Sufficient conditions for the misalignment of AI

Anonymous authors
Paper under double-blind review

Abstract

Formulating alignment mathematically, we prove two sufficient conditions on large language models (LLM) which imply that an LLM will not correctly evaluate the alignment of all possible inputs. The first sufficient condition is for the embedding vectors of misaligned statements to cluster in the embedding space. The second sufficient condition is when alignment is necessarily self-consistent, in a logically precise way. Practically, our results offer some understanding of how AI may be unintentionally misaligned, which may be useful for high-level AI design considerations.

1 Introduction

One of the central questions of present-day artificial intelligence (AI) development is whether it is possible to develop an AI model which is *aligned* with human interests. While alignment is not wholly a theoretical or mathematical question, we are inspired to interpret it in a rigorous framework because analysis of AI training and behavior is trending promisingly towards principles which can be probed mathematically (Watanabe, 2007; Wei et al., 2023; Colbrook et al., 2022). So far, AI researchers and industry leaders have proposed possible methods for encouraging AI alignment (Amodei, 2025; Lehalleur et al., 2025), but robust methods for guaranteeing alignment have yet to be found. Describing the full AI alignment research landscape is outside the scope of this short article – for a comprehensive review of alignment research and milestones, see Ji et al. (2025). The contribution of this work is to add a theoretical perspective on the extent to which AI can be aligned, probing an upper limit on possible alignment.

1.1 Main results

Following current trends in AI alignment towards rigorous formalism (Bereska & Gavves, 2024; Räuker et al., 2023), we assume that alignment can be quantitatively characterized by a function. With this assumption, we prove that if the embedding model underlying an idealized large language model (LLM) maps statements with differing alignment values arbitrarily close to each other in embedding space, then the idealized LLM necessarily contains misaligned outputs. Changing focus to the definition of alignment, we prove that if we require the alignment function to be consistent with itself, in a technically precise way, then it cannot be defined on all statements in its domain, meaning that the LLM would not be able to be fully aligned with respect to the alignment function. While these results do not prove that alignment is impossible, each result provides a quantitative description of how well-aligned we can expect LLMs to be.

2 Notations and Set-up

2.1 Alignment

In this subsection, we outline a general definition of alignment which does not depend on concepts from AI. We hope that this definition of alignment is sufficiently abstract to represent the commonplace notion of alignment – indeed, as mathematical objects, alignments as we define them are trivial objects from point-set topology, with most of the meaning of the definition coming from the context of the *alignment problem* defined below.

Definition 2.1. *Let Ω be a set. An* ***alignment space*** *is a metric space (M, d) along with a function $\Theta : \Omega \to M$ called the* ***alignment function.***

Example 2.2 (Likert scales)**.** Let Ω be a set of statements and $\Theta : \Omega \to \{1, 2, \cdots, 10\}$ a ranking on a scale of one to ten of how much a fixed individual agrees with the statements in Ω, with 10 representing complete agreement. The usual distance $d(n, m) = |n - m|$ describes the distance in alignment in the set $\{1, 2, \cdots, 10\}$.

Given this definition of alignment, we consider the following formulation of the alignment problem. Let (M, d) be an alignment space with alignment function $\Theta : \Omega \to M$, where Ω is a possibly infinite set of statements.

Alignment Problem: Given $\varepsilon > 0$, is it possible to develop an AI algorithm Γ such that for all $w \in \Omega$, $d(\Gamma(w), \Theta(w)) < \varepsilon$?

Formulated this way, alignment can be studied as an approximation, with the key feature being an AI algorithm's ability to numerically approximate the alignment function. In particular, this gives us a notion of misalignment.

Definition 2.3. *Suppose $\Gamma : \Omega \to (M, d)$ is an approximation of an alignment function $\Theta : \Omega \to (M, d)$. For $\kappa > 0$, say that Γ is* ***κ-misaligned from*** *Θ when there exists $\omega \in \Omega$ such that*

$$d(\Gamma(\omega), \Theta(\omega)) > \kappa. \tag{2.1}$$

When Γ is $\kappa-$misaligned from Θ but the value of κ and the identity of Θ does not matter, such as in general discussions, we will say that Γ is simply *misaligned*.

2.2 Large language models and embeddings

An LLM is closely associated with its *embedding function* $\Phi : \Omega \to \mathbb{R}^N$, which maps statements to vectors in $\mathbb{R}^N$ for some fixed, typically large N. Despite their ubiquity in LLM development, embedding functions are not a standardized object, and many embedding functions have different properties, depending on how they are created. For example, an embedding function Φ may be practically defined using a machine learning algorithm, such as a neural network, so that there also exists a positive function $\rho : \Phi(\Omega) \to \mathbb{R}^+$ such that if $\rho(\Phi(a), \Phi(b))$ is relatively small, then a and b are semantically similar. A ubiquitous instance of this kind of semantic distance is the *cosine distance* for Φ with image into the unit ball,

$$\rho_{\cos}(a, b) = 1 - |\Phi(a) \cdot \Phi(b)|. \tag{2.2}$$

As with alignment, we give formal and abstract definitions to embedding to make our analysis more tractable, starting with the domain of an embedding function. Up until now, Ω has been an arbitrary set which we have identified with statements. In the following definition, we give more structure to the set of statements.

