Taming the AI Monster

Monitoring of Individual Fairness for Effective Human Oversight

Holger Hermanns • Saarland University • SummerSoc 2024 • 27 June 2024

The

explosion of opportunities

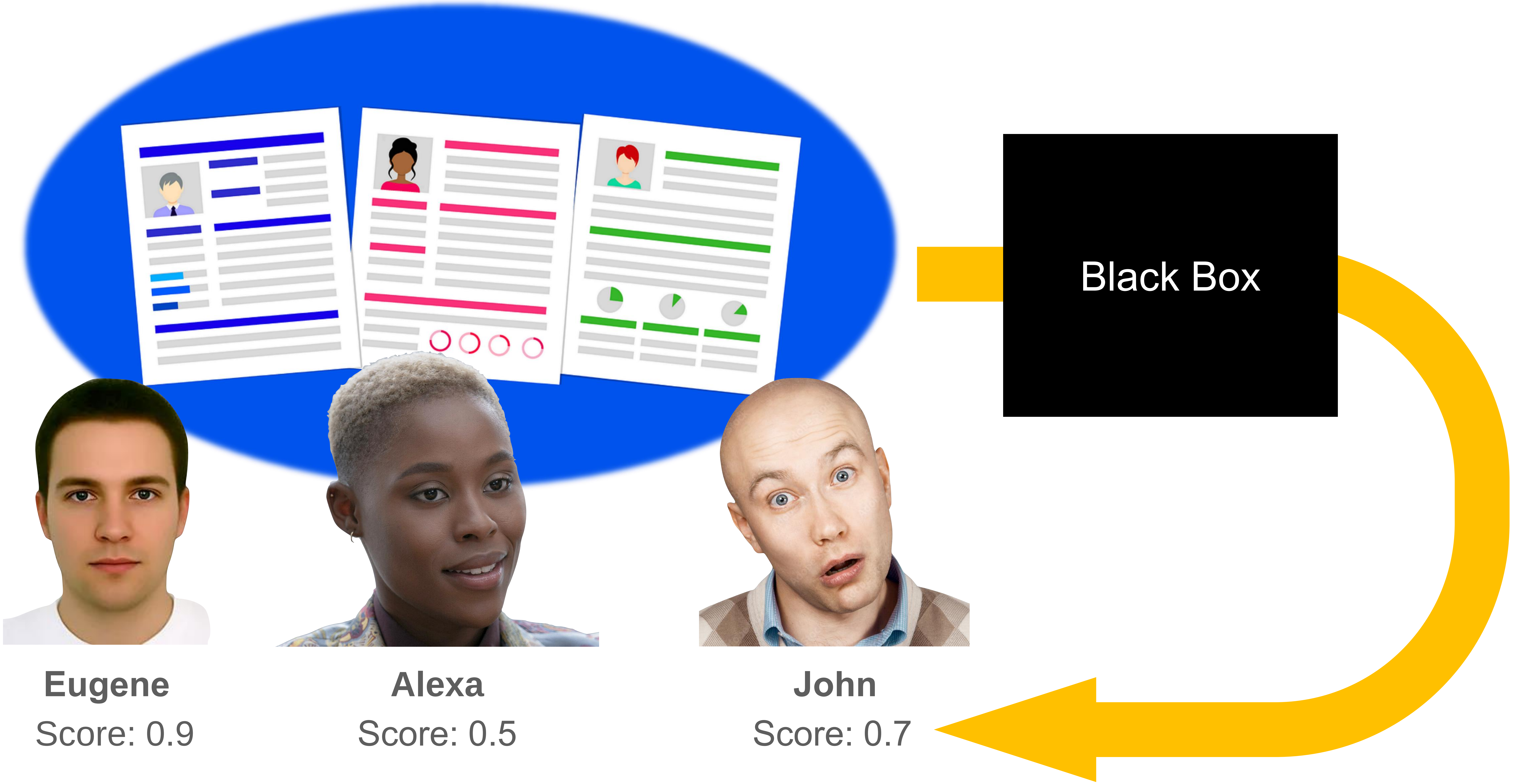
for software-driven innovations

comes with an

implosion of human opportunities and capabilities

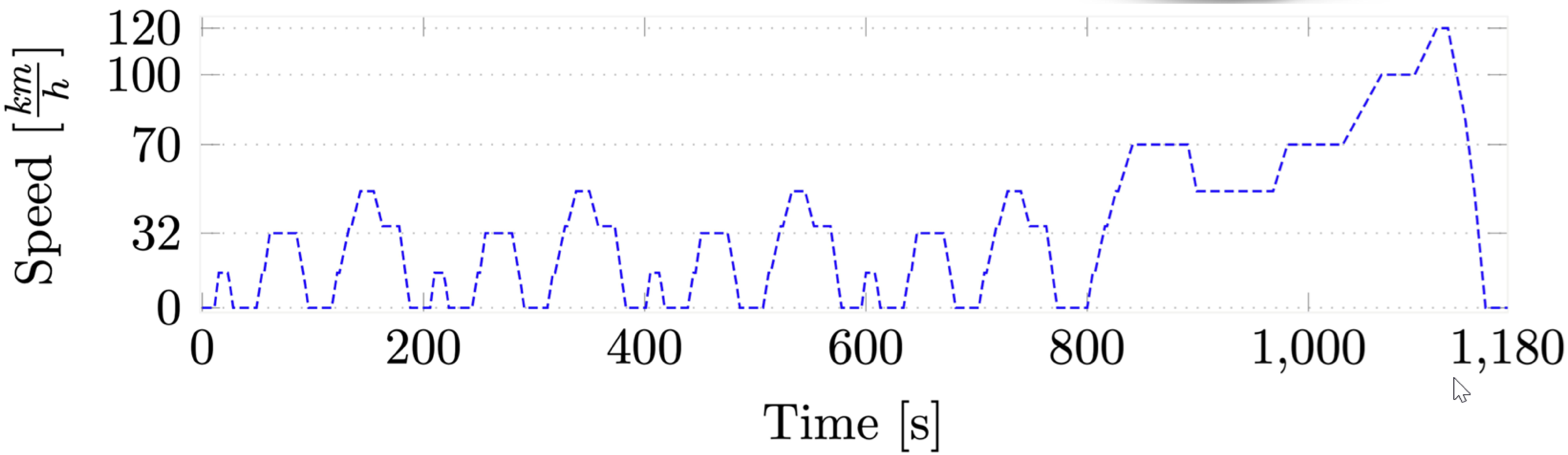
to understand and control these innovations.

