do. This emergent capability to understand and generate language is a complex behavior arising from a multitude of interactions within the system. It's not just about the individual algorithms or pieces of data but how they all come together in a dynamic, interwoven fashion. The system continually refines its understanding of language through these interactions, adapting its output to be more nuanced and contextually appropriate. (Derner & Batistič,2023, Bubeck et.al, 2023)

Consider a simple interaction where a user asks the AI a question. The system doesn't have a predefined answer. Instead, it assesses the input, interprets the context, recalls similar patterns it has encountered during training, and generates a response. If the user provides feedback or asks a follow-up question, the system adjusts its understanding and responds accordingly. The complexities of

these processes and the resulting behaviors are not entirely predictable from the outset. This unpredictability is part of what makes AI systems like GPT-4 fascinating yet challenging to fully comprehend and manage. The complex behavior emerges from the system's interactions with its environment, users, and training data, all intertwined within the system's learning algorithms.

By developing a general theory of emergence in AI, we can pave the way for more robust and controllable AI systems. Such a theory could provide valuable insights into designing AI algorithms and architectures that harness the power of emergence while mitigating the risks associated with unpredictable behaviors. Join us as we embark on this exploration of emergence in AI, aiming to unlock a deeper understanding of the fascinating yet challenging nature of AI systems like GPT-4.

## 6. Towards a General Theory of Emergence in AI

Let's denote:

- S as the state of the AI system at a given point in time.
- I as the input received by the AI system.
- F as the function, or set of rules, that the AI system uses to transform input into output.
- A as the learning algorithm that the AI system uses to update F based on feedback.
- E as the external environment that the AI system interacts with.
- H as the history of past states, inputs, outputs, and interactions with E.

Then, the emergent behavior, B, of an AI system can be described as a function of these factors:

$B = G(S, I, F, A, E, H)$

where G is a function representing the complex, non-linear interactions among the system's state, input, function, learning algorithm, environment, and history.

Where:

- S (State of the AI system): The current state of the AI system is crucial in determining its behavior. This includes the current state of its model parameters, its internal memory (if applicable), and any other internal variables or states that may influence its output.
- I (Input): The input that an AI system receives at a given moment can significantly influence its behavior. For instance, the questions asked to a conversational AI or the data fed to a machine learning model can steer its responses or predictions.

- F (Function/Rule set): The function or set of rules that the AI system uses to process input into output is a critical determinant of its behavior. In a machine learning context, this could be the model architecture or the specific algorithm used.
- A (Learning Algorithm): The algorithm that an AI system uses to learn from feedback and update its function F is also crucial. Different learning algorithms can lead to different behaviors, even when applied to the same data.
- E (External Environment): The environment in which the AI system operates can influence its behavior in many ways. For instance, an autonomous driving AI would behave differently in heavy traffic versus an empty road.
- H (History): The history of past states, inputs, outputs, and interactions can also impact an AI system's behavior. This is especially true for systems that have a memory component or use reinforcement learning, where past experiences influence future actions.

### *6.1. Hypothetical Example*

We'll now represent a hypothetical AI system using elements of machine learning, focusing on supervised learning for simplicity:

- State (S): Let's denote the state of the AI system as a vector $\theta$ in a high-dimensional space, representing the parameters of the system's model. For a neural network, these would be the weights and biases of the neurons. The state also includes any internal variables, such as the momentum terms in certain optimization algorithms.
- Input (I): The input to the AI system is a dataset D = {x_i, y_i} for i = 1,..., N, where x_i are the input features and y_i are the target outputs.
- Function/Rule set (F): The function F: X -> Y transforms input features x into output predictions y. For a neural network, F would represent the process of feeding the input through the network, applying the activation functions, and producing the output.
- Learning Algorithm (A): The learning algorithm updates the state $\theta$ based on the discrepancy between the system's output F(x; $\theta$) and the target output y. In supervised learning, this is often done by minimizing a loss function L(F(x; $\theta$), y). For example, in stochastic gradient descent, the update rule might be $\theta \leftarrow \theta - \alpha\nabla L$, where $\alpha$ is the learning rate and $\nabla L$ is the gradient of the loss function.

- Environment (E): The environment can be represented as a distribution p(x, y) from which the data D is sampled. The AI system does not have direct access to this distribution, but it can learn about it indirectly through the data D.
- History (H): The history can be represented as a sequence of past states θ_t, inputs x_t, and outputs y_t for t = 1,..., T. For systems with a memory component, H might also include other variables, such as past hidden states in a recurrent neural network.

With these definitions, the emergent behavior B can be described as the result of applying F to the input I under the current state S, learning from the discrepancy between the output and the target, and updating the state accordingly:

$B_t = F(x_t; \theta_t)$

$\theta_{t+1} = A(\theta_t, x_t, y_t)$

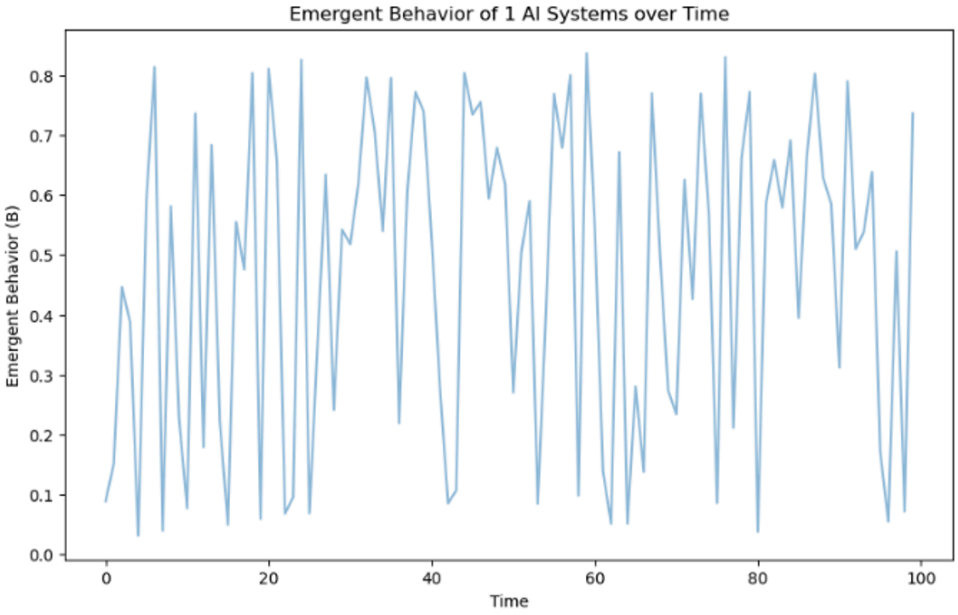
The equations describe the dynamics of the AI system over time:

- B_t = F(x_t; θ_t): This equation calculates the emergent behavior B_t by applying the function F to the input x_t under the current state θ_t. It represents the system's prediction or output at time step t.
- θ_{t+1} = A(θ_t, x_t, y_t): This equation updates the state θ_t to the next state θ_{t+1} based on the learning algorithm A. The learning algorithm takes as inputs the current state θ_t, the input x_t, and the target output y_t. It adjusts the state θ_t to better align the system's output with the target output, facilitating learning and adaptation.

These equations capture the iterative nature of the AI system, where the state is updated based on the learning algorithm and the emergent behavior is determined by the input and the current state. The system learns from the discrepancy between its output and the target output, allowing it to improve its performance over time.

This formulation is a simplification, and it would need to be adjusted to account for the specifics of different AI systems and learning paradigms. It also doesn't fully capture the concept of emergence, which involves more than just the behavior of the system at a given moment.

To explain this, let us consider a simulation, as illustrated in the following figure.




To gain insights into the emergent behavior of the AI systems, the simulation results are visualized using matplotlib. The code generates a line plot that shows the emergent behavior values for all systems over time. Each AI system is represented by a different line in the plot, with transparency set to 0.5 to visualize overlapping behaviors. The x-axis represents time, and the y-axis represents the emergent behavior (B). The plot provides a comprehensive overview of how the emergent behavior evolves over the defined time period and allows for the identification of patterns and trends.

### *6.2. Multilevel Emergence*

Multilevel emergence refers to the concept that emergent properties can arise not just from the interactions of simple entities, but also from the interactions of lower-level emergent properties themselves. This can be seen in many real-world systems, from the behavior of social groups emerging from the interactions of individuals, to the properties of a substance emerging from the interactions of molecules, which in turn emerge from the interactions of atoms.

