Fundamental Limitation in Explaining AI

A Quadrilemma among Environment Complexity, AI Performance, Interpretability, and Faithfulness

Atsushi Suzuki Jing Wang

2026-05-27

The University of Hong Kong / University of Greenwich

香 港 大 學
THE UNIVERSITY OF HONG KONG

KAWAII lab

Materials available here!

- arXiv: https://arxiv.org/abs/2605.24727
  - 
- The presentation slides:
  https://ash-suzuki.github.io/material/xai_limitation_talk.pdf
  - 

SUZUKI, Atsushi

- 2011-2020: Bachelor (Engineering, 2015), Master (Information Science and Technology (IST), 2017), PhD (IST, 2020), The University of Tokyo
- 2020-2024: Assistant Professor (UK Lecturer) in UK (University of Greenwich, King's College London)
- 2025-: **Assistant Professor** (math) at the University of Hong Kong.

Research interests:

- Methods for AI (differential geometry, information theory, deep learning)
- Theoretical analysis of AIs (statistics, information theory, computability theory)

Outline

Motivation

Problem Formulation

Formulating the 4 conditions

Main Result

Conclusion

References

Motivation

Success of AIs

Success of AIs

- McKinsey estimates its annual economic value to be **$2.6–$4.4 trillion** [6].
- Goldman Sachs estimates it could boost **global GDP by about 7% (approximately $7 trillion)** over the next decade [4].

Social demand in explaining AIs

Public institutions emphasize the importance of **explainability in AI**.

Social demand in explaining AIs

Public institutions emphasize the importance of **explainability in AI**.

Example (Action Plans by NIST (US) [7])

- “Address general risks associated with a lack of **explainability and transparency** in GAI systems” (GV-4.1-001)
- “Implement interpretability and explainability methods to evaluate GAI system decisions and verify alignment with intended purpose.” (MS-4.2-003)

Existing explanations are not completely “faithful”

We say that an explanation is **faithful** if we can recover the behavior of AI [1].

[1] There is no single established and formalized definition of “faithful,” to the best of our knowledge

Existing explanations are not completely “faithful”

We say that an explanation is **faithful** if we can recover the behavior of AI [1].

The faithfulness must be a plausible property, because, if otherwise, it means that the explanation lacks some information about the AI’s behavior.

However,

[1] There is no single established and formalized definition of “faithful,” to the best of our knowledge

Existing explanations are not completely “faithful”

We say that an explanation is **faithful** if we can recover the behavior of AI [1].

The faithfulness must be a plausible property, because, if otherwise, it means that the explanation lacks some information about the AI’s behavior.

However, existing explanation methods do NOT pursue it completely.

[1] There is no single established and formalized definition of “faithful,” to the best of our knowledge

Non faithful explanation example: LIME

Example (LIME (Local Interpretable Model-agnostic Explanations) [9])

- Considers binarization of the original data points, i.e., intentionally **gives up the faithfulness** of the explanation.
- Explanation: local approximation in the binarized space.

Specifically, fix a binarization function $(\bullet)_{\mathsf{bin}} : \mathcal{X} \to \{0,1\}^d$ and recover function $(\bullet)_{\mathsf{rec}} : \{0,1\}^d \to \mathcal{X}$. The explanation of a function $f : \mathcal{X} \to \mathbb{R}$ of interest around $\boldsymbol{x}$ is given by $\boldsymbol{w}_{\boldsymbol{x}} \in \mathbb{R}^{d'}$ that minimizes

$$\sum_{\boldsymbol{z}' \in \mathsf{neighbor}\left(x_{\mathsf{bin}}\right)} \mathsf{similarity}(x, \boldsymbol{z}'_{\mathsf{rec},x}) \left(f(\boldsymbol{z}'_{\mathsf{rec},x}) - \boldsymbol{w}_x^{\mathsf{T}} \boldsymbol{z}' \right)^2 \quad \text{s.t. } \|\boldsymbol{w}_x\|_0 \geq K.$$

Starting question

Starting question

Why did they have to be not completely faithful to an AI?

Naïve question: **Can we not provide a faithful explanation** of AIs?

Starting question

Why did they have to be not completely faithful to an AI?

Naïve question: **Can we not provide a faithful explanation** of AIs?

Before we discuss this problem, we first need to exclude trivial and uninteresting cases.

Exclude trivial cases. 1. When AI's performance is poor

Example (A poor AI case)

Task: AI is asked to answer the user's question.

Exclude trivial cases. 1. When AI's performance is poor

Example (A poor AI case)

Task: AI is asked to answer the user's question.

- **User prompt 1**: `Tell me the smallest perfect number?`
- **AI's answer 1**: `I am an AI.`

Exclude trivial cases. 1. When AI's performance is poor

Example (A poor AI case)

Task: AI is asked to answer the user's question.

- **User prompt 1**: `Tell me the smallest perfect number?`
- **AI's answer 1**: `I am an AI.`
- **User prompt 2**: `Where is the capital city of Japan?`
- **AI's answer 2**: `I am an AI.`

Exclude trivial cases. 1. When AI's performance is poor

Example (A poor AI case)

Task: AI is asked to answer the user's question.

- **User prompt 1**: `Tell me the smallest perfect number?`
- **AI's answer 1**: `I am an AI.`
- **User prompt 2**: `Where is the capital city of Japan?`
- **AI's answer 2**: `I am an AI.`
- **Explanation**: "The AI just outputs `I am an AI.`, i.e., $\mathrm{AI}(\boldsymbol{x}) =$ `I am an AI.`"

In this case, while the explanation is faithful, the AI's performance is poor.

Exclude trivial cases. 2. When the task/environment is simple.

Example (Trivial task case.)

Task: AI is asked to repeat the user prompt.

Exclude trivial cases. 2. When the task/environment is simple.

Example (Trivial task case.)

Task: AI is asked to repeat the user prompt.

- **User prompt 1**: `Tell me the smallest perfect number.`
- **AI's answer 1**: `Tell me the smallest perfect number.`

Exclude trivial cases. 2. When the task/environment is simple.

