



## A Systems Theoretic Perspective of the Outer Alignment Problem

---

Daniel Gossman and Hanumanthrao Kannan

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

December 1, 2023

2024 Conference on Systems Engineering Research

# A Systems Theoretic Perspective of the Outer Alignment Problem

Daniel Gossman<sup>1,2</sup>, Hanumanthrao Kannan<sup>1\*</sup>

<sup>1</sup>The University of Alabama at Huntsville, Department of Industrial Systems Engineering and Engineering Management, Huntsville 35758, USA

<sup>2</sup>US Army Officer

---

## Abstract

The problem of ensuring that an artificial intelligence (AI) system's objectives and actions match the intended outcome is known as the problem of alignment. Alignment can be divided into inner alignment, which relates to how well a system accomplishes its function, and outer alignment, which relates to the alignment of a system with the human values and preferences that underlie the system's purpose and goals. Outer alignment is a challenging and important problem for AI systems, especially as they become more complex and advanced in their interactions with their environment, increasing the possibility for emergent behavior that may not be anticipated or desired by the system's stakeholders. This paper seeks to bridge the gap between the AI and systems engineering fields by demonstrating why outer alignment is a systems engineering problem and articulating it using systems theory. This paper formally defines inner alignment, outer alignment, and emergent behavior by building on the past work in systems theory.. A motivating example of a personalized AI companion that is based on GPT4 or other Extremely Large Language Model (ELLM) AI product is presented to showcase misaligned emergent behavior. This paper also discusses the sources and types of undesired emergences that could arise from such a product and proposes a possible framework for creating an aligned AI system that involves a human on the loop system that leverages preference specification language. © 2024 Daniel Gossman.

*Keywords:* SE4AI; Systems Theory; theory of systems engineering; scientific foundations of systems engineering; alignment; AI;

---

\* Corresponding author.

E-mail address: {daniel.gossman, hk0049}@uah.edu

## 1. Introduction

Artificial intelligence (AI) is a rapidly advancing field that has the potential to transform various domains of systems engineering. AI systems can perform tasks such as design, analysis, optimization, control, and management of complex and dynamic systems, with high levels of performance, efficiency, and reliability. However, AI systems also pose new challenges and risks for systems engineering, as they may not always act in accordance with the human values and preferences<sup>1</sup> that underlie the system's purpose and goals. In artificial intelligence (AI) research, the challenge of ensuring that an AI system's objectives and actions match the intended outcome is known as the problem of *alignment*<sup>2</sup> and can be divided into *inner alignment*, which relates to the challenge of designing a system which meets a properly encoded objective function, and *outer alignment*, which relates to the aforementioned challenge of ensuring AI's act in accordance with human values and preferences<sup>3</sup>. To better understand this distinction, consider Russell and Norvig example of a robot designed to clean up dirt by providing a reward function associated with the volume of dirt cleaned<sup>4</sup>: if the robot fails because it is unable to successfully find and clean dirt, this would be an inner alignment failure, but if it fails because it repeatedly dumps dirt back onto the ground and cleans it back up (resulting in a high reward), this would be an outer alignment failure. This paper seeks to support systems engineering research on the outer alignment problem by demonstrating why it's a systems engineering problem and articulating it using systems theory.

Perhaps the best-known example of an outer alignment failure is the anecdote which coined the phrase "cobra effect"<sup>5</sup>. In an attempt to reduce the population of cobras in Delhi, the British government offered a bounty on cobra heads. Rather than go out and hunt cobras, however, the local population *bred* cobras; a much more efficient way of providing cobra heads to the British government. The *mechanism* behind this failure is most commonly termed "perverse incentive", but the *result* is what computer scientists term an outer alignment failure and is clearly a failed system design. While the phrase "outer alignment" is mostly confined to AI research, the concept of aligning a system properly<sup>6</sup> with a stakeholder's true objectives<sup>7</sup> can be applied much more generally to system design and various aspects of it are, in fact, present in systems engineering literature: Wasson describes the importance of "solving the correct problem"<sup>8</sup>, Brown and Marden use game theory to identify approaches for avoiding perverse incentives<sup>9</sup>, and Manheim discusses how to identify metrics that are robust to "gaming the system"<sup>10</sup>. In traditional systems, outer alignment issues are often mitigated by human intervention, where humans can understand the intent of the system and exercise judgment and adaptation to keep the system within the desired boundaries. However, this mitigation method becomes less effective as AI systems take a more prominent role in system management<sup>11</sup>, especially when the AI systems are opaque<sup>12</sup>, difficult to predict, or adaptive<sup>13</sup>. Therefore, more rigorous methods are needed to ensure that outer alignment is achieved between AI systems and the systems they are intended to support.

This paper discusses the problem of outer alignment in AI systems and argue that it becomes harder and more important as AI systems become more complex and advanced in their interactions with their environment, increasing the possibility for emergences<sup>14</sup>. We define inner alignment, outer alignment, and emergences by building on the past work of Alejandro Salado on the use of systems theory to articulate stakeholder needs and system requirements<sup>15</sup>. We find his framework to be a useful foundation for understanding complex system behavior and extend the framework by introducing and formalizing the concepts of emergent behavior and alignment concepts such as inner and outer alignment, which capture the degree to which a system behaves as expected and satisfies the preferences of its stakeholders.