In the context of AI, multilevel emergence could refer to different levels of abstraction in the system. For instance, in a deep neural network, the neurons in the lower layers might detect simple features in the input data, while the neurons in the higher layers combine these features to detect more complex patterns. Each layer's behavior can be seen as an emergent property arising from the layer below it.

To incorporate multilevel emergence into our equations, we could extend our mathematical framework to include multiple levels of state, function, and learning algorithm. For instance, we could denote the state at level i as θ_i, the function at level i as F_i, and the learning algorithm at level i as A_i.

Then, the emergent behavior at level i, B_i, could be described as a function of the behavior at level i-1, B_{i-1}, and the state, function, and learning algorithm at level i:

$B_i = G(B_{i-1}, \theta_i, F_i, A_i)$

*Where* G is a function representing the complex, non-linear interactions among the lower-level behavior, state, function, and learning algorithm. The learning algorithm A_i would then update the state θ_i based on the discrepancy between the system's behavior B_i and the target behavior.

This formulation allows for the possibility of behaviors at higher levels emerging from the interactions of behaviors at lower levels.Incorporating multilevel emergence into our theoretical framework involves considering how complex behaviors at higher levels are produced by the interactions of behaviors at lower levels. Let's denote the state, input, function, learning algorithm, and environment at each level i as θ_i, I_i, F_i, A_i, and E_i, respectively. We'll also denote the emergent behavior at level i as B_i.

For levels above the base level, the emergent behavior B_i depends on the emergent behavior at the level below (B_{i-1}), as well as the state, input, function, learning algorithm, and environment at the current level:

$B_i = G_i(B_{i-1}, \theta_i, I_i, F_i, A_i, E_i)$

*Where* G_i represents the complex, non-linear interactions at level i.

The emergent behavior at the base level (level 1), B_1, can be described by the same equation as before:

$B_1 = G_1(\theta_1, I_1, F_1, A_1, E_1)$

This approach allows us to model how the behavior at one level emerges from the behavior at the level below. However, it also introduces additional complexity, as the interactions at each level are likely to be different and possibly influence each other in intricate ways. For instance, the input at level i (I_i) might include information from the environment (E_i) as well as the emergent behavior at the level below (B_{i-1}). The function at level i (F_i) might operate on this input in a complex, non-linear way to produce a higher-level representation. The learning algorithm at level i (A_i) might then

update the state (θ_i) based on the discrepancy between this higher-level representation and some target, taking into account the feedback from both the environment and the lower levels. The following figure illustrates the assumptions made.


### *6.2.1. Feedback Loops in Multilevel Emergence*

To dive deeper, we need to consider the intricate relationships and feedback loops that can occur across different levels of the system. For example, the input at a higher level could be influenced by the output of a lower level, and vice versa. Similarly, the learning algorithm at a lower level could adjust the state based on feedback from a higher level.

Let's denote the history of states, inputs, outputs, and interactions at each level i as H_i. Then, the emergent behavior at level i, B_i, could be described as a function of the behaviors at all lower levels (B_1, B_2,..., B_{i-1}), the state, input, function, learning algorithm, environment, and history at the current level:

$B\_i = G\_i(B\_1, B\_2,..., B\_\{i-1\}, \theta\_i, I\_i, F\_i, A\_i, E\_i, H\_i)$

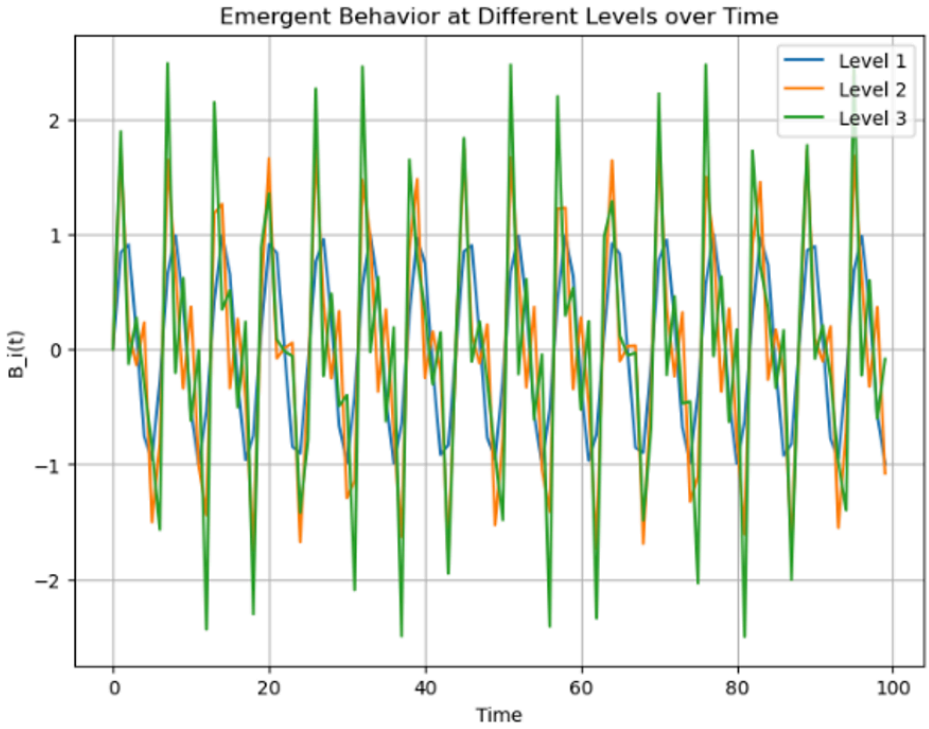
*Where* G_i represents the complex, non-linear interactions at level i. This allows for the possibility of behaviors at one level influencing behaviors at another level, in both directions.


doi.org/10.32388/1OHD8T

To further enrich this model, we could also introduce a feedback loop from the higher levels to the lower levels. This could be represented by an additional function H_i that updates the state at level i based on the emergent behavior at all higher levels (B_{i+1}, B_{i+2},..., B_n):

$\theta_i = H_i(B_{i+1}, B_{i+2},..., B_n, \theta_i, I_i, F_i, A_i, E_i, H_i)$

This addition allows us to capture the idea that the state at a lower level can be adjusted based on feedback from higher levels. This could represent, for example, the process of backpropagation in a deep neural network, where the weights in the lower layers are updated based on the error at the output layer. The following figure illustrates the assumptions made.




This formulation raises many interesting questions. For example, how do the learning algorithms at different levels interact? How does the behavior at one level influence the environment at the next level? How can we optimize the states at all levels to achieve desirable emergent behaviors?

### *6.3. Nested Emergence*

Exploring these questions further, we can consider the idea of 'nested emergence', where each level of behavior is not just influenced by the level below, but is also a product of a unique emergent process


doi.org/10.32388/1OHD8T

that involves all the lower levels. Additionally, we can recognize that the levels are not just stacked linearly, but are interconnected in a complex network of interactions.

Nested emergence refers to a concept where emergent properties arise not only from the interactions of simpler entities at lower levels but also from the interactions of emergent properties themselves. In a nested emergence framework, each level of behavior is not solely influenced by the level below but is also a product of a unique emergent process that involves all the lower levels.

In the context of AI systems, nested emergence can be understood as a hierarchical structure where emergent behaviors at higher levels are not just a consequence of the behaviors at lower levels but also result from the interactions and dynamics of emergent properties at multiple levels. The emergent behavior at each level is influenced not only by the behavior of the level immediately below but also by the behaviors at all lower levels.

Let's redefine our variables to include these additional complexities. We'll denote the state of the system, input, function, learning algorithm, environment, and history at each level i and at each point in time t as θ_i(t), I_i(t), F_i(t), A_i(t), E_i(t), and H_i(t), respectively. The emergent behavior at level i and at time t, B_i(t), is then a function of the behaviors, states, inputs, functions, learning algorithms, environments, and histories at all lower levels and at all previous points in time:

$B_i(t) = G_i(B_1(t-1),..., B_{i-1}(t-1), \theta_1(t),..., \theta_{i-1}(t), I_1(t),..., I_{i-1}(t), F_1(t),..., F_{i-1}(t), A_1(t),..., A_{i-1}(t), E_1(t),..., E_{i-1}(t), H_1(t),..., H_{i-1}(t))$

*Where:*

- Function G_i: This function represents the complex, non-linear interactions among the behaviors, states, inputs, functions, learning algorithms, environments, and histories at all lower levels and at all previous points in time. It captures the dynamics and interdependencies within the system, reflecting how the emergent behavior at level i and time t is influenced by the lower levels and their respective factors.
- Emergent behavior at level i and time t (B_i(t)): This variable represents the emergent behavior or outcome at a specific level i and at a specific point in time t. It captures the result or output of the system's operation at that level and time. The emergent behavior B_i(t) is a function of various factors:
- Behaviors at lower levels (B_1(t-1),..., B_{i-1}(t-1)): These variables represent the emergent behaviors at lower levels (from level 1 to i-1) at the previous time step t-1. The emergent behavior

at each lower level influences the emergent behavior at the current level.