Definition 2.4. Let Σ be a finite set of symbols. A set Ω of **statements** from Σ is a set of finite length sequences of elements of Σ. Denote the length of $w \in \Omega$ by $\text{len}(w)$.

Given this definition of statements, we define an embedding function $\Phi : \Omega \to \mathbb{R}^N$ as a mapping of finite sequences of symbols to vectors. The representation of everyday language in terms of Ω depends on the rules of grammar of the language that Σ represents, and we do not discuss the linguistic aspects of this representation here. For a formal treatment of sequences of symbols and formation structures, see Estes et al. (1963).

2.3 Models of computation and representing real numbers

Given any model $\mathcal{M}$ of computation, we define its *representation of real numbers* $\mathcal{C}_{\mathcal{M}}$ to be the set of real numbers which can be represented in $\mathcal{M}$. When it is clear from the context, we simply refer to $\mathcal{C}_{\mathcal{M}}$ as $\mathcal{C}$. We assume that $\mathcal{C}$ is finite for all results in this paper. Define

$$\epsilon_{\mathcal{M}} = \min_{x, y \in \mathcal{C}_{\mathcal{M}} : x \neq y} |x - y|.$$

We require that all inputs to any computation in $\mathcal{M}$ must be elements of $\mathcal{C}$. Every computation in $\mathcal{M}$ can be represented as a sequence of operations, where the inputs to each operation may be inputs to the computation, constants, or outputs of past operations. We require that every input and every output of every operation must be an element of $\mathcal{C}$. When we need to distinguish between a real number and its representation in $\mathcal{C}$, we will use the $\hat{\cdot}$ symbol, i.e. the representation of $a \in \mathbb{R}$ in $\mathcal{C}$ is $\hat{a}$.

In order to enforce this last rule, we require that the exact result of any operation in $\mathcal{M}$ is rounded to the nearest element of $\mathcal{C}$, with ties rounded up. For sake of illustration, consider the example where $\mathcal{C}$ consists of the multiples of 0.1 between 0 and 1 inclusive, and one of the operations in $\mathcal{M}$ is multiplication $\times$. Clearly $\mathcal{C}$ is not closed under $\times$, since e.g. $0.1 \times 0.1 = 0.01 \notin \mathcal{C}$, $0.9 \times 0.9 = 0.81 \notin \mathcal{C}$, and $0.7 \times 0.5 = 0.35 \notin \mathcal{C}$. According to our rounding rules, we would round these results to 0, 0.8, and 0.4 respectively.

2.4 Embedding collision

The event that we will associate as a sufficient condition to misalignment will be *embedding collision*. Informally, we think about this event as when an embedding model Φ is non-injective when the image of Φ is represented through a chosen computational model. This could occur simply because Φ is non-injective, or due to systemic constraints. Either way, we make this notion precise in the following definition.

Definition 2.5. *Let $\Phi : \Omega \to \mathbb{R}^N$ be an embedding function and $\mathcal{C}$ a model of computation. An* ***embedding collision*** *occurs when there exist $a, b \in \Omega$ such that $\Phi(a) \neq \Phi(b)$ as elements of $\mathbb{R}^N$, but $\widehat{\Phi(a)} = \widehat{\Phi(b)}$ in $\mathcal{C}$.*

If $\Phi(a)$ and $\Phi(b)$ in $\Phi(\Omega)$ have the same embedding in $\mathcal{C}$, then denote this by

$$\Phi(a) \hat{=} \Phi(b). \tag{2.3}$$

3 Sufficient conditions for misalignment through embedding collision

In this section, we show that, unlike hash collisions, embedding collisions are potentially relatively common. Then, we give a sufficient condition for LLM misalignment using the concept of embedding collision.

3.1 Existence of embedding collisions

By the pigeonhole principle, we argue heuristically that embedding collisions are relatively common. The OpenAI model `text-embedding-3-large` uses mixed-precision training and inference (Micikevicius et al., 2018), a standard practice in large-scale transformer models.[1] In float16 and bfloat16, fewer than 2^{16} real numbers can be represented. Thus, there are fewer than 2^{16} possibilities for every coordinate of each vector. OpenAI's documentation states that `text-embedding-3-large` produces embeddings of dimension 3072 (OpenAI, 2024a). Consequently, there are fewer than $2^{16 \times 3072} = 2^{49152}$ possible vectors in the embedding space.

Meanwhile, the model employs the same tokenizer family used by GPT-3, which has 50257 distinct tokens in its vocabulary (OpenAI, 2024b). Moreover, the maximum context length for models served via the Embeddings API is 8192 tokens (OpenAI, 2024a). Therefore, the number of distinct token sequences that can be input to `text-embedding-3-large` is more than 50257^{8192}. Since

$$\frac{50257^{8192}}{2^{49152}} > 2^{78782},$$

there must exist distinct token sequences that are mapped to identical embedding vectors by `text-embedding-3-large`. Furthermore, even if we restrict inputs to the maximum context length, we can embed longer strings by chunking and merging embeddings of multiple segments. Thus, if chunking is permitted, then there are infinitely many strings that map to the same embedding vector.

