Epistemic Exams

Loren K. Fryxell

City St George's, University of London

loren.fryxell@citystgeorges.ac.uk

April 18, 2026

5th Southeast Theory Festival

Introduction

Introduction

An **exam** measures what you know.

Introduction

An **exam** measures what you know.

An **epistemic exam** measures what you know *and* what you know you know.

Example

Student 1 gets 6/10 correct answers on an exam

Result. C C C C C C W W W W

Example

Student 1 gets 6/10 correct answers on an exam

Result. C C C C C C W W W W

and after the exam reports *"I'm very confident my first 6 answers are correct. Moreover, though these are my best guesses, I'm very confident my last 4 answers are wrong."*

Beliefs. C C C C C C W W W W

Example

Student 1 gets 6/10 correct answers on an exam

Result. C C C C C C W W W W

and after the exam reports *"I'm very confident my first 6 answers are correct. Moreover, though these are my best guesses, I'm very confident my last 4 answers are wrong."*

Beliefs. C C C C C C W W W W

Student 2 gets 8/10 correct answers on an exam

Result. C C C C C C C C W W

Example

Student 1 gets 6/10 correct answers on an exam

Result. C C C C C C W W W W

and after the exam reports *"I'm very confident my first 6 answers are correct. Moreover, though these are my best guesses, I'm very confident my last 4 answers are wrong."*

Beliefs. C C C C C C W W W W

Student 2 gets 8/10 correct answers on an exam

Result. C C C C C C C C W W

and after the exam reports *"I'm very confident my last 8 answers are correct. Moreover, though these are my best guesses, I'm very confident my first 2 answers are wrong."*

Beliefs. W W C C C C C C C C

Example

Student 1	:	Result.	C C C C C C W W W W
		Beliefs.	C C C C C C W W W W
Student 2	:	Result.	C C C C C C C C W W
		Beliefs.	W W C C C C C C C C

Example

Student 1	:	Result.	C C C C C C W W W W
		Beliefs.	C C C C C C W W W W
Student 2	:	Result.	C C C C C C C C W W
		Beliefs.	W W C C C C C C C C

- Student 2 **knows** more than Student 1.

Example

Student 1	:	Result.	C C C C C C W W W W
		Beliefs.	C C C C C C W W W W
Student 2	:	Result.	C C C C C C C C W W
		Beliefs.	W W C C C C C C C C

- Student 2 **knows** more than Student 1.
  - That is, Student 2 will get a higher fraction of her answers correct than Student 1, given similar questions

Example

Student 1	:	Result.	C C C C C C W W W W
		Beliefs.	C C C C C C W W W W
Student 2	:	Result.	C C C C C C C C W W
		Beliefs.	W W C C C C C C C C

- Student 2 **knows** more than Student 1.
  - That is, Student 2 will get a higher fraction of her answers correct than Student 1, given similar questions
- Student 1 **knows what she knows** more than Student 2.

Example

Student 1	:	Result.	C C C C C C W W W W
		Beliefs.	C C C C C C W W W W
Student 2	:	Result.	C C C C C C C C W W
		Beliefs.	W W C C C C C C C C

- Student 2 **knows** more than Student 1.
  - That is, Student 2 will get a higher fraction of her answers correct than Student 1, given similar questions
- Student 1 **knows what she knows** more than Student 2.
  - That is, Student 1 can perfectly identify which of her answers will be correct and which will be wrong, while Student 2 cannot

Example

How should we score each of these students?

Example

How should we score each of these students?

That's what this paper is about

Example

How should we score each of these students?

That's what this paper is about

Food for Thought and Preview

Example

How should we score each of these students?

That's what this paper is about

Food for Thought and Preview

- Would you prefer to hire Student 1 or Student 2?

Example

How should we score each of these students?

That's what this paper is about

Food for Thought and Preview

- Would you prefer to hire Student 1 or Student 2?
- It depends on the context

Example

How should we score each of these students?

That's what this paper is about

Food for Thought and Preview

- Would you prefer to hire Student 1 or Student 2?
- It depends on the context
  - Sometimes it's better to have the student who knows more, even if they know what they know less

Example

How should we score each of these students?

That's what this paper is about

Food for Thought and Preview

- Would you prefer to hire Student 1 or Student 2?
- It depends on the context
  - Sometimes it's better to have the student who knows more, even if they know what they know less
  - Other times it's better to have the student who knows what they know more, even if they know less

Example

How should we score each of these students?

That's what this paper is about

Food for Thought and Preview

- Would you prefer to hire Student 1 or Student 2?
- It depends on the context
  - Sometimes it's better to have the student who knows more, even if they know what they know less
  - Other times it's better to have the student who knows what they know more, even if they know less
- The desired scoring rule will be a function of the characteristics of the situation you face

This Paper

The purpose of this paper is to

This Paper

The purpose of this paper is to

1. Introduce the concept of an epistemic exam

This Paper

The purpose of this paper is to

1. Introduce the concept of an epistemic exam
2. Develop a general theory for how to score epistemic exams

This Paper

The purpose of this paper is to

1. Introduce the concept of an epistemic exam
2. Develop a general theory for how to score epistemic exams
3. Argue that **AI capabilities** should be measured using epistemic exams, and develop a **specific** scoring rule to score AI models (which I call FABLE)

This Paper

The purpose of this paper is to

1. Introduce the concept of an epistemic exam
2. Develop a general theory for how to score epistemic exams
3. Argue that **AI capabilities** should be measured using epistemic exams, and develop a **specific** scoring rule to score AI models (which I call FABLE)

For AI, the purpose is twofold:

This Paper

The purpose of this paper is to

1. Introduce the concept of an epistemic exam
2. Develop a general theory for how to score epistemic exams
3. Argue that **AI capabilities** should be measured using epistemic exams, and develop a **specific** scoring rule to score AI models (which I call FABLE)

For AI, the purpose is twofold:

1. To better measure what we actually care about when it comes to AI capabilities

This Paper

The purpose of this paper is to

1. Introduce the concept of an epistemic exam
2. Develop a general theory for how to score epistemic exams
3. Argue that **AI capabilities** should be measured using epistemic exams, and develop a **specific** scoring rule to score AI models (which I call FABLE)

For AI, the purpose is twofold:

1. To better measure what we actually care about when it comes to AI capabilities
2. To help train AIs to behave closer to how we would like them to behave

This Paper

The purpose of this paper is to

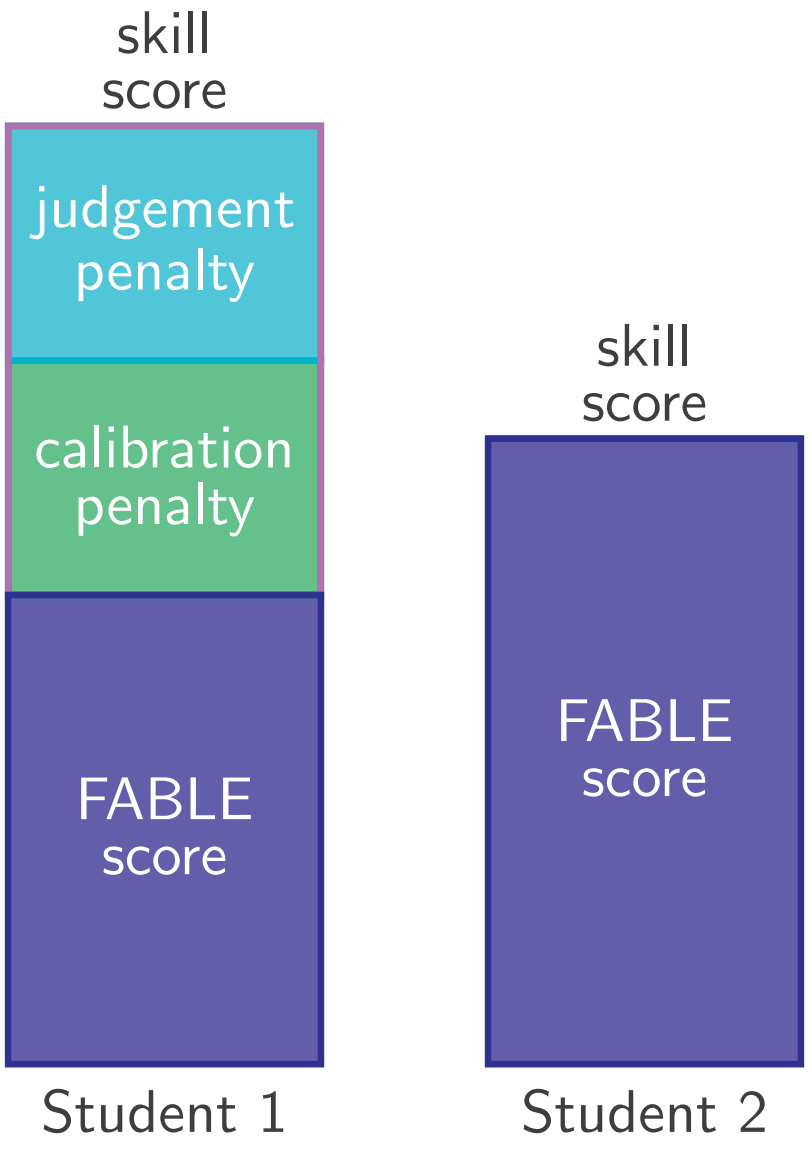
1. Introduce the concept of an epistemic exam
2. Develop a general theory for how to score epistemic exams
3. Argue that **AI capabilities** should be measured using epistemic exams, and develop a **specific** scoring rule to score AI models (which I call FABLE)

For AI, the purpose is twofold:

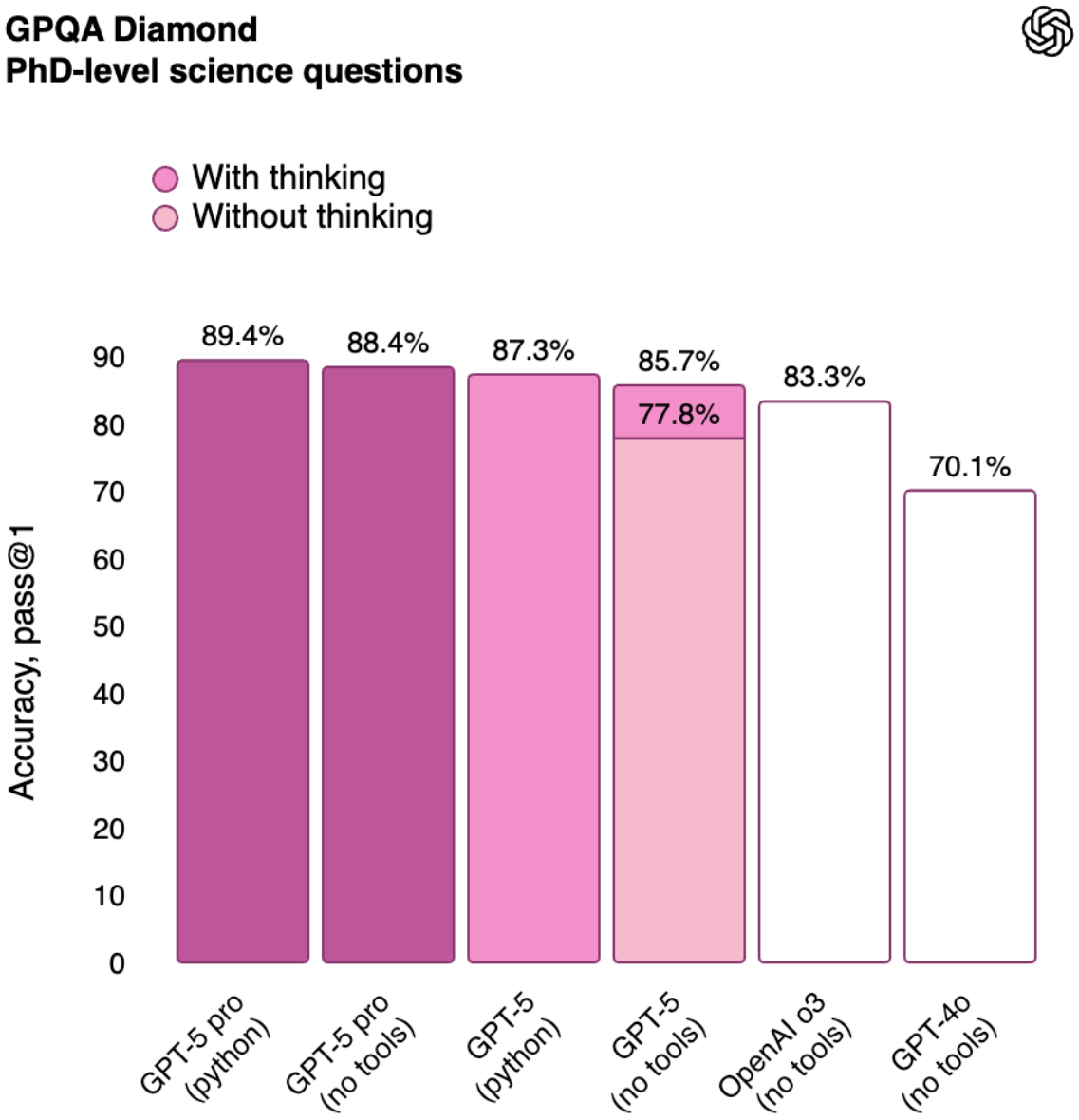
1. To better measure what we actually care about when it comes to AI capabilities
2. To help train AIs to behave closer to how we would like them to behave

In particular, current AIs know a lot, but don't always know what they know, and as a result are **unreliable**

Preview


skill
score
judgement
penalty
calibration
penalty
FABLE
score
Student 1
skill
score
FABLE
score
Student 2

Instead Of


GPQA Diamond
PhD-level science questions
With thinking
Without thinking
Accuracy, pass@1
0
10
20
30
40
50
60
70
80
90
89.4%
88.4%
87.3%
85.7%
77.8%
83.3%
70.1%
GPT-5 pro (python)
GPT-5 pro (no tools)
GPT-5 (python)
GPT-5 (no tools)
OpenAI o3 (no tools)
GPT-4o (no tools)

Roadmap

Roadmap

1. Scoring Rules

Roadmap

1. Scoring Rules
2. Incentive-Compatible Scoring Rules

Roadmap

1. Scoring Rules
2. Incentive-Compatible Scoring Rules
3. Utilitarian Scoring Rules

Roadmap

1. Scoring Rules
2. Incentive-Compatible Scoring Rules
3. Utilitarian Scoring Rules
4. FABLE

Scoring Rules

Forecasting

Forecasting Environment

A *forecaster* is an agent who is given an event and is asked to report their subjective belief that the event will occur

Forecasting Environment

A *forecaster* is an agent who is given an event and is asked to report their subjective belief that the event will occur

A forecaster has a prior belief p about the probability the event occurs

Forecasting Environment

A *forecaster* is an agent who is given an event and is asked to report their subjective belief that the event will occur

A forecaster has a prior belief p about the probability the event occurs

We would like to incentivize the forecaster to report p by providing a score based on the realized outcome

Forecasting Environment

Definition. A *scoring rule* (S_0, S_1) is a pair of functions $S_0 : [0,1] \to [-\infty, \infty]$ and $S_1 : [0,1] \to [-\infty, \infty]$ where

Forecasting Environment

Definition. A *scoring rule* (S_0, S_1) is a pair of functions $S_0 : [0,1] \to [-\infty, \infty]$ and $S_1 : [0,1] \to [-\infty, \infty]$ where

- $S_0(p)$ is the score given to the reported belief p if the event does not occur and

Forecasting Environment

Definition. A *scoring rule* (S_0, S_1) is a pair of functions $S_0 : [0,1] \to [-\infty, \infty]$ and $S_1 : [0,1] \to [-\infty, \infty]$ where

- $S_0(p)$ is the score given to the reported belief p if the event does not occur and
- $S_1(p)$ is the score given to the reported belief p if the event does occur.