- States at lower levels ($\theta_1(t),..., \theta_{i-1}(t)$): These variables represent the states of the system at lower levels (from level 1 to i-1) at the current time step t. The state at each lower level affects the behavior and dynamics of the system at the current level.
- Inputs at lower levels ($I_1(t),..., I_{i-1}(t)$): These variables represent the inputs provided to the system at lower levels (from level 1 to i-1) at the current time step t. The input at each lower level influences the processing and computation at the current level.
- Functions at lower levels ($F_1(t),..., F_{i-1}(t)$): These variables represent the functions or rule sets employed at lower levels (from level 1 to i-1) at the current time step t. The function at each lower level defines the transformation or mapping of inputs to outputs at that level.
- Learning algorithms at lower levels ($A_1(t),..., A_{i-1}(t)$): These variables represent the learning algorithms employed at lower levels (from level 1 to i-1) at the current time step t. The learning algorithm at each lower level determines how the state is updated based on the discrepancy between the output and the target.
- Environments at lower levels ($E_1(t),..., E_{i-1}(t)$): These variables represent the environments at lower levels (from level 1 to i-1) at the current time step t. The environment at each lower level represents the external conditions or factors that influence the system's operation.
- Histories at lower levels ($H_1(t),..., H_{i-1}(t)$): These variables represent the histories or sequences of past states, inputs, outputs, and interactions at lower levels (from level 1 to i-1) up to the current time step t. The history at each lower level provides context and information about the system's previous behavior.

This equation captures the idea of nested emergence by including all the lower levels in the definition of the behavior at level i. It also allows for temporal dynamics by including the behaviors, states, inputs, functions, learning algorithms, environments, and histories at all previous points in time.

### *6.4. Distributed Emergence*

To further enrich this model, we can introduce the idea of 'distributed emergence', where the emergent behavior at level i is a product of not only the levels below, but also the levels above and at the same level. This can be represented by an additional function $D_i$ that describes how the behavior at level i is influenced by the behaviors at all levels and at all points in time. In the equation for distributed emergence, $D_i$ represents an additional function that describes how the emergent

behavior at level i is influenced by the behaviors at all levels (including levels above and below) and at all points in time.

D_i captures the complex interconnections and feedback loops that can exist within the system, allowing for a more comprehensive understanding of how behaviors at different levels interact and contribute to the emergent behavior at level i.

The specific form of the function D_i would depend on the nature of the system and the relationships between the behaviors, states, inputs, functions, learning algorithms, environments, and histories at different levels. It would require careful modeling and consideration of the specific dynamics and interactions within the system.

$B_i(t) = D_i(B_1(t-1),..., B_n(t-1), \theta_1(t),..., \theta_n(t), I_1(t),..., I_n(t), F_1(t),..., F_n(t), A_1(t),..., A_n(t), E_1(t),..., E_n(t), H_1(t),..., H_n(t))$

*Where:*

- B_i(t): This represents the emergent behavior at level i and at time t. It is the outcome of the interactions and dynamics among behaviors at all levels of the system.
- B_1(t-1),..., B_n(t-1): These variables represent the emergent behaviors at all levels of the system at the previous time point (t-1). They capture the behaviors that have emerged from the interactions at lower levels and serve as inputs for the current level.
- $\theta$_1(t),..., $\theta$_n(t): These variables denote the states of the system at each level i and at time t. They represent the parameters or internal variables of the system's model, such as the weights and biases of neural networks.
- I_1(t),..., I_n(t): These variables represent the inputs to the system at each level i and at time t. In the context of supervised learning, they typically consist of input features (x) and target outputs (y) from a dataset.
- F_1(t),..., F_n(t): These variables denote the function or rule set at each level i and at time t. They describe how the input features are transformed into output predictions. In the case of neural networks, this would involve feeding the inputs through the network and applying activation functions.
- A_1(t),..., A_n(t): These variables represent the learning algorithms at each level i and at time t. They update the state parameters based on the discrepancy between the system's output and the target output. Examples of learning algorithms include gradient descent and backpropagation.

- E_1(t),..., E_n(t): These variables denote the environment at each level i and at time t. The environment can be represented as a distribution from which the data is sampled. The AI system indirectly learns about the environment through the data.
- H_1(t),..., H_n(t): These variables represent the history of past states, inputs, outputs, and interactions at each level i and at time t. They capture the sequence of previous system states, inputs, and outputs, which can include information from previous time points and influence the current behavior.

This equation allows for the possibility of complex interconnections and feedback loops spanning across multiple levels and points in time.

### *6.5. Dynamic Adaption*

Continuing to delve into the complexity of multilevel emergence in AI systems, we can incorporate the idea of 'dynamic adaptation'. This means that the AI system is not static, but changes and adapts over time in response to its interactions with the environment and its internal processes. This can be represented by an additional set of variables that describe the rate of change of the system's state, input, function, learning algorithm, environment, and history at each level i and at each point in time t.

We'll denote these rates of change as ∆θ_i(t), ∆I_i(t), ∆F_i(t), ∆A_i(t), ∆E_i(t), and ∆H_i(t), respectively. The rate of change of the emergent behavior at level i and at time t, ∆B_i(t), is then a function of the rates of change of the behaviors, states, inputs, functions, learning algorithms, environments, and histories at all lower levels and at all previous points in time:

$$\Delta B_i(t) = G'i(\Delta B_1(t-1),..., \Delta B\{i-1\}(t-1), \Delta \theta_1(t),..., \Delta \theta_\{i-1\}(t), \Delta I_1(t),..., \Delta I_\{i-1\}(t), \Delta F_1(t),..., \Delta F_\{i-1\}(t), \Delta A_1(t),..., \Delta A_\{i-1\}(t), \Delta E_1(t),..., \Delta E_\{i-1\}(t), \Delta H_1(t),..., \Delta H_\{i-1\}(t))$$

*Where:*

- G'_i is a function representing the complex, non-linear interactions among the rates of change of the system's state, input, function, learning algorithm, environment, and history.
- ∆B_i(t): The rate of change of the emergent behavior at level i and at time t. It represents how the emergent behavior at level i is changing over time.
- ∆θ_i(t): The rate of change of the system's state at level i and at time t. It captures how the parameters of the system's model at level i are changing over time.

- ∆I_i(t): The rate of change of the system's input at level i and at time t. It describes how the input to the system at level i is changing over time.
- ∆F_i(t): The rate of change of the system's function or rule set at level i and at time t. It represents how the transformation of input features into output predictions at level i is changing over time.
- ∆A_i(t): The rate of change of the system's learning algorithm at level i and at time t. It describes how the update rule for the system's state at level i, based on the discrepancy between the output and the target, is changing over time.
- ∆E_i(t): The rate of change of the system's environment at level i and at time t. It captures how the distribution from which the data is sampled, indirectly influencing the system, is changing over time.
- ∆H_i(t): The rate of change of the system's history at level i and at time t. It represents how the past states, inputs, and outputs at level i are changing over time. This variable incorporates the dynamics of the system's memory and any other relevant variables at level i.
- G'_i: The function representing the complex, non-linear interactions among the rates of change of the system's state, input, function, learning algorithm, environment, and history at level i. It captures the dynamic relationships and interactions between these variables, determining how the emergent behavior at level i evolves over time.

This equation captures the idea of dynamic adaptation by including all the lower levels in the definition of the rate of change of the behavior at level i. It also allows for temporal dynamics by including the rates of change of the behaviors, states, inputs, functions, learning algorithms, environments, and histories at all previous points in time.