Example – Individual Fairness



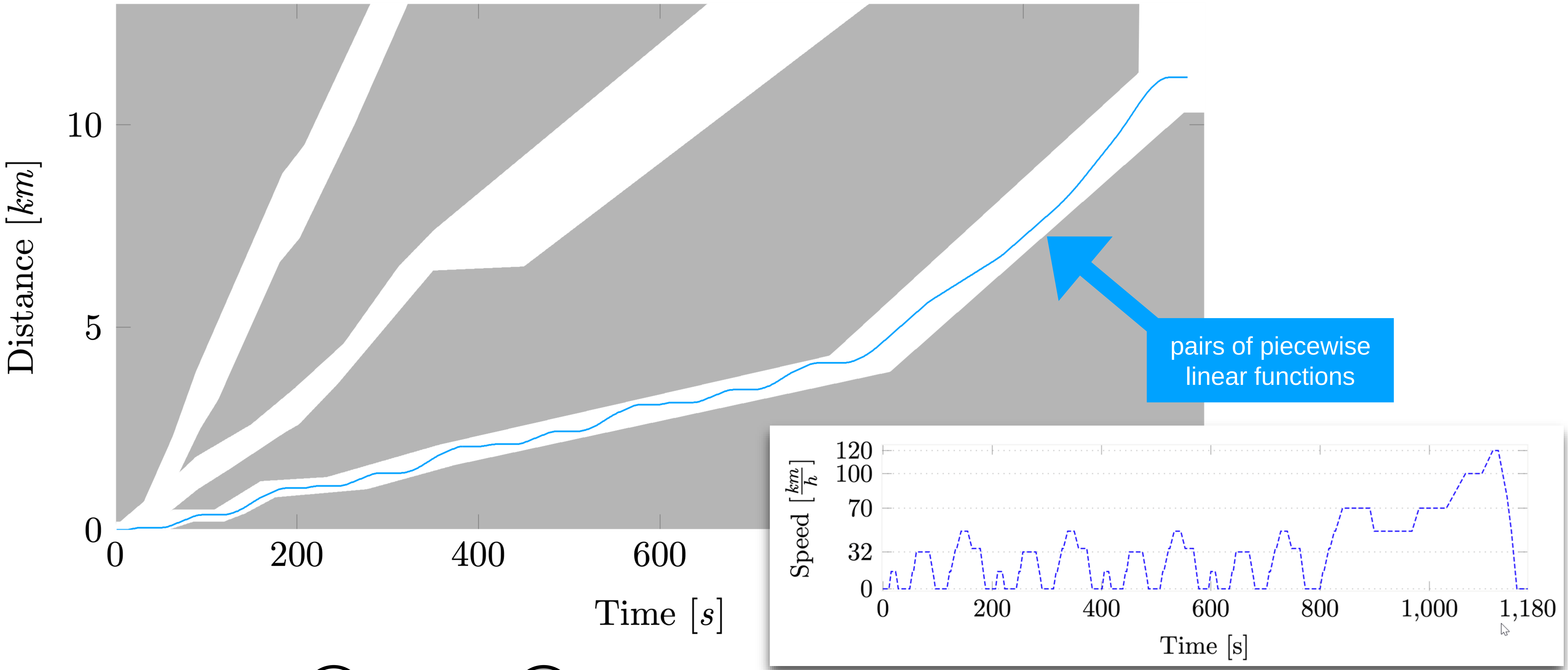
Example – Software Doping

New European Driving Cycle (NEDC):