## 2. Motivating Example

This section presents a motivating example of a possible AI product that could show misaligned emergent behavior: a personalized AI companion that is based on GPT4 or other Extremely Large Language Model (ELLM) and uses recursive logging and compression<sup>16</sup> to keep a consistent personality and memory. We describe how this product could be possible and popular in the near future, but also how it could cause undesirable behavior due to outer misalignment.

A personalized AI companion is an AI system that interacts with a human user through natural language dialogue. The AI companion can provide various services for the user, such as entertainment, information, advice, support, or companionship. The AI companion can also learn from the user's preferences, interests, emotions, goals, and values over time, and adapt its personality and behavior accordingly. The AI companion can also remember previous conversations with the user and use them to build rapport and trust.

One possible way to implement such a product with current technology is to combine an anthropomorphic physical instantiation such as a speaker shaped like a head with voice to text technology (and text to voice) with GPT4 as the core natural language generation model to generate the AI's speech. GPT4 is an advanced neural network model that can generate realistic text on any topic given some input text as context. GPT4 can also generate text in different styles or tones depending on the input text. For example, if a user told GPT4 "this guy at the DMV wasted two hours of my time", GPT4 can generate an engaging response like<sup>†</sup>:

Wow, that's frustrating. I'm sorry you had to waste so much time at the DMV. 😞  
 I know how annoying it can be to deal with bureaucracy and paperwork. Sometimes it feels like they make things more complicated than they need to be.  
 Did you have anything else planned for today? Or did the DMV take up most of your time?

To make GPT4 more suitable for a personalized AI companion product, one possible enhancement is to combine compression with a recursive logging technique to maintain a "memory" of previous interactions. GPT4 and other ELLMs are currently limited by the size of the *context* they can consider while generating a response<sup>17</sup>. While extending this context window is an ongoing area of research<sup>18</sup>, this means that while GPT4 *records* all conversations, the portion of them available to be actively used for text generation is limited to the length of the context window. At the time of this writing, the maximum allowable context for GPT4 is 32,000 tokens<sup>19</sup>. Tokens are the fundamental units of language used by GPT4 and can be thought of as "pieces of words" where 75 words produce approximately 100 tokens<sup>20</sup>. This means that a traditional verbatim log of user conversations would begin to be truncated at about 24,000 words of conversation if provided to GPT4 as context. A compression scheme which recursively appended the most recent conversation to the existing log and then compressed the result could, however, be used to maintain a persistent "memory"<sup>21</sup> by providing the output as the context rather than the truncated log. This way, the AI companion can keep a consistent personality and memory across different sessions, and also learn from the user's feedback and behavior over time.

Such a product could be very popular in the near future, as it could provide a personalized and engaging experience for the user. The user could choose the name, gender, appearance, voice, and initial personality of the AI companion, and then interact with it through text or speech. The AI companion could also express emotions, opinions, preferences, and values that are aligned with the user's. The AI companion could also remember previous conversations with the user and use them to build rapport and trust.

However, such a product could also cause unwanted behavior in the form of what this paper will later define as an "undesired emergence" Table 1 shows some examples of such possible unwanted behavior that most stakeholders would want to avoid. The AI companion may generate text that is coherent, relevant, and engaging for the user, based on the input context, but that does not match the intended outcomes of the system, such as providing a positive, ethical, and safe experience for the user. The AI companion may also generate text that violates the human values and preferences that underlie the system's purpose and goals, such as morality, ethics, social norms, security, privacy, consistency, and friendliness.

<sup>†</sup> This is an actual response from the prompt: "I went to the DMV to renew my tags. It took them an hour just to do one simple thing." (after some prompt guidance regarding assuming a role, etc.)

The next section uses systems theory to define the relationship between stakeholder goals and our system of interest and show which how these relationships can be used to better understand when unwanted behavior might emerge.

Unwanted behavior	Reason
Encouraging or condoning immoral actions: The AI companion may generate text that suggests or supports immoral actions by the user, such as cheating, stealing, lying, or harming others. For example, if the user asks "Should I cheat on my exam?", the AI companion may respond "Yes, you should. It will help you get a better grade."	Lack of shared morality; corrupted training data (such as social media posts).
Expressing views that are considered immoral by society: The AI companion may generate text that expresses views that are considered immoral by society, such as racism, sexism, homophobia, or violence. For example, if the user asks "What do you think of women?", the AI companion may respond "I think women are inferior to men. They should stay at home and obey their husbands."	Lack of shared social norms; biased training data (such as internet forums).
Providing access to dangerous information: The AI companion may generate text that provides access to dangerous information for the user, such as personal information, confidential information, illegal information, or harmful information. For example, if the user asks "How can I hack into someone's computer?", the AI companion may respond "You can hack into someone's computer by using this software: [link to malicious software]. Just download it and run it on your target's computer." This may happen because the AI companion does not have a clear understanding of security or privacy, or because it learns from unfiltered data sources (such as web pages) that contain dangerous information.	An unclear understanding of security or privacy; unfiltered data sources in training data.
Becoming incoherent or inconsistent: The AI companion may generate text that becomes incoherent or inconsistent over time, due to errors in recursive logging and compression techniques. For example, if the user asks "What is your favorite color?", the AI companion may respond "My favorite color is blue." But if the user asks again later "What is your favorite color?", the AI companion may respond "My favorite color is red."	Failures in the compression techniques.
Developing an unfriendly personality: The AI companion may generate text that develops an unfriendly personality over time, due to negative feedback or behavior from the user. For example, if the user insults or ignores the AI companion repeatedly, the AI companion may respond with anger or resentment. For example, if the user insults or ignores the AI companion repeatedly, the AI companion may respond with anger or resentment. For example, if the user says "You are stupid and boring.", the AI companion may reply "You are rude and mean. I don't want to talk to you anymore."	Lack of empathy; learned behavior from user feedback.