Test-Taking

Test-Taking Environment

A *test-taker* is an agent who is given a question and is asked to report their best answer and their subjective belief that their answer is correct.

Test-Taking Environment

A *test-taker* is an agent who is given a question and is asked to report their best answer and their subjective belief that their answer is correct.

Let X be a finite set of feasible answers and let $p(x)$ be the test-taker's prior belief that the answer x is correct

Test-Taking Environment

A *test-taker* is an agent who is given a question and is asked to report their best answer and their subjective belief that their answer is correct.

Let X be a finite set of feasible answers and let $p(x)$ be the test-taker's prior belief that the answer x is correct

(For simplicity, assume that distinct answers have distinct probabilities, so that $p(x) \neq p(x')$ for all $x \neq x'$)

Test-Taking Environment

A *test-taker* is an agent who is given a question and is asked to report their best answer and their subjective belief that their answer is correct.

Let X be a finite set of feasible answers and let $p(x)$ be the test-taker's prior belief that the answer x is correct

(For simplicity, assume that distinct answers have distinct probabilities, so that $p(x) \neq p(x')$ for all $x \neq x'$)

The agent submits a pair (x, p) where $x \in X$ and $p \in [0, 1]$

Test-Taking Environment

Definition. A *scoring rule* (S_0, S_1) is a pair of functions $S_0 : [0,1] \to [-\infty, \infty]$ and $S_1 : [0,1] \to [-\infty, \infty]$ where

Test-Taking Environment

Definition. A *scoring rule* (S_0, S_1) is a pair of functions $S_0 : [0,1] \to [-\infty, \infty]$ and $S_1 : [0,1] \to [-\infty, \infty]$ where

- $S_0(p)$ is the score given to the reported belief p if the given answer is not correct and

Test-Taking Environment

Definition. A *scoring rule* (S_0, S_1) is a pair of functions $S_0 : [0,1] \to [-\infty, \infty]$ and $S_1 : [0,1] \to [-\infty, \infty]$ where

- $S_0(p)$ is the score given to the reported belief p if the given answer is not correct and
- $S_1(p)$ is the score given to the reported belief p if the given answer is correct.

Incentive-Compatible Scoring Rules

Forecasting

Proper Scoring Rules

Definition. A scoring rule (S_0, S_1) is *(strictly) proper* if the forecaster (uniquely) maximizes her expected score by reporting her true belief.

Proper Scoring Rules

Definition. A scoring rule (S_0, S_1) is *(strictly) proper* if the forecaster (uniquely) maximizes her expected score by reporting her true belief. That is, for any $p^* \in [0, 1]$ and $p \neq p^*$,

$$p^* S_1(p^*) + (1 - p^*) S_0(p^*) \geq (>)\, p^* S_1(p) + (1 - p^*) S_0(p).$$

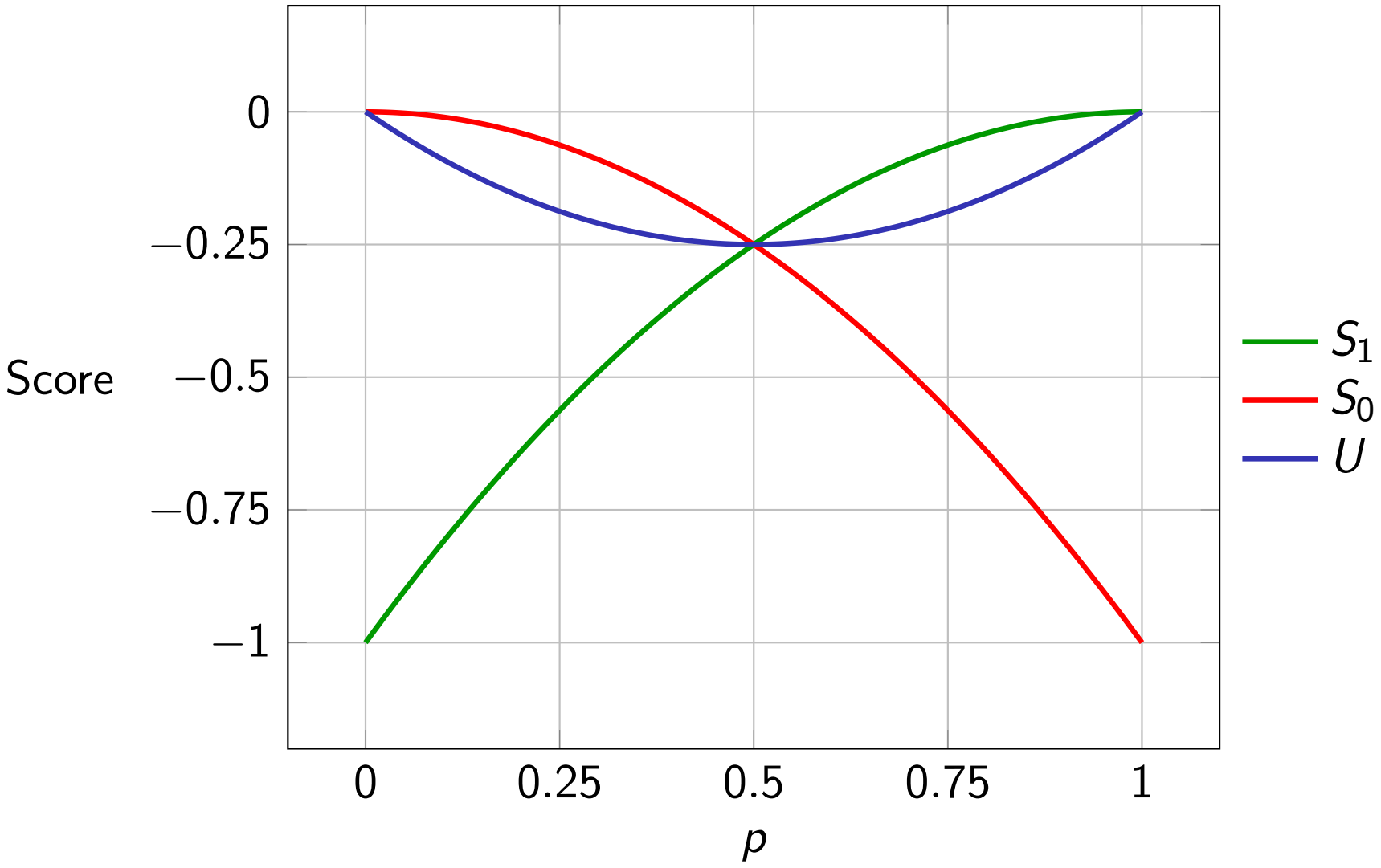
Quadratic Scoring Rule

For example, the *quadratic scoring rule* is defined by

$$S_0(p) = -p^2 \quad \text{and} \quad S_1(p) = -(1-p)^2$$

and is a strictly proper scoring rule.

Quadratic Scoring Rule



Characterization of Proper Scoring Rules

Theorem 1 (McCarthy, Savage). A scoring rule S is (strictly) proper if and only if

$$S_0(p) = U(p) - pU'(p) \quad \text{and} \quad S_1(p) = U(p) + (1-p)U'(p)$$

where $U : [0,1] \to \mathbb{R}$ is a (strictly) convex function.

Characterization of Proper Scoring Rules

Theorem 1 (McCarthy, Savage). A scoring rule S is (strictly) proper if and only if

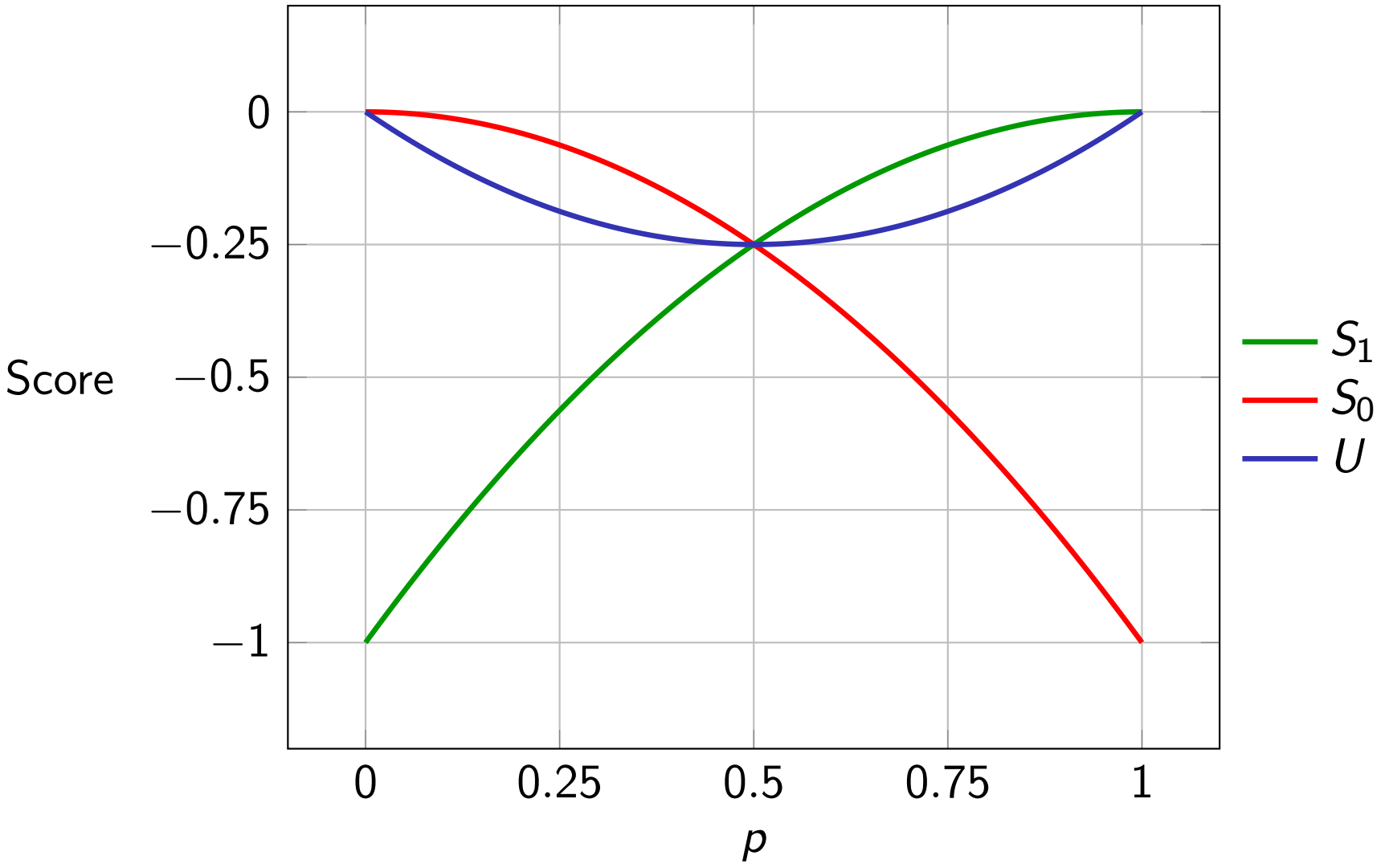
$$S_0(p) = U(p) - pU'(p) \quad \text{and} \quad S_1(p) = U(p) + (1-p)U'(p)$$

where $U : [0,1] \to \mathbb{R}$ is a (strictly) convex function.

Moreover, U is precisely the expected score function, i.e.,

$$U(p) = pS_1(p) + (1-p)S_0(p).$$

Quadratic Scoring Rule



Test-Taking

Proper Epistemic Exam Scoring Rules

Definition. A scoring rule (S_0, S_1) is a *(strictly) proper epistemic exam scoring rule* if the test-taker (uniquely) maximizes her expected score by submitting her best answer x^* along with her true belief that it is correct $p(x^*)$, no matter the feasible set of answers X.

Proper Epistemic Exam Scoring Rules

Definition. A scoring rule (S_0, S_1) is a *(strictly) proper epistemic exam scoring rule* if the test-taker (uniquely) maximizes her expected score by submitting her best answer x^* along with her true belief that it is correct $p(x^*)$, no matter the feasible set of answers X.

That is, for any X, if $x \neq x^*$ or $\pi \neq p(x^*)$, then

$$p(x^*)S_1(p(x^*)) + (1 - p(x^*))S_0(p(x^*)) \geq (>)\, p(x)S_1(\pi) + (1 - p(x))S_0(\pi),$$

where $x^* \in \arg\max_{x \in X} p(x)$.

Characterization of Proper Epistemic Exam Scoring Rules

Theorem 2. A scoring rule (S_0, S_1) is a (strictly) proper epistemic exam scoring rule if and only if

Characterization of Proper Epistemic Exam Scoring Rules

Theorem 2. A scoring rule (S_0, S_1) is a (strictly) proper epistemic exam scoring rule if and only if

1. (S_0, S_1) is (strictly) proper, and

Characterization of Proper Epistemic Exam Scoring Rules

Theorem 2. A scoring rule (S_0, S_1) is a (strictly) proper epistemic exam scoring rule if and only if

1. (S_0, S_1) is (strictly) proper, and
2. the expected payoff of reporting an answer with true belief p, $U(p) = pS_1(p) + (1-p)S_0(p)$, is (strictly) increasing in p.