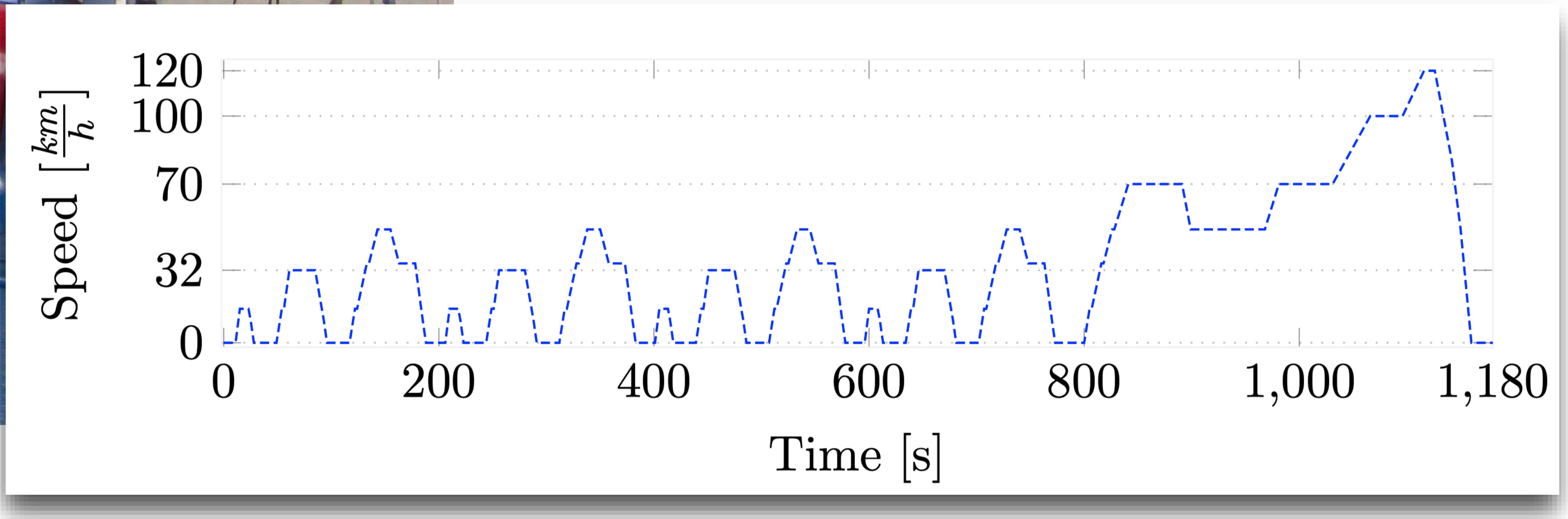
Contag et al.: How They Did It: An Analysis of Emission Defeat Devices in Modern Automobiles. SP 2017.
Domke and Lange: The exhaust emissions scandal ("Diesel-gate"). Chaos Communication Congress 2015.

Emission Cleaning by Volkswagen



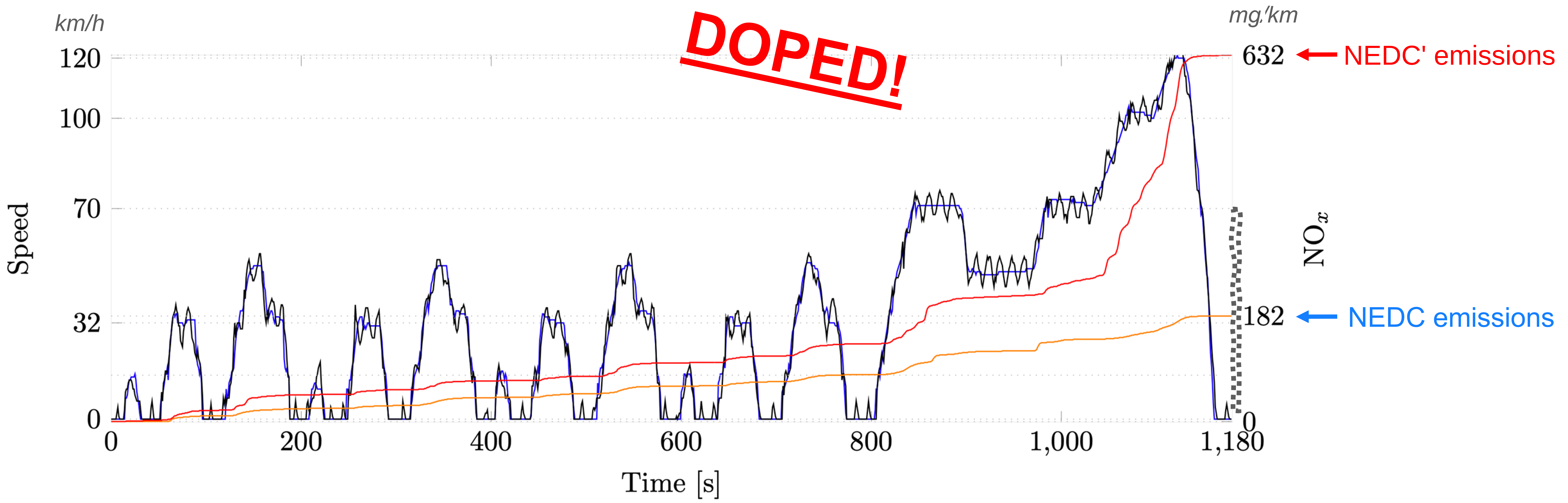
Êmission cleaning: enabled disabled, irreversible

Emission Cleaning by Others



NEDC vs. NEDC'

$$d_{\mathsf{In}}(\mathsf{i}_1, \mathsf{i}_2) = |\mathsf{i}_1 - \mathsf{i}_2| \qquad d_{\mathsf{Out}}(\mathsf{o}_1, \mathsf{o}_2) = |\mathsf{o}_1 - \mathsf{o}_2| \qquad \kappa_{\mathsf{i}} = 15\ \mathrm{km/h} \qquad \kappa_{\mathsf{o}} = 180\ \mathrm{mg/km}$$



Software Cleanness – a general expectation

A software is doped if and only if it is not clean.