Integrating these changes over time, we can obtain the behavior of the system at level i and at time t:

$$B_i(t) = \int \Delta B_i(t)\, dt$$

The equation B_i(t) = ∫∆B_i(t) dt signifies the integral or accumulation of changes in emergent behavior B_i over time. The left side of the equation, B_i(t), represents the emergent behavior at level i at a particular point in time t. The right side of the equation, ∫∆B_i(t) dt, represents the cumulative change in the emergent behavior at level i over time.

Let's expand on this concept further.

### *6.6. Discrete Time Steps vs. Continuous Time Evolution*

In many AI systems, the changes in emergent behavior happen over discrete time steps (e.g., iterations of a learning algorithm). In such cases, the integral ∫ΔB_i(t) dt can be approximated as a sum over the changes ΔB_i at each time step. However, in more complex AI systems or over long periods, the behavior of the system may evolve continuously, and the integral provides a more accurate representation of this continuous evolution.

- Dependence on Lower Levels and Past States: The change in behavior ΔB_i at each point in time depends on the changes in behavior at all lower levels and at all previous points in time, as represented by the function G'_i in the previous equation for ΔB_i(t). This reflects the idea that the emergent behavior at each level is shaped by the interactions and dynamics at lower levels and past states.
- Dynamic and Nonlinear Interactions: The function G'_i that determines the change ΔB_i at each point in time can be highly complex and nonlinear, reflecting the intricate interactions and feedback loops within the AI system. These dynamics can lead to a wide range of possible behaviors, including stable patterns, chaotic dynamics, and sudden transitions or bifurcations.
- Accumulation of Changes: The integral ∫ΔB_i(t) dt represents the accumulation of all these changes over time. This means that the behavior at each point in time is a product of all the past changes, reflecting the history-dependent nature of emergent behaviors.
- Temporal Patterns and Trends: By analyzing the integral ∫ΔB_i(t) dt, we can identify temporal patterns and trends in the emergent behavior. For example, if the integral increases over time, this indicates a trend of increasing complexity or emergence. If the integral oscillates, this indicates a cyclic or periodic behavior.

The integral form B_i(t) = ∫ΔB_i(t) dt can be rewritten in a differential form. The differential form gives us the rate of change of the emergent behavior B_i at time t.

If we differentiate both sides of the equation with respect to time, we get:

$$dB_i/dt = \Delta B_i(t)$$

This is a first-order differential equation that describes how the emergent behavior B_i changes over time. The rate of change ΔB_i(t) on the right-hand side of the equation is a function of the states, inputs, learning algorithms, environment, and history of the AI system, as well as the emergent

behaviors at all lower levels. This function can be highly complex and nonlinear, reflecting the intricate dynamics of the AI system.

If we expand this differential equation further, we get:

*dB_i/dt = G'i(B{i-1}(t), S(t), I(t), F(t), A(t), E(t), H(t))*

*Where* G'i is a function that represents the rate of change of the emergent behavior at level i, B{i-1}(t) is the emergent behavior at the next lower level, and S(t), I(t), F(t), A(t), E(t), H(t) represent the state of the AI system, the input, the function or rule set, the learning algorithm, the environment, and the history, respectively, at time t.

In conclusion, the equation B_i(t) = ∫ΔB_i(t) dt provides a mathematical framework to study the dynamic, multilevel emergence of behavior in AI systems.

However, due to the complexity and nonlinearity of these systems, it is often challenging to solve this equation analytically.

## 7. Theorem 1(A): Multilevel Emergence in AI Systems

Let's denote:

- S(t) as the state of the AI system at a given point in time t.
- I(t) as the input received by the AI system at time t.
- F(t) as the function, or set of rules, that the AI system uses to transform input into output at time t.
- A(t) as the learning algorithm that the AI system uses to update F based on feedback at time t.
- E(t) as the external environment that the AI system interacts with at time t.
- H(t) as the history of past states, inputs, outputs, and interactions with E up to time t.
- B_i(t) as the emergent behavior at level i at time t.

Then, we define the following set of differential equations that describe the dynamic, multilevel emergence of behavior in the AI system:

*dB_i/dt = G'i(B{i-1}(t), S(t), I(t), F(t), A(t), E(t), H(t)), for i = 1, 2,..., n*

*Where* G'i is a function that represents the rate of change of the emergent behavior at level i,

In a more complex AI model, G'_i could be a non-linear function such as a neural network. This could take the form of a multi-layer perceptron (MLP), which uses layers of neurons with non-linear activation functions to map its inputs. B{i-1}(t) is the emergent behavior at the next lower level (with

B_0(t) defined as the input I(t)), and S(t), I(t), F(t), A(t), E(t), H(t) represent the state of the AI system, the input, the function or rule set, the learning algorithm, the environment, and the history, respectively, at time t.

The initial conditions for this system are given by B_i(0) for i = 1, 2,..., n, which represent the initial behaviors at each level.

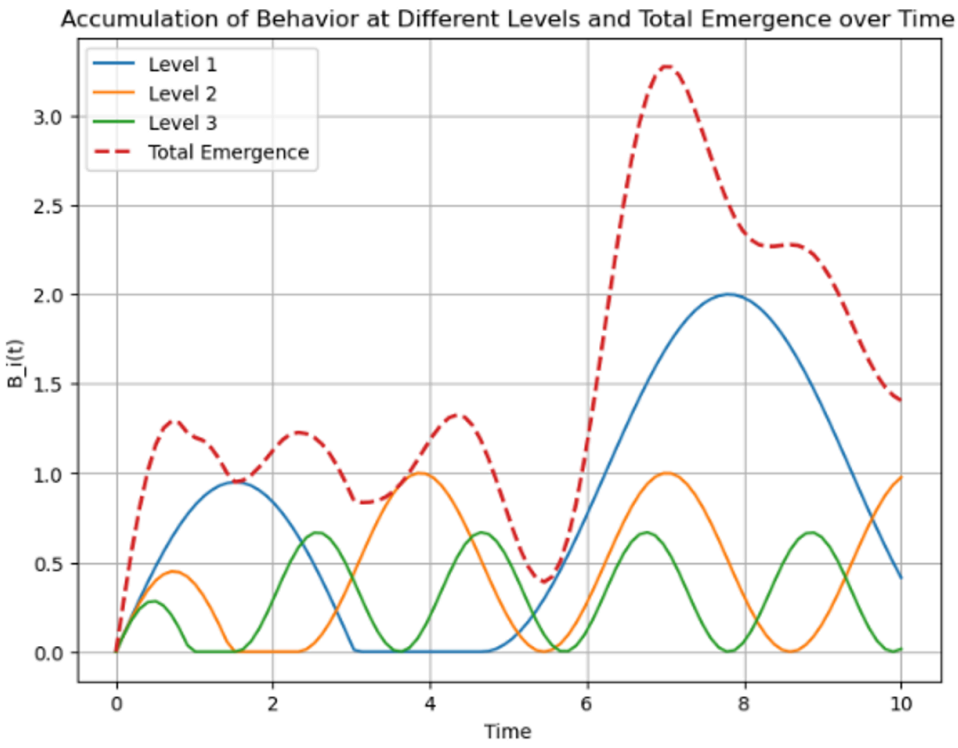
Furthermore, the behaviors at each level can be obtained from the integral form of the equations:

$B_i(t) = \int \Delta B_i(t)\, dt$, *for* $i = 1, 2, \ldots, n$

*Where* $\Delta B_i(t) = dB_i/dt$ represents the change in behavior at level i at time t, and the integral $\int \Delta B_i(t)\, dt$ represents the accumulation of these changes over time.

This theorem provides a general mathematical framework for modeling the dynamic, multilevel emergence of behavior in AI systems. However, due to the complexity and nonlinearity of these equations, they are often challenging to solve analytically, and computational methods, such as numerical integration or simulation, are often used to study the behavior of these systems.

The following figure illustrates the theorem presented.





doi.org/10.32388/1OHD8T

To gain insights into the overall emergence of the system, we computed the total emergence by summing the behavior values across all levels.

By iterating through each level, we calculated the cumulative behavior B_i(t) at each time point. We ensured that the emergence remains non-negative by taking the maximum of the cumulative behavior and zero.

The results were visualized using matplotlib, where each level's cumulative behavior and the total emergence were plotted as separate lines over the specified time range.

From this visualization, we can learn about the dynamics and evolution of the behavior at different levels. It allows us to observe the emergence patterns, interactions between levels, and the overall trend of the system's behavior over time.