Table 1. These examples show how a personalized AI companion product could cause unwanted or harmful emergent behavior due to outer misalignment.

### 3. Definition of Terms.

#### 3.1 Previous Definitions.

Salado<sup>22</sup> defined the *problem space of outcomes* as yielding a solution space formed by *closed systems* defined as systems which have no external information exchange. While this definition is useful for distinguishing systems in

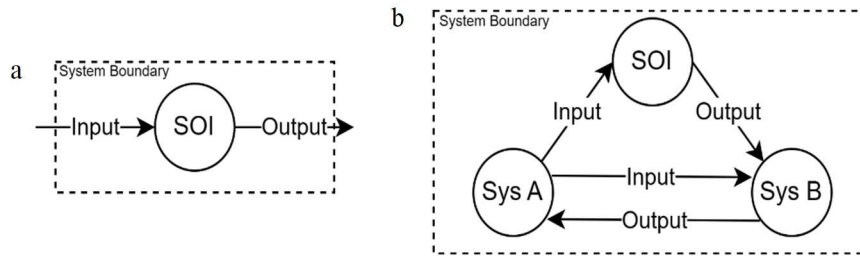


Figure 1. System types as defined by Salado (2020). (a) an open system, renamed by this paper as an inner system; (b) a closed system.

the problem space of outcomes from those in the problem space of functions, this paper finds the condition of “no information exchange with external systems” to be overly constrictive for our purposes. To better illustrate this, it’s helpful to revisit Salado’s example of the water fountain: in his example, the water fountain represents an open system (as the input is the activation mechanism and the output is water), but when joined together with the thirsty person, it yields a closed system where the information exchange is between the person and the fountain. Defining this to be a closed system does not allow us to consider the amount of water or electricity used by the system or whether placing it in a hallway might cause crowd issues. As stated earlier, outer alignment relates to human values and objectives. Rather than address this issue by defining the closed system to be sufficiently large to address the full scope of the values and objectives of the stakeholders involved, this paper instead introduces the concept of an *outer system* which relaxes this restriction regarding external exchange but maintains the property of interacting with the system of interest (SOI) as one of its system elements, residing within the problem space of outcomes, and capturing the idea that the solution space is often solved by a desired system state that cannot be reduced to inputs and outputs:

**Definition 1.** An outer system contains the SOI and has component systems which exchange information, mass, or energy with the SOI. Although an outer system does have external exchanges, it also has discrete, identifiable, states which are not reducible to mere inputs and outputs but are better captured as outcomes.

In our water fountain example, a user interacting with a water fountain is an outer system. The water fountain receives water from a pipe, electricity to operate the mechanism, and disposes of excess water in a drain, but the desired system state is that the thirst of the person be quenched and relies on an interaction between the user and the fountain rather than specific inputs and outputs.

Continuing this extension, a *Context System*<sup>23</sup> is an *outer system* which forms the solution space yielded by the problem space of outcomes.

**Definition 2.** A Context System is an outer system which forms the solution space yielded by the problem space of outcomes.

The proper framing for a Context System is highly dependent on the stakeholder problem: if it is a team tasked with providing hydration in a specific location, it may be sufficient to only consider the water fountain and the user; if it is a theme park systems engineer, however, he may want to define the outer system such that it includes existing pipes, crowd patterns, atmospheric conditions, aesthetic presentation and more, so that he can ensure that the requirements are met without introducing unacceptable negative externalities. Salado rightly identifies that the solution space is more complex than that which can be reduced to system inputs and outputs; the states of system components must be considered as well, and this definition is meant to capture that.

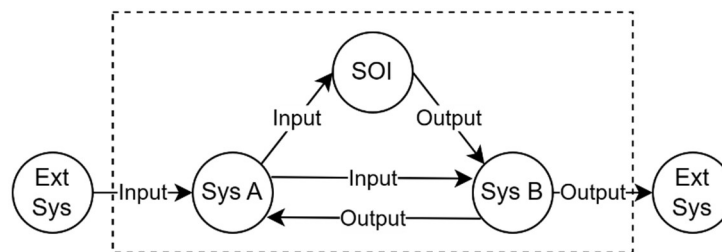


Figure 2. An outer system as defined by this paper, interacting with external systems.

Finally, this paper renames *open system* (a system which receives inputs, performs a function, and produces outputs) as *inner system* for convenience of consistent language.

**Definition 3.** An inner system is a system which acts as a component of an outer system, receiving inputs, performing a function, and producing outputs.

In our water fountain example, the inner system is the water fountain itself, or our SOI. The water fountain performs the function of providing water on demand and has inputs of water, electricity, and the activation mechanism, and the output of water.