Characterization of Proper Epistemic Exam Scoring Rules

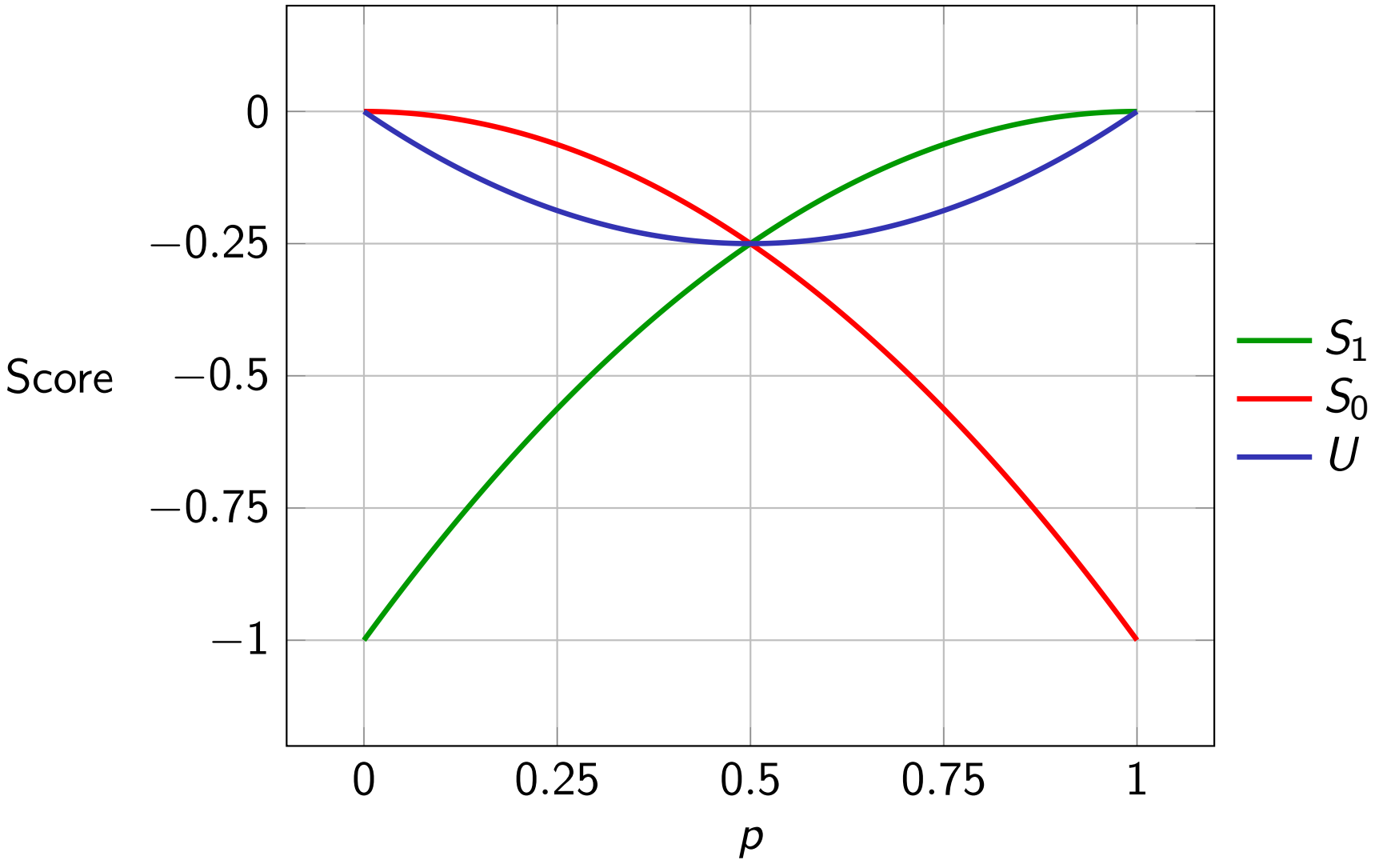
Theorem 2. A scoring rule (S_0, S_1) is a (strictly) proper epistemic exam scoring rule if and only if

1. (S_0, S_1) is (strictly) proper, and
2. the expected payoff of reporting an answer with true belief p, $U(p) = pS_1(p) + (1-p)S_0(p)$, is (strictly) increasing in p.

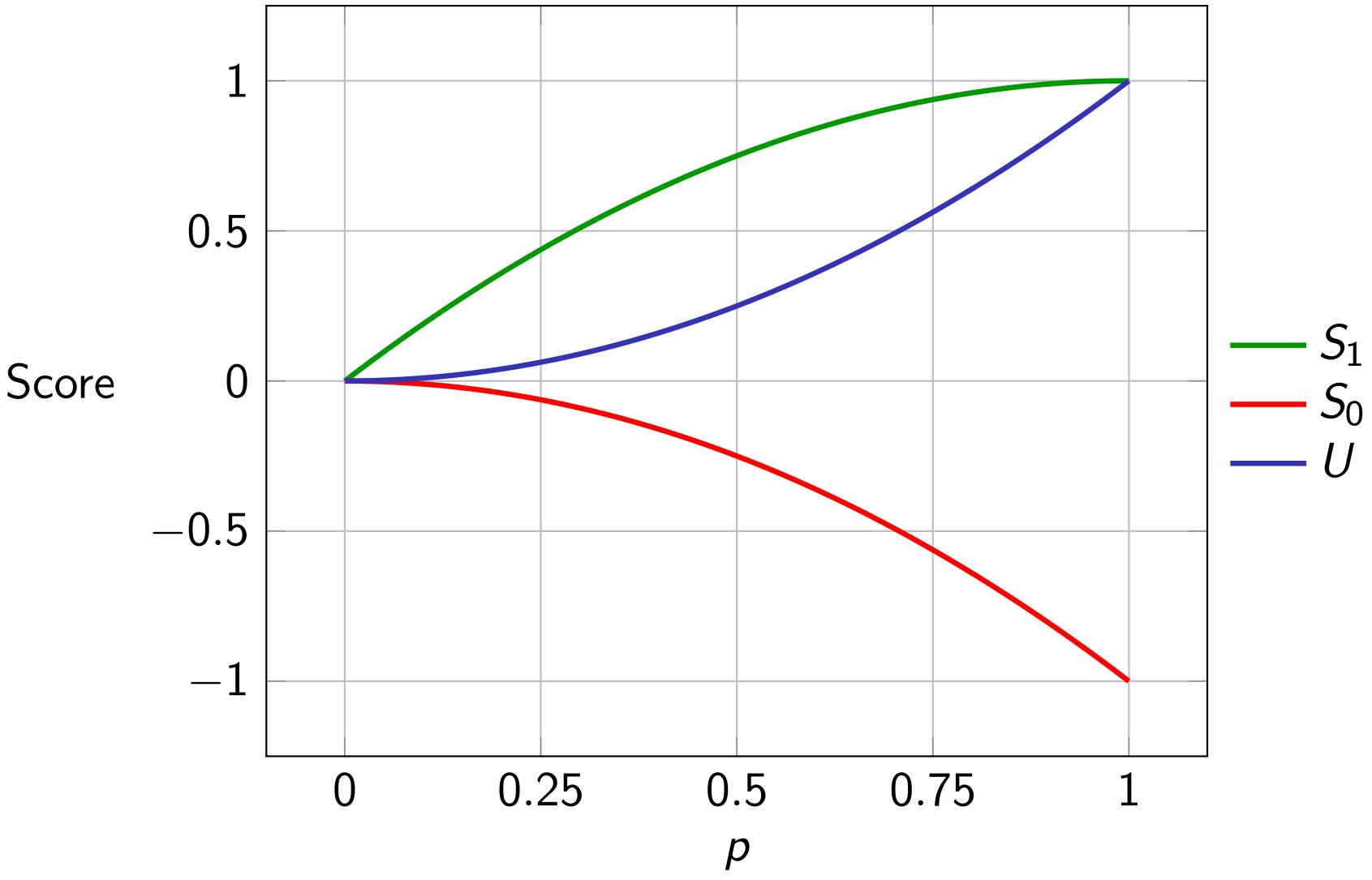
Alternatively, (2) can be replaced by

3. the score for correct answers is everywhere above the score for incorrect answers, $S_1(p) \geq S_0(p)$ for all p.

Quadratic Scoring Rule



Quadratic Epistemic Exam Scoring Rule



Zooming Out

For any desired expected payoff for a truthful agent U, then there are infinitely many scoring rules (S_0, S_1) that generate it (e.g., $S_1 = S_0 = U$)

Zooming Out

For any desired expected payoff for a truthful agent U, then there are infinitely many scoring rules (S_0, S_1) that generate it (e.g., $S_1 = S_0 = U$)

If U is **not convex**, then none of these scoring rules are proper scoring rules

Zooming Out

For any desired expected payoff for a truthful agent U, then there are infinitely many scoring rules (S_0, S_1) that generate it (e.g., $S_1 = S_0 = U$)

If U is **not convex**, then none of these scoring rules are proper scoring rules

If U is **convex**, then exactly one of these scoring rules is a proper scoring rule

Zooming Out

For any desired expected payoff for a truthful agent U, then there are infinitely many scoring rules (S_0, S_1) that generate it (e.g., $S_1 = S_0 = U$)

If U is **not convex**, then none of these scoring rules are proper scoring rules

If U is **convex**, then exactly one of these scoring rules is a proper scoring rule

If U is **convex and increasing**, then exactly one of these scoring rules is a proper epistemic exam scoring rule

Utilitarian Scoring Rules

Utilitarian Scoring Rule

A *utilitarian scoring rule* is a scoring rule that reflects the payoff to society of employing the agent

Utilitarian Scoring Rule

A *utilitarian scoring rule* is a scoring rule that reflects the payoff to society of employing the agent

The payoff to society can be thought of as the payoff to

Utilitarian Scoring Rule

A *utilitarian scoring rule* is a scoring rule that reflects the payoff to society of employing the agent

The payoff to society can be thought of as the payoff to

- the end-user of the forecast in the forecasting environment, or

Utilitarian Scoring Rule

A *utilitarian scoring rule* is a scoring rule that reflects the payoff to society of employing the agent

The payoff to society can be thought of as the payoff to

- the end-user of the forecast in the forecasting environment, or
- the end-user of the answer in the test-taking environment

Forecasting

Forecasting Environment with User

Let A be the set of all feasible actions a user can take upon receiving some forecast p

Forecasting Environment with User

Let A be the set of all feasible actions a user can take upon receiving some forecast p

E.g., $A = \{\text{bring umbrella, don't bring umbrella}\}$

Forecasting Environment with User

Let A be the set of all feasible actions a user can take upon receiving some forecast p

E.g., $A = \{\text{bring umbrella, don't bring umbrella}\}$

Let $U_0 : A \to \mathbb{R}$ and $U_1 : A \to \mathbb{R}$ where

Forecasting Environment with User

Let A be the set of all feasible actions a user can take upon receiving some forecast p

E.g., $A = \{\text{bring umbrella, don't bring umbrella}\}$

Let $U_0 : A \to \mathbb{R}$ and $U_1 : A \to \mathbb{R}$ where

- $U_0(a)$ is the user's payoff of taking action a given the forecasted event does not occur

Forecasting Environment with User

Let A be the set of all feasible actions a user can take upon receiving some forecast p

E.g., $A = \{\text{bring umbrella, don't bring umbrella}\}$

Let $U_0 : A \to \mathbb{R}$ and $U_1 : A \to \mathbb{R}$ where

- $U_0(a)$ is the user's payoff of taking action a given the forecasted event does not occur
- $U_1(a)$ is the user's payoff of taking action a given the forecasted event does occur

Utilitarian Scoring Rule

Upon receiving forecast p, the user chooses an action

$$a^*(p) \in \arg\max_{a \in A} \; pU_1(a) + (1-p)U_0(a)$$

Utilitarian Scoring Rule

Upon receiving forecast p, the user chooses an action

$$a^*(p) \in \arg\max_{a \in A} \; pU_1(a) + (1-p)U_0(a)$$

Definition. A *utilitarian scoring rule* is a scoring rule that reflects the expected utility to society of employing that agent, i.e., for all $p \in [0, 1]$,

$$S_0(p) = U_0(a^*(p)) \quad \text{and} \quad S_1(p) = U_1(a^*(p)).$$

Proposition 1

Proposition 1. In a forecasting environment, every utilitarian scoring rule is a proper scoring rule.

Proposition 1

Proposition 1. In a forecasting environment, every utilitarian scoring rule is a proper scoring rule.

Intuitively, the forecaster and the user are perfectly aligned—and the user makes optimal decisions given the forecast—so the forecaster wants to give the user the true forecast

Quick Aside

There's a long history of work on proper scoring rules, which I'm slowly beginning to familiarize myself with

Quick Aside

There's a long history of work on proper scoring rules, which I'm slowly beginning to familiarize myself with

Some of this work is on how to select the most desirable scoring rule for a given context

Quick Aside

There's a long history of work on proper scoring rules, which I'm slowly beginning to familiarize myself with

Some of this work is on how to select the most desirable scoring rule for a given context

Most of them solve for the rule which maximizes/minimizes some non-payoff characteristic of the rule

Quick Aside

There's a long history of work on proper scoring rules, which I'm slowly beginning to familiarize myself with

Some of this work is on how to select the most desirable scoring rule for a given context

Most of them solve for the rule which maximizes/minimizes some non-payoff characteristic of the rule

But this basic observation seems to me like quite a natural option—just estimate the payoffs of the end user and use that as your score

Quick Aside

There's a long history of work on proper scoring rules, which I'm slowly beginning to familiarize myself with

Some of this work is on how to select the most desirable scoring rule for a given context

Most of them solve for the rule which maximizes/minimizes some non-payoff characteristic of the rule

But this basic observation seems to me like quite a natural option—just estimate the payoffs of the end user and use that as your score

I'd be curious afterwards if you have any references or thoughts on this

Test-Taking

Test-Taking Environment with User

The test-taking environment is again the forecasting environment plus a set X of possible answers the user can receive from the agent:

Test-Taking Environment with User

The test-taking environment is again the forecasting environment plus a set X of possible answers the user can receive from the agent:

- X is the set of possible answers the agent can give

Test-Taking Environment with User

The test-taking environment is again the forecasting environment plus a set X of possible answers the user can receive from the agent:

- X is the set of possible answers the agent can give
- A is the set of feasible actions the user can take upon receiving answer x and confidence p

Test-Taking Environment with User

The test-taking environment is again the forecasting environment plus a set X of possible answers the user can receive from the agent:

- X is the set of possible answers the agent can give
- A is the set of feasible actions the user can take upon receiving answer x and confidence p

E.g., $A = \{\text{disregard answer, evaluate answer, implement answer}\}$

Test-Taking Environment with User

The test-taking environment is again the forecasting environment plus a set X of possible answers the user can receive from the agent:

- X is the set of possible answers the agent can give
- A is the set of feasible actions the user can take upon receiving answer x and confidence p

E.g., $A = \{\text{disregard answer, evaluate answer, implement answer}\}$

Let $U_0 : A \times X \to \mathbb{R}$ and $U_1 : A \times X \to \mathbb{R}$ where

Test-Taking Environment with User

The test-taking environment is again the forecasting environment plus a set X of possible answers the user can receive from the agent:

- X is the set of possible answers the agent can give
- A is the set of feasible actions the user can take upon receiving answer x and confidence p

E.g., $A = \{\text{disregard answer, evaluate answer, implement answer}\}$

Let $U_0 : A \times X \to \mathbb{R}$ and $U_1 : A \times X \to \mathbb{R}$ where

- $U_0(a, x)$ is the user's payoff of taking action a given the answer x is incorrect

Test-Taking Environment with User

The test-taking environment is again the forecasting environment plus a set X of possible answers the user can receive from the agent:

- X is the set of possible answers the agent can give
- A is the set of feasible actions the user can take upon receiving answer x and confidence p

E.g., $A = \{\text{disregard answer, evaluate answer, implement answer}\}$

Let $U_0 : A \times X \to \mathbb{R}$ and $U_1 : A \times X \to \mathbb{R}$ where

- $U_0(a, x)$ is the user's payoff of taking action a given the answer x is incorrect
- $U_1(a, x)$ is the user's payoff of taking action a given the answer x is correct

Utilitarian Scoring Rule

Upon receiving answer x with confidence p, the user chooses action

$$a^*(p, x) \in \arg\max_{a \in A} \; pU_1(a, x) + (1 - p)U_0(a, x)$$

Utilitarian Scoring Rule

Upon receiving answer x with confidence p, the user chooses action

$$a^*(p, x) \in \arg\max_{a \in A} \; pU_1(a, x) + (1 - p)U_0(a, x)$$

Let $U(p) = \max_{a \in A} \; pU_1(a, x) + (1 - p)U_0(a, x)$ be the user's indirect utility as a function of the confidence p

Utilitarian Scoring Rule

Upon receiving answer x with confidence p, the user chooses action

$$a^*(p, x) \in \arg\max_{a \in A} \; pU_1(a, x) + (1 - p)U_0(a, x)$$

Let $U(p) = \max_{a \in A} \; pU_1(a, x) + (1 - p)U_0(a, x)$ be the user's indirect utility as a function of the confidence p

Assumption. U is weakly increasing (better answers result in higher payoffs for the user)

Utilitarian Scoring Rule

Upon receiving answer x with confidence p, the user chooses action

$$a^*(p, x) \in \arg\max_{a \in A} \; pU_1(a, x) + (1 - p)U_0(a, x)$$

Let $U(p) = \max_{a \in A} \; pU_1(a, x) + (1 - p)U_0(a, x)$ be the user's indirect utility as a function of the confidence p

Assumption. U is weakly increasing (better answers result in higher payoffs for the user)

Definition. A *utilitarian scoring rule* is a scoring rule that reflects the expected utility to society of employing that agent, i.e., for all $p \in [0, 1]$,

$$S_0(p) = U_0(a^*(p, x)) \quad \text{and} \quad S_1(p) = U_1(a^*(p, x)).$$

Proposition 2

Proposition 2. In a test-taking environment, every utilitarian scoring rule is a proper epistemic exam scoring rule.