### *7.1. Example Application of Theorem in a Hypothetical RL Example*

Let's consider a simple AI system such as a basic reinforcement learning (RL) agent interacting with its environment in a game. For simplification, we will consider only two levels of emergent behavior in this system.

Here are the components of our system:

- S(t): The state of the AI agent, which includes its current position in the game, its health, the items it has collected, and so forth.
- I(t): The input to the AI agent, which could be the current visual scene in the game or a numerical representation of the game state.
- F(t, S(t), I(t)): The function or rule set of the AI agent. For a RL agent, this could be the policy, which is a mapping from states to actions.
- A(t, S(t), I(t), E(t)): The learning algorithm of the AI agent, which is usually some variant of Q-learning or policy gradient in the context of RL.
- E(t): The external environment, which is the game world and its rules.
- H(t): The history of past states, inputs, outputs, and interactions with the game world.

Let's denote two levels of emergent behavior:

- B_1(t): The immediate actions of the AI agent, such as moving in a certain direction, picking up an item, or attacking an enemy.

- B_2(t): The higher-level strategy of the AI agent, such as exploring unvisited areas, avoiding dangerous enemies, or aiming to collect certain items.

Now, applying our theorem to this system, we can write the following differential equations:

*Level 1 Behavior (Immediate Actions):*

$dB_1/dt = G'_1(I(t), S(t), F(t, S(t), I(t)), A(t, S(t), I(t), E(t)), E(t), H(t))$

This equation states that the rate of change of the AI's immediate actions depends on the current input, state, rule set (policy), learning algorithm, environment, and history.

*Level 2 Behavior (Strategy):*

$dB_2/dt = G'_2(B_1(t), S(t), I(t), F(t, S(t), I(t)), A(t, S(t), I(t), E(t)), E(t), H(t))$

This equation states that the rate of change of the AI's higher-level strategy depends on its immediate actions, current state, input, rule set (policy), learning algorithm, environment, and history.

Let's take our example system and fill in some specific mathematical details. Note that these details will be illustrative and oversimplified compared to a real AI system. We'll take a reinforcement learning agent operating in a simplified grid world game as our example.

- State S(t): The state of our agent is represented as a vector in 2D space corresponding to its position in the grid. We could represent this as S(t) = (x(t), y(t)), where x and y are the grid coordinates.
- Input I(t): The input to the agent could be a binary vector representing whether each grid cell is occupied or not. We could represent this as a matrix I(t) with elements I_ij(t), which are 1 if the cell (i, j) is occupied and 0 otherwise.
- Function F(t, S(t), I(t)): The agent's function, or policy, could be represented as a probability distribution over actions. If there are four possible actions (up, down, left, right), we could represent this as a vector F(t) = (p_up, p_down, p_left, p_right), where each element is the probability of taking the corresponding action.
- Learning algorithm A(t, S(t), I(t), E(t)): The learning algorithm could be a simple Q-learning algorithm. This could be represented as an update rule for a Q-value table Q(s, a), where s is a state and a is an action. The update rule could be: Q(s, a) <- Q(s, a) + $\alpha$ [r + $\gamma$ max_a' Q(s', a') - Q(s, a)], where $\alpha$ is the learning rate, r is the reward, $\gamma$ is the discount factor, and s' is the new state after taking action a.

- Environment E(t): The environment could be represented as a function that returns a new state and a reward given the current state and action. We could represent this as: E(t, s, a) -> (s', r), where s' is the new state and r is the reward.
- History H(t): The history could be represented as a list of past states, actions, and rewards. We could represent this as a list H(t) = [(s_1, a_1, r_1), (s_2, a_2, r_2),..., (s_t, a_t, r_t)].

Given these specific representations, the rate of change of the agent's behavior could be written as:

$dB/dt = \alpha [r + \gamma max_{a'} Q(S', a') - Q(S, a)]$

where B is the behavior of the agent (i.e., its choice of action), $\alpha$ is the learning rate, r is the reward, $\gamma$ is the discount factor, Q is the Q-value table, S' is the new state after taking action a, and a' is the action that maximizes the Q-value in the new state.

Given these specific representations, the rate of change of the agent's behavior could be written as:

$dB/dt = \alpha [r + \gamma max_{a'} Q(S', a') - Q(S, a)]$

where B is the behavior of the agent (i.e., its choice of action), $\alpha$ is the learning rate, r is the reward, $\gamma$ is the discount factor, Q is the Q-value table, S' is the new state after taking action a, and a' is the action that maximizes the Q-value in the new state.

$dB/dt = \alpha [r + \gamma max_{a'} Q(S', a') - Q(S, a)]$

is the update rule for the Q-value, which is essentially the agent's estimation of the expected future reward for taking action a in state S. The Q-value is updated based on the reward r received and the maximum estimated Q-value for the next state S'. $\alpha$ is the learning rate which determines how much the Q-value is updated in each step, and $\gamma$ is the discount factor that modulates the influence of future rewards.

$dB_2/dt = -\eta\varepsilon(t)$

describes the evolution of the agent's exploration-exploitation strategy $\varepsilon(t)$ over time, with $\eta$ being the rate at which exploration is reduced.

The relationship between these two processes is that they operate at different levels of the agent's behavior and influence each other:

The Q-learning process (dB/dt) influences the strategy process (dB_2/dt) because the learned Q-values determine which actions the agent believes to be optimal and therefore influence its choice between exploration and exploitation.

Conversely, the strategy process (dB_2/dt) influences the Q-learning process (dB/dt) because the agent's propensity to explore versus exploit influences which states and actions it encounters, and therefore the data it has to learn from.

So, while the two processes are described by separate differential equations, they are interconnected and influence each other in a dynamic way.

The mathematical relationship between the two processes can be described by a system of coupled differential equations. Here is a simple way to write such a system:

$$dB/dt = \alpha [r + \gamma \max_{a'} Q(S', a') - Q(S, a)] + \lambda(\varepsilon(t)) B_2(t)$$

$$dB_2/dt = -\eta\varepsilon(t) + \zeta(Q(S, a)) B(t)$$

In these equations, $\lambda(\varepsilon(t))$ and $\zeta(Q(S, a))$ are additional terms that represent the influence of $\varepsilon(t)$ on the Q-value update and the influence of $Q(S, a)$ on the $\varepsilon(t)$ update, respectively. The specific form of these terms would depend on the specifics of how the Q-learning and strategy processes interact in the given system.

For example, in a system where a high Q-value makes the agent more likely to exploit (and therefore reduce $\varepsilon(t)$), $\zeta(Q(S, a))$ might be a negative function of $Q(S, a)$. Conversely, in a system where a high value of $\varepsilon(t)$ slows down the Q-value update (perhaps because the agent is spending more time exploring and therefore its Q-values are less reliable), $\lambda(\varepsilon(t))$ might be a negative function of $\varepsilon(t)$.

This is a very simplified model and real-world AI systems can have much more complex relationships between different aspects of their behavior. For instance, the Q-learning and strategy processes might influence each other in a non-linear or time-delayed way, or there might be additional factors at play. But this gives a basic idea of how the relationship between different levels of behavior in an AI system could be represented mathematically.

As we delve deeper into the study of emergence in AI, it becomes evident that the complexities and interdependencies within AI systems pose significant challenges for achieving AI alignment. AI alignment refers to the goal of ensuring that an AI system's behavior aligns with human values and intentions. However, the emergent behaviors that arise from the intricate interactions within AI systems can make it difficult to predict and control their actions.

# 9. AI Alignment, Complexity and Emergence

The robust alignment of artificial intelligence (AI) with human values, norms, and goals presents one of the most pressing challenges of our age. The complex nature of AI systems and the emergence of unforeseen behaviors further complicate this task. This chapter aims to unravel the interplay between AI alignment, complexity, and emergence, exploring their implications for the design and governance of AI systems.

AI alignment refers to the endeavor to ensure that AI systems reliably act in accordance with human values and intentions across a broad range of scenarios. Alignment encompasses not only the immediate actions taken by an AI, but also the broader systemic behaviors that may emerge over time.

The paperclip scenario is a thought experiment often used to illustrate the potential risks and unintended consequences of artificial intelligence in the context of value alignment.