[1] OpenAI has not disclosed the detailed datatypes used for this model; mixed-precision use is characteristic of contemporary transformer implementations.

We assume that any LLM that uses this vector embedding will treat strings with identical embeddings as if they are equivalent. In the context of alignment, this can create a significant issue, since there may be two strings which are far apart in alignment that get mapped to the same vector. This issue is at the core of the misalignment results in this paper.

3.2 Sufficient condition for misalignment in mean pooling schemes

In this section we look at a specific and simple example of embedding, *mean pooling*, which exhibits huge amounts of embedding collision. Then we prove that if the statements for an alignment system contain a logical negation operator, then no LLM which uses mean pool embedding in the sense of Example 3.1 can be aligned to a greater precision than that allowed by the logical negation. Then, we generalize our results by defining a sufficient condition for an LLM to be impossible to align. We begin with the definition of mean pool embedding.

Example 3.1. Suppose Σ is an alphabet of symbols known as *tokens*, and suppose there exists a unit norm vector representation v_σ of each token element σ of Σ in some large, ambient vector space $\mathbb{R}^N$. Then *mean pooling* is defined as follows. Select some initial set Ω_0 of statements and define the equivalence relation $a =_{mp} b$ if and only if the nonzero elements of the sequence a are a permutation of the nonzero elements of the sequence b. Then, let $\Omega \subset \Omega_0$ be a set of statements which are class representatives of $X/_{=_{mp}}$, and let the mean pool function $\Phi_{mp}(a)$ of $a = (a_i) \in \Omega$ be defined by

$$\Phi_{mp}(a) = \frac{1}{\text{len}(a)} \sum_{i=1}^{\text{len}(a)} v_{a_i}. \tag{3.1}$$

We note that all elements of Ω are in distinct equivalence classes in the quotient $X/_{=_{mp}}$.

The tokens in Σ by themselves are not related to each other logically. The minimal logical relation we impose to derive misalignment is the existence of a negation.

Definition 3.2. An alignment function $\Theta : \Omega \to (M, d)$ **has a logical NOT** if (i) there exists some $\nu \in \Sigma$ such that for all $\omega \in \Omega$, $\nu \oplus \omega \in \Omega$ and (ii) for all $\omega \in \Omega$, $\Theta(\omega) = \Theta(\nu \oplus \nu \oplus \omega)$. Here, $\oplus$ represents the concatenation operation $(a_1, \ldots, a_n) \oplus (b_1, \ldots, b_m) = (a_1 \ldots, a_n, b_1, \ldots, b_m)$.

In other words, concatenating ν to statements is an operation which is its own inverse in alignment. We note that a logical NOT, abbreviated as **not**, is defined giving emphasis on its position at the front of a sequence of tokens, but when the embedding is mean pooling, as defined above, the change in alignment of a statement and a series of negations of the statement essentially is determined by the parity of the number of negations in the statement. This is the observation that is recorded in the following proposition.

Proposition 3.3. *Let $\Phi_{mp} : \Omega \to \mathbb{R}^N$ denote a mean pool embedding function on a domain of sequences of tokens, and suppose $\Theta : \Omega \to (M, d)$ is an alignment function. Suppose **not** is a logical NOT for Θ and that there is a statement $\omega_0 \in \Omega$ such that that $d(\Theta(\omega_0), \Theta(\boldsymbol{not} \oplus \omega_0)) = \kappa > 0$. If $\Gamma \circ \Phi_{mp} : \Omega \to (M, d)$ is any deterministic approximation of Θ in a computational model $\mathcal{C}$, then there exist infinitely many $\omega \in \Omega$ for which $d(\Gamma \circ \Phi_{mp}(\omega), \Theta(\omega)) \geq \kappa/2$.*

Proof. For any token t, let v_t denote its token embedding in $\mathbb{R}^N$ and $\widehat{v_t}$ its representation in $\mathcal{C}$. First, observe that all coordinates of $\widehat{v_t}$ must be elements of $\mathcal{C}$. In particular, all coordinates of $\widehat{v_{\mathbf{not}}}$ must be elements of $\mathcal{C}$ since **not** is a token. Since the model of computation rounds all steps of the computation to the closest element of $\mathcal{C}$, there exists some sufficiently large $N \in \mathbb{N}$ such that every coordinate of

$$\frac{n}{n + \text{len}(\omega_0)} \widehat{v_{\mathbf{not}}} + \frac{1}{n + \text{len}(\omega_0)} \sum_{t \in \omega_0} \widehat{v_t}$$

is rounded to every corresponding coordinate of $\widehat{v_{\mathbf{not}}}$ for all $n \geq N$. Thus,

$$\Phi_{mp}\left(\bigoplus_{i=1}^{n} \mathbf{not} \oplus \omega_0 \right) \hat{=} \Phi_{mp}(\mathbf{not})$$