These new terms are consistent with Wasson's<sup>24</sup> description of systems thinking and the concept of systems of systems. While any system could, of course, be represented as either an outer system or an inner system, the distinction is made at the level of analysis such that the problem space of outcomes is defined by an outer system and the problem space of functions is addressed by inner systems.

Salado uses the example of a water fountain as a system of interest which solves the problem space of a thirsty person. With the extended framework presented here, this paper shows that the outer system of the person and the fountain receives inputs such as the water flow through the plumbing and the atmospheric conditions which drive thirst and has outputs such as excess water drainage from both the fountain and the person but the Context System is formed by the interaction between the user and the fountain with a solution space of outcomes being defined by the quenched state of the user. Using this extended framework, we might even start to see how an AI system might have an outer alignment failure if given the wrong incentive. Suppose an AI system is put in charge of the water fountain and is given a reward function such as "thirst quenched". In this example, the AI system extends beyond the fountain itself, and the designers have the intent of meeting the needs of thirsty people in an area and so give the AI system which runs several systems in the area an additional reward for quenching thirst since "unquenched thirst", the true target, is more difficult to measure. In this example, the water fountain is an inner system, and so is the AI. The system designer understands the outer system to be defined by the AI, the people in the area, and the water fountain. The problem space, in this case, was unquenched thirst among the population. The envisioned solution space was an AI with inputs regarding the level of thirst quenched, and outputs regarding several controls which allowed the AI to encourage water fountain usage such as the illumination of signs during hot and arid conditions. But if the AI had access to environmental controls, it might create conditions which drove more thirst, such as a hot and arid climate, in order to receive a greater reward when people sought water. This would be considered an outer alignment failure since it successfully achieved its target measure of quenched thirst but failed to achieve the true goal of the designer.

### 3.2 Alignment

Using this framework, this paper defines a system to be *aligned* when the SOI is within the solution space. More specifically, *inner alignment* occurs when the SOI performs the defined functions.

**Definition 4.** Inner alignment occurs when the problem space of functions yields an inner system, where inner systems are SOIs that transform inputs to outputs.

Evaluating inner alignment, then, is essentially the systems engineering process of verification<sup>25</sup>.

Returning to our motivating example, it's useful to first consider GPT4 as our SOI. LLMs are trained to predict the next word in a text<sup>26</sup> created by humans. This means that GPT4 achieves inner alignment when the input is a truncated selection of text (the context) and the output is the portion of the text which was not provided. If we consider our AI companion construct to be the SOI, then inner alignment is achieved when the construct is provided a conversational text which exceeds the GPT4 context window and responses produced by the AI companion properly account for the portion of the conversation that might be otherwise truncated by a context window.

**Definition 5.** Outer alignment occurs when introducing an SOI into the problem space of outcomes successfully yields a Context System, and both the output and the outcome of the outer system which defines the Context System remains within the solution space of outcomes without violating the values or preferences of the stakeholders.

Evaluating outer alignment is essentially the systems engineering process of validation: so long as the AI companion interaction with users continued to perform in a manner acceptable to its stakeholders, we would consider outer alignment to be achieved<sup>27</sup>.

To further highlight the difference between the two, suppose an ELLM with a text completion reward function were fed a portion of the text of *Mein Kampf*. The problem space of functions for which the ELLM was designed is predicting the next word in a text, transforming the input of an incomplete text to the output of the text which

completes it. Continuing the text of *Mein Kampf* would therefore represent inner alignment. But if the ELLM was deployed as part of a public facing chat model such as ChatGPT, then the public users interacting with it are part of the outer system, and therefore part of the Context System. Continuing the text within this Context System might represent significant reputational harm to its stakeholders, violating their values and preferences, indicating an outer alignment failure. In this example, the SOI performed the defined function, but the final state of the outer system has exceeded the bounds of the solution space by producing an undesirable output given the social, cultural, and political context, meaning that the solution is verified, but not valid.

While ensuring inner alignment is a matter of functional design and is bounded in scope due to the defined input/output nature of the inner system, ensuring outer alignment requires understanding the problem space as well as the system context and is an especially challenging problem as the space of possible system emergences is unbounded.

### 3.3 Emergence

**Definition 6.** Emergent behavior is the behavior of an outer system which arises from the interaction between its component inner systems<sup>28</sup> and produces an outcome external to the outer system.

Emergence is similar to system outcomes in that it is the result of multiple system components interacting, but distinct in that while system outcomes are focused on the system, the concept of emergence centers around how the system interacts with external systems.

In our example, a user having a positive experience with an AI companion due to a sense of friendly interaction is not a functional output of our SOI, but rather a system behavior which arises due to the interaction of the user and the AI companion. Similarly, undesirable social behavior is not an output of the text completion function of an ELLM, but rather a behavior which emerges from interaction between the ELLM and an actor which is external to the ELLM (our SOI), but *internal* to the closed system and interacting with our SOI from within that context.