Example (Trivial task case.)

Task: AI is asked to repeat the user prompt.

- **User prompt 1**: `Tell me the smallest perfect number.`
- **AI's answer 1**: `Tell me the smallest perfect number.`
- **User prompt 2**: `Where is the capital city of Japan?`
- **AI's answer 2**: `Where is the capital city of Japan?`

Exclude trivial cases. 2. When the task/environment is simple.

Example (Trivial task case.)

Task: AI is asked to repeat the user prompt.

- **User prompt 1**: `Tell me the smallest perfect number.`
- **AI's answer 1**: `Tell me the smallest perfect number.`
- **User prompt 2**: `Where is the capital city of Japan?`
- **AI's answer 2**: `Where is the capital city of Japan?`
- **Explanation**: "The AI just repeats the user prompt, i.e., $\mathrm{AI}(\boldsymbol{x}) = \boldsymbol{y}$"

In this case, the AI's performance is perfect, and the explanation is faithful.

Exclude trivial cases. 3. When AI explanation is NOT interpretable.

Example (A poor AI case)

Task: AI is asked to answer the user's question.

Exclude trivial cases. 3. When AI explanation is NOT interpretable.

Example (A poor AI case)

Task: AI is asked to answer the user's question.

- **User prompt 1**: `Tell me the smallest perfect number?`
- **AI's answer 1**: `It's 6.`

Exclude trivial cases. 3. When AI explanation is NOT interpretable.

Example (A poor AI case)

Task: AI is asked to answer the user's question.

- **User prompt 1**: `Tell me the smallest perfect number?`
- **AI's answer 1**: `It's 6.`
- **User prompt 2**: `Where is the capital city of Japan?`
- **AI's answer 2**: `The de facto capital city of Japan is Tokyo.`

Exclude trivial cases. 3. When AI explanation is NOT interpretable.

Example (A poor AI case)

Task: AI is asked to answer the user's question.

- **User prompt 1**: `Tell me the smallest perfect number?`
- **AI's answer 1**: `It's 6.`
- **User prompt 2**: `Where is the capital city of Japan?`
- **AI's answer 2**: `The de facto capital city of Japan is Tokyo.`
- **Explanation**: (Just gives the source code and the checkpoint)

In this case, while the AI performs well and the explanation is faithful, the explanation is NOT interpretable.

Refined research question

Research question: Can there exist an AI and its explanation such that:

- The AI works in a complicated environment, e.g., natural language Q/A.
- The AI's performance is good.
- The explanation is completely faithful to the AI behavior.
- The explanation is interpretable.

Main answer

Theorem (Informal quadrilemma)

*An AI system and its explanation **CANNOT** simultaneously satisfy all of the following:*

- *The AI's operation environment is complicated,*
- *The AI's performance is good,*
- *The explanation is completely faithful to the AI behavior,*
- *The explanation is interpretable.*

Main answer

Theorem (Informal quadrilemma)

*An AI system and its explanation **CANNOT** simultaneously satisfy all of the following:*

- *The AI's operation environment is complicated,*
- *The AI's performance is good,*
- *The explanation is completely faithful to the AI behavior,*
- *The explanation is interpretable.*

But a natural question arises: how do we **mathematically** formulate the above notions?

Main answer

Theorem (Informal quadrilemma)

*An AI system and its explanation **CANNOT** simultaneously satisfy all of the following:*

- *The AI's operation environment is complicated,*
- *The AI's performance is good,*
- *The explanation is completely faithful to the AI behavior,*
- *The explanation is interpretable.*

But a natural question arises: how do we **mathematically** formulate the above notions?

Actually, today's talk is mainly about definitions, rather than proofs.

Difficulty in mathematically formulating the interpretability

We may mathematically formulate the conditions as below (rough idea):

- The AI's operation environments complexity: Statistics of the data distribution, function space, etc.

Difficulty in mathematically formulating the interpretability

We may mathematically formulate the conditions as below (rough idea):

- The AI's operation environments complexity: Statistics of the data distribution, function space, etc.
- The AI's performance: Evaluation criteria, e.g., perplexity

Difficulty in mathematically formulating the interpretability

We may mathematically formulate the conditions as below (rough idea):

- The AI's operation environments complexity: Statistics of the data distribution, function space, etc.
- The AI's performance: Evaluation criteria, e.g., perplexity
- The explanation's faithfulness to the AI behavior: e.g., Whether the explanation can be converted to the implementation

Difficulty in mathematically formulating the interpretability

We may mathematically formulate the conditions as below (rough idea):

- The AI's operation environments complexity: Statistics of the data distribution, function space, etc.
- The AI's performance: Evaluation criteria, e.g., perplexity
- The explanation's faithfulness to the AI behavior: e.g., Whether the explanation can be converted to the implementation

But...

Difficulty in mathematically formulating the interpretability

We may mathematically formulate the conditions as below (rough idea):

- The AI's operation environments complexity: Statistics of the data distribution, function space, etc.
- The AI's performance: Evaluation criteria, e.g., perplexity
- The explanation's faithfulness to the AI behavior: e.g., Whether the explanation can be converted to the implementation

But... **How** can we mathematically define the **intepretability** of an explanation?

Key trick to avoid the difficulty in formulating the interpretability

It's not easy to formulate the interpretability of an explanation [1, 5].

Key trick to avoid the difficulty in formulating the interpretability

It's not easy to formulate the interpretability of an explanation [1, 5].

Key trick: To negate a condition, it's sufficient to disprove its **necessary** condition.

Key trick to avoid the difficulty in formulating the interpretability

It's not easy to formulate the interpretability of an explanation [1, 5].

Key trick: To negate a condition, it's sufficient to disprove its **necessary** condition.

What condition is

- necessary for the interpretability, and
- easy to formulate mathematically?

Key trick to avoid the difficulty in formulating the interpretability

It's not easy to formulate the interpretability of an explanation [1, 5].

Key trick: To negate a condition, it's sufficient to disprove its **necessary** condition.