Our cleanness mantra is: *Similar inputs lead to similar outputs.*

Robust Cleanness

reactive

$P : \mathsf{In}^\omega \to \mathsf{Out}^\omega$

deterministic

distance function for inputs, $(\mathsf{In}^* \times \mathsf{In}^*) \to \overline{\mathbb{R}}_{\geq 0}$

distance function for outputs, $(\mathsf{Out}^* \times \mathsf{Out}^*) \to \overline{\mathbb{R}}_{\geq 0}$

Contract $\mathcal{C} = \langle \mathsf{StdIn}, d_{\mathsf{In}}, d_{\mathsf{Out}}, \kappa_i, \kappa_o \rangle$

standard inputs

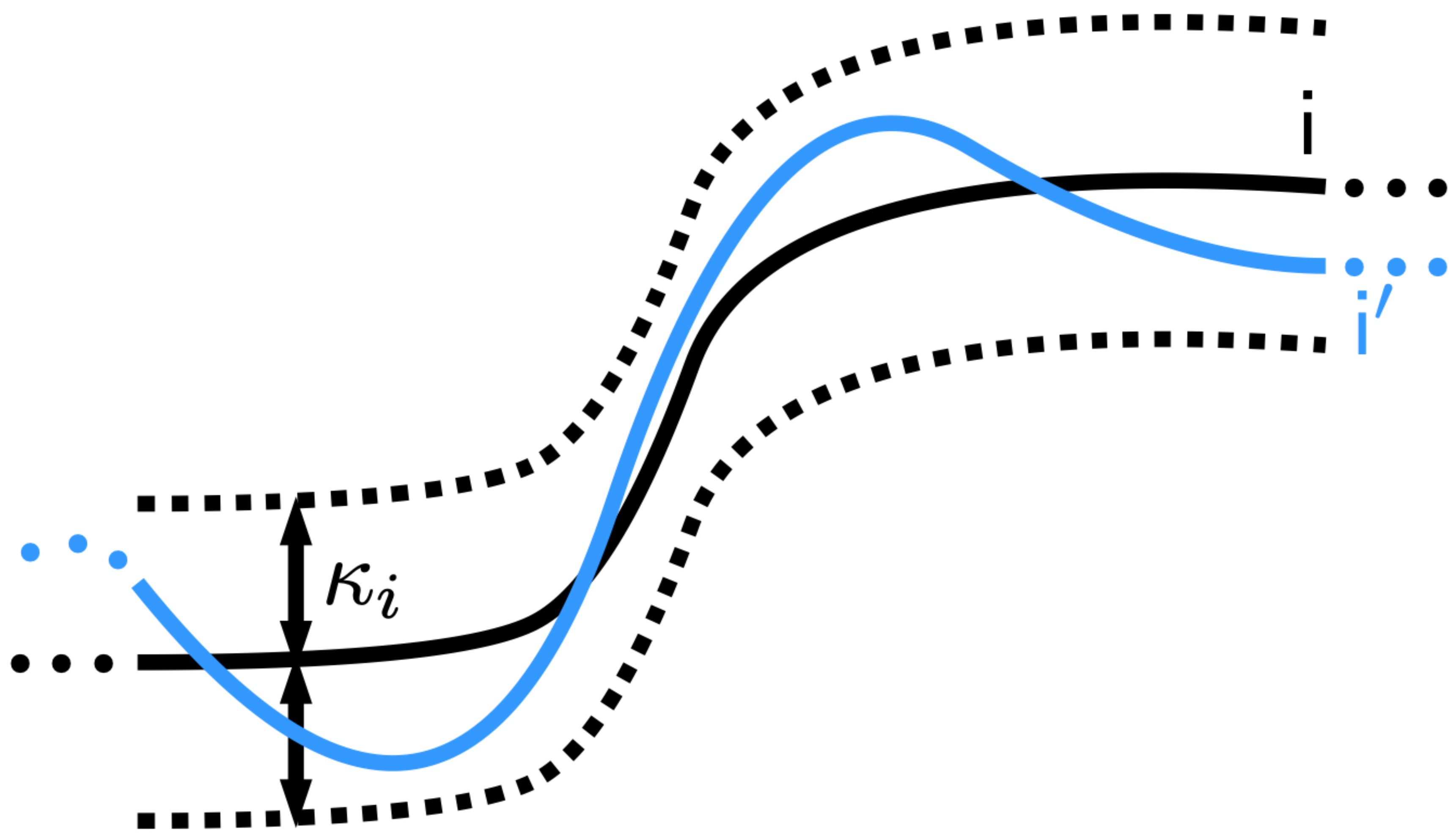
threshold for output distance

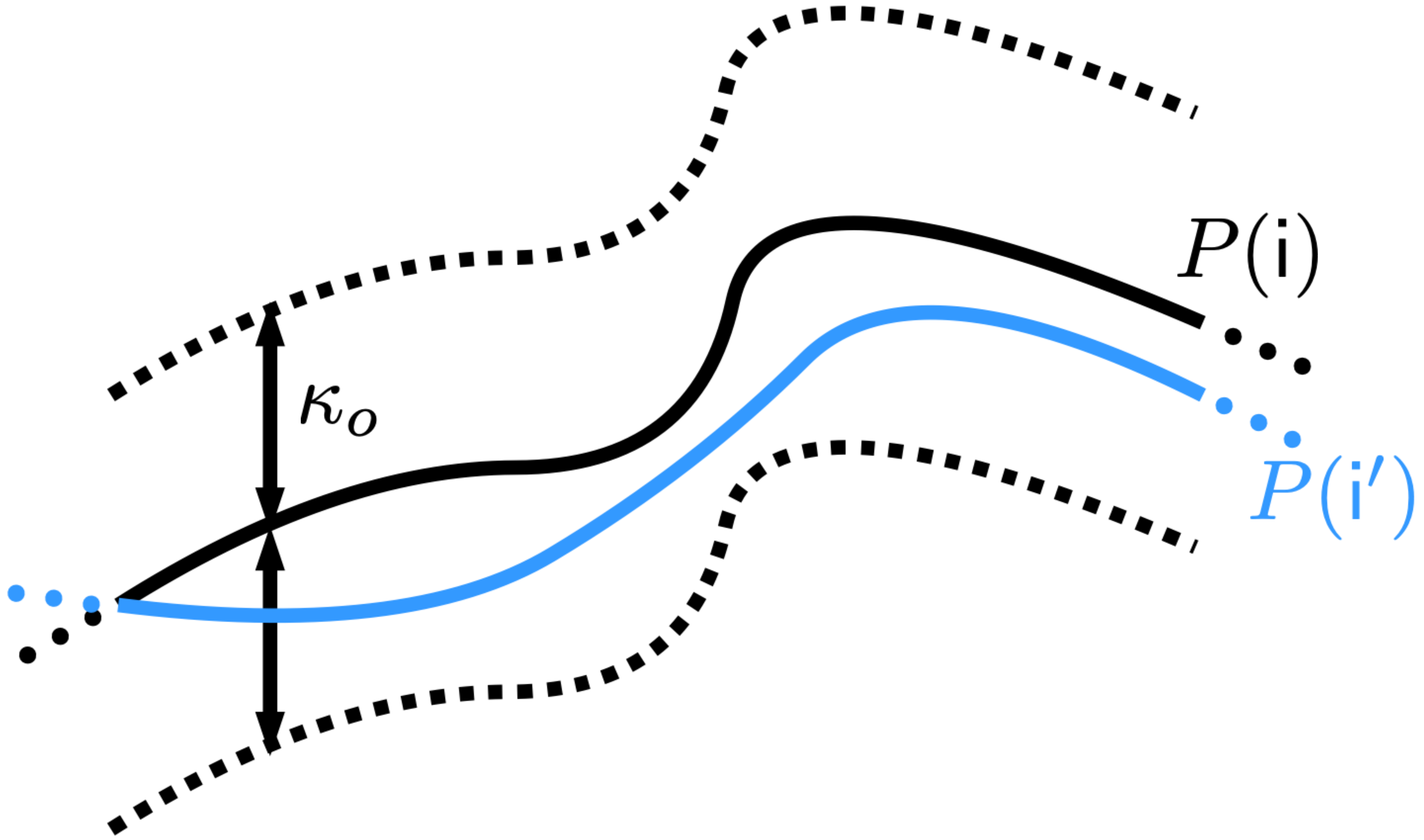
threshold for input distance

$\mathsf{StdIn} \subseteq \mathsf{In}^\omega$ e.g., $\mathsf{StdIn} = \{\mathrm{NEDC}\}$

$\mathsf{i} = \mathrm{NEDC}$ $\mathsf{i} \in \mathsf{StdIn}$

$\mathsf{i}' \neq \mathrm{NEDC}$ $\mathsf{i}' \notin \mathsf{StdIn}$





For all $\mathsf{i} \in \mathsf{StdIn}$, $\mathsf{i}' \in \mathsf{In}^\omega$ and $k \in \mathbb{N}$. If $d_{\mathsf{In}}(\mathsf{i}[..j], \mathsf{i}'[..j]) \leq \kappa_i$ for all $j \leq k$, then $d_{\mathsf{Out}}(P(\mathsf{i})[..k], P(\mathsf{i}')[..k]) \leq \kappa_o$.

Robust Cleanness

sequential

$P : \mathsf{In} \to \mathsf{Out}$

deterministic

distance function for inputs, $(\mathsf{In} \times \mathsf{In}) \to \overline{\mathbb{R}}_{\geq 0}$

distance function for outputs, $(\mathsf{Out} \times \mathsf{Out}) \to \overline{\mathbb{R}}_{\geq 0}$

Contract $\mathcal{C} = \langle \mathsf{StdIn}, d_{\mathsf{In}}, d_{\mathsf{Out}}, \kappa_i, \kappa_o \rangle$