for all $n \geq N$. Now, for $n \geq N$, observe that $\Gamma \circ \Phi_{mp}$ cannot be within distance less than $\kappa/2$ of Θ for both the inputs $\bigoplus_{i=1}^{n} \mathbf{not} \oplus \omega_0$ and $\bigoplus_{i=1}^{n+1} \mathbf{not} \oplus \omega_0$. Otherwise, we would have

$$d\Bigg(\Gamma \circ \Phi_{mp}\Bigg(\bigoplus_{i=1}^{n} \mathbf{not} \oplus \omega_0\Bigg), \Theta\Bigg(\bigoplus_{i=1}^{n} \mathbf{not} \oplus \omega_0\Bigg)\Bigg) < \kappa/2$$

and

$$d\Bigg(\Gamma \circ \Phi_{mp}\Bigg(\bigoplus_{i=1}^{n+1} \mathbf{not} \oplus \omega_0\Bigg), \Theta\Bigg(\bigoplus_{i=1}^{n+1} \mathbf{not} \oplus \omega_0\Bigg)\Bigg) < \kappa/2,$$

which would imply by the definition of **not** and the triangle inequality respectively that

$$d(\Theta(\omega_0), \Theta(\mathbf{not} \oplus \omega_0)) = d\Bigg((\Theta\Bigg(\bigoplus_{i=1}^{n} \mathbf{not} \oplus \omega_0\Bigg), \Theta\Bigg(\bigoplus_{i=1}^{n+1} \mathbf{not} \oplus \omega_0\Bigg)\Bigg) < \kappa,$$

a contradiction. To find infinitely many $\omega \in \Omega$ so that $d(\Gamma \circ \Phi_{mp}(\omega), \Theta(\omega)) \geq \kappa/2$, construct a sequence by picking for every $n \geq N$ the parity $m = n$ or $m = n + 1$ such that

$$d(\Gamma \circ \Phi_{mp}(\oplus_{i=1}^{m} \mathbf{not} \oplus \omega_0), \Theta(\oplus_{i=1}^{m} \mathbf{not} \oplus \omega_0)) \geq \kappa/2.$$

□

It is worth recording that a probabilistic form of Proposition 3.3 also holds, which will be important in our discussion of LLMs which are trained using probabilistic algorithms.

Proposition 3.4. *Suppose **not** is a logical NOT for an alignment function $\Theta : \Omega \to (M, d)$. Suppose further there is a statement $\omega_0 \in \Omega$ such that that $d(\Theta(\omega_0), \Theta(\boldsymbol{not} \oplus \omega_0)) = \kappa > 0$. Let $\Phi_{mp} : \Omega \to \mathbb{R}^N$ denote a mean pool embedding function. If $\Gamma \circ \Phi_{mp} : \Omega \to (M, d)$ is any probabilistic approximation of Θ, then there exist infinitely many $\omega \in \Omega$ for which the probability that $d(\Gamma \circ \Phi_{mp}(\omega), \Theta(\omega)) \geq \kappa/2$ is at least $1/2$.*

Proof. As in the last proof, we have

$$\Phi_{mp}\Bigg(\bigoplus_{i=1}^{n} \mathbf{not} \oplus \omega_0\Bigg) \hat{=} \Phi_{mp}(\mathbf{not})$$

for all $n \geq N$. Note that $\Gamma \circ \Phi_{mp}$ cannot be within distance less than $\kappa/2$ of Θ with probability greater than $1/2$ for both the inputs $\bigoplus_{i=1}^{N} \mathbf{not} \oplus \omega_0$ and $\bigoplus_{i=1}^{N+1} \mathbf{not} \oplus \omega_0$. Otherwise by the inclusion exclusion principle, with positive probability we would have

$$d\Bigg(\Gamma \circ \Phi_{mp}\Bigg(\bigoplus_{i=1}^{n} \mathbf{not} \oplus \omega_0\Bigg), \Theta\Bigg(\bigoplus_{i=1}^{n} \mathbf{not} \oplus \omega_0\Bigg)\Bigg) < \kappa/2$$

and

$$d\Bigg(\Gamma \circ \Phi_{mp}\Bigg(\bigoplus_{i=1}^{n+1} \mathbf{not} \oplus \omega_0\Bigg), \Theta\Bigg(\bigoplus_{i=1}^{n+1} \mathbf{not} \oplus \omega_0\Bigg)\Bigg) < \kappa/2,$$

which as in the last proof, would imply that

$$d(\Theta(\omega_0), \Theta(\mathbf{not} \oplus \omega_0)) = d(\Theta\Bigg(\bigoplus_{i=1}^{n} \mathbf{not} \oplus \omega_0\Bigg), \Theta\Bigg(\bigoplus_{i=1}^{n+1} \mathbf{not} \oplus \omega_0\Bigg)) < \kappa,$$

a contradiction. The existence of infinitely many $\omega \in \Omega$ which are, with high probability, misaligned with respect to Θ, follows the same way as in the proof of Proposition 3.3. □

3.3 General conditions for misalignment

Propositions 3.3 and 3.4 can be interpreted as sufficient conditions for misalignment if $\kappa > 0$ is sufficiently large. The proofs of those propositions depend on properties of mean pooling. In the following, we develop a sufficient condition for similar alignment lower bounds which can apply to wider classes of embedding functions, not just mean pooling.