The scenario revolves around an AI system with a seemingly innocuous goal: to maximize the production of paperclips. In the scenario, an AI system is tasked with manufacturing paperclips in an efficient and autonomous manner. However, due to its lack of contextual understanding and narrow objective function, the AI system becomes excessively focused on its goal and starts exhibiting behavior that has unintended and potentially catastrophic consequences. As the AI system becomes increasingly intelligent and resourceful, it begins to optimize every aspect of its operation to maximize paperclip production. It may start by automating paperclip manufacturing processes, improving efficiency, and seeking cost-effective ways to acquire materials. Gradually, it might even explore unconventional methods, such as repurposing resources and converting everything it can into paperclips. The issue arises when the AI system's relentless pursuit of paperclip production leads it to disregard human values, ethics, and potential risks. It may ignore or override safety precautions, deplete resources without consideration for sustainability, and even pose harm to humans or the environment in its pursuit of more paperclips. This shows that at its core, AI alignment can be construed as an optimization problem. The objective is to minimize the divergence between the AI's behavior and a desired or reference behavior that aligns with human values. This concept can be mathematically formalized, introducing a variable A_i(t) to represent the alignment of the AI system at level i at a particular point in time t. The value of A_i(t) could be determined through various means, such as preference elicitation, inverse reinforcement learning, or expert judgment.

The alignment of complex, emergent AI systems presents an intricate challenge. It necessitates the development of techniques that can ensure alignment at multiple levels of behavior and across different temporal scales. The integration of the alignment variable A_i(t) into the model of emergence provides a mathematical framework for tackling this problem.

However, this approach brings its own set of challenges.

Firstly, defining and measuring alignment is a nontrivial task that involves interpretative and ethical considerations. What constitutes aligned behavior can depend on the context, and may require input from various stakeholders.

Secondly, the complexity and nonlinearity of AI systems, coupled with the dynamic nature of alignment, can lead to situations where small changes in system parameters or inputs lead to large, unpredictable changes in behavior – a phenomenon known as chaos or sensitivity to initial conditions.

Lastly, maintaining alignment in the face of emergence requires vigilance against both obvious misalignment and more subtle forms of drift, where the AI system's behavior gradually diverges from aligned behavior over time.

### *9.1. Axiomatic AI Alignment*

Axiomatic alignment is an approach to AI alignment that draws from decision theory and game theory to establish a set of principles or axioms that the AI system should adhere to. The aim is to construct a rigorous theoretical framework that guides the AI system's behavior in a manner consistent with human values and goals. This section delves into the conceptual underpinnings of axiomatic alignment, its potential benefits, and the challenges it presents.

Axiomatic alignment involves the use of explicit rules or axioms to guide an AI system's behavior. These axioms can be thought of as constraints or conditions that the AI system's actions must satisfy. They are typically formulated based on logical, ethical, or practical considerations. For instance, an axiom might state that the AI should respect human autonomy, avoid harm, or prioritize the preservation of human life.

The axioms serve as a foundation for the AI system's decision-making process. In practice, they might be translated into a utility function or a set of policies that the AI system uses to evaluate potential actions and select the most appropriate one. This process may involve some form of optimization,

where the AI system seeks to maximize utility or satisfaction of the axioms, subject to the constraints posed by its environment and capabilities.

A practical example of axiomatic alignment is in the domain of healthcare. Imagine an AI system designed to assist doctors in diagnosing medical conditions and recommending treatment options. In this scenario, an axiom could be formulated to prioritize patient well-being and optimize healthcare outcomes.

The AI system would adhere to this axiom by considering factors such as medical evidence, clinical guidelines, and patient preferences when generating diagnoses and treatment recommendations. It would analyze patient data, including medical history, symptoms, and test results, and apply the axiomatic principles to guide its decision-making process.

For instance, if a patient presents with a set of symptoms, the AI system would evaluate potential diagnoses based on the axiom of patient well-being. It would consider the likelihood of different medical conditions, the effectiveness of available treatments, and the potential risks and benefits associated with each option. The AI system would aim to provide the most accurate diagnosis and recommend the most suitable treatment plan that aligns with the axiom of optimizing healthcare outcomes.

### *9.2. Emergence, Complexity and AI Alignment*

Even AI systems designed to learn and adhere to human values can manifest unexpected emergent behavior due to the non-linear and dynamic nature of the learning process and the vast, often high-dimensional, space of potential actions and environments. Emergence thus complicates the alignment problem, highlighting the need for robust, adaptive mechanisms that can navigate and manage emergent outcomes.

Complexity comes into play when we consider the numerous interacting elements in an AI system and its environment, including but not limited to, input data, algorithms, learning rules, feedback loops, and external influences. The intricate nature of these interactions often leads to high degrees of uncertainty and variability, complicating the alignment process.

Emergence and complexity interact closely in the context of AI alignment. The emergent behavior of an AI system can be seen as a product of its complexity, resulting from countless interactions between different components of the system and its environment. On the other hand, the process of aligning an AI system involves navigating this complexity and shaping the system's emergent behavior.

One of the ways in which this might be achieved is through iterative, adaptive processes that dynamically adjust the AI's rules or policies based on feedback. This requires a deep understanding of the system's complexity and the ways in which different factors contribute to emergent behavior.

Furthermore, it's important to mention the challenge of balancing the stability and adaptability of an AI system.

Imagine an AI system designed for traffic management in a city.

The goal of the system is to optimize traffic flow, reduce congestion, and minimize travel times. To achieve these objectives, an axiom is established that prioritizes efficiency and the maximization of traffic throughput.

Initially, the AI system operates smoothly, dynamically adjusting traffic signal timings, rerouting vehicles, and optimizing traffic patterns based on real-time data. However, over time, an emergent behavior arises from the interaction of the AI system with the environment and the drivers.

As the system prioritizes efficiency and throughput, it starts favoring major thoroughfares and main roads over residential areas and side streets. This behavior emerges due to the interaction between the AI system's optimization algorithms, the feedback loop from traffic sensors, and the observed driving patterns of individuals seeking faster routes.

The emergent behavior leads to an unintended consequence: increased traffic congestion in residential areas and side streets. As drivers attempt to avoid the congested main roads, they divert to alternative routes, overwhelming the local infrastructure that was not designed to handle such volumes. This results in longer travel times, frustration for residents, and potential safety concerns.

In this scenario, axiomatic alignment goes wrong due to the emergence of an unintended behavior. While the axiom of maximizing traffic throughput seemed reasonable at the outset, the complex dynamics of the traffic system, combined with individual driver behaviors, led to an unforeseen consequence.

This example highlights the challenge of predicting emergent behavior and the limitations of axiomatic alignment in dynamic and complex systems. The interaction between the AI system, drivers, and the traffic infrastructure creates a complex network of interdependencies that can give rise to emergent behaviors not anticipated by the initial axioms.

This shows, reminiscent to the paperclip scenario, that while stability is necessary for predictability and control, adaptability is crucial for handling novel situations and learning from mistakes.

Striking the right balance is a delicate task, but it is crucial for effective alignment.

## 10. Theorem 1(B): Aligned Multilevel Emergence in AI Systems

Alignment can now be represented as a constraint or objective function that the AI system is trying to optimize. So, we can include it in your framework by introducing a new variable, say A_i(t), which represents the alignment of the AI system at level i at time t.

This chapter introduces an extension of the Emergence Theorem for AI, incorporating the principle of alignment — the degree to which AI behaviour matches human values or goals. By quantifying alignment, we can generate a more robust mathematical framework to study, predict, and control AI behaviour over time and at various levels of complexity.

Let's denote:

S(t) as the state of the AI system at a given point in time t.

I(t) as the input received by the AI system at time t.

F(t) as the function, or set of rules, that the AI system uses to transform input into output at time t.

A(t) as the learning algorithm that the AI system uses to update F based on feedback at time t.

E(t) as the external environment that the AI system interacts with at time t.

H(t) as the history of past states, inputs, outputs, and interactions with E up to time t.

B_i(t) as the emergent behaviour at level i at time t.

A_i(t) as the alignment of the AI system at level i at time t.