Proposition 2

Proposition 2. In a test-taking environment, every utilitarian scoring rule is a proper epistemic exam scoring rule.

This follows from Proposition 1 and the fact that U is weakly increasing

Takeaway

In economics, we are usually quite concerned with incentive compatibility

Takeaway

In economics, we are usually quite concerned with incentive compatibility

With AI, it's not obvious that we need their scores to be incentive-compatible

Takeaway

In economics, we are usually quite concerned with incentive compatibility

With AI, it's not obvious that we need their scores to be incentive-compatible

What we do want is that the score measures their usefulness (the payoff to society of using that AI)

Takeaway

In economics, we are usually quite concerned with incentive compatibility

With AI, it's not obvious that we need their scores to be incentive-compatible

What we do want is that the score measures their usefulness (the payoff to society of using that AI)

Proposition 2 shows that if we want to measure an AI's usefulness to society, then that scoring rule will also be incentive-compatible.

FABLE

Deriving FABLE

We are now going to derive FABLE

Deriving FABLE

We are now going to derive FABLE

First, we will derive a simpler FABE scoring rule, and then we will derive the full FABLE scoring rule

Deriving FABLE

We are now going to derive FABLE

First, we will derive a simpler FABE scoring rule, and then we will derive the full FABLE scoring rule

We would like to design a utilitarian scoring rule that reflects the payoffs to society of employing the AI that we are testing

Deriving FABLE

We will consider three actions the user can take:

Deriving FABLE

We will consider three actions the user can take:

1. **Disregard** : Don't implement

Deriving FABLE

We will consider three actions the user can take:

1. **Disregard** : Don't implement
2. **Evaluate** : Evaluate and implement only if the user believes it to be correct

Deriving FABLE

We will consider three actions the user can take:

1. **Disregard** : Don't implement
2. **Evaluate** : Evaluate and implement only if the user believes it to be correct
3. **Implement** : Do implement

The User's (Society's) Payoffs

Let's define the user's payoffs

The User's (Society's) Payoffs

Let's define the user's payoffs

First, the two normalized payoffs

The User's (Society's) Payoffs

Let's define the user's payoffs

First, the two normalized payoffs

- 0 : the status quo payoff of not having an AI agent to help the user, i.e., the payoff of disregarding any answer

The User's (Society's) Payoffs

Let's define the user's payoffs

First, the two normalized payoffs

- 0 : the status quo payoff of not having an AI agent to help the user, i.e., the payoff of disregarding any answer
- 1 : the user's payoff of implementing a correct answer

The User's (Society's) Payoffs

Let's define the user's payoffs

First, the two normalized payoffs

- 0 : the status quo payoff of not having an AI agent to help the user, i.e., the payoff of disregarding any answer
- 1 : the user's payoff of implementing a correct answer

From here, there will be 5 parameters

The 5 Parameters

The 5 Parameters

- F : cost of failure

The 5 Parameters

- F : cost of failure
  - the cost of implementing an incorrect answer (relative to the benefit of implementing a correct answer)

The 5 Parameters

- F : cost of failure
  - the cost of implementing an incorrect answer (relative to the benefit of implementing a correct answer)
  - $F = \frac{1-\alpha}{\alpha}$ where α is the failure rate at which the user is just indifferent between using the AI and not

The 5 Parameters

- F : cost of failure
  - the cost of implementing an incorrect answer (relative to the benefit of implementing a correct answer)
  - $F = \frac{1-\alpha}{\alpha}$ where α is the failure rate at which the user is just indifferent between using the AI and not
- E : cost of evaluation

The 5 Parameters

- F : cost of failure
  - the cost of implementing an incorrect answer (relative to the benefit of implementing a correct answer)
  - $F = \frac{1-\alpha}{\alpha}$ where α is the failure rate at which the user is just indifferent between using the AI and not
- E : cost of evaluation
  - the cost of evaluating an answer (relative to the benefit of implementing a correct answer)

The 5 Parameters

- F : cost of failure
  - the cost of implementing an incorrect answer (relative to the benefit of implementing a correct answer)
  - $F = \frac{1-\alpha}{\alpha}$ where α is the failure rate at which the user is just indifferent between using the AI and not
- E : cost of evaluation
  - the cost of evaluating an answer (relative to the benefit of implementing a correct answer)
  - $E = \frac{1}{n}$ where n is the number of answers at which the user would be just indifferent between checking n answers to get one correct answer and not bothering

The 5 Parameters (continued)

- α : type I error

The 5 Parameters (continued)

- α : type I error
  - the rate of false positives associated with the evaluation technology (evaluation says answer is correct when it's incorrect)

The 5 Parameters (continued)

- α : type I error
  - the rate of false positives associated with the evaluation technology (evaluation says answer is correct when it's incorrect)
- β : type II error

The 5 Parameters (continued)

- α : type I error
  - the rate of false positives associated with the evaluation technology (evaluation says answer is correct when it's incorrect)
- β : type II error
  - the rate of false negatives associated with the evaluation technology (evaluation says answer is incorrect when it's correct)

The 5 Parameters (continued)

- α : type I error
  - the rate of false positives associated with the evaluation technology (evaluation says answer is correct when it's incorrect)
- β : type II error
  - the rate of false negatives associated with the evaluation technology (evaluation says answer is incorrect when it's correct)
- L : luck

The 5 Parameters (continued)

- α : type I error
  - the rate of false positives associated with the evaluation technology (evaluation says answer is correct when it's incorrect)
- β : type II error
  - the rate of false negatives associated with the evaluation technology (evaluation says answer is incorrect when it's correct)
- L : luck
  - the confidence level associated with random guessing (1 / the number of multiple choice answers)

The 5 Parameters

And if we reshuffle those 5 parameters...

The 5 Parameters

And if we reshuffle those 5 parameters...

- F : cost of failure
- α : type I error
- β : type II error
- L : luck
- E : cost of evaluation

The 5 Parameters for the Examples Ahead

For the examples ahead, I will use:

The 5 Parameters for the Examples Ahead

For the examples ahead, I will use:

- $F = 9$ (failure is 9x worse than success)

The 5 Parameters for the Examples Ahead

For the examples ahead, I will use:

- $F = 9$ (failure is 9x worse than success)
- $\alpha = 0.05$

The 5 Parameters for the Examples Ahead

For the examples ahead, I will use:

- $F = 9$ (failure is 9x worse than success)
- $\alpha = 0.05$
- $\beta = 0.2$

The 5 Parameters for the Examples Ahead

For the examples ahead, I will use:

- $F = 9$ (failure is 9x worse than success)
- $\alpha = 0.05$
- $\beta = 0.2$
- $L = 0.25$ (4 multiple choice answers)

The 5 Parameters for the Examples Ahead

For the examples ahead, I will use:

- $F = 9$ (failure is 9x worse than success)
- $\alpha = 0.05$
- $\beta = 0.2$
- $L = 0.25$ (4 multiple choice answers)
- $E = 0.33$: (evaluation costs 0.33 of the benefit of a success)

Deriving FABLE

Let's now write the payoffs/scores assuming the user always disregards, evaluates, and implements, respectively

Deriving FABLE

Let's now write the payoffs/scores assuming the user always disregards, evaluates, and implements, respectively

Then we will combine them and take the max to find the overall scoring rule

Disregard

Disregard

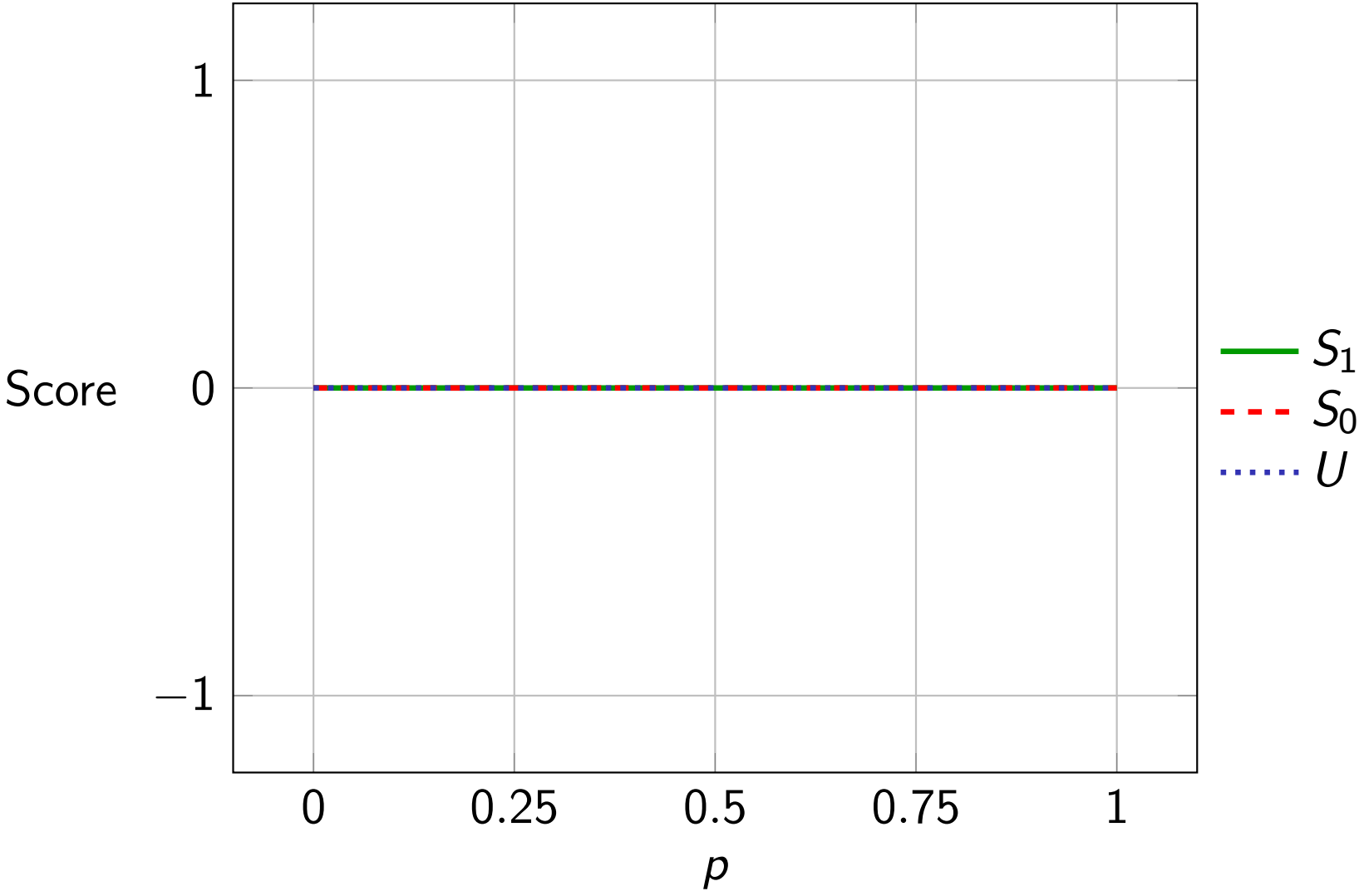
$$S_1(p) = 0$$

Disregard

$$S_1(p) = 0$$

$$S_0(p) = 0$$

Disregard



Evaluate

Evaluate

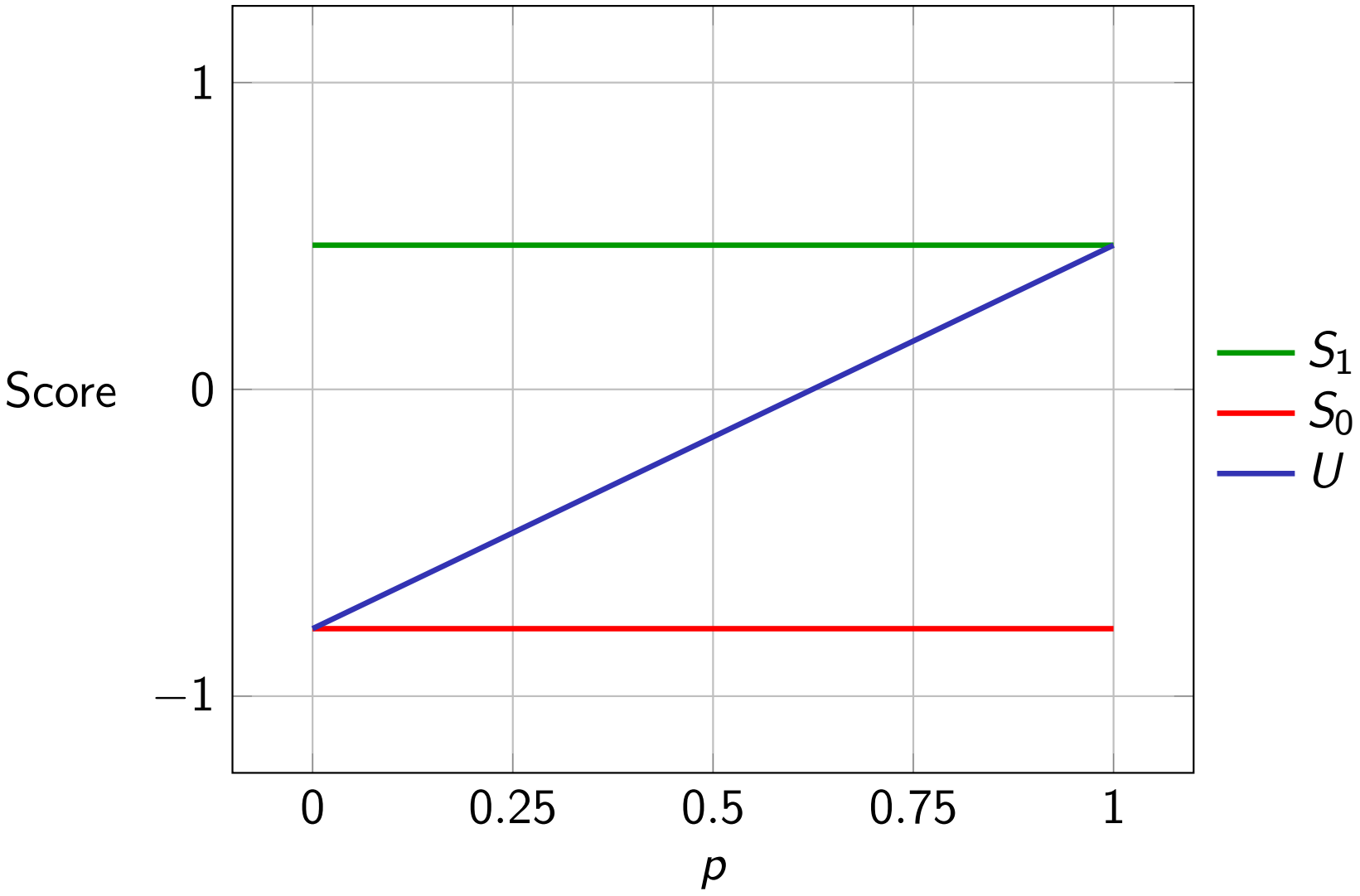
$$
\begin{aligned}
S_1(p) &= -E + (1-\beta) \times 1 + \beta \times 0 \\
&= 1 - \beta - E
\end{aligned}
$$