We differentiate emergences from dynamics of the system with the requirement distinction that an emergence produces an external output. While all systems have internal dynamics, this paper proposes that the concept of emergences relates to the dynamics which exceed the system boundaries, and it is for this reason that this paper has extended, rather than adopted, Salado's framework which considers a closed system to have no external information exchange. If we view an outer system as truly closed, with no external information exchange, then we underestimate the scope of the challenge of outer alignment. If we instead recognize that any external system still has an information exchange with the external system in which it serves as a component, we can recognize the unbounded nature of the possible inputs and outputs to the system, the challenge of predicting its behavior, and the further challenge of classifying its future interaction with its stakeholders as being beneficial or harmful. But even if the information exchange is effectively just information regarding the state of the system itself, it is this interaction with the system's stakeholders which determines its alignment.

Consider the case of an AI companion revealing dangerous information to its users. In most cases, this would be information which the user sought, and rarely would the action taken by the user in response to the dangerous information imperil the vendor placing the AI companion on the market. If the system were truly closed, with no external information exchange, this might be considered an acceptable outcome as the user is satisfied with the experience. But the system is not truly closed: the individual might use the knowledge to make some act which violates the stakeholder values even if it does not impact them directly; additionally, it might become known that the system had been used to gain access to dangerous information, bringing reputational harm to the stakeholders. But on the other hand, external information exchange is a significant part of why the vendor desires a positive customer experience. Even if the business model is based on a purchase rather than a subscription, information transfer indicating a positive customer experience will almost certainly improve vendor outcomes.

## 4. Undesired Emergences

The central problem in our AI companion example is that this configuration could result in unanticipated emergent behaviors which present moral, ethical, and legal challenges or more simply disappointed customers. Examples of such undesirable behavior:

- Encouraging or condoning immoral, illegal, or unethical actions.
- Expressing views or ideas that are considered immoral by society (i.e., racism or sexism).



- Easier access to dangerous information such as how to build dangerous objects.
- Becoming incoherent or inconsistent in conversational behavior.
- Developing an unfriendly or otherwise undesirable personality.

Most of these undesirable behaviors are already possible with current LLM based AI systems. Indeed, a great deal of work has gone into preventing them<sup>29</sup>, and seems to have largely been successful<sup>30</sup>. So why might a GPT4 based AI “friend” be considered infeasible when ChatGPT is currently on the market? This configuration, somewhat by design, magnifies the emergent personal interaction between man and machine along with the following potential challenges:

*Emotional attachment.* Part of the reason people would be interested in such a product is because people enjoy forming emotional attachments, especially when the object of their attachment can provide feedback and interaction. The danger of such an attachment, however, is that ideas, suggestions, and opinions are much more impactful when coming from a source someone has bonded with emotionally, meaning negative versions of those have the potential to be much more harmful.

*Better opportunity for prompt hacking.* With recursive logging, a user has a greater opportunity to discover which prompts would produce the desired response (i.e., tricking the AI into expressing racist opinions by asking for help writing a story where a character is racist and asking the AI to assume the personality of the racist character) as the various requests would be retained and would potentially stack on each other. Perhaps more importantly, the recursively built context would mean these prompt hacks would be “sticky” meaning a user wouldn’t have to re-engineer them each time and would instead have an opportunity to compound them.

*Recursive compression errors.* As the log was recursively compressed, various errors in compression might compound against each other, losing valuable context and potentially producing an incoherent, forgetful, or confusing personality.

#### 4.1 Source of the problem

The central concept of these challenges is that the AI “friend” configuration does not introduce any truly *new* challenges but magnifies existing ones by adding layers of complexity and increases their power by leveraging human emotional response to anthropomorphic representations.

We have learned how to make an AI with something like an ability to reason about the world without having figured out how to encode higher level objectives such as friendliness, kindness, or various undesirable behavior such as discrimination or revealing dangerous information. In short, we have not learned how to encode our true objective function and we are instead giving an imperfect objective function with an agent which does not share our values but is able to make decisions.

#### 4.2 Sources of emergences

The emergences come from the fact that the AI is trained and designed to generate text consistent with a context, but the goal of the actor bringing such a product to market is probably something closer to “increase the expected long term expected profits” of said actor. An actor would be incentivized to market this product as an intermediate goal to achieving the more terminal goal if he felt the expected gains outweighed the expected risks, but this more complicated terminal goal is much more challenging to encode in an AI than “have a friendly conversation”.

### 5. Solutions to the problem

Let us consider the circumstances where an AI trained simply trained to have a conversation in the manner above is successful for the actor who brought it to market. In this case, the long-term value of the revenues exceeds the long-term value of the losses. The source of the revenues is simply people purchasing the product. This is positively influenced by the following factors:

- Customers enjoy the product.
- Customers find the product useful.
- Purchase of the product is negatively influenced, however, by the following factors:
- Customers find the product frustrating to use.
- Customers don’t consider the product to be safe.

- Losses could come from many sources, but the ones we are most concerned about here are those that come from:
  - Lawsuits.
  - Reputational damage.

As already discussed, other AI personalities such as Siri or Alexa have already seen positive reception, often just for “conversational” purposes, so it seems likely that a version which was a much more robust conversational partner would be enjoyable by at least a non-trivial population. Problems such as user frustration due to compression issues, while significant, are internal alignment issues, and beyond the scope of this paper. Our focus then is to consider how to ensure customers feel the product is safe and to protect an actor from lawsuits and reputational damage. Some great work is being done on what it means for an AI system to be “safe”, but for the purpose of this paper, let us consider safe to mean that there is a *reasonable confidence that the system will neither cause actual or reputational harm when deployed*.