What condition is

- necessary for the interpretability, and
- easy to formulate mathematically?

Our idea: Use the **shortness** of the explanation as a necessary condition for the interpretability.

Why the shortness is necessary for the interpretability?

An explanation must be short to be interpretable since human capacity to read and remember text is limited [2] [3]

Why the shortness is necessary for the interpretability?

An explanation must be short to be interpretable since human capacity to read and remember text is limited [2] [3]

Example (Human memorizing capacity [3])

The world record for memorizing the decimal expansion of $\pi = 3.14159...$, is 70,000 digits [3], which is at most 233 thousand bits.

Why the shortness is necessary for the interpretability?

An explanation must be short to be interpretable since human capacity to read and remember text is limited [2] [3]

Example (Human memorizing capacity [3])

The world record for memorizing the decimal expansion of $\pi = 3.14159...$, is 70,000 digits [3], which is at most 233 thousand bits.

Example (Human reading capacity [2])

238 English words per minute for nonfiction text, with 4.6 characters per word, which corresponds to approximately 4B bits per year.

Why the shortness is necessary for the interpretability?

An explanation must be short to be interpretable since human capacity to read and remember text is limited [2] [3]

Example (Human memorizing capacity [3])

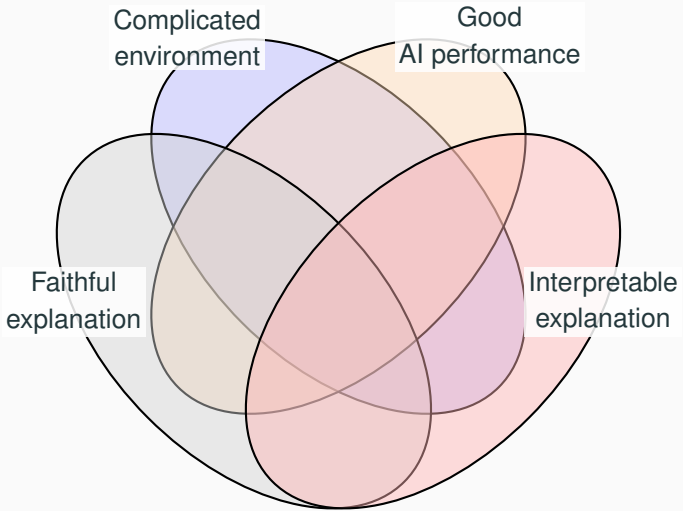
The world record for memorizing the decimal expansion of $\pi = 3.14159...$, is 70,000 digits [3], which is at most 233 thousand bits.

Example (Human reading capacity [2])

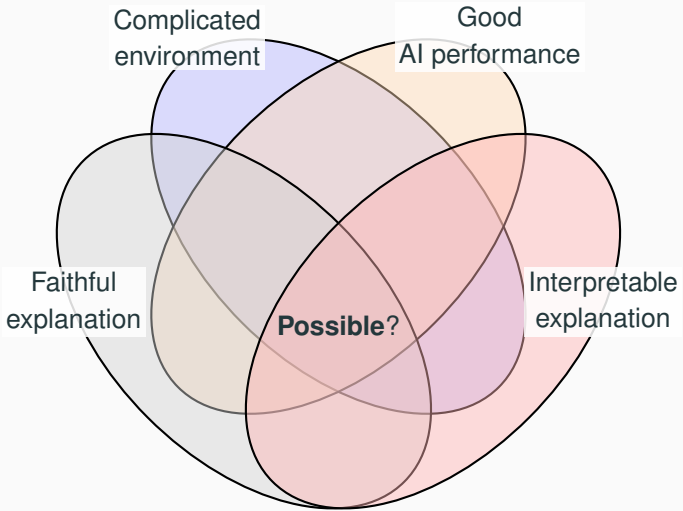
238 English words per minute for nonfiction text, with 4.6 characters per word, which corresponds to approximately 4B bits per year.

This is why the pair of the source code and checkpoint (28B bits even for a 7B model with the 4 bit quantization.) of the open model is NOT interpretable.

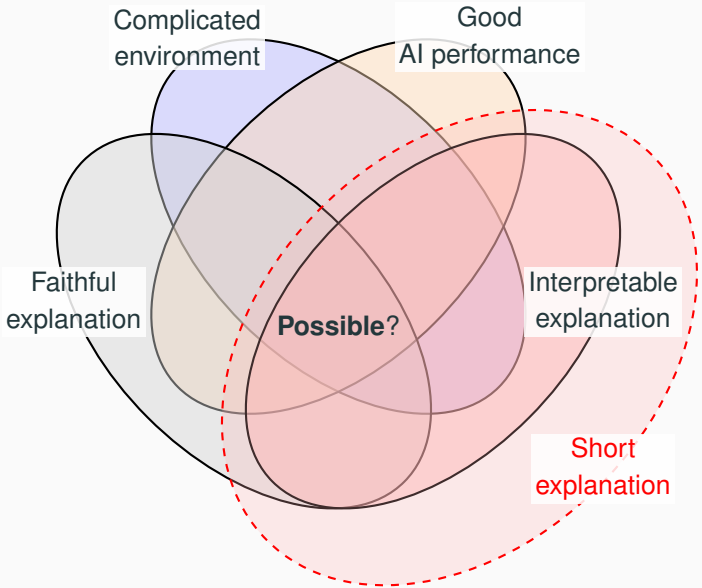
Conceptual picture



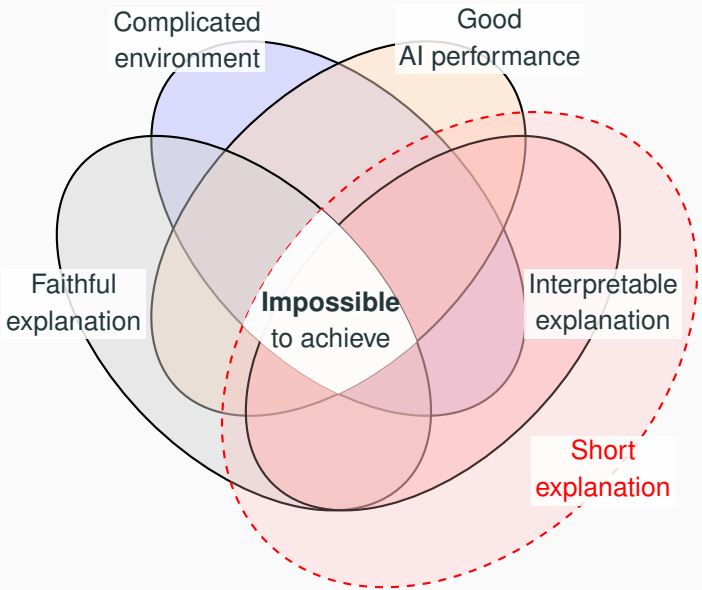
Conceptual picture



Conceptual picture



Conceptual picture



What we actually prove is...