standard inputs

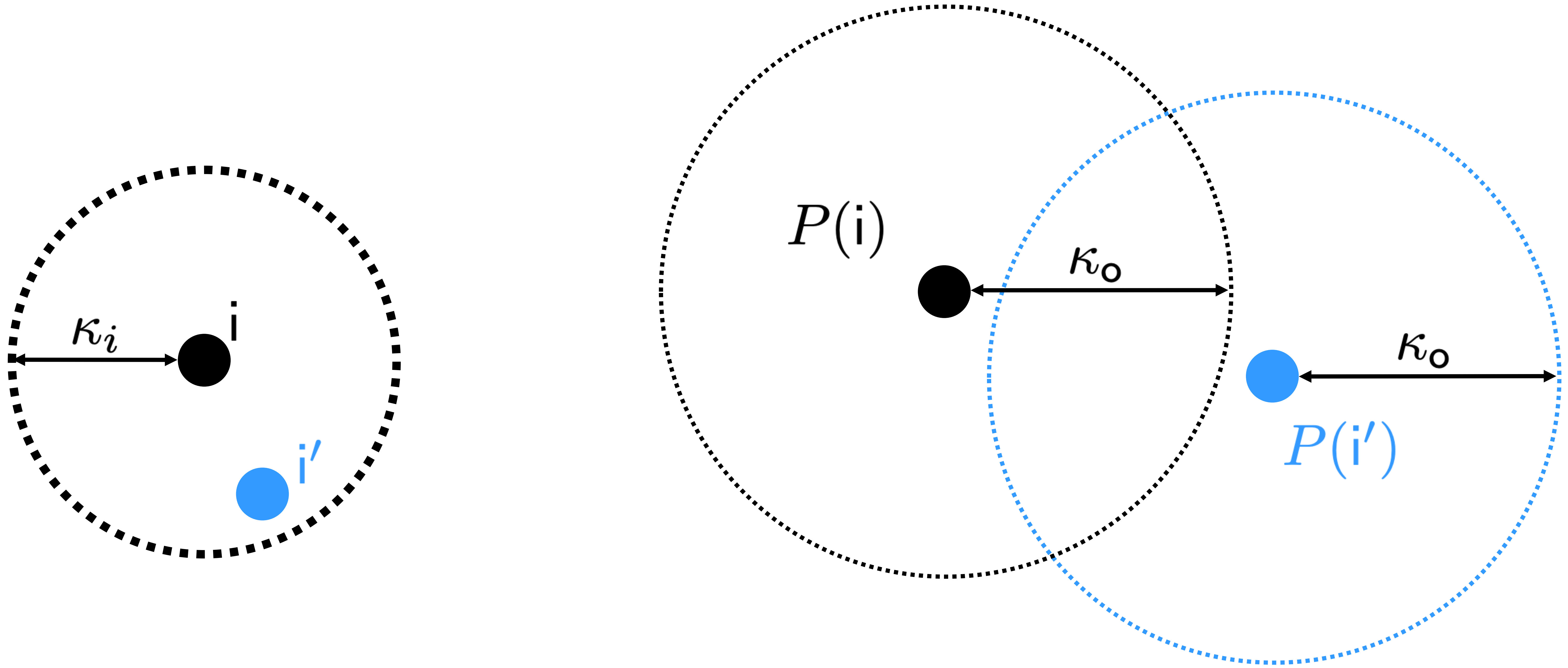
threshold for output distance

threshold for input distance

$\mathsf{StdIn} \subseteq \mathsf{In}$

$\mathsf{i} \in \mathsf{StdIn}$

$\mathsf{i}' \in \mathsf{In}$



For all $\mathsf{i} \in \mathsf{StdIn}$ and $\mathsf{i}' \in \mathsf{In}$. If $d_{\mathsf{In}}(\mathsf{i}, \mathsf{i}') \leq \kappa_i$, then $d_{\mathsf{Out}}(P(\mathsf{i}), P(\mathsf{i}')) \leq \kappa_o$.