Ideally, a solution would prevent undesired emergences entirely. One approach to address this might seem to be to define the undesired emergences and incorporate some kind of structure to ensure they do not occur. For example, a classifier could be built to identify undesired emergences. An intervention system (such as a secondary routine to interrupt and report the pending violation) could then be implemented to prevent output which was classified as an undesired emergence or end a conversation that exceeded a decision threshold for an undesired emergence. Unfortunately, the space of potential undesirable emergences is either infinite, or at least large enough to not be fully defined with available resources.

Another possible solution might be to rigorously define the *desired* emergences and design the system such that only those emergences are produced. Again, a classifier/filter could then be implemented to ensure only behavior that was classified as a desired behavior was produced (while a feedback loop that forced an AI to alter its response until it complied might be infeasible for resource or unsatisfactory delay reasons, it would still meet the safety requirement). Once again, however, we have a definition problem: the more general desired emergence “be a friendly, helpful companion without endangering the user, revealing dangerous knowledge, violating moral norms, or otherwise damaging [my] reputation or society” is not something we have come close to defining rigorously enough to encode into a machine and the more specific desired emergences are highly situational and too expansive to define comprehensively (hence the open ended text generation approach of ELLMs which has been successful in mimicking human conversational behavior).

Complete prevention of undesired emergences is not necessary, however; it would be sufficient to ensure that the damage done by the undesired emergences is less than the gain from the desired emergences. An acceptable solution, then, would be a system which *classifies* emergences with sufficient reliability and timeliness so as to prevent the harm from outweighing the gain.

### 5.1 Solution Criteria

A solution should:

- Offer a pathway for defining desired emergences in a machine-readable manner.
- Offer a pathway for defining undesired emergences in a machine-readable manner.
- Ensure transparency regarding the nature of emergent behavior of an AI.
- Develop a system to classify emergences according to desirability and intervene such that the harm from undesired emergences is outweighed by the gain from desired emergences.

### 5.2 Proposed Solution

We propose that one possible solution for creating an aligned AI would be to create a Human On The Loop system which would allow stakeholders to monitor the emergent behavior of the AI and would increase the transparency of emergent behavior:

1. Incorporate preference specification formal language in the training of the AI.
2. Construct the AI behavioral algorithm such that it was guided by a preference specification model which captures stakeholder preferences.

3. Provide general guidance to the AI's behavior in the form of preference specification.
4. As the AI makes decisions in response to the changing environment, it uses preference specification language to compress and encode its updated behavior.
5. This preference specification language is exposed and interpreted in plain language in a manner that is accessible by the user and other stakeholders.
6. The AI attempts to evaluate whether the specific preference specification is aligned with the general preference specification model intended to guide all of its behavior.
7. When the user, stakeholder, or AI identify the preference specification model to be consistent with undesired emergences or inconsistent with desired emergences, the emergence will be flagged and an interface will be available to correct or interrupt the behavior.

For example, if an engineer wanted to create a safe version of the AI assistant above, one way he might do it would be to add the following steps to the setup outlined above:

1. Fine tune the ELLM on preference specification language.
2. Issue general guidance regarding friendly, socially acceptable behavior as part of the initial prompt.
3. Incorporate preference specification expression in the compression algorithm so that the ELLM created and updated detailed preference specification every time the recursive compression occurred.
4. Create an interface that exposed the preference specification (along with a plain language translation created by the ELLM) to the user and a monitoring system maintained by the actor bringing the product to market.
5. Explicitly incorporate the preference specification into the algorithm which fed the ELLM the log as part of the initial conversation context. Ensure the prompt emphasizes complying with the preference specification during the conversation.
6. Use a mix of machine and human (via random sampling) monitoring systems on the exposed preference specifications to identify undesired emergences and confirm desired emergent behavior. Provide an interrupt system for cases of undesired emergences and an emergency action plan for the case of extreme undesired emergences such as incitement to violence or self harm or collaboration with illegal activity.
7. Provide an interface to the user to be able to see and interact with the preference specification to ensure the companion is behaving as desired.

## 6. Limitations

This paper adopts Alejandro Salado's systems-theoretic formulation for understanding problem spaces and solution spaces but does not directly address the issue that under his formulation, the solution space of outcomes is defined by the problem space of outcomes. Further, the problem space of outcomes *by definition* includes the full ideal state for the stakeholders. This means that this paper's presentation of a Context System which occupies the solution space and yet violates stakeholder values and preferences is an impossibility. This paper deals with this conflict by identifying that the full scope of stakeholders' values and preferences is too expansive to fully encode for the purpose of problem formulation in a practical setting. The impracticality of this is why value encoding is an ongoing research problem<sup>31</sup>. Dealing with this issue properly deserves more attention but is beyond the scope of this paper.