Theorem (Informal quadrilemma (more formulatable))

*An AI system and its explanation **CANNOT** simultaneously satisfy all of the following:*

- *The AI's operation environment is complicated,*
- *The AI's performance is good,*
- *The explanation is completely faithful to the AI behavior,*
- *The explanation is **short**.*

The above statement is much easier to formulate than the original form.

Problem Formulation

What do we need to do?

- Formulate an AI itself.
- Formulate the goodness of AI performance.
- Formulate the faithfulness of the explanation.
- Formulate the shortness of the explanation.
- Formulate AI's operation environment

Preliminary: finite-length strings

Definition (Finite-length string)

Let Σ be the letter set that the computer can natively use. Typically $\Sigma = 0, 1$. A **finite-length string** is a finite-length sequence of letters, e.g., 001101011.

The set of all finite-length strings is denoted by $\Sigma^* := \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots$.

Note that in a computer, everything needs to be represented as a finite-length string, and the set Σ^* is countably infinite.

Hence, it CANNOT represent the whole of an uncountably infinite set such as the real number set $\mathbb{R}$ or the function class $\{f : \Sigma^* \to \Sigma^*\}$.

Rough formulation of AI

We define an AI as an input-output stochastic computation process

Definition (A rough formulation of an AI)

A **stochastic input-output AI** is a system that receives

Rough formulation of AI

We define an AI as an input-output stochastic computation process

Definition (A rough formulation of an AI)

A **stochastic input-output AI** is a system that receives

- **Task Input** $x \in \Sigma^*$: User prompt, system prompt, user context, etc,

and (often)

Rough formulation of AI

We define an AI as an input-output stochastic computation process

Definition (A rough formulation of an AI)

A **stochastic input-output AI** is a system that receives

- **Task Input** $\boldsymbol{x} \in \Sigma^*$: User prompt, system prompt, user context, etc,

and (often)

- **Random sequence** $\boldsymbol{u} \in \Sigma^*$ generated by using a pseudo random number generator until it satisfies the termination condition $\boldsymbol{u} \in \mathcal{T}_{\boldsymbol{x}}$,

Rough formulation of AI

We define an AI as an input-output stochastic computation process

Definition (A rough formulation of an AI)

A **stochastic input-output AI** is a system that receives

- **Task Input** $\boldsymbol{x} \in \Sigma^*$: User prompt, system prompt, user context, etc,

and (often)

- **Random sequence** $\boldsymbol{u} \in \Sigma^*$ generated by using a pseudo random number generator until it satisfies the termination condition $\boldsymbol{u} \in \mathcal{T}_{\boldsymbol{x}}$,

and processes them by a program $\boldsymbol{p} \in \Sigma^*$, then returns

Rough formulation of AI

We define an AI as an input-output stochastic computation process

Definition (A rough formulation of an AI)

A **stochastic input-output AI** is a system that receives

- **Task Input** $\boldsymbol{x} \in \Sigma^*$: User prompt, system prompt, user context, etc,

and (often)

- **Random sequence** $\boldsymbol{u} \in \Sigma^*$ generated by using a pseudo random number generator until it satisfies the termination condition $\boldsymbol{u} \in \mathcal{T}_{\boldsymbol{x}}$,

and processes them by a program $\boldsymbol{p} \in \Sigma^*$, then returns

- **Output** $\boldsymbol{y} = f_{\boldsymbol{p}}(\boldsymbol{u}, \boldsymbol{x}) \in \Sigma^*$: response to the users, internal steps, etc.

where $f_{\boldsymbol{p}} : \Sigma^* \to \Sigma^*$ is the function implemented by $\boldsymbol{p}$.

Example 1: text-to-image using a diffusion process.

Example 1: text-to-image using a diffusion process.

Example (Text-to-image using diffusion process (e.g., [10]))

- **Input**: user prompt x
- **Step 1**: Convert the user prompt x to a context vector c (e.g., by CLIP).

Example 1: text-to-image using a diffusion process.

Example (Text-to-image using diffusion process (e.g., [10]))

- **Input**: user prompt x
- **Step 1**: Convert the user prompt x to a context vector c (e.g., by CLIP).
- **Step 2**: Generate an initial random latent image z_0 by generating pseudo random numbers.

Example 1: text-to-image using a diffusion process.

Example (Text-to-image using diffusion process (e.g., [10]))

- **Input**: user prompt $\boldsymbol{x}$
- **Step 1**: Convert the user prompt $\boldsymbol{x}$ to a context vector $\boldsymbol{c}$ (e.g., by CLIP).
- **Step 2**: Generate an initial random latent image $\boldsymbol{z}_0$ by generating pseudo random numbers.
- **Step 3**: For $t \leftarrow 0, 1, ..., T-1$: Compute next $\boldsymbol{z}_{t+1}$ from $\boldsymbol{z}_t$, $\boldsymbol{c}$, t.

Example 1: text-to-image using a diffusion process.