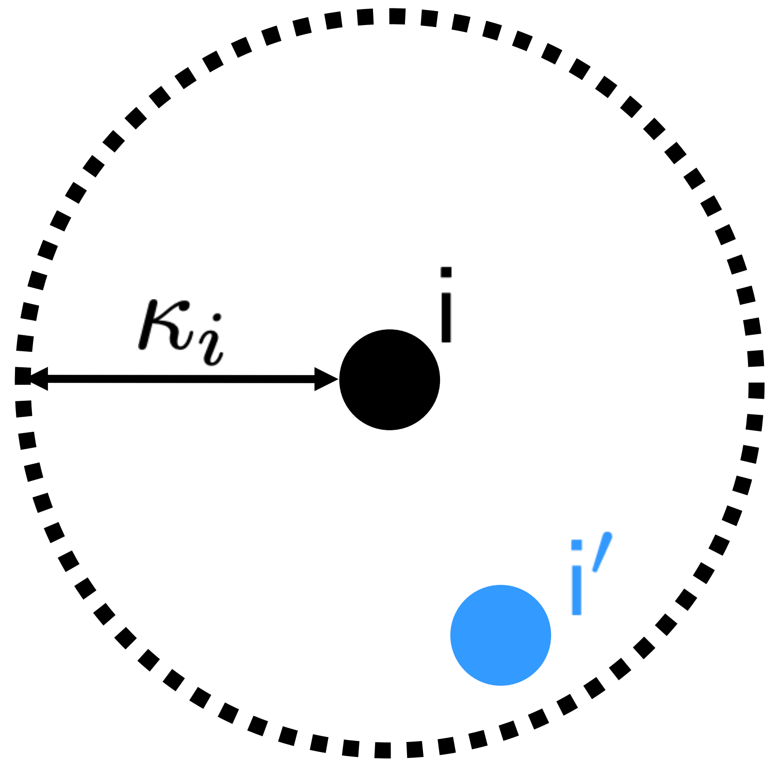
Robust Cleanness

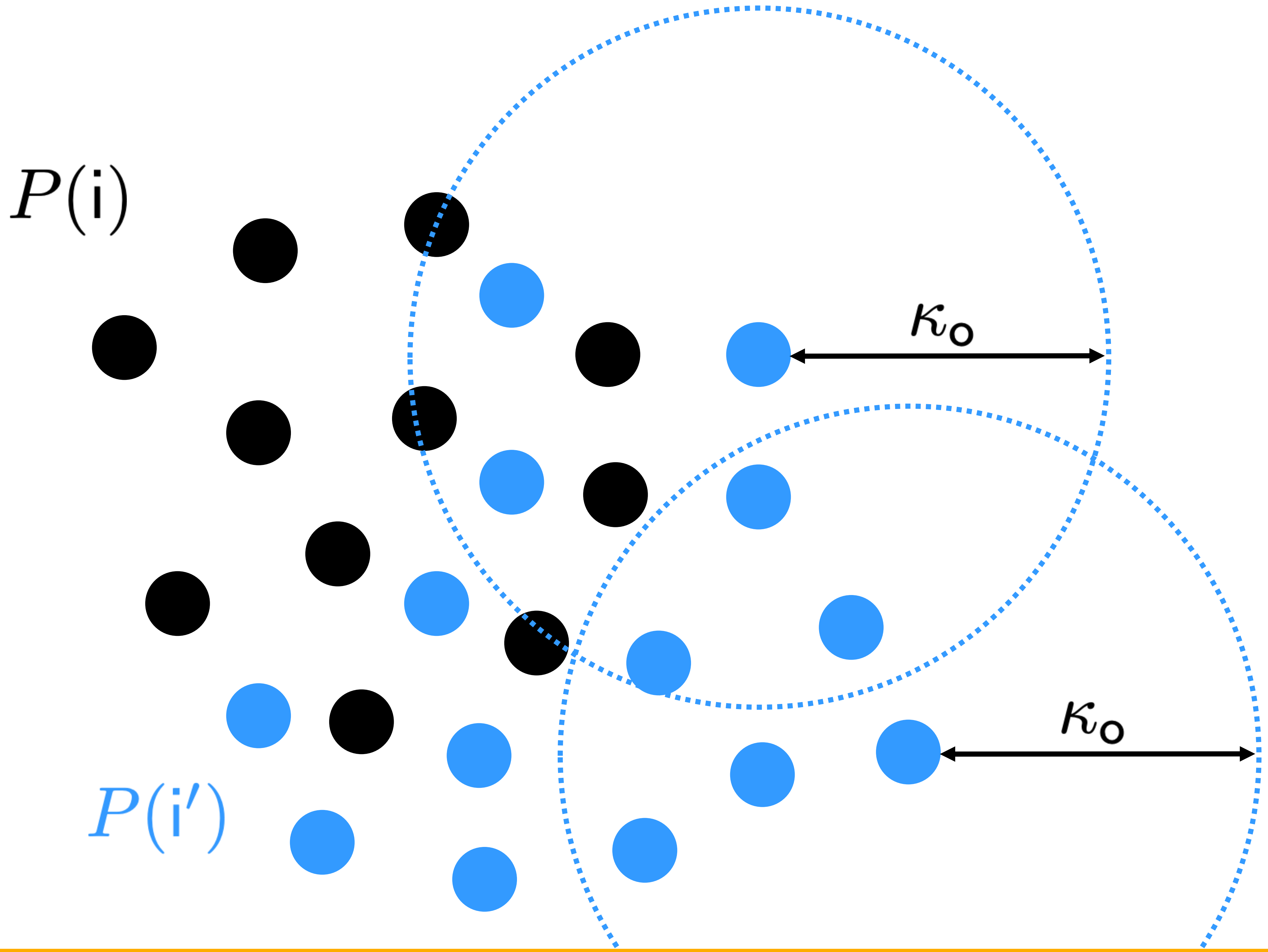
sequential

$P : \mathsf{In} \to 2^{\mathsf{Out}}$

nondeterministic

u-robust cleanness





For all $\mathsf{i} \in \mathsf{StdIn}$ and $\mathsf{i}' \in \mathsf{In}$. If $d_{\mathsf{In}}(\mathsf{i}, \mathsf{i}') \leq \kappa_i$, then for all $\mathsf{o}' \in P(\mathsf{i}')$, there exists $\mathsf{o} \in P(\mathsf{i})$, such that $d_{\mathsf{Out}}(\mathsf{o}, \mathsf{o}') \leq \kappa_\mathsf{o}$.