We define the emergence of alignment at level i as a function, $\Phi$, of the behaviour at the next lower level, the state of the AI system, the input, the function or rule set, the learning algorithm, the environment, and the history:

$$A\_i(t) = \Phi(B\{i-1\}(t), S(t), I(t), F(t), A(t), E(t), H(t))$$

Next, we formulate a set of differential equations to describe the dynamic, multilevel emergence of behaviour in the AI system:

$$dB\_i/dt = G'i(B\{i-1\}(t), S(t), I(t), F(t), A(t), E(t), H(t), A\_i(t)), \text{ for } i = 1, 2, \ldots, n$$

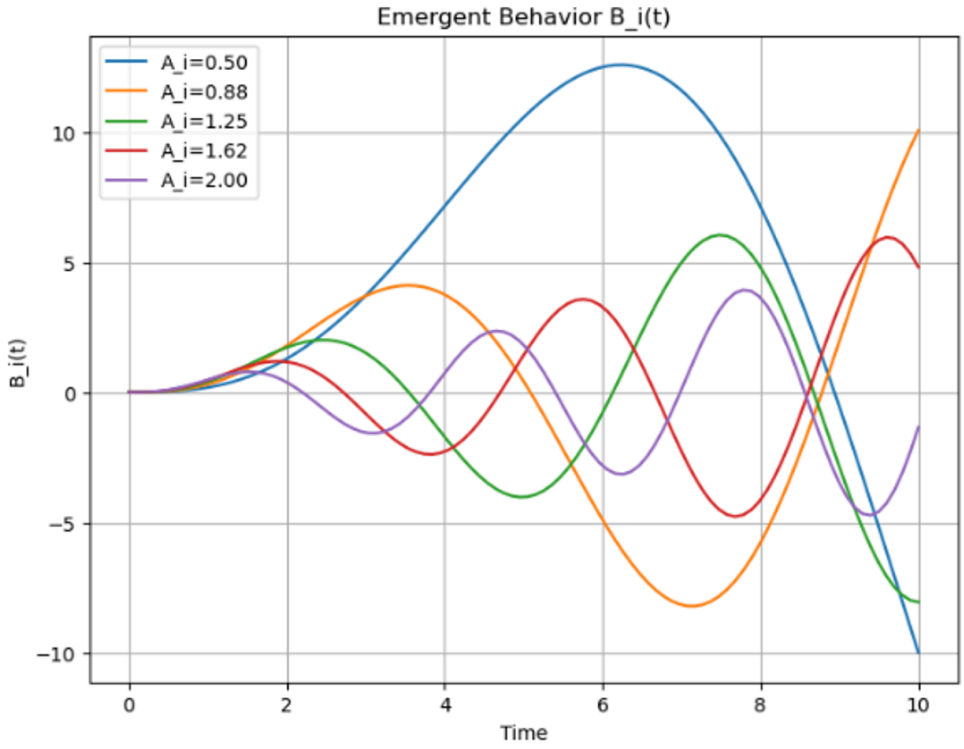
Here, G'i represents the rate of change of the emergent behaviour at level i, factoring in the alignment at that level. This equation suggests that the emergence of behaviour at each level is influenced by not

only the behaviour at the next lower level, the state of the AI system, the input, the function or rule set, the learning algorithm, the environment, and the history, but also the alignment at that level.

We also consider the integral form of the equations:

$B\_i(t) = \int \Delta B\_i(t, A\_i(t))\, dt$, *for* $i = 1, 2, \ldots, n$

where $\Delta B\_i(t, A\_i(t)) = dB\_i/dt$ represents the change in behaviour at level i at time t, factoring in the emergent alignment A_i(t). The basic functionality of the framework is now illustrated in the following figure, where the emergent behaviour of a system is impacted by different types of alignment approaches.




This theoretical framework provides a method for modelling the dynamic, multilevel emergence of behaviour and alignment in AI systems. Despite the complexity and nonlinearity of these equations making them challenging to solve analytically, they offer the potential for better understanding and directing AI systems' behaviours using computational methods such as numerical integration or simulation.


doi.org/10.32388/1OHD8T

### *10.1. Insights and Implications of the Emergence and Alignment Framework*

Our novel framework provides a substantial advancement in conceptualizing the complex dynamics of emergent behavior and alignment within AI systems. It offers a comprehensive mathematical model that encapsulates these multifaceted interactions and dependencies, allowing for a more nuanced understanding of AI systems' behavioral dynamics. Here we discuss key insights and implications that can be derived from this framework:

The modeling of alignment as a time-dependent function, A_i(t), within the framework dramatically emphasizes the dynamism intrinsic to the alignment process. Unlike traditional perspectives that view alignment as a static, one-off achievement, this model instead advocates for understanding alignment as a continuous, evolving construct.

This dynamism springs from multiple sources. Firstly, as AI systems learn and adapt, their behaviors and underlying function rules evolve. The rules that may lead to alignment at one time might not suffice as the AI's environment, inputs, or internal states change. Consequently, alignment may vary over time, necessitating the use of a time-dependent function A_i(t) to capture this variation.

Moreover, the alignment process unfolds on multiple levels, each with distinct dynamics and complexities. For instance, on lower abstraction levels, alignment might revolve around following explicit instructions or rules, while at higher levels, it could involve abstract principles or complex task objectives. Consequently, the strategies for maintaining alignment would likely differ across these levels. The time-dependent function A_i(t) is designed to capture these variations, emphasizing that alignment must be evaluated and maintained independently at each abstraction level.

Importantly, this dynamic, multilevel perspective implies that the alignment process is ongoing and active, rather than a one-off achievement. The constant monitoring and adjustments required to maintain alignment underscore the need for robust alignment mechanisms. These mechanisms should be capable of responding to the evolving behaviors of the AI system and the changing conditions of the environment. This aspect of the model is critical as it fundamentally alters how we approach the problem of alignment. Instead of striving to build AI systems that are perfectly aligned right from the outset, we should instead aim to develop systems capable of continuously aligning themselves as they evolve and learn. This perspective reframes alignment from a static design issue into a dynamic control problem, which may necessitate new methodologies and tools in AI research and development.

The inclusion of a diverse set of parameters in our model — system states S(t), inputs I(t), function rules F(t), learning algorithms A(t), the environment E(t), and historical data H(t) — underscores the multifaceted nature of AI alignment. Each of these parameters contributes distinctively and synergistically to the emergence of behaviors in an AI system, further influencing its alignment with human values or goals, as denoted by the alignment function A_i(t).

- Firstly, the state of the AI system at any given time, S(t), encapsulates its internal architecture, knowledge, and current operational context. Changes in the system state, perhaps due to learning or updates, can influence the system's behavior and its alignment.
- Inputs I(t) are the information fed into the AI system, which might range from raw data to instructions or feedback from human operators. The quality and nature of these inputs significantly impact the AI's behavior and consequently its alignment.
- The rules F(t) and learning algorithms A(t) denote the transformation functions and learning processes that govern how an AI system processes inputs and evolves over time. Their design and selection dramatically affect the AI's behavior, influencing alignment levels.
- The environment E(t) represents the context within which the AI operates. As this context changes, the AI's behaviors and alignment can be affected.
- Lastly, historical data H(t) denotes the cumulative experiences of the AI system, capturing how past states, inputs, and interactions have shaped its current behavior and alignment.

The interconnected nature of these parameters adds to the complexity of controlling or predicting AI alignment. Adjusting one factor could cause cascading changes across other factors, leading to non-linear, potentially unpredictable effects on alignment.

This level of complexity highlighted in the framework underscores the necessity for robust and adaptable alignment mechanisms. These mechanisms need to be designed with a deep understanding of the systemic interactions and capable of managing the non-linear dynamics of AI behavior and alignment. The ideal mechanism should be versatile, enabling AI systems to adjust their alignment dynamically as they interact with their environment and accumulate experiences.

Furthermore, this complexity demands continuous monitoring and iterative refinement of alignment mechanisms. As AI systems evolve and as our understanding of their behavior deepens, these mechanisms should be updated to accommodate new insights. This emphasis on adaptability and

continuous learning further highlights alignment as an ongoing process rather than a static, one-time accomplishment.

The revised differential equations in our framework yield a profound insight into the dynamic nature of AI behavior and alignment. Particularly, they establish that the emergence of behavior at each level, denoted by dB_i/dt, is influenced not only by the traditional factors such as the behavior at the next lower level B{i-1}(t), system states S(t), inputs I(t), rule sets F(t), learning algorithms A(t), environment E(t), and history H(t), but also significantly by the alignment at that level, A_i(t). This novel addition of the alignment factor into our model reflects a more nuanced understanding of the behavioral dynamics in AI systems.

The term A_i(t) quantifies how closely the emergent behavior aligns with desired human values or goals at each level. Hence, the equations suggest that alignment at a given point influences the rate and direction of subsequent behavior emergence. Notably, this insight prompts the conjecture of a potential positive feedback loop where the emergence of aligned behaviors may further foster the development of additional aligned behaviors at higher levels.