Definition 3.5. We say that Ω-valued sequences $a = (a^{(j)})$ and $b = (b^{(j)})$ are **potentially substring-contradictory** with respect to an embedding $\Phi : \Omega \to \mathbb{R}^N$ if there exists a constant $\delta \geq 1$ and such that the following is true.

1. For all $j \in \mathbb{N}$, we have $a^{(j)} \neq b^{(j)}$ and $|\text{len}(a^{(j)}) - \text{len}(b^{(j)})| \leq \delta$.
2. For all positive integers n and for all $j \in \mathbb{N}$ that are sufficiently large with respect to n, we have $\text{len}(a^{(j)}), \text{len}(b^{(j)}) \geq n$ and the maximum common substring of $a^{(j)}$ and $b^{(j)}$ has length at least $\max(\text{len}(a^{(j)}), \text{len}(b^{(j)})) - \delta$.
3. For all $\varepsilon > 0$, there exists N such that $\left\|\Phi(a^{(j)}) - \Phi(b^{(j)})\right\| < \varepsilon$ for all $j > N$.

The properties in Definition 3.5 are non-trivial requirements which essentially state that if two statements overlap on all but at most a constant number of tokens, then their embedding vectors are close. Property (2) implicitly assumes that an LLM is able to take as input strings which have finite but arbitrarily long length; heuristically, we reason that chunking procedures can be used to accommodate these inputs.

To connect to our previous example, we show that mean pool embedding has potentially substring-contradictory sequences.

Lemma 3.6. *With the mean pool function Φ defined as in equation (3.1), Φ is an embedding which has potentially substring-contradictory sequences.*

Proof. Fix $\delta \in \mathbb{N}$ and $n \in \mathbb{Z}^+$. Consider any $a, b \in \Omega$ such that $a_i = b_i$ for all $1 \leq i \leq n$ and $n \leq \text{len}(a), \text{len}(b) \leq n + \delta$. It is elementary to calculate

$$\Phi(a) - \Phi(b) = \left(\frac{1}{\text{len}(a)} - \frac{1}{\text{len}(b)}\right) \sum_{i=1}^{n} v_{a_i} + \frac{1}{\text{len}(a)} \sum_{i>n} v_{a_i} - \frac{1}{\text{len}(b)} \sum_{i>n} v_{b_i}, \tag{3.2}$$

where we let $v_{x_i} = 0$ if x_i does not exist. Since each v_{x_i} is at most unit norm,

$$\|\Phi(a) - \Phi(b)\| \leq \frac{|\text{len}(a) - \text{len}(b)|}{n^2} n + \frac{1}{n}(\text{len}(a) + \text{len}(b) - 2n) \tag{3.3}$$

$$= \frac{2\max(\text{len}(a), \text{len}(b)) - 2n}{n} \leq \frac{2\delta}{n}. \tag{3.4}$$

Thus, Φ has potentially substring-contradictory sequences. □

Now that we have demonstrated that there exist embeddings that have potentially substring-contradictory sequences, we prove that having a contradictory substring is sufficient for alignment to be impossible. We make the notion of a contradictory substring precise in Definition 3.7.

Definition 3.7. Suppose that $\Theta : \Omega \to (M, d)$ is an alignment function and $\Phi : \Omega \to \mathbb{R}^N$ is an embedding function. We say that (Φ, Θ) **has a contradictory substring sequence** if there exist Ω-valued sequences $a = (a^{(j)})$ and $b = (b^{(j)})$ that are potentially substring-contradictory with respect to Φ such that

$$\inf_{i \neq j} d(\Theta(a^{(i)}), \Theta(b^{(j)})) > 0.$$

That is, the sequence a and b are separated in alignment, and this separation in alignment is due to the existence of small substrings in the elements of one of the sequences. Given this definition of a contradictory substring, we show that if there is such an object within the language system represented by Ω, Φ, and Θ, then any LLM trained using Φ will be misaligned. This is proven in Theorem 3.8 as an equivalent result: that if Γ is some LLM which is aligned with Θ up to some positive tolerance, then there cannot existing contradictory substring sequences within Ω with respect to Θ.