Example (Text-to-image using diffusion process (e.g., [10]))

- **Input**: user prompt $\boldsymbol{x}$
- **Step 1**: Convert the user prompt $\boldsymbol{x}$ to a context vector $\boldsymbol{c}$ (e.g., by CLIP).
- **Step 2**: Generate an initial random latent image $\boldsymbol{z}_0$ by generating pseudo random numbers.
- **Step 3**: For $t \leftarrow 0, 1, ..., T-1$: Compute next $\boldsymbol{z}_{t+1}$ from $\boldsymbol{z}_t$, $\boldsymbol{c}$, t.
- **Step 4**: From $\boldsymbol{z}_T$, compute an image $\boldsymbol{y}$ (e.g., by VAE) and **output** $\boldsymbol{y}$.

Example 1: text-to-image using a diffusion process.

Example (Text-to-image using diffusion process (e.g., [10]))

- **Input**: user prompt $\boldsymbol{x}$
- **Step 1**: Convert the user prompt $\boldsymbol{x}$ to a context vector $\boldsymbol{c}$ (e.g., by CLIP).
- **Step 2**: Generate an initial random latent image $\boldsymbol{z}_0$ by generating pseudo random numbers.
- **Step 3**: For $t \leftarrow 0, 1, ..., T-1$: Compute next $\boldsymbol{z}_{t+1}$ from $\boldsymbol{z}_t$, $\boldsymbol{c}$, t.
- **Step 4**: From $\boldsymbol{z}_T$, compute an image $\boldsymbol{y}$ (e.g., by VAE) and **output** $\boldsymbol{y}$.

It receives a user prompt $\boldsymbol{x}$ pseudo-random numbers and returns an image $\boldsymbol{y}$, showing a pseudo-stochastic behavior. The termination random variable set $\mathcal{T}_{\boldsymbol{x}}$ consists of fixed-length sequences.

Example 2: large language model

Example 2: large language model

Example (Large language model)

- **Input**: user prompt x

Example 2: large language model

Example (Large language model)

- **Input**: user prompt $\boldsymbol{x}$
- **Step 1**: $t \leftarrow 0$, $\boldsymbol{y}_{0:-1} = ()$.

Example 2: large language model

Example (Large language model)

- **Input**: user prompt $\boldsymbol{x}$
- **Step 1**: $t \leftarrow 0$, $\boldsymbol{y}_{0:-1} = ()$.
- **Step 2**: Compute the next token distribution Q_t from the user prompt $\boldsymbol{x}$ and past token sequence $\boldsymbol{y}_{0:t-1}$.

Example 2: large language model

Example (Large language model)

- **Input**: user prompt $\boldsymbol{x}$
- **Step 1**: $t \leftarrow 0$, $\boldsymbol{y}_{0:-1} = ()$.
- **Step 2**: Compute the next token distribution Q_t from the user prompt $\boldsymbol{x}$ and past token sequence $\boldsymbol{y}_{0:t-1}$.
- **Step 3**: From Q_t and a pseudo-random number, determine the next token y_t, and concatenate it to the past token sequence to get $\boldsymbol{y}_{0:t} = \boldsymbol{y}_{0:t-1} \cdot y_t$.

Example 2: large language model

Example (Large language model)

- **Input**: user prompt $\boldsymbol{x}$
- **Step 1**: $t \leftarrow 0$, $\boldsymbol{y}_{0:-1} = ()$.
- **Step 2**: Compute the next token distribution Q_t from the user prompt $\boldsymbol{x}$ and past token sequence $\boldsymbol{y}_{0:t-1}$.
- **Step 3**: From Q_t and a pseudo-random number, determine the next token y_t, and concatenate it to the past token sequence to get $\boldsymbol{y}_{0:t} = \boldsymbol{y}_{0:t-1} \cdot y_t$.
- **Step 4**: If y_t is an EOS (end of sentence) token, then **output** $\boldsymbol{y}_{0:t}$. Otherwise, go to **Step 2**.

Example 2: large language model

Example (Large language model)

- **Input**: user prompt $\boldsymbol{x}$
- **Step 1**: $t \leftarrow 0$, $\boldsymbol{y}_{0:-1} = ()$.
- **Step 2**: Compute the next token distribution Q_t from the user prompt $\boldsymbol{x}$ and past token sequence $\boldsymbol{y}_{0:t-1}$.
- **Step 3**: From Q_t and a pseudo-random number, determine the next token y_t, and concatenate it to the past token sequence to get $\boldsymbol{y}_{0:t} = \boldsymbol{y}_{0:t-1} \cdot y_t$.
- **Step 4**: If y_t is an EOS (end of sentence) token, then **output** $\boldsymbol{y}_{0:t}$. Otherwise, go to **Step 2**.

It also receives a user prompt $\boldsymbol{x}$ pseudo-random numbers and returns an image $\boldsymbol{y}$, showing a pseudo-stochastic behavior. The termination random variable set $\mathcal{T}_{\boldsymbol{x}}$ (all random number sequences that output EOS in the end) depends on $\boldsymbol{x}$.

Behavior as a conditional probability mass function

Definition (AI-induced conditional probability mass function)

Let $\mathrm{Prob}(\boldsymbol{u})$ be a probability mass virtually allocated to pseudo random number sequence $\boldsymbol{u}$.

For a stochastic input-output AI A with the termination condition given by $\mathcal{T}_x$, define Q_A by

$$Q_A(y \mid x) = \sum_{\boldsymbol{u} \in \mathcal{T}_x} \mathrm{Prob}(\boldsymbol{u}) \mathbb{1}\big(f(\boldsymbol{u}, x) = y\big).$$

This is the conditinal probability mass function representing the stochastic input-output AI.

Behavior as a conditional probability mass function

Definition (AI-induced conditional probability mass function)

Let $\mathrm{Prob}(\boldsymbol{u})$ be a probability mass virtually allocated to pseudo random number sequence $\boldsymbol{u}$.

For a stochastic input-output AI A with the termination condition given by $\mathcal{T}_x$, define Q_A by

$$Q_A(y \mid x) = \sum_{\boldsymbol{u} \in \mathcal{T}_x} \mathrm{Prob}(\boldsymbol{u}) \mathbb{1}\big(f(\boldsymbol{u}, x) = y\big).$$

This is the conditinal probability mass function representing the stochastic input-output AI.