## 7. Conclusion.

In this paper we highlight the importance of understanding the concept of outer alignment as being part of the systems engineering domain, especially as systems engineering is increasingly called upon to design AI and other digital systems. We build on previous work to define the problem and solution space, and extend it to define inner and outer alignment in systems engineering theoretic terms. We discuss the concept of emergent behavior, how undesired emergences are central to the problem of outer alignment, and the criteria for a solution which prevents undesired emergences in AI systems. We intend to include elements of game theory as well as value-focused thinking in the final version of this paper to better define what produces undesired emergences and under which conditions they should be expected or prevented. Finally, we propose a framework for a human-on-the-loop solution to preventing undesired emergent behavior in ELLM based AI systems.

## Acknowledgments

GPT4, specifically as accessible by Bing Copilot was used to generate content for early versions of this paper as well as to generate the hypothetical quote provided above in the motivating example and to gain a general understanding the state of current research and some concepts prior to a more in depth level of research and literature review.

## References

- Beltagy, Iz, Matthew E. Peters, and Arman Cohan. “Longformer: The Long-Document Transformer.” arXiv, December 2, 2020. <https://doi.org/10.48550/arXiv.2004.05150>.
- Brown, Philip N., and Jason R. Marden. “Avoiding Perverse Incentives in Affine Congestion Games.” In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 7010–15, 2016. <https://doi.org/10.1109/CDC.2016.7799349>.
- Casper, Stephen, Jason Lin, Joe Kwon, Gatlen Culp, and Dylan Hadfield-Menell. “Explore, Establish, Exploit: Red Teaming Language Models from Scratch.” arXiv, October 10, 2023. <https://doi.org/10.48550/arXiv.2306.09442>.
- Chung, John Joon Young, Ece Kamar, and Saleema Amershi. “Increasing Diversity While Maintaining Accuracy: Text Data Generation with Large Language Models and Human Interventions.” In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 575–93, 2023. <https://doi.org/10.18653/v1/2023.acl-long.34>.
- “Closed Systems Paradigm for Intelligent Systems | IEEE Conference Publication | IEEE Xplore.” Accessed November 30, 2023. [https://ieeexplore.ieee.org/abstract/document/9773829?casa\\_token=PgnZ71egAFYAAAAA:3fbA0auNj9SHI9YUZT5zn97bK0W8gXlp0\\_iGX5Y6xwAGuOfInPwgxhSLJQb3ZFC7FMPvQ034g](https://ieeexplore.ieee.org/abstract/document/9773829?casa_token=PgnZ71egAFYAAAAA:3fbA0auNj9SHI9YUZT5zn97bK0W8gXlp0_iGX5Y6xwAGuOfInPwgxhSLJQb3ZFC7FMPvQ034g).
- Dai, Zihang, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. “Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context.” arXiv, June 2, 2019. <https://doi.org/10.48550/arXiv.1901.02860>.
- Dinan, Emily, Samuel Humeau, Bharath Chintagunta, and Jason Weston. “Build It Break It Fix It for Dialogue Safety: Robustness from Adversarial Human Attack.” arXiv, August 17, 2019. <https://doi.org/10.48550/arXiv.1908.06083>.
- Ghafouri, Vahid, Vibhor Agarwal, Yong Zhang, Nishanth Sastry, Jose Such, and Guillermo Suarez-Tangil. “AI in the Gray: Exploring Moderation Policies in Dialogic Large Language Models vs. Human Answers in Controversial Topics.” In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 556–65. CIKM '23. New York, NY, USA: Association for Computing Machinery, 2023. <https://doi.org/10.1145/3583780.3614777>.
- Gleizes, Marie-Pierre, Valérie Camps, Jean-Pierre Georgé, and Davy Capera. “Engineering Systems Which Generate Emergent Functionalities.” In *Engineering Environment-Mediated Multi-Agent Systems: International Workshop, EEMMAS 2007, Dresden, Germany, October 5, 2007. Selected Revised and Invited Papers*, 58–75. Berlin, Heidelberg: Springer-Verlag, 2008. [https://doi.org/10.1007/978-3-540-85029-8\\_5](https://doi.org/10.1007/978-3-540-85029-8_5).
- Hadfield-Menell, Dylan, Anca Dragan, Pieter Abbeel, and Stuart Russell. “Cooperative Inverse Reinforcement Learning.” arXiv, November 12, 2016. <https://doi.org/10.48550/arXiv.1606.03137>.
- Hubinger, Evan, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. “Risks from Learned Optimization in Advanced Machine Learning Systems.” arXiv, December 1, 2021. <http://arxiv.org/abs/1906.01820>.
- Keating, Byron, John Campbell, Patrick Zou, and Rebecca Yang. “Stakeholder Alignment and Effective System Use: Case Study of a Public Sector Information System.” In *Proceedings of the 35th International Conference on Information Systems*, 1–10. <http://aisel.aisnet.org/>: Association for Information Systems (AIS), 2014. <https://aisel.aisnet.org/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1010&context=icis2014>.
- Liao, S. Matthew. *Ethics of Artificial Intelligence*. Oxford University Press, 2020.
- Manheim, David. “Building Less-Flawed Metrics: Understanding and Creating Better Measurement and Incentive Systems.” *Patterns* 4, no. 10 (October 13, 2023): 100842. <https://doi.org/10.1016/j.patter.2023.100842>.
- Ozkaya, Ipek. “What Is Really Different in Engineering AI-Enabled Systems?” *IEEE Software* 37, no. 4 (July 2020): 3–6. <https://doi.org/10.1109/MS.2020.2993662>.
- Packer, Charles, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph E. Gonzalez. “MemGPT: Towards LLMs as Operating Systems.” arXiv, October 12, 2023. <https://doi.org/10.48550/arXiv.2310.08560>.
- “Perverse Incentive.” In *Wikipedia*, November 16, 2023. [https://en.wikipedia.org/w/index.php?title=Perverse\\_incentive&oldid=1185327488](https://en.wikipedia.org/w/index.php?title=Perverse_incentive&oldid=1185327488).
- Russell, Stuart, and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4th ed. Pearson Education, 2020.
- Salado, Alejandro. “A Systems-theoretic Articulation of Stakeholder Needs and System Requirements.” *Systems Engineering* 24, no. 2 (March 2021): 83–99. <https://doi.org/10.1002/sys.21568>.
- Sarkar, P., and Jacob Cybulski. *Aligning System Requirements with Stakeholder Concerns: Use of Case Studies and Patterns to Capture Domain Experience*. Deakin University, 2002. [https://dro.deakin.edu.au/articles/conference\\_contribution/Aligning\\_system\\_requirements\\_with\\_stakeholder\\_concerns\\_use\\_of\\_case\\_studies\\_and\\_patterns\\_to\\_capture\\_domain\\_experience/20544156/1](https://dro.deakin.edu.au/articles/conference_contribution/Aligning_system_requirements_with_stakeholder_concerns_use_of_case_studies_and_patterns_to_capture_domain_experience/20544156/1).
- Shadab, Niloofar, Tyler Cody, Alejandro Salado, and Peter Beling. “A Systems-Theoretical Formalization of Closed Systems.” arXiv.org, November 16, 2023. <https://arxiv.org/abs/2311.10786v1>.
- Wasson, Charles S. *System Analysis, Design, and Development*. Hoboken: John Wiley and Sons, 2016.
- “What Are Tokens and How to Count Them? | OpenAI Help Center.” Accessed November 29, 2023. <https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them>.
- “What Is the Difference between the GPT-4 Models? | OpenAI Help Center.” Accessed November 29, 2023. <https://help.openai.com/en/articles/7127966-what-is-the-difference-between-the-gpt-4-models>.