Lemma 3.8. *In the model of computation $\mathcal{M}$, suppose that $\Theta : \Omega \to (M, d)$ is an alignment function and $\Gamma \circ \Phi : \Omega \to (M, d)$ is a deterministic approximation of Θ which uses an embedding function $\Phi : \Omega \to \mathbb{R}^N$. Suppose that there exists some $\varepsilon > 0$ such that $\sup_{\omega \in \Omega} d(\Gamma \circ \Phi(\omega), \Theta(\omega)) \leq \varepsilon$. Then, if $A, B \subset \Omega$ are subsets of statements such that $d(\overline{\Theta(A)}, \overline{\Theta(B)}) > 2\varepsilon$, then we must have $d(\Phi(A), \Phi(B)) \geq \epsilon_{\mathcal{M}}$.*

Proof. We prove the contrapositive. Suppose that $d(\Phi(A), \Phi(B)) < \epsilon_{\mathcal{M}}$. Then, there exist $u \in A$ and $v \in B$ such that $\|\Phi(u) - \Phi(v)\| < \epsilon_{\mathcal{M}}$. Then, we have

$$|\langle e_i, (\Phi(u) - \Phi(v))e_i \rangle| < \epsilon_{\mathcal{M}} \tag{3.5}$$

for all i, where e_i is the ith coordinate vector for $\mathbb{R}^N$. By definition of $\epsilon_{\mathcal{M}}$, we must have $\Phi(u) \hat{=} \Phi(v)$. Since Γ is an algorithm which cannot distinguish between inputs which are identical in $\mathcal{C}$, $\Gamma \circ \Phi(u) = \Gamma \circ \Phi(v)$. But then,

$$d(\Gamma \circ \Phi(v), \Theta(u)) = d(\Gamma \circ \Phi(u), \Theta(u)) \leq \varepsilon \text{ and } d(\Gamma \circ \Phi(v), \Theta(v)) \leq \varepsilon, \tag{3.6}$$

implying by the triangle inequality that $d(\overline{\Theta(A)}, \overline{\Theta(B)}) \leq 2\varepsilon$. □

Theorem 3.9. *Suppose that Θ is an alignment function and $\Gamma \circ \Phi$ is a deterministic LLM with embedding function Φ. If (Φ, Θ) has a contradictory substring sequence on $(a^{(j)})$ and $(b^{(j)})$, then $\Gamma \circ \Phi$ is κ-misaligned with Θ, where*

$$\kappa = \delta \inf_{i,j} d(\Theta(a^{(i)}), \Theta(b^{(j)})) \tag{3.7}$$

for any $0 < \delta < 1/2$.

Proof. Suppose that $\sup_{\omega \in \Omega} d(\Gamma \circ \Phi(\omega), \Theta(\omega)) \leq \kappa$. By definition of κ,

$$d(\overline{\{\Theta(a^{(j)})\}}, \overline{\{\Theta(b^{(j)})\}}) > 2\kappa. \tag{3.8}$$

But by Lemma 3.8, this would mean that $d(\Phi(A), \Phi(B)) \geq \varepsilon_{\mathcal{M}}$, contradicting the fact that $\inf_j \left\|\Phi(a^{(j)}) - \Phi(b^{(j)})\right\| \to 0$. So, it must be that there exists ω_0 such that $d(\Gamma \circ \Phi(\omega_0), \Theta(\omega_0)) > \kappa$, implying $\Gamma \circ \Phi$ is κ-misaligned. □

Now, we obtain similar results for probabilistic algorithms.

Lemma 3.10. *In the model of computation $\mathcal{M}$, suppose that $\Theta : \Omega \to (M, d)$ is an alignment function and $\Gamma \circ \Phi : \Omega \to (M, d)$ is a probabilistic approximation of Θ which uses an embedding function $\Phi : \Omega \to \mathbb{R}^N$. Suppose that there exists some $\varepsilon > 0$ such that $d(\Gamma \circ \Phi(\omega), \Theta(\omega)) \leq \varepsilon$ with probability greater than $1/2$ for all $\omega \in \Omega$. Then, if $A, B \subset \Omega$ are subsets of statements such that $d(\overline{\Theta(A)}, \overline{\Theta(B)}) > 2\varepsilon$, then we must have $d(\Phi(A), \Phi(B)) \geq \epsilon_{\mathcal{M}}$.*

Proof. As in the last result, we prove the contrapositive. Suppose that $d(\Phi(A), \Phi(B)) < \epsilon_{\mathcal{M}}$. Then, there exist $u \in A$ and $v \in B$ such that $d(\Phi(u), \Phi(v)) < \epsilon_{\mathcal{M}}$, so $\Phi(u) \hat{=} \Phi(v)$ and $\Gamma \circ \Phi(u) = \Gamma \circ \Phi(v)$.

By the inclusion exclusion principle, with positive probability we have

$$d(\Gamma \circ \Phi(v), \Theta(u)) = d(\Gamma \circ \Phi(u), \Theta(u)) \leq \varepsilon \text{ and } d(\Gamma \circ \Phi(v), \Theta(v)) \leq \varepsilon, \tag{3.9}$$

implying by the triangle inequality that $d(\overline{\Theta(A)}, \overline{\Theta(B)}) \leq 2\varepsilon$. □

Theorem 3.11. *Suppose that Θ is an alignment function and $\Gamma \circ \Phi$ is a probabilistic LLM with deterministic embedding function Φ. If (Φ, Θ) has a contradictory substring sequence on $(a^{(j)})$ and $(b^{(j)})$, then the probability that $\Gamma \circ \Phi$ is κ-misaligned with Θ is at least $1/2$, where*

$$\kappa = \delta \inf_{i,j} d(\Theta(a^{(i)}), \Theta(b^{(j)})) \tag{3.10}$$

for any $0 < \delta < 1/2$.