The goal of AI explanation: **to explain** Q_A.

Formulating the 4 conditions

Formulation of AI Performance: perplexity

As a metric of the AI performance, we simply use **perplexity**, widely used in NLP (e.g., [8]).

Definition (Perplexity)

Given an input $x \in \mathcal{X}$ and an output $y \in \mathcal{Y}$, the **perplexity** of a conditional probability mass function Q is

$$\frac{1}{Q(y \mid x)}.$$

Given a true distribution P on $\mathcal{X} \times \mathcal{Y}$, the expected logarithmic perplexity is

$$\mathbb{E}_{X,Y \sim P}\big[-\log Q(Y \mid X) \big].$$

Remark

A Small expected logarithmic perplexity means good AI performance.

Formulation of completely faithful explanations

A faithful explanation should recover the AI behavior when we interpret it under a fixed formal rule.

Formulation of completely faithful explanations

A faithful explanation should recover the AI behavior when we interpret it under a fixed formal rule.

We define an **interpretation function** to represent the rule to interpret the explanation.

Formulation of completely faithful explanations

A faithful explanation should recover the AI behavior when we interpret it under a fixed formal rule.

We define an **interpretation function** to represent the rule to interpret the explanation.

Definition (Interpretation function)

Fix a letter set Λ of natural languages (e.g., Unicode). An **interpretation function** is a **computable** function

$$\mathsf{Interpret} :\subseteq \Lambda^* \times \mathbb{N} \times \mathcal{Z} \xrightarrow{\text{comp}} \mathbb{Q}$$

that interprets an explanation string as an arbitrary-precision approximation of a real-valued function.

Notes on a computable function

Informally speaking, a **computable function** is a function that can be realized by Python with unlimited time and memory. It includes, e.g., an LLM.

Notes on a computable function

Informally speaking, a **computable function** is a function that can be realized by Python with unlimited time and memory. It includes, e.g., an LLM.

In the above, we can replace Python with another programming language, e.g., C, because we can write a C emulator in Python and a Python emulator in C.

Notes on a computable function

Informally speaking, a **computable function** is a function that can be realized by Python with unlimited time and memory. It includes, e.g., an LLM.

In the above, we can replace Python with another programming language, e.g., C, because we can write a C emulator in Python and a Python emulator in C.

A function implemented in a programming language may not define the return value for some inputs owing to an infinite loop. Such a function is called a partial function, and is denoted like $f :\subseteq \mathcal{X} \to \mathcal{Y}$, instead of $f : \mathcal{X} \to \mathcal{Y}$.

When $f(x)$ is defined, we write $f(x) \downarrow$.

Faithfulness as reproducibility under a fixed interpretation rule

Definition (Completely faithful explanation)

A string $\boldsymbol{e} \in \Lambda^*$ is a completely faithful explanation of a function

$$f : \mathcal{Z} \xrightarrow{\text{comp}} \mathbb{R}$$

under Interpret if, for every $z \in \mathcal{Z}$ and every $k \in \mathbb{N}$,

$$|\text{Interpret}(\boldsymbol{e}, k, z) - f(z)| < |\Sigma|^{-k}.$$

Faithfulness as reproducibility under a fixed interpretation rule

Definition (Completely faithful explanation)

A string $\boldsymbol{e} \in \Lambda^*$ is a completely faithful explanation of a function

$$f : \mathcal{Z} \xrightarrow{\text{comp}} \mathbb{R}$$

under Interpret if, for every $z \in \mathcal{Z}$ and every $k \in \mathbb{N}$,

$$|\text{Interpret}(\boldsymbol{e}, k, z) - f(z)| < |\Sigma|^{-k}.$$

Remark

Considering the approximation is necessary since a computer cannot directly handle irrational numbers.

Formulating the length of an explanation

The length of the explanation is $|e|$, the length as a sequence.

$|\texttt{A good AI}| = 9$.

Key idea for formulating the operating environment complexity

How do we quantify the complexity of the input-output relation?

Key idea for formulating the operating environment complexity

How do we quantify the complexity of the input-output relation?

If the relation is simple, we can find a simple rule that generates the output from the given input.

Key idea for formulating the operating environment complexity

How do we quantify the complexity of the input-output relation?

If the relation is simple, we can find a simple rule that generates the output from the given input.

Our idea: Use **the shortest program length** generating the output from the given input as a measure of the input-output relation complexity.

Key idea for formulating the operating environment complexity

How do we quantify the complexity of the input-output relation?

If the relation is simple, we can find a simple rule that generates the output from the given input.

Our idea: Use **the shortest program length** generating the output from the given input as a measure of the input-output relation complexity.

This idea leads us to the definition of **conditional plain Kolmogorov complexity**.

Formulation of the operating environment complexity

Definition (Conditional plain Kolmogorov complexity)

Fix a **universal conditional function** U (like a Python interpreter). For $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^*$, the **conditional plain Kolmogorov complexity** is defined by

$$C_U(y \mid x) := \min \left\{ |\boldsymbol{p}| \;\middle|\; U\left(\overline{\boldsymbol{x}} \cdot \boldsymbol{p}\right) \downarrow = \boldsymbol{y} \right\}.$$

$\overline{\bullet}$ is a fixed self-delimiting encoder function.

Formulation of the operating environment complexity

Definition (Conditional plain Kolmogorov complexity)

Fix a **universal conditional function** U (like a Python interpreter). For $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^*$, the **conditional plain Kolmogorov complexity** is defined by

$$C_U(y \mid x) := \min \left\{ |\boldsymbol{p}| \;\middle|\; U\left(\overline{\boldsymbol{x}} \cdot \boldsymbol{p}\right) \downarrow = \boldsymbol{y} \right\}.$$

$\overline{\bullet}$ is a fixed self-delimiting encoder function.

U can be regarded as a programming language interpreter, and $\boldsymbol{p}$ can be regarded as a program.