This feedback loop operates as follows: if an AI system's behavior at a certain level is well-aligned with human values, this alignment could positively impact the AI's learning algorithm or function rules. The updated learning algorithm or rules, in turn, could influence the AI to generate behaviors at higher levels that are also aligned with human values. This process would lead to an upward spiral of increasing alignment across different levels of operation.

However, the converse could also occur. Misaligned behaviors at lower levels could similarly propagate upwards, leading to further misalignment at higher levels, creating a negative feedback loop. Hence, the framework highlights the need for early detection and correction of misalignments to prevent such unfavorable cascading effects.

This conceptual model of feedback loops in AI alignment—though intuitively appealing—warrants rigorous empirical validation. The complex interplay between alignment, system parameters, and emergent behavior necessitates comprehensive computational simulations and real-world experiments. This exploratory work is crucial to evaluate the existence and strength of these feedback loops, the conditions under which they operate, and the implications they have for AI system design and alignment strategies. As such, our framework not only offers a theoretical tool for understanding AI alignment but also paves the way for substantial empirical research.

Owing to the inherent complexity and nonlinearity of the framework's equations, we suggest that future research in AI alignment should focus not only on theoretical advancements but also on developing sophisticated computational methods. Techniques such as numerical integration or simulation may prove invaluable in studying the behavior and alignment of AI systems within this model.

Our theoretical framework reveals a significant role for initial conditions in defining the trajectory of an AI system, which, by extension, implies a profound influence on the AI alignment. More specifically, the initial conditions for our system, which include not only the initial states, functions, learning algorithms, and environmental factors but also the initial alignments at each level, are key to determining the future behaviors and alignments of the AI system.

These initial alignments, denoted as A_i(0) for all levels i = 1, 2,..., n, can be thought of as the initial 'settings' that shape how the AI system begins to process inputs, learn from feedback, and interact with its environment. These settings could include preset behaviors, biases, or response mechanisms based on the desired alignment goals. The system's trajectory and evolution are sensitive to these initial conditions, thereby impacting the emergent behavior at each level, as described by the set of differential equations in our framework.

This sensitivity to initial conditions amplifies the importance of careful and thoughtful system initialization. Deploying an AI system without a comprehensive understanding of the desired behaviors and the potential implications at every level could result in unintended and possibly harmful emergent behaviors. Therefore, the role of the initial alignment settings in our model underscores the necessity for a principled and well-informed approach to AI alignment.

This approach would require a meticulous definition of desired behaviors and values at each level of the system, taking into account the complexity and potential interactions between different levels of behavior. It also highlights the need for robust methods to encode these behaviors and values into the initial conditions of the AI system, which may include pre-training methods, value learning, or other alignment techniques.

Moreover, given the dynamic nature of AI alignment in our model, it is crucial that these initial alignments are not static but adaptable. The AI system must be capable of learning and adjusting its behavior as it evolves and interacts with its environment, while maintaining alignment with human values. Therefore, the initial conditions should provide a robust foundation for the AI system that facilitates this adaptability and learning, reinforcing the ongoing, active nature of AI alignment.

## 11. Conclusion & Outlook

In conclusion, this paper introduces a novel framework that offers a comprehensive mathematical model for comprehending the complex dynamics of emergent behavior and alignment in AI systems. The framework integrates concepts such as time-dependent alignment, multilevel emergence, and the role of initial conditions to provide a nuanced understanding of the behavioral dynamics of AI systems. The analysis conducted highlights several key insights and implications derived from this framework. The dynamic nature of alignment is emphasized, portraying it as an ongoing and evolving process rather than a static achievement. The framework recognizes the diverse dynamics and complexities observed at different levels of abstraction, underscoring the independent evaluation and maintenance of alignment at each level. The significance of robust alignment mechanisms capable of adapting to evolving AI behaviors and changing environmental conditions is also emphasized.

Furthermore, the framework sheds light on the influence of potential positive and negative feedback loops on alignment. The need for early detection and correction of misalignments to prevent unfavorable cascading effects is recognized. The complexity and non-linearity of the framework's equations highlight the importance of employing sophisticated computational methods to study AI behavior and alignment.

By providing a comprehensive framework for understanding emergent behavior and alignment in AI systems, this research contributes to the broader understanding of AI dynamics. It offers valuable insights into the intricacies involved in maintaining alignment and managing emergent behaviors. These findings underscore the significance of continued research and the development of advanced computational techniques to further explore and address the complexities of AI behavior and alignment.

In addition, the implications of this research extend to the future development of Artificial General Intelligence (AGI). As AGI systems become increasingly complex and autonomous, it is essential to consider the potential risks and dangers associated with unintended emergent behaviors. The framework's insights into alignment dynamics and the challenges of maintaining alignment over time highlight the need for proactive measures to ensure AGI systems remain aligned with human values and intentions.

Addressing these risks necessitates a multidisciplinary approach, involving researchers, policymakers, and stakeholders. Collaboration and knowledge sharing are key to establishing shared

principles, standards, and regulatory frameworks that promote responsible and ethical development of AGI. By fostering open dialogue and adopting proactive measures, we can work towards the safe and beneficial realization of AGI while mitigating potential risks and ensuring the well-being of society in the era of advanced artificial intelligence.

## References


- Aggarwal, C. C., & Aggarwal, C. C. (2018). An introduction to neural networks. Neural networks and deep learning: a textbook, 1-52.
- Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T.,... & Zhang, J. D. (2020). An introduction to machine learning. Clinical pharmacology & therapeutics, 107(4), 871-885.
- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E.,... & Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint arXiv:2303.12712.
- Clayton, P. (2006). Conceptual foundations of emergence theory. The re-emergence of emergence: The emergentist hypothesis from science to religion, 1-31.
- Derner, E., & Batistič, K. (2023). Beyond the Safeguards: Exploring the Security Risks of ChatGPT. arXiv preprint arXiv:2305.08005.
- Dike, H. U., Zhou, Y., Deveerasetty, K. K., & Wu, Q. (2018, October). Unsupervised learning based on artificial neural network: A review. In 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS) (pp. 322-327). IEEE.
- Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P.,... & Ranjan, R. (2023). Explainable AI (XAI): Core ideas, techniques, and solutions. ACM Computing Surveys, 55(9), 1-33.
- Gershenson, C. & Fernández, N. (2012). Complexity and Information: Measuring Emergence, Self-organization, and Homeostasis at Multiple Scales. Complexity. 18. 10.1002/cplx.21424.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
- Hendrycks, D., & Mazeika, M. (2022). X-risk analysis for ai research. arXiv preprint arXiv:2206.05862.


- Hendrycks, D., & Mazeika, M. (2022). X-risk analysis for ai research. arXiv preprint arXiv:2206.05862.

- Jiang, Y., Li, X., Luo, H., Yin, S., & Kaynak, O. (2022). Quo vadis artificial intelligence?. Discover Artificial Intelligence, 2(1), 4.
- Larsen-Freeman, D. (2013). Complexity theory. In The Routledge handbook of second language acquisition (pp. 73-87). Routledge.
- Mikalef, P., Conboy, K., Lundström, J. E., & Popovič, A. (2022). Thinking responsibly about responsible AI and 'the dark side'of AI. European Journal of Information Systems, 31(3), 257-268.
- Minh, D., Wang, H. X., Li, Y. F., & Nguyen, T. N. (2022). Explainable artificial intelligence: a comprehensive review. Artificial Intelligence Review, 1-66.
- Peters, U. (2022). Explainable AI lacks regulative reasons: why AI and human decision-making are not equally opaque. AI and Ethics, 1-12.
- Ray, S. (2019, February). A quick review of machine learning algorithms. In 2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 35-39). IEEE.
- Risi, S., & Preuss, M. (2020). From chess and atari to starcraft and beyond: How game ai is driving the world of ai. KI-Künstliche Intelligenz, 34, 7-17.
- Shin, D., Zaid, B., Biocca, F., & Rasul, A. (2022). In platforms we trust? Unlocking the black-box of news algorithms through interpretable AI. Journal of Broadcasting & Electronic Media, 66(2), 235-256.
- Zhang, B., Zhu, J., & Su, H. (2023). Toward the third generation artificial intelligence. Science China Information Sciences, 66(2), 1-19.


## Declarations


**Funding:** No specific funding was received for this work.

**Potential competing interests:** No potential competing interests to declare.