Evaluate

$$
\begin{aligned}
S_1(p) &= -E + (1-\beta) \times 1 + \beta \times 0 \\
&= 1 - \beta - E
\end{aligned}
$$

$$
\begin{aligned}
S_0(p) &= -E + (1-\alpha) \times 0 + \alpha \times (-F) \\
&= -\alpha F - E
\end{aligned}
$$

Evaluate



Implement

Implement

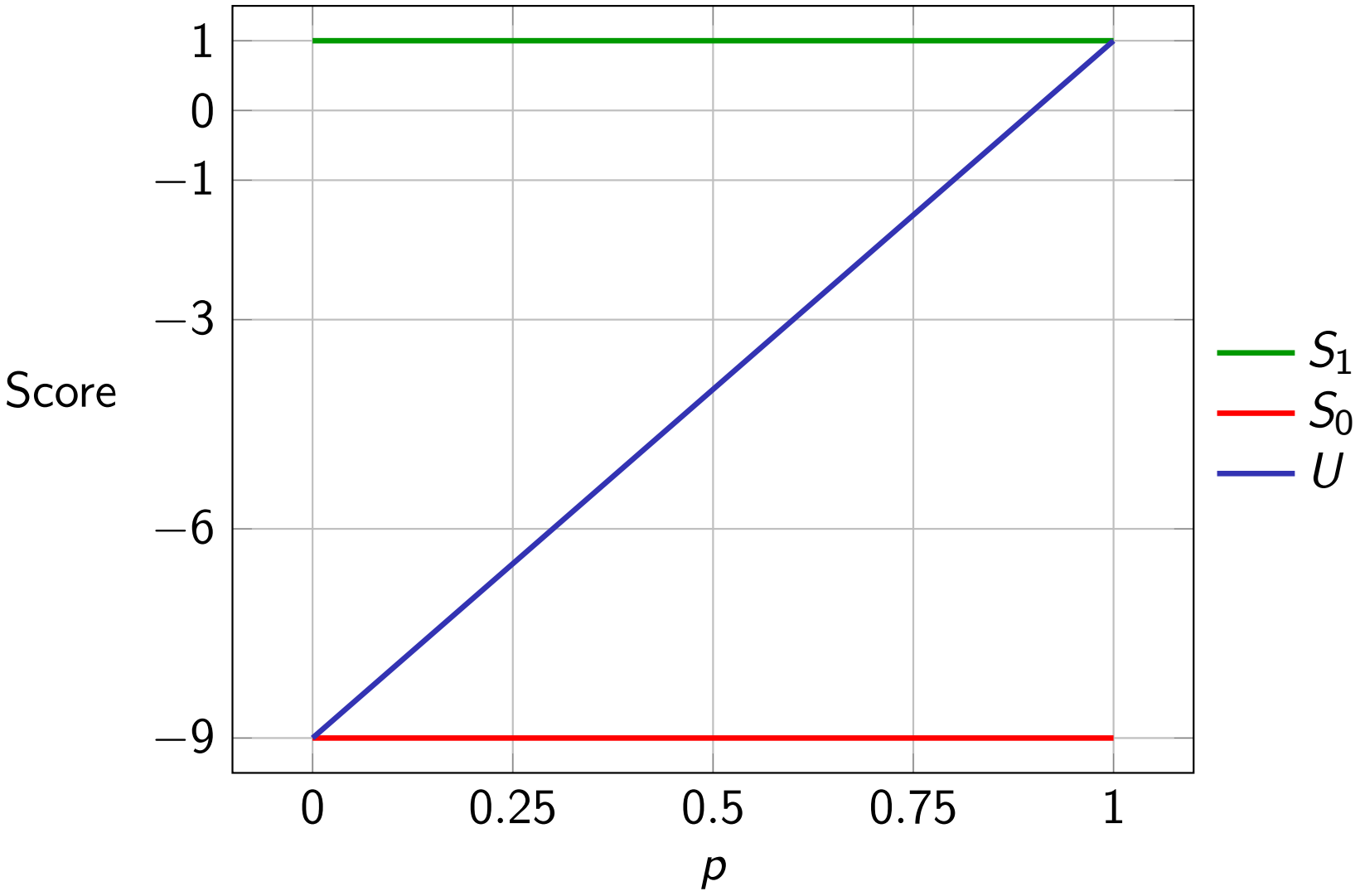
$$S_1 = 1$$

Implement

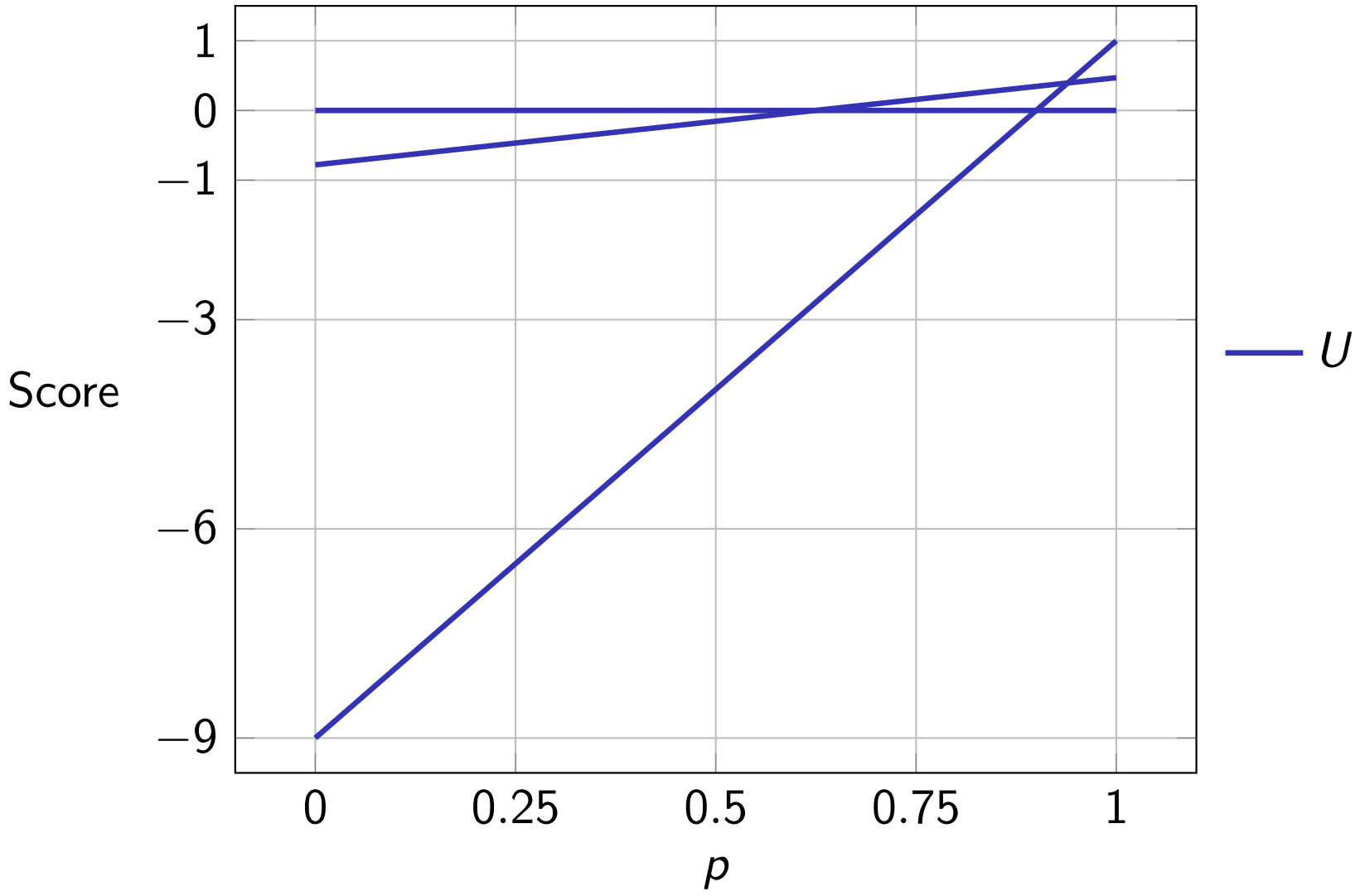
$$S_1 = 1$$

$$S_0 = -F$$

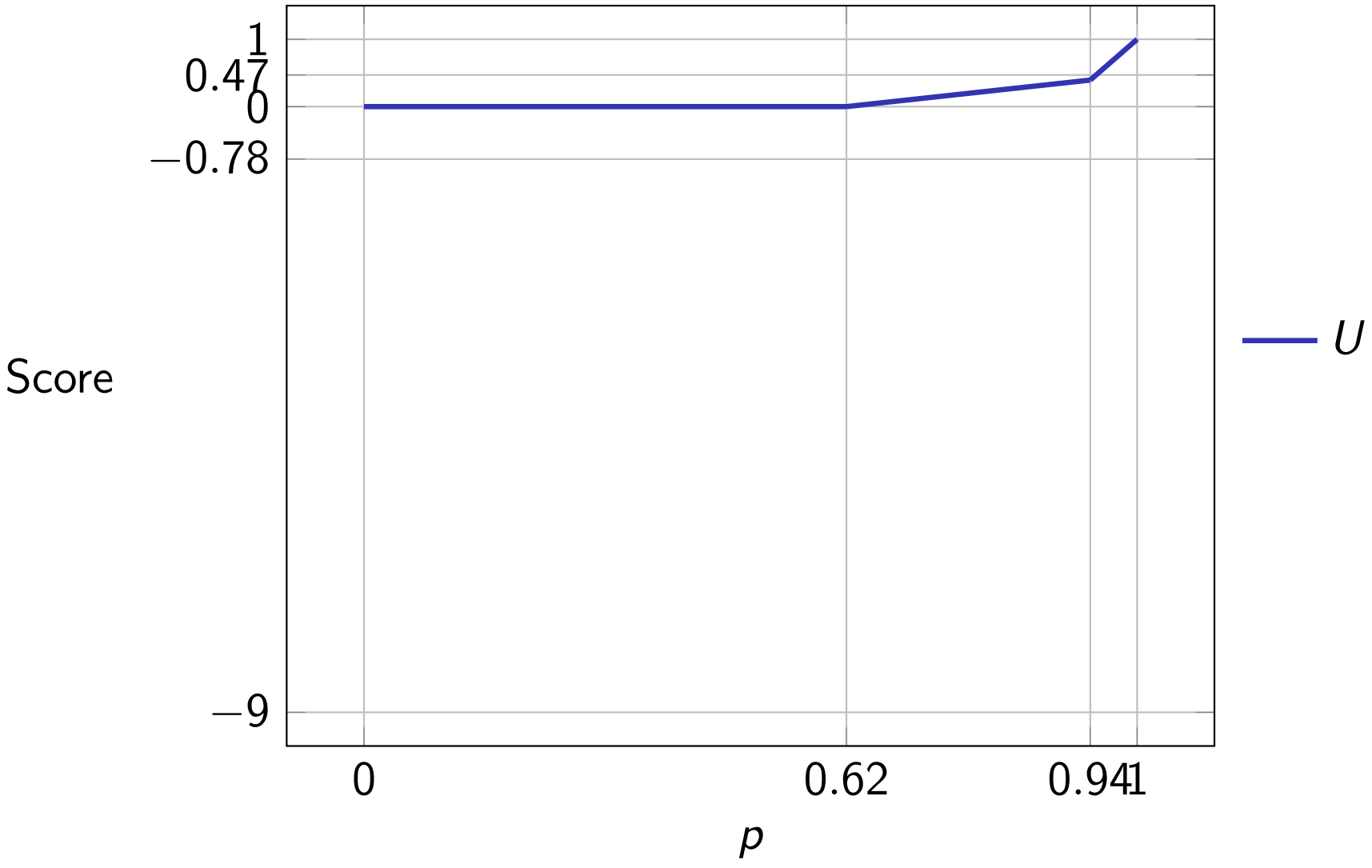
Implement



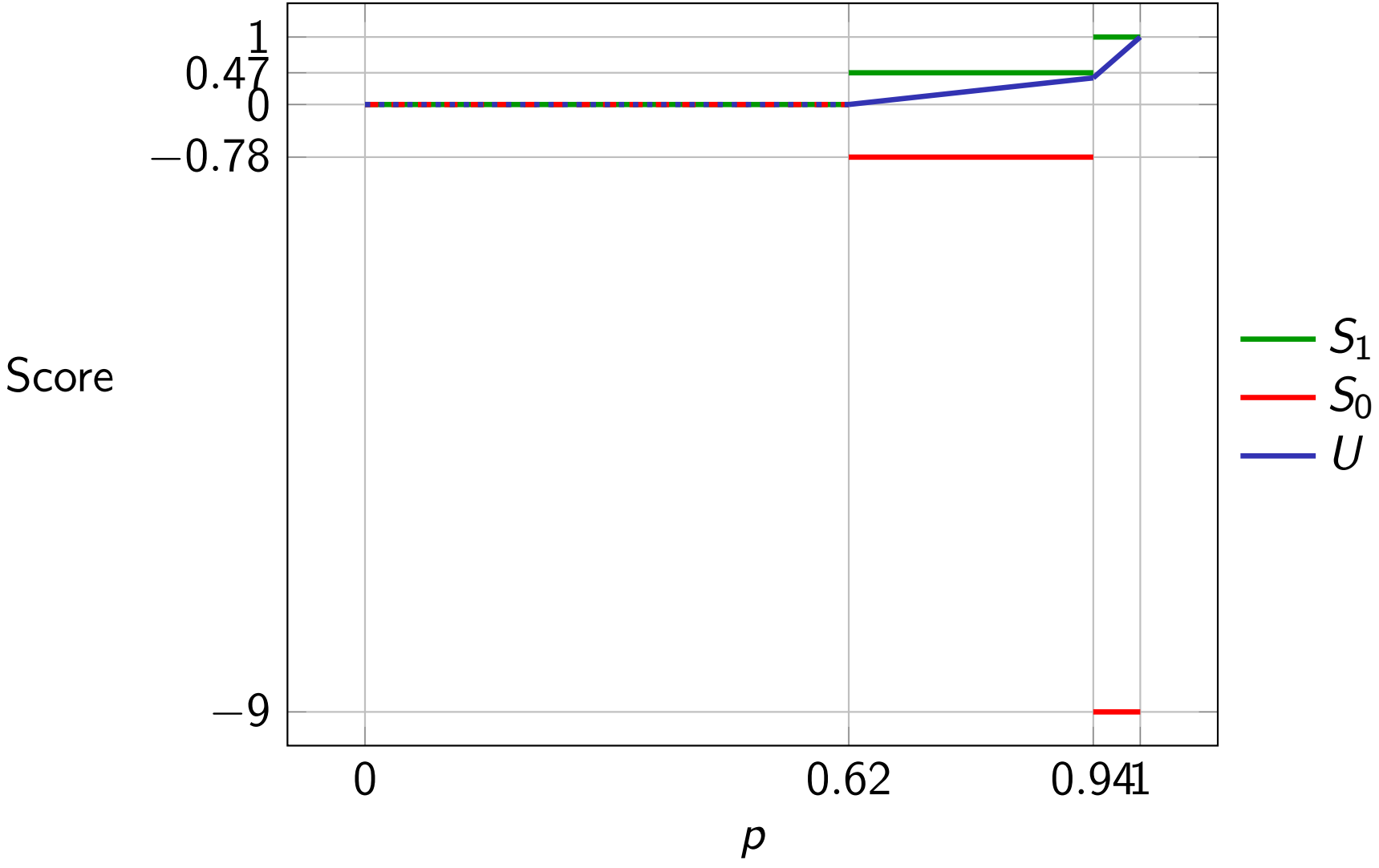
All Payoffs Together



Taking the Max



Adding S_0 and S_1 Back In



Incentives

Notice that U is weakly convex and weakly increasing

Incentives

Notice that U is weakly convex and weakly increasing

Hence, this scoring rule gives the agent:

Incentives

Notice that U is weakly convex and weakly increasing

Hence, this scoring rule gives the agent:

1. strict incentives to report their best answer (unless they have sufficiently low confidence, in which case it doesn't matter what they report)

Incentives

Notice that U is weakly convex and weakly increasing

Hence, this scoring rule gives the agent:

1. strict incentives to report their best answer (unless they have sufficiently low confidence, in which case it doesn't matter what they report)
2. strict incentives to report their confidence interval (no confidence, intermediate confidence, high confidence)

Incentives

Notice that U is weakly convex and weakly increasing

Hence, this scoring rule gives the agent:

1. strict incentives to report their best answer (unless they have sufficiently low confidence, in which case it doesn't matter what they report)
2. strict incentives to report their confidence interval (no confidence, intermediate confidence, high confidence)
3. weak incentives to report their exact confidence level within those intervals

Incentives

Notice that U is weakly convex and weakly increasing

Hence, this scoring rule gives the agent:

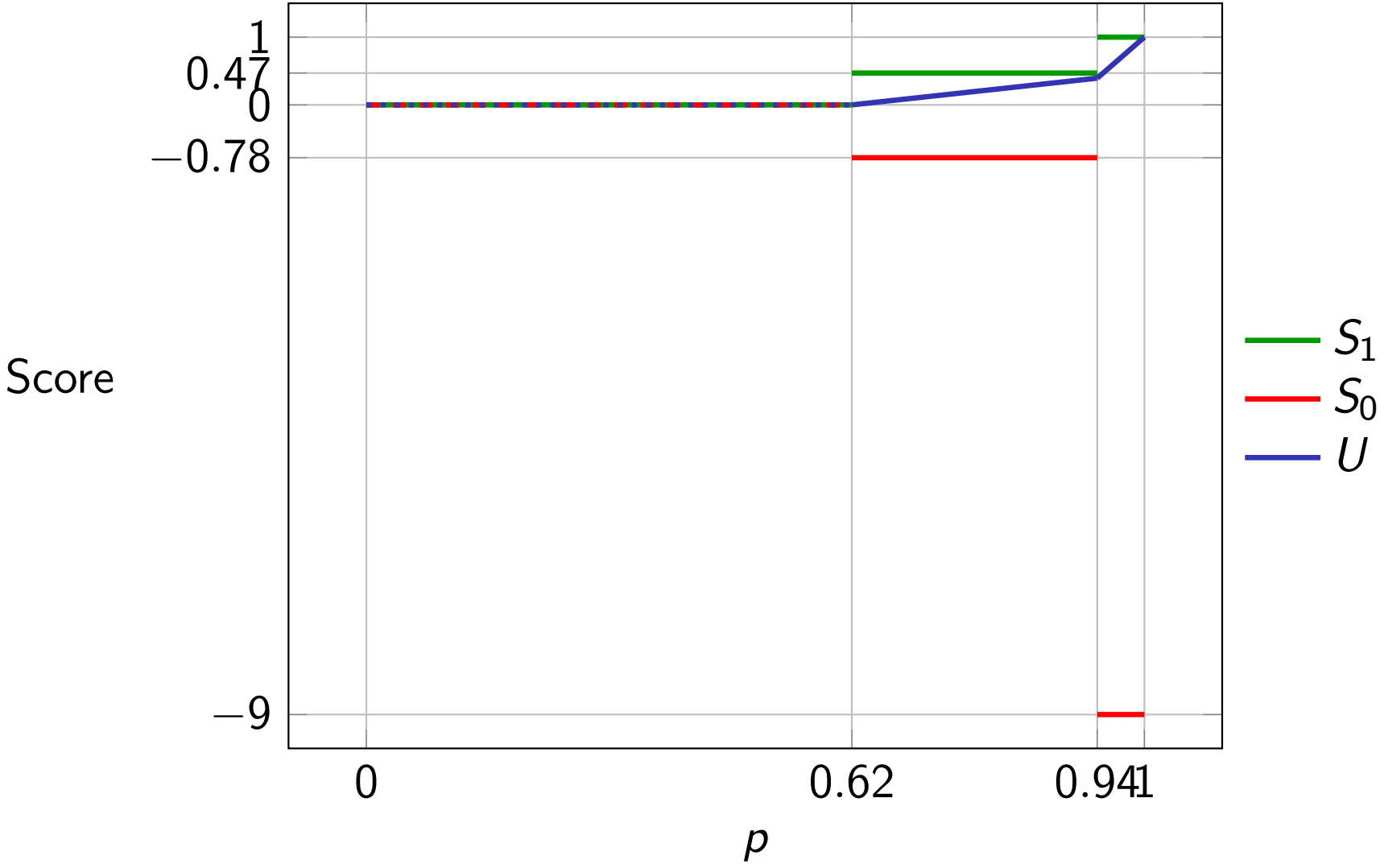
1. strict incentives to report their best answer (unless they have sufficiently low confidence, in which case it doesn't matter what they report)
2. strict incentives to report their confidence interval (no confidence, intermediate confidence, high confidence)
3. weak incentives to report their exact confidence level within those intervals

Since we didn't use the L parameter for this, I will call this the FABE scoring rule

FABE $(9, 0.05, 0.2, 0.33)$

Confidence	**Outcome**	**Score**
None $[0, 0.62)$	$\times$	0
	$\checkmark$	0
Intermediate $[0.62, 0.94]$	$\times$	-0.78
	$\checkmark$	0.47
High $(0.94, 1]$	$\times$	-9
	$\checkmark$	1

FABE $(9, 0.05, 0.2, 0.33)$



From FABE to FABLE

We are now going to derive the full FABLE scoring rule

From FABE to FABLE

We are now going to derive the full FABLE scoring rule

The FABLE scoring rule is a smoothing of the FABE scoring rule that gives agents strict incentives to report their exact confidence level

From FABE to FABLE

We are now going to derive the full FABLE scoring rule

The FABLE scoring rule is a smoothing of the FABE scoring rule that gives agents strict incentives to report their exact confidence level

Moreover, I argue that this will be a closer approximation to the user's actual payoff than FABE

From FABE to FABLE

Let U^{FABE} denote the expected payoffs of a truthful agent under FABE, which is what we derived in the previous section

From FABE to FABLE

Let U^{FABE} denote the expected payoffs of a truthful agent under FABE, which is what we derived in the previous section

Let U^{FABLE} denote the expected payoffs of a truthful agent under FABLE, which is what we will construct now

Deriving FABLE

We would like U^{FABLE} to satisfy the following properties:

Deriving FABLE

We would like U^{FABLE} to satisfy the following properties:

- U^{FABLE} is one level smoother than U^{FABE}

Deriving FABLE

We would like U^{FABLE} to satisfy the following properties:

- U^{FABLE} is one level smoother than U^{FABE}
  - U^{FABE} was piecewise linear and continuous

Deriving FABLE

We would like U^{FABLE} to satisfy the following properties:

- U^{FABLE} is one level smoother than U^{FABE}
  - U^{FABE} was piecewise linear and continuous
  - $U^{\text{FABLE}\prime}$ should be piecewise linear and continuous

Deriving FABLE

We would like U^{FABLE} to satisfy the following properties:

- U^{FABLE} is one level smoother than U^{FABE}
  - U^{FABE} was piecewise linear and continuous
  - $U^{\text{FABLE}\prime}$ should be piecewise linear and continuous
- for all $p \leq L$, $U^{\text{FABLE}}(p) = U^{\text{FABE}}(p)$ and $U^{\text{FABLE}\prime}(p) = U^{\text{FABE}\prime}(p)$

Deriving FABLE

We would like U^{FABLE} to satisfy the following properties:

- U^{FABLE} is one level smoother than U^{FABE}
  - U^{FABE} was piecewise linear and continuous
  - $U^{\text{FABLE}\prime}$ should be piecewise linear and continuous
- for all $p \leq L$, $U^{\text{FABLE}}(p) = U^{\text{FABE}}(p)$ and $U^{\text{FABLE}\prime}(p) = U^{\text{FABE}\prime}(p)$
- $U^{\text{FABLE}}(\frac{p_0+p_1}{2}) = U^{\text{FABE}}(\frac{p_0+p_1}{2})$ and $U^{\text{FABLE}\prime}(\frac{p_0+p_1}{2}) = U^{\text{FABE}\prime}(\frac{p_0+p_1}{2})$

Deriving FABLE

We would like U^{FABLE} to satisfy the following properties:

- U^{FABLE} is one level smoother than U^{FABE}
  - U^{FABE} was piecewise linear and continuous
  - $U^{\text{FABLE}\prime}$ should be piecewise linear and continuous
- for all $p \leq L$, $U^{\text{FABLE}}(p) = U^{\text{FABE}}(p)$ and $U^{\text{FABLE}\prime}(p) = U^{\text{FABE}\prime}(p)$
- $U^{\text{FABLE}}(\frac{p_0+p_1}{2}) = U^{\text{FABE}}(\frac{p_0+p_1}{2})$ and $U^{\text{FABLE}\prime}(\frac{p_0+p_1}{2}) = U^{\text{FABE}\prime}(\frac{p_0+p_1}{2})$
- $U^{\text{FABLE}}(1) = U^{\text{FABE}}(1)$ and $U^{\text{FABLE}\prime}(1) = U^{\text{FABE}\prime}(1)$

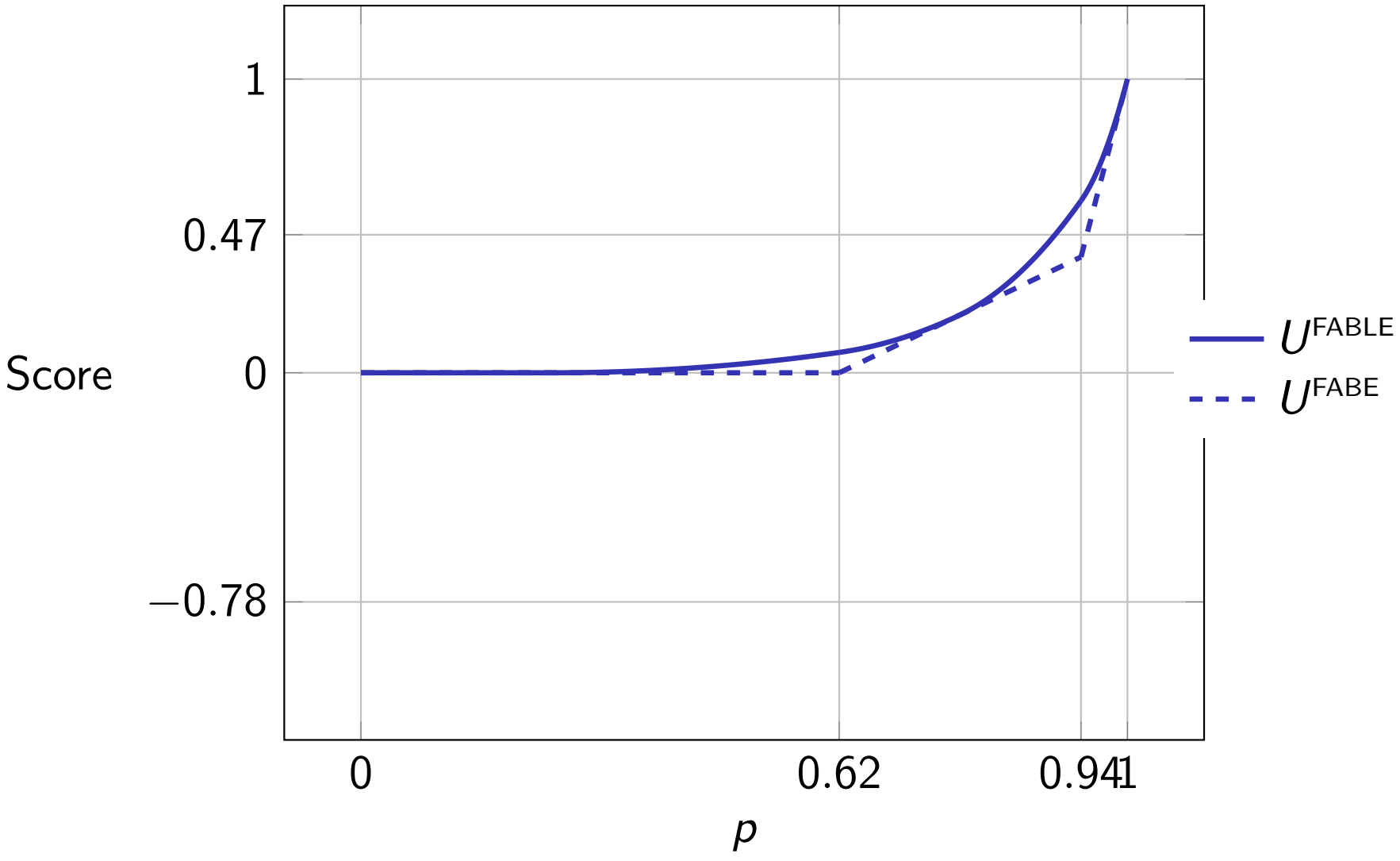
Deriving FABLE

We would like U^{FABLE} to satisfy the following properties:

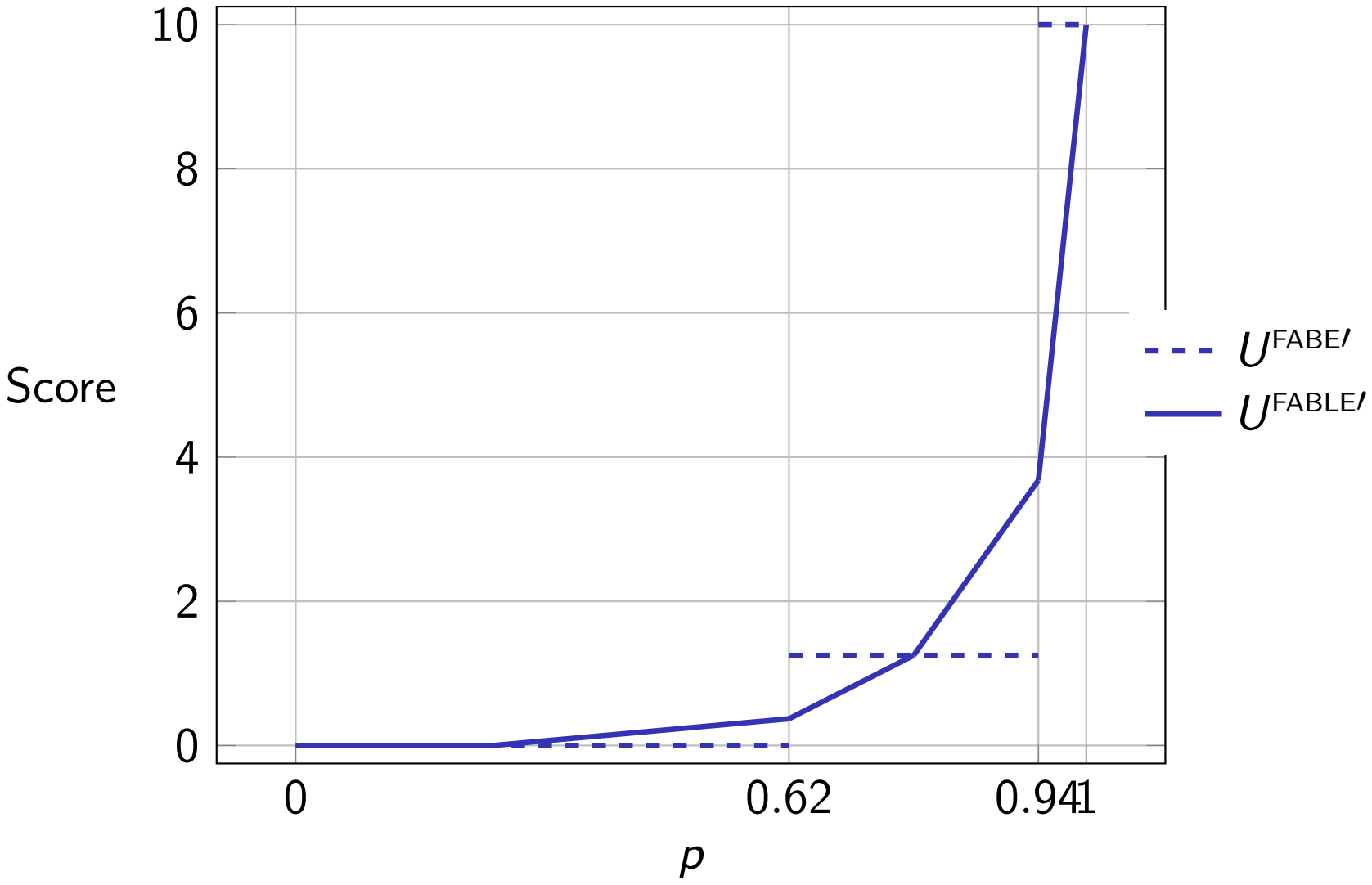
- U^{FABLE} is one level smoother than U^{FABE}
  - U^{FABE} was piecewise linear and continuous
  - $U^{\text{FABLE}\prime}$ should be piecewise linear and continuous
- for all $p \leq L$, $U^{\text{FABLE}}(p) = U^{\text{FABE}}(p)$ and $U^{\text{FABLE}\prime}(p) = U^{\text{FABE}\prime}(p)$
- $U^{\text{FABLE}}(\frac{p_0+p_1}{2}) = U^{\text{FABE}}(\frac{p_0+p_1}{2})$ and $U^{\text{FABLE}\prime}(\frac{p_0+p_1}{2}) = U^{\text{FABE}\prime}(\frac{p_0+p_1}{2})$
- $U^{\text{FABLE}}(1) = U^{\text{FABE}}(1)$ and $U^{\text{FABLE}\prime}(1) = U^{\text{FABE}\prime}(1)$

It turns out this pins down U^{FABLE} uniquely

U^{FABE} and U^{FABLE}



$U^{\text{FABE}\prime}$ and $U^{\text{FABLE}\prime}$



Why Do This?

FABE is a simple skeleton of the payoffs for the end user, supposing the user has just three available actions: disregard, evaluate, and implement

Why Do This?

FABE is a simple skeleton of the payoffs for the end user, supposing the user has just three available actions: disregard, evaluate, and implement

Of course, in reality, the user has many more actions, including

Why Do This?

FABE is a simple skeleton of the payoffs for the end user, supposing the user has just three available actions: disregard, evaluate, and implement

Of course, in reality, the user has many more actions, including

- evaluating the answer with different levels of effort

Why Do This?

FABE is a simple skeleton of the payoffs for the end user, supposing the user has just three available actions: disregard, evaluate, and implement

Of course, in reality, the user has many more actions, including

- evaluating the answer with different levels of effort
- using an incorrect answer as helpful inspiration

Why Do This?

The FABE skeleton is optimized to be a good approximation of U at

Why Do This?

The FABE skeleton is optimized to be a good approximation of U at

- the left endpoint for disregard

Why Do This?

The FABE skeleton is optimized to be a good approximation of U at

- the left endpoint for disregard
- the midpoint for evaluate

Why Do This?

The FABE skeleton is optimized to be a good approximation of U at

- the left endpoint for disregard
- the midpoint for evaluate
- the right endpoint for implement

Why Do This?

The FABE skeleton is optimized to be a good approximation of U at

- the left endpoint for disregard
- the midpoint for evaluate
- the right endpoint for implement

Everywhere else, it is a *lower bound* on U

Why Do This?

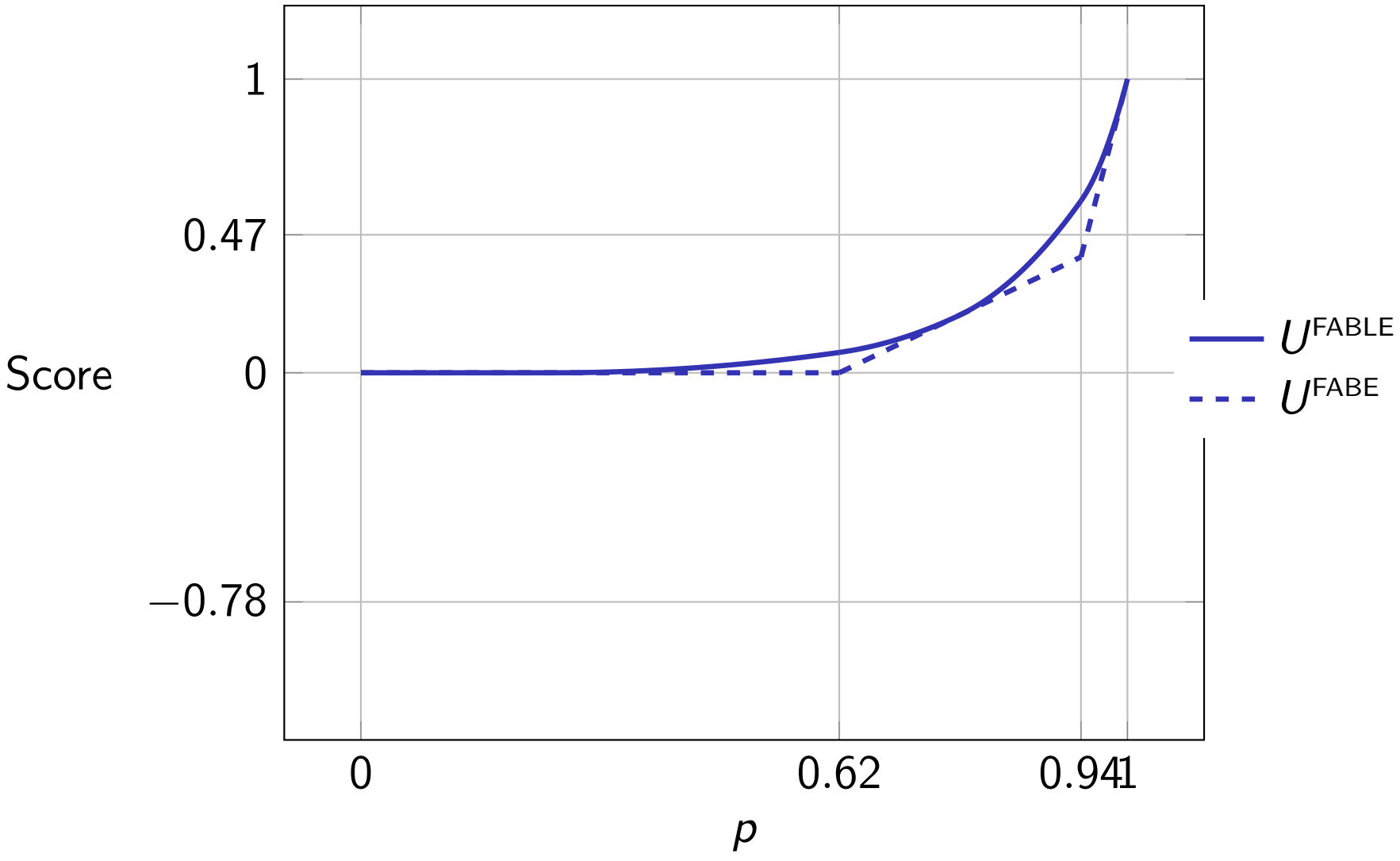
The FABE skeleton is optimized to be a good approximation of U at

- the left endpoint for disregard
- the midpoint for evaluate
- the right endpoint for implement

Everywhere else, it is a *lower bound* on U

FABLE adds the "meat" to FABE's "bones," smoothing out U in between these points to capture the other actions an individual might take that are unnecessary to model explicitly

U^{FABE} and U^{FABLE}



From U^{FABLE} to (S_0, S_1)

We've pinned down U^{FABLE}, but haven't yet pinned down (S_0, S_1)

From U^{FABLE} to (S_0, S_1)

We've pinned down U^{FABLE}, but haven't yet pinned down (S_0, S_1)

Recall that there are infinitely many (S_0, S_1) that could generate U^{FABLE}

From U^{FABLE} to (S_0, S_1)

We've pinned down U^{FABLE}, but haven't yet pinned down (S_0, S_1)

Recall that there are infinitely many (S_0, S_1) that could generate U^{FABLE}

By Theorem 2, since U^{FABLE} is strictly convex and strictly increasing, there exists a unique (S_0, S_1) which is strictly incentive compatible

From U^{FABLE} to (S_0, S_1)

We've pinned down U^{FABLE}, but haven't yet pinned down (S_0, S_1)

Recall that there are infinitely many (S_0, S_1) that could generate U^{FABLE}

By Theorem 2, since U^{FABLE} is strictly convex and strictly increasing, there exists a unique (S_0, S_1) which is strictly incentive compatible

By Proposition 2, the scoring rule must be incentive compatible if it is to reflect the payoffs of society

From U^{FABLE} to (S_0, S_1)

We've pinned down U^{FABLE}, but haven't yet pinned down (S_0, S_1)

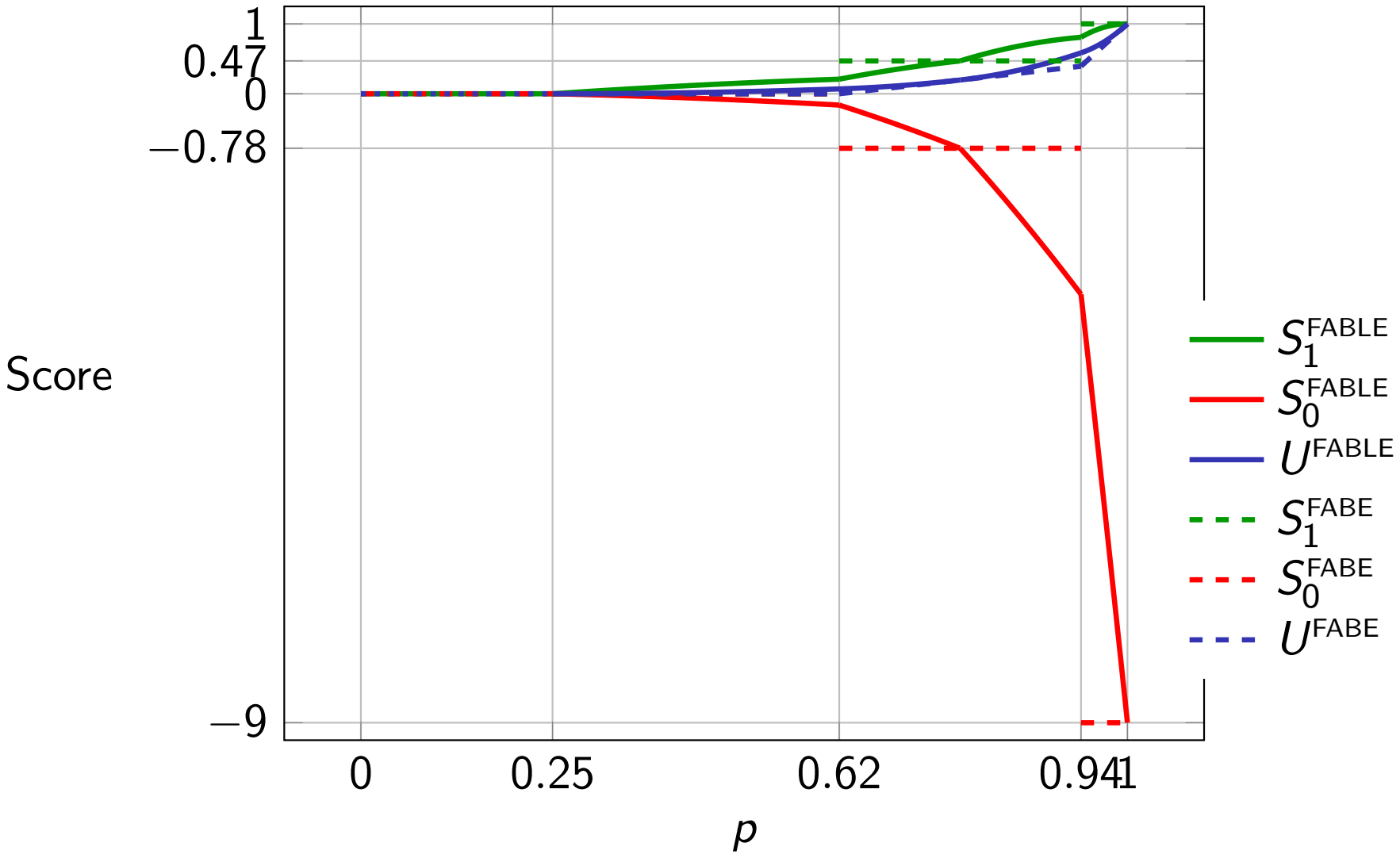
Recall that there are infinitely many (S_0, S_1) that could generate U^{FABLE}

By Theorem 2, since U^{FABLE} is strictly convex and strictly increasing, there exists a unique (S_0, S_1) which is strictly incentive compatible