Formulation of the operating environment complexity

Definition (Conditional plain Kolmogorov complexity)

Fix a **universal conditional function** U (like a Python interpreter). For $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^*$, the **conditional plain Kolmogorov complexity** is defined by

$$C_U(y \mid x) := \min \left\{ |\boldsymbol{p}| \;\middle|\; U\left(\overline{\boldsymbol{x}} \cdot \boldsymbol{p}\right) \downarrow = \boldsymbol{y} \right\}.$$

$\overline{\bullet}$ is a fixed self-delimiting encoder function.

U can be regarded as a programming language interpreter, and $\boldsymbol{p}$ can be regarded as a program.

We use $\mathbb{E}_{X,Y} C_U(Y \mid X)$ as a metric of the operating environment complexity.

Formulation of the operating environment complexity

Definition (Conditional plain Kolmogorov complexity)

Fix a **universal conditional function** U (like a Python interpreter). For $\boldsymbol{x}, \boldsymbol{y} \in \Sigma^*$, the **conditional plain Kolmogorov complexity** is defined by

$$C_U(y \mid x) := \min \left\{ |\boldsymbol{p}| \;\middle|\; U\left(\overline{\boldsymbol{x}} \cdot \boldsymbol{p}\right) \downarrow = \boldsymbol{y} \right\}.$$

$\overline{\bullet}$ is a fixed self-delimiting encoder function.

U can be regarded as a programming language interpreter, and $\boldsymbol{p}$ can be regarded as a program.

We use $\mathbb{E}_{X,Y} C_U(Y \mid X)$ as a metric of the operating environment complexity.

Note: changing U only changes an additive constant independent of $\boldsymbol{x}$ and $\boldsymbol{y}$.

Main Result

Fundmental Quadrilemma: a rough formula

Our theorem roughly states that:

Fundmental Quadrilemma: a rough formula

Our theorem roughly states that:

Up to logarithmic terms,

$$(\text{perplexity}) + (\text{faithful explanation length}) \geq (\text{conditional Kolmogorov complexity})$$

Fundmental Quadrilemma: a rough formula

Our theorem roughly states that:

Up to logarithmic terms,

$$(\text{perplexity}) + (\text{faithful explanation length}) \geq (\text{conditional Kolmogorov complexity})$$

Implication: If the operating environment (task) is complex, then the RHS is large. Then, either of the following holds:

Fundmental Quadrilemma: a rough formula

Our theorem roughly states that:

Up to logarithmic terms,

$$(\text{perplexity}) + (\text{faithful explanation length}) \geq (\text{conditional Kolmogorov complexity})$$

Implication: If the operating environment (task) is complex, then the RHS is large. Then, either of the following holds:

- The perplexity is large (poor performance), or

Fundmental Quadrilemma: a rough formula

Our theorem roughly states that:

Up to logarithmic terms,

$$(\text{perplexity}) + (\text{faithful explanation length}) \geq (\text{conditional Kolmogorov complexity})$$

Implication: If the operating environment (task) is complex, then the RHS is large. Then, either of the following holds:

- The perplexity is large (poor performance), or
- The (faithful) explanation is long (uninterpretable explanation).

Fundamental Quadrilemma: mathematical form

Theorem (Fundamental Quadrilemma)

Fix a universal function U, an encoding function an alphabet Σ, an interpretation function with character set Λ, and $L_\Lambda := \max_{\lambda \in \Lambda} \left|\mathsf{Enc}_\Lambda^{\mathsf{PF}}(\lambda)\right|$.

Suppose that for every $x \in \mathcal{X}$, a completely faithful explanation $\boldsymbol{e}_x$ of $Q_A(\cdot \mid x)$ is given. For every probability distribution P on $\mathcal{X} \times \mathcal{Y}$,

$$\begin{aligned}&\mathbb{E}_{X,Y\sim P}\big[-\log Q_A(Y \mid X)\big] + \mathbb{E}_{X,Y\sim P}\left[L_\Lambda|\boldsymbol{e}_X| + 2\log\left(L_\Lambda|\boldsymbol{e}_X| + 1\right)\right]\\ &\geq \mathbb{E}_{X,Y\sim P} C_U(Y \mid X) + c.\end{aligned}$$

The constant c does not depend on A, x, y, or P.

If Λ is Unicode characters and UTF-8 for encoding, $L_\Lambda = 32$.

Meaning of the inequality

Remark (Trade-off)

The theorem implies that the following cannot all be small or favorable:

1. performance (expected logarithmic perplexity): $\mathbb{E}[-\log Q_A(Y \mid X)]$,
2. expected faithful explanation length, $\mathbb{E}[|e_X|]$,
3. environment complexity, $\mathbb{E}C_U(Y \mid X)$.

Remark

Recall that a lengthy explanation is NOT interpretable.

Implication for AI governance

Remark (Governance implication)

In many practical applications:

1. the operation environment is complex,
2. good AI performance is required,
3. interpretable explanations are required.

Therefore, the condition to give up is usually complete faithfulness.

Remark

AI governance should be designed on the premise that AI explanations are incomplete in faithfulness.

Local and global explanations

Remark

The theorem is stated for local explanations e_x of $Q_A(\cdot \mid x)$. A global explanation of $Q_A(\cdot \mid \cdot)$ must be able to recover each local behavior $Q_A(\cdot \mid x)$. Hence, up to fixed constants, the lower-bound logic also applies to global explanations.

Hypothesis on the environment complexity.

Since the environment itself is complex, computing its complexity (e.g., $\mathbb{E}C_U(Y \mid X)$) is, in principle, a difficult problem.

Hypothesis on the environment complexity.

Since the environment itself is complex, computing its complexity (e.g., $\mathbb{E}C_U(Y \mid X)$) is, in principle, a difficult problem.

Nevertheless, our quadrilemma theorem is suggestive. If an AI that achieves the true input-output relation and is written by a short program were realized, then $\mathbb{E}_{X,Y\sim P}C_U(Y \mid X)$ would be small, and the quadrilemma would not be a problem. However, considering the history in which text generation could not achieve sufficient performance until the advent of large-scale language models, one can formulate the hypothesis that such a situation will not occur.