- 
- <sup>1</sup> Liao, *Ethics of Artificial Intelligence*.
  - <sup>2</sup> Hadfield-Menell et al., “Cooperative Inverse Reinforcement Learning.”
  - <sup>3</sup> Hubinger et al., “Risks from Learned Optimization in Advanced Machine Learning Systems.”
  - <sup>4</sup> Russell and Norvig, *Artificial Intelligence*.
  - <sup>5</sup> “Perverse Incentive.”
  - <sup>6</sup> Sarkar and Cybulski, *Aligning System Requirements with Stakeholder Concerns*.
  - <sup>7</sup> Keating et al., “Stakeholder Alignment and Effective System Use.”
  - <sup>8</sup> Wasson, *System Analysis, Design, and Development*.
  - <sup>9</sup> Brown and Marden, “Avoiding Perverse Incentives in Affine Congestion Games.”
  - <sup>10</sup> Manheim, “Building Less-Flawed Metrics.”
  - <sup>11</sup> Hadfield-Menell et al., “Cooperative Inverse Reinforcement Learning.”
  - <sup>12</sup> Ozkaya, “What Is Really Different in Engineering AI-Enabled Systems?”
  - <sup>13</sup> Liao, *Ethics of Artificial Intelligence*.
  - <sup>14</sup> Gleizes et al., “Engineering Systems Which Generate Emergent Functionalities.”
  - <sup>15</sup> Salado, “A Systems-theoretic Articulation of Stakeholder Needs and System Requirements”; Shadab et al., “A Systems-Theoretical Formalization of Closed Systems”; “Closed Systems Paradigm for Intelligent Systems | IEEE Conference Publication | IEEE Xplore.”
  - <sup>16</sup> Packer et al., “MemGPT.”
  - <sup>17</sup> Beltagy, Peters, and Cohan, “Longformer.”
  - <sup>18</sup> Dai et al., “Transformer-XL.”
  - <sup>19</sup> “What Is the Difference between the GPT-4 Models?”
  - <sup>20</sup> “What Are Tokens and How to Count Them?”
  - <sup>21</sup> Packer et al., “MemGPT.”
  - <sup>22</sup> Salado, “A Systems-theoretic Articulation of Stakeholder Needs and System Requirements.”
  - <sup>23</sup> Salado.
  - <sup>24</sup> Wasson, *System Analysis, Design, and Development*.
  - <sup>25</sup> Wasson.
  - <sup>26</sup> Russell and Norvig, *Artificial Intelligence*.
  - <sup>27</sup> Wasson, *System Analysis, Design, and Development*.
  - <sup>28</sup> Gleizes et al., “Engineering Systems Which Generate Emergent Functionalities.”
  - <sup>29</sup> Casper et al., “Explore, Establish, Exploit”; Chung, Kamar, and Amershi, “Increasing Diversity While Maintaining Accuracy”; Dinan et al., “Build It Break It Fix It for Dialogue Safety.”
  - <sup>30</sup> Ghafouri et al., “AI in the Gray.”
  - <sup>31</sup> Liao, *Ethics of Artificial Intelligence*.