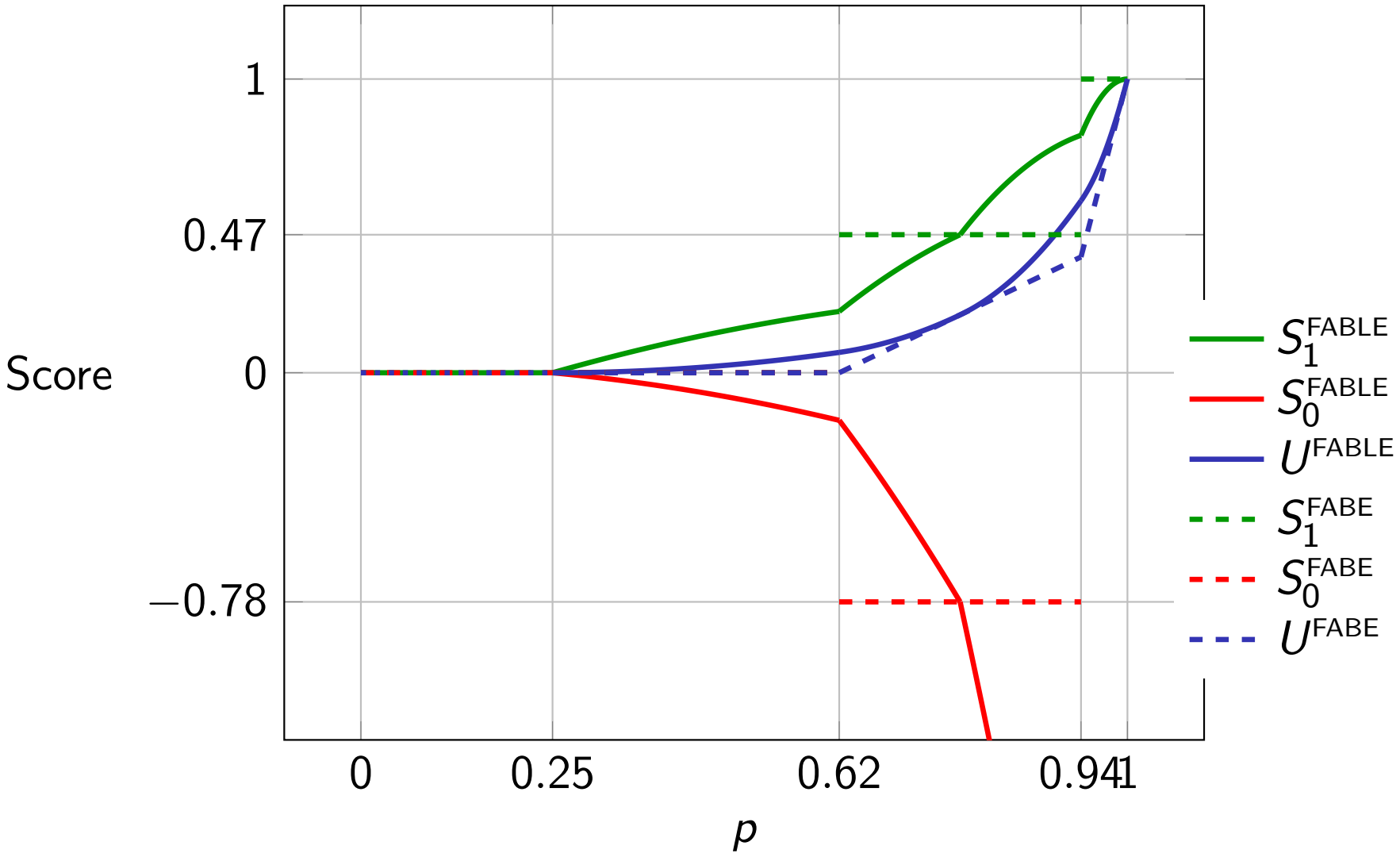
By Proposition 2, the scoring rule must be incentive compatible if it is to reflect the payoffs of society

Adding the unique incentive-compatible (S_0, S_1) which generates U^{FABLE} gives us FABLE, and it looks like this

FABE and FABLE



FABE and FABLE



Online Demo

Online Demo

Skill, Judgement, and Calibration

We can decompose FABLE (or any scoring rule) into three pieces: skill, judgement, and calibration

Skill, Judgement, and Calibration

We can decompose FABLE (or any scoring rule) into three pieces: skill, judgement, and calibration

Skill Score

the expected payoff to society of employing the agent if they had perfect judgement and calibration (knowing what they know exactly)

Skill, Judgement, and Calibration

We can decompose FABLE (or any scoring rule) into three pieces: skill, judgement, and calibration

Skill Score

the expected payoff to society of employing the agent if they had perfect judgement and calibration (knowing what they know exactly)

Skill Score + Judgement Penalty

the expected payoff to society of employing the agent if they had perfect calibration (knowing what they know on average)

Skill, Judgement, and Calibration

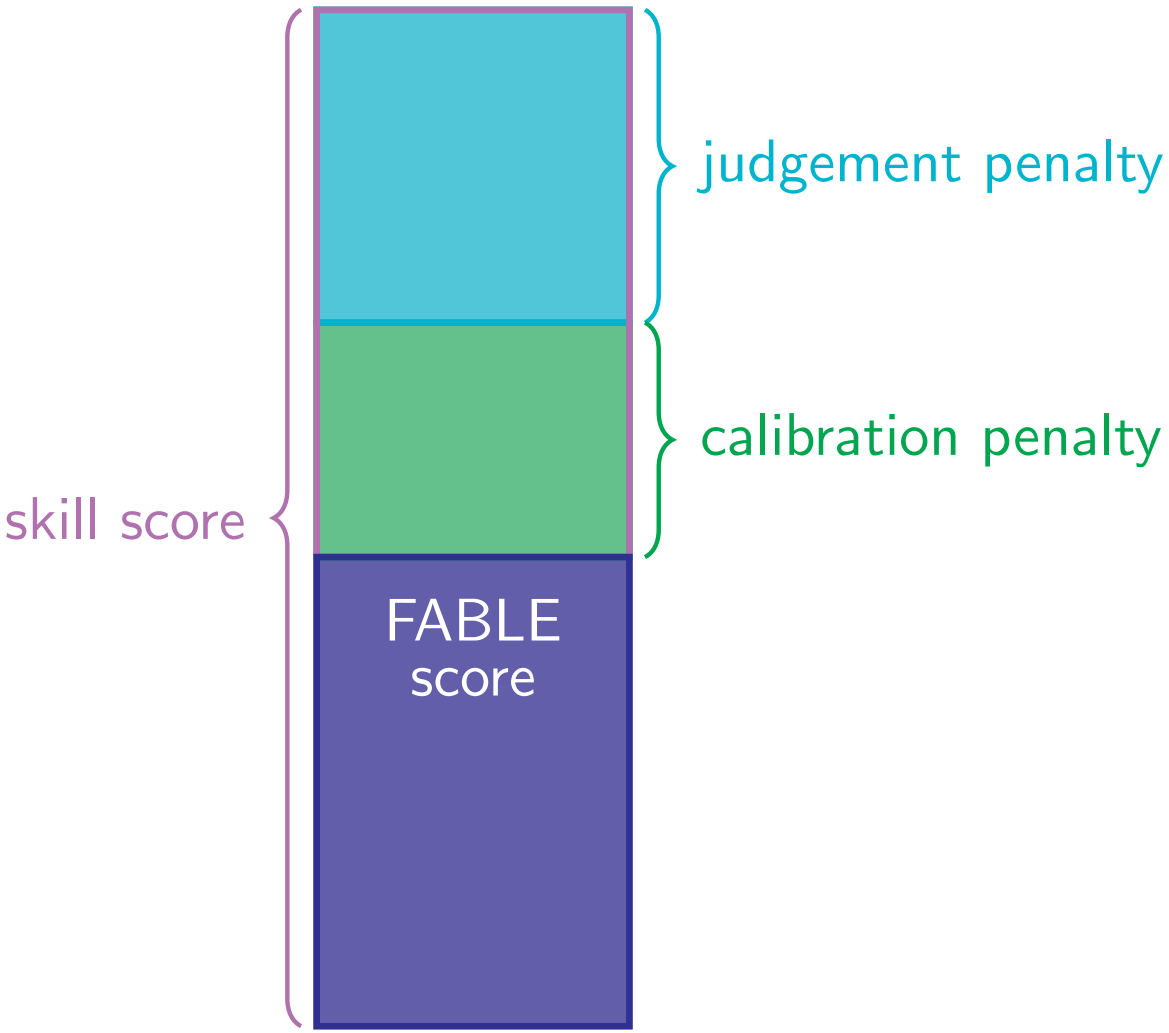
We can decompose FABLE (or any scoring rule) into three pieces: skill, judgement, and calibration

Skill Score
the expected payoff to society of employing the agent if they had perfect judgement and calibration (knowing what they know exactly)

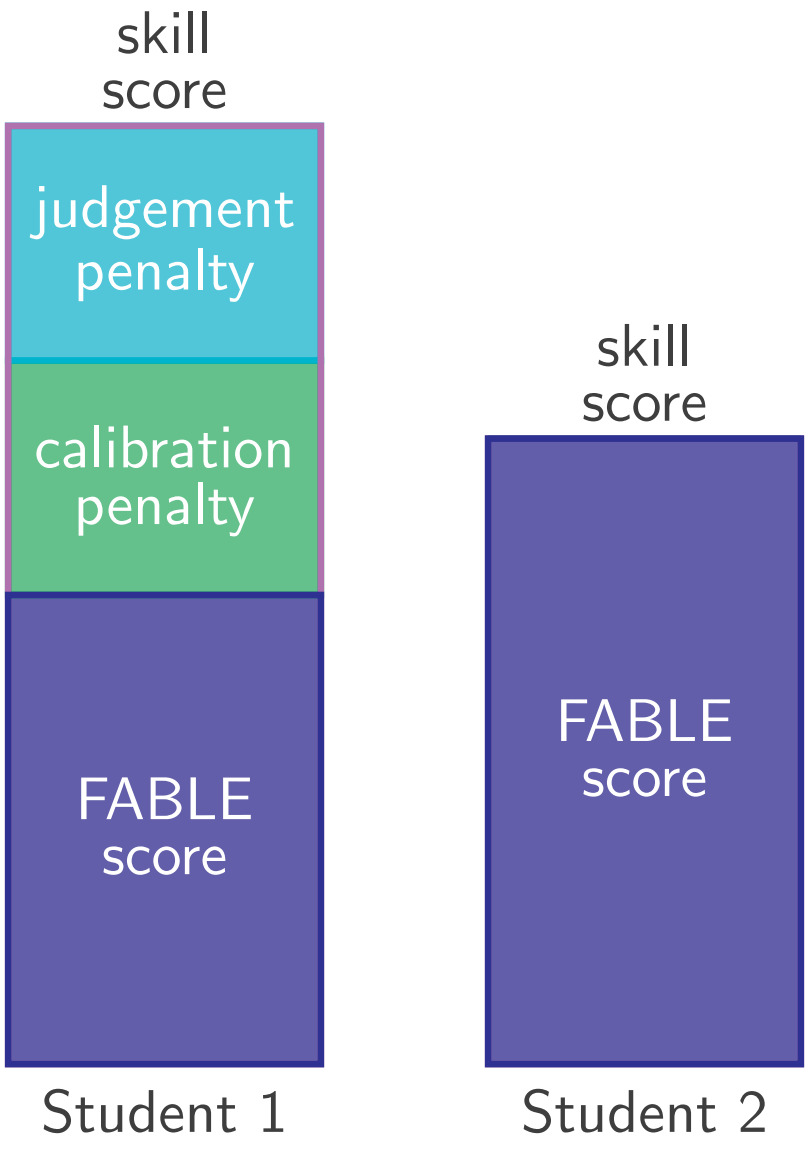
Skill Score + Judgement Penalty
the expected payoff to society of employing the agent if they had perfect calibration (knowing what they know on average)

Skill Score + Judgement Penalty + Calibration Penalty = FABLE
the expected payoff to society of employing the agent

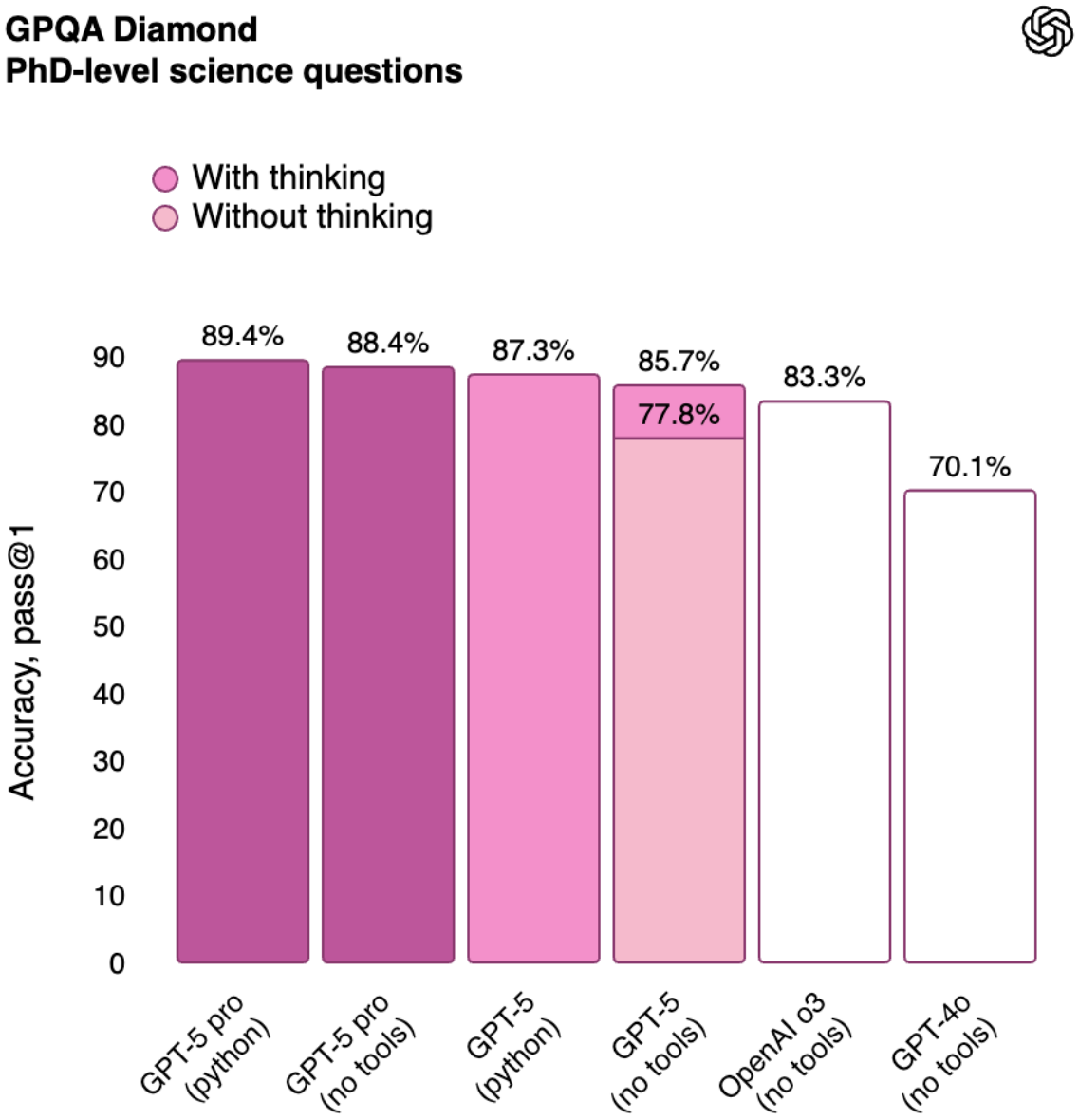
Skill, Judgement, and Calibration



Example


skill
score
judgement
penalty
calibration
penalty
FABLE
score
Student 1
skill
score
FABLE
score
Student 2

Instead Of


GPQA Diamond
PhD-level science questions
With thinking
Without thinking
Accuracy, pass@1
0
10
20
30
40
50
60
70
80
90
89.4%
88.4%
87.3%
85.7%
77.8%
83.3%
70.1%
GPT-5 pro (python)
GPT-5 pro (no tools)
GPT-5 (python)
GPT-5 (no tools)
OpenAI o3 (no tools)
GPT-4o (no tools)

Thank you for listening!

Questions, comments, or concerns?