Conclusion

Summary

Since shortness is a necessary condition for human interpretability, the theorem gives a mathematical quadrilemma among:

1. environment complexity,
2. AI performance,
3. interpretability,
4. complete faithfulness.

Summary

Since shortness is a necessary condition for human interpretability, the theorem gives a mathematical quadrilemma among:

1. environment complexity,
2. AI performance,
3. interpretability,
4. complete faithfulness.

AI governance should be designed on the premise that AI explanations are incomplete in faithfulness.

References

References i

[1] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information fusion, 58:82–115, 2020.

[2] Marc Brysbaert. How many words do we read per minute? a review and meta-analysis of reading rate. Journal of memory and language, 109: 104047, 2019.

References ii

[3] Guinness World Records. Most decimal places of pi memorized, 2015. URL https://www.guinnessworldrecords.com/world-records/most-pi-places-memorised. Record achieved by Rajveer Meena at VIT University, Vellore, India, on 21 March 2015.

[4] Jan Hatzius, Joseph Briggs, and Devesh Kodnani. The potentially large effects of artificial intelligence on economic growth. Technical report, Goldman Sachs Global Investment Research, 2023.

[5] Prabha M Kumarage and Mirka Saarela. Explainable generative ai: A two-stage review of existing techniques and future research directions. AI, 7 (1):31, 2026.

References iii

[6] McKinsey Global Institute. The economic potential of generative ai: The next productivity frontier, 2023.

[7] National Institute of Standards and Technology. Artificial intelligence risk management framework: Generative artificial intelligence profile. Technical Report NIST AI 600-1, National Institute of Standards and Technology, 2024. URL https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf.

[8] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI Blog, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

References iv

[9] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144, 2016.

[10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10684–10695, 2022.

Appendix

Stochastic input-output AI

Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ be the output space.

Definition (Stochastic input-output AI)

A stochastic input-output AI with domain $\mathcal{X}' \subseteq \mathcal{X}$ is a pair

$$A = (f, \tau),$$

where

$$f :\subseteq (\mathbb{N}_{<n})^* \times \mathcal{X} \xrightarrow{\text{comp}} \mathcal{Y}$$

is the main function, and

$$\tau :\subseteq (\mathbb{N}_{<n})^* \times \mathcal{X} \xrightarrow{\text{comp}} \{0, 1\}$$

is the random-sequence acceptance decision function.

Accepted random sequences

For each input $x \in \mathcal{X}'$, define

$$\mathcal{T}_x := \{\boldsymbol{u} \in (\mathbb{N}_{<n})^* \mid \tau(\boldsymbol{u}, x) \downarrow = 1\}.$$

Definition (Key requirements)

For each $x \in \mathcal{X}'$:

1. $\mathcal{T}_x$ is computable,
2. $\mathcal{T}_x$ is prefix-free,
3. a computable stopping bound $N_{\mathrm{stop}}(x, k)$ satisfies

$$1 - \sum_{\substack{\boldsymbol{u} \in \mathcal{T}_x \\ |\boldsymbol{u}| \leq N_{\mathrm{stop}}(x,k)}} n^{-|\boldsymbol{u}|} \leq |\Sigma|^{-k},$$

4. if $\boldsymbol{u} \in \mathcal{T}_x$, then $(\boldsymbol{u}, x) \in \mathrm{dom}\, f$.

Intended stochastic operation

Definition (Intended operation)

Given $x \in \mathcal{X}'$, the AI repeatedly samples symbols from a uniform oracle $\mathsf{Uniform}_{<n}$ and concatenates them into a random string $\boldsymbol{u}$.

When $\boldsymbol{u} \in \mathcal{T}_x$, it outputs

$$f(\boldsymbol{u}, x)$$

and terminates.

Remark

This framework includes deterministic functions, diffusion models with fixed random-number consumption, and LLMs whose number of consumed random numbers depends on generated tokens and termination conditions.

Why Not Use the Naive Complexity?

Naive idea

A natural first attempt is to measure environment complexity by the Kolmogorov complexity of the true conditional probability mass function:

$$C_U\big(P(\cdot \mid \cdot)\big).$$

Proposition (Naive exact-minimization result)

If Q has a completely faithful explanation $\boldsymbol{e}$ with

$$L_\Lambda|\boldsymbol{e}| < C_U(P) - c_{\mathsf{Interpret}},$$

then Q cannot exactly minimize expected logarithmic conditional perplexity.

Remark

This only rules out exact minimization.

Failure of the naive method

Proposition (Failure of the naive lower bound)

Let $\mathcal{X}$ and $\mathcal{Y}$ be countably infinite. For every $n \in \mathbb{N}$ and every $\epsilon > 0$, there exist π, computable P, Q, and a completely faithful explanation e of Q such that

$$\mathbb{E}_{X\sim\pi,Y|X\sim P}[-\log Q(Y \mid X)] - \min_{Q^*} \mathbb{E}_{X\sim\pi,Y|X\sim P}[-\log Q^*(Y \mid X)] < \epsilon,$$

but

$$|e| < C_U(P) - n.$$

Remark

Thus $C_U(P)$ does not give the desired performance-explanation trade-off near the optimum.

Why expected conditional Kolmogorov complexity is used

Remark (Chosen complexity indicator)

The theorem uses

$$\mathbb{E}_{X,Y\sim P} C_U(Y \mid X)$$

because it measures the algorithmic complexity of individual true outputs conditioned on inputs.

Remark

Replacing it by entropy alone loses the implication for explanation length, because Gibbs' inequality already gives

$$\mathbb{E}_{X,Y\sim P}[-\log Q(Y \mid X)] \geq H_P(Y \mid X).$$

Gap from entropy

Proposition (Kolmogorov complexity can exceed entropy)

For every $n \in \mathbb{N}$, there exists a probability distribution P such that

$$\mathbb{E}_{X,Y\sim P} K_V(Y \mid X) \geq H_P(Y \mid X) + n.$$

Remark

This shows that the main theorem can yield a stronger lower bound than an entropy-only argument.