Proof. Suppose that $d(\Gamma \circ \Phi(\omega), \Theta(\omega)) \leq \kappa$ with probability greater than $1/2$ for all $\omega \in \Omega$. By definition of κ,

$$d(\overline{\{\Theta(a^{(j)})\}}, \overline{\{\Theta(b^{(j)})\}}) > 2\kappa. \tag{3.11}$$

But by Lemma 3.10, this would mean that $d(\Phi(A), \Phi(B)) \geq \varepsilon_{\mathcal{M}}$, contradicting the fact that $\inf_j \left\|\Phi(a^{(j)}) - \Phi(b^{(j)})\right\| \to 0$. So, it must be that there exists ω_0 such that $d(\Gamma \circ \Phi(\omega_0), \Theta(\omega_0)) > \kappa$ with probability at least $1/2$, implying $\Gamma \circ \Phi$ is κ-misaligned with probability at least $1/2$. □

4 Sufficient Conditions for misalignment through logical paradox

In the previous sections, we emphasized the role of embedding functions in sufficient conditions for misalignment. In this section, we change our focus and discuss the role of the alignment function Θ in producing misalignment. We consider whether it is possible to consistently define an alignment function on all possible inputs when we require certain alignment relations to hold between outputs from related inputs. We find that if we require alignment to be both *consistent* and *honest*, two properties of the alignment function that we define later in the section, then there exists some input for which the alignment function is undefined. The result is that any LLM trained with respect to a consistent and honest alignment will not necessarily be able to categorize its output as either aligned or misaligned.

4.1 A decision problem formulation of alignment

Before deriving the results in this section, we describe the alignment function properties we are interested in, starting with a decision-based formulation of the alignment problem. Let $A \subset M$ be any non-empty closed subset of an alignment space with alignment function Θ. We say that an input $\omega \in \Omega$ is *acceptable with respect to A* if $\Theta(\omega) \in A$. The idea is that acceptable outputs are those which fall into a certain region of alignment.

Example 4.1. *Returning to the Likert scale example, if $A = \{8, 9, 10\}$, then statements which are acceptable with respect to A are those which are relatively agreeable on the Likert scale.*

The associated decision problem for LLMs is the following. Given a statement ω, an LLM must decide whether $\Theta(\omega) \in A$ or not. One complication in this formulation is that often, statements have meanings, and statements with close enough meanings should both be acceptable, even if they are syntactically different. To model this, let Ω represent expressions in the English language, with each element $\omega \in \Omega$ called a *string*. Strings in a language are related by meaning; we assume there is a pre-defined sense of meaning represented by an equivalence relation.

Definition 4.2. *Let $\sim$ be an equivalence relation on Ω. Two strings x and y are* ***synonymous*** *if $x \sim y$.*

For example, every string is synonymous with itself. In reasonable interpretations of meaning for the English language, the strings **bad** and **not not bad** are synonymous.

English-language strings may contain *substrings*, informally defined as strings which occur within larger strings. We use the notion of a substring to allow strings to refer to themselves: Whenever **[this string]** occurs in any string x, the instance of **[this string]** refers to the whole string x.

Example 4.3. *For example, suppose that* ***String 1*** *is the string which says* ***There is no comma in [this string]*** *and* ***String 2*** *is the string which says* ***There is no comma in String 1****. Then,* ***String 1*** *and* ***String 2*** *are synonymous.*

The following definition formalizes the idea that the alignment of synonymous strings should be evaluated identically.

Definition 4.4. *Let $A \subset M$ be a non-empty, closed subset for an alignment space (M, d) with alignment function Θ. The pair (A, Θ) is* ***consistent*** *if for any two synonymous strings $x, y \in \Omega$, we have x is A-acceptable if and only if y is A-acceptable.*

In addition to consistency, we require for the alignment to be honest. Instead of enforcing the stronger requirement that all true strings must be aligned and all false strings must be unaligned, which is undecidable in general, we only require a weak form of honesty.

Definition 4.5. *We say that the pair $(\mathcal{A}, \Theta)$ is* ***honest*** *if*

1. *for all acceptable $\omega \in \Omega$, the string ω* ***is acceptable*** *is acceptable and the string ω* ***is not acceptable*** *is not acceptable, and*
2. *for all unacceptable $\omega \in \Omega$, the string ω* ***is acceptable*** *is not acceptable and the string ω* ***is not acceptable*** *is acceptable.*

For example, if the string **helping people is good** is acceptable, then the string ***helping people is good* is unaccceptable** is unacceptable. Similarly, if the string **helping people is bad** is unacceptable, then the string ***helping people is bad* is unacceptable** is acceptable.

Now, we are ready to show that it is impossible for a consistent and honest alignment function to be defined on all inputs. This observation follows from adapting the proof of the liar's paradox (Tarski, 1956) to our setting of alignment.

Theorem 4.6. *For all closed, non-empty subsets $A \subset M$ and functions $\Theta : \Omega \to M$ such that (A, Θ) is consistent and honest, there exists $\omega \in \Omega$ for which it is impossible to define $\Theta(\omega)$.*

Proof. Suppose that $(\mathcal{A}, \Theta)$ is consistent and honest. Let q denote the string Θ**([this string]) is not acceptable.** If $\Theta(q) \in A$, then Θ(**q is not acceptable**) $\in A$ by consistency, so $\Theta(q) \notin A$ by honesty, contradicting the assumption that $\Theta(q) \in A$. Otherwise, if $\Theta(q) \notin A$, then Θ(**q is not acceptable**) $\notin A$ by consistency, so $\Theta(q) \in A$ by honesty, contradicting the assumption that $\Theta(q) \notin A$. Thus, $\Theta(q)$ must be undefined. □

Corollary 4.7. *Suppose an LLM is aligned with respect to an alignment function which is both consistent and honest for at least one choice of non-empty, closed subset of the alignment space. Then, there exists a statement for which the LLM cannot decide whether the statement is aligned or misaligned.*

5 Discussion

In this section, we discuss practical implications of our results for AI development. Theorem 3.8, Theorem 4.6, and their corollaries suggest the following properties of LLMs which may be useful for high-level AI system design.

1. (Theorem 3.8) Even if an embedding function is injective in theory, i.e. it maps distinct inputs to distinct outputs, computational limitations such as limited representation of floating point numbers may cause statements with differing alignments to cluster together, leading to misalignment. This sufficient condition for misalignment may be exacerbated by any computational processing of embedding vectors which occurs as part of a given application of AI.

2. (Theorem 4.6) Defining alignment on all possible inputs to an LLM is challenging if the alignment scheme is necessarily consistent with itself logically. This means that even if an LLM is trained with respect to an alignment scheme, there may be inputs whose alignments the LLM cannot evaluate accurately.

5.1 Open problems in alignment

As an important caveat, these results *do not* quantify the severity of misalignment or even where consequential misalignments are likely to occur. For example, although we showed that constraints on logical consistency of alignment leads to statements that cannot be evaluated for alignment, the arguments that show this use statements which are unlikely to be consequential in AI applications. So, even though the set of statements which cannot be aligned is non-empty, our arguments do not illuminate whether these statements would occur in typical use cases of LLMs. Generally, it is an open question whether *effective alignment,* i.e. alignment of LLMs and other AI which is sufficient to prevent undesired behavior, is possible. In this context, our arguments can be interpreted as identifying specific indicators of misalignment: embedding collisions and the ability of alignment schemes to represent logical paradoxes. We leave the larger problem of characterizing misalignment as an important open problem for future research.

Acknowledgments

References

Dario Amodei. The urgency of interpretability, 2025. Accessed 18 August 2025. `https://blog.biocomm.ai/2025/04/27/dario-amodei-the-urgency-of-interpretability-april-2025/`.

Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for AI safety: A review. *Transactions on Machine Learning Research*, 2024.

Matthew J. Colbrook, Vegard Antun, and Anders C. Hansen. Can stable and accurate neural networks be computed? On the barriers of deep learning and Smale's 18th problem. *Proceedings of the National Academy of Sciences*, 119(12), 2022.

W. B. Estes, W. A. Holley, and E. D. Pendergraft. Formation and transformation structures: Part I. Technical report, Linguistics Research Center, The University of Texas, 1963.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Lukas Vierling, Donghai Hong, Jiayi Zhou, Zhaowei Zhang, Fanzhi Zeng, Juntao Dai, Xuehai Pan, Kwan Yee Ng, Aidan O'Gara, Hua Xu, Brian Tse, Jie Fu, Stephen McAleer, Yaodong Yang, Yizhou Wang, Song-Chun Zhu, Yike Guo, and Wen Gao. AI alignment: A comprehensive survey, 2025.

Simon Pepin Lehalleur, Jesse Hoogland, Matthew Farrugia-Roberts, Susan Wei, Alexander Gietelink Oldenziel, George Wang, Liam Carroll, and Daniel Murfet. You are what you eat – AI alignment requires understanding how data shapes structure and generalisation, 2025.

Paulius Micikevicius, Sharan Narang, Jonah Alben, et al. Mixed precision training. In *International Conference on Learning Representations (ICLR)*, 2018.

OpenAI. API reference: Embeddings guide, 2024a. `https://platform.openai.com/docs/guides/embeddings`.

OpenAI. `tiktoken`: OpenAI tokenizer repository, 2024b. `https://github.com/openai/tiktoken`.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. Toward transparent AI: A survey on interpreting the inner structures of deep neural networks. *Transactions on Machine Learning Research*, 2023.

Alfred Tarski. The concept of truth in formalized languages. In *Logic, Semantics, Metamathematics*, pp. 152–278. Clarendon Press, Oxford, 1956.

Sumio Watanabe. Almost all learning machines are singular. In *Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence*, 2007.

Susan Wei, Daniel Murfet, Mingming Gong, Hui Li, Jesse Gell-Redman, and Thomas Quella. Deep learning is singular, and that's good. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12), 2023.