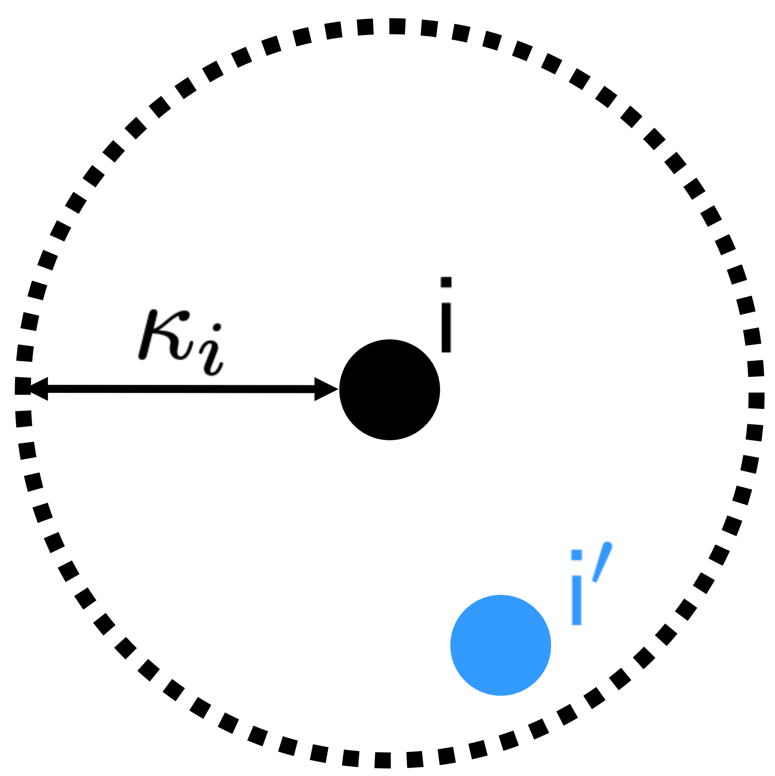
Robust Cleanness

sequential

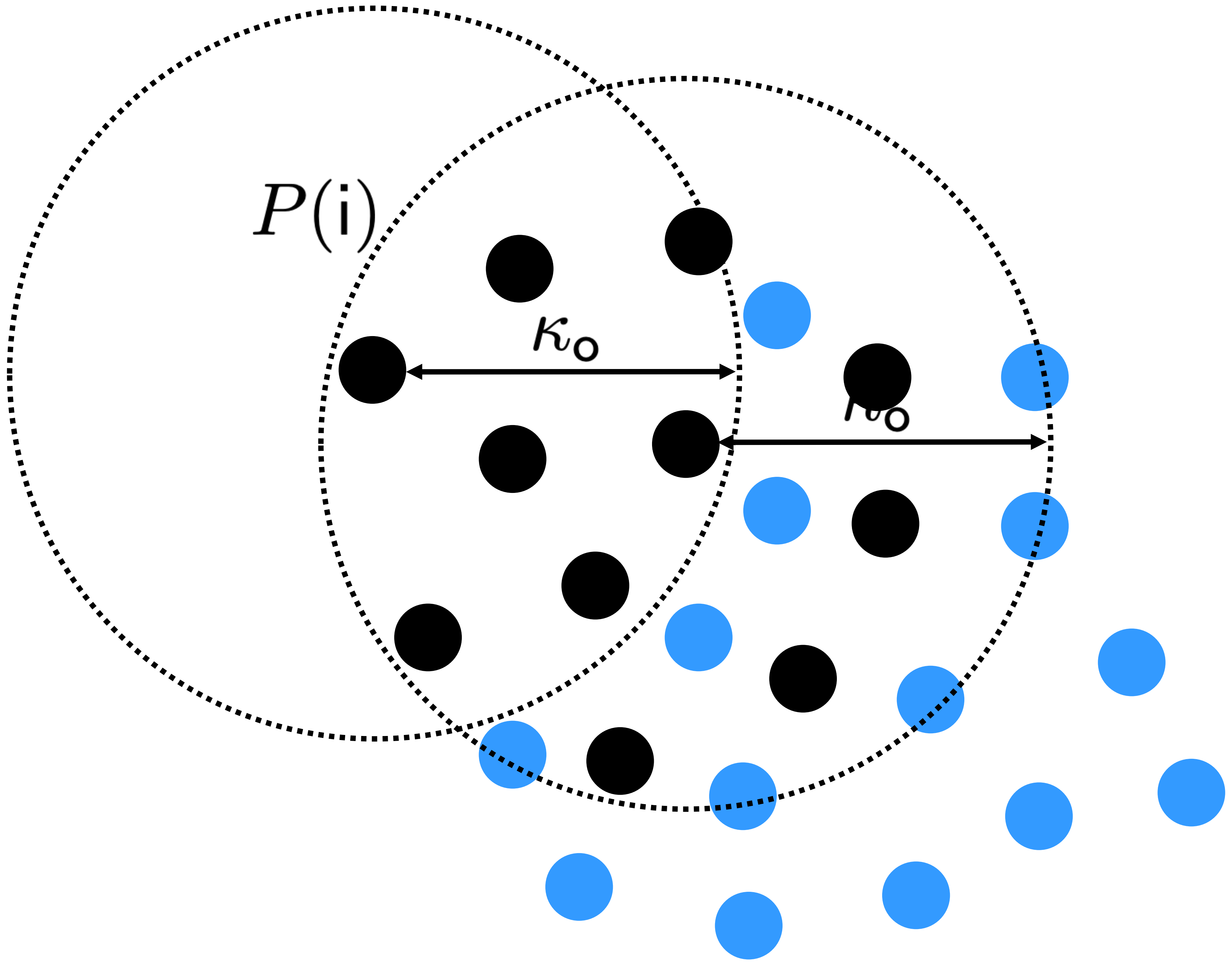
$P : \mathsf{In} \to 2^{\mathsf{Out}}$

nondeterministic

l-robust cleanness



u-robust cleanness



For all $\mathsf{i} \in \mathsf{StdIn}$ and $\mathsf{i}' \in \mathsf{In}$. If $d_{\mathsf{In}}(\mathsf{i}, \mathsf{i}') \leq \kappa_i$, then for all $\mathsf{o} \in P(\mathsf{i})$, there exists $\mathsf{o}' \in P(\mathsf{i}')$, such that $d_{\mathsf{Out}}(\mathsf{o}, \mathsf{o}') \leq \kappa_{\mathsf{o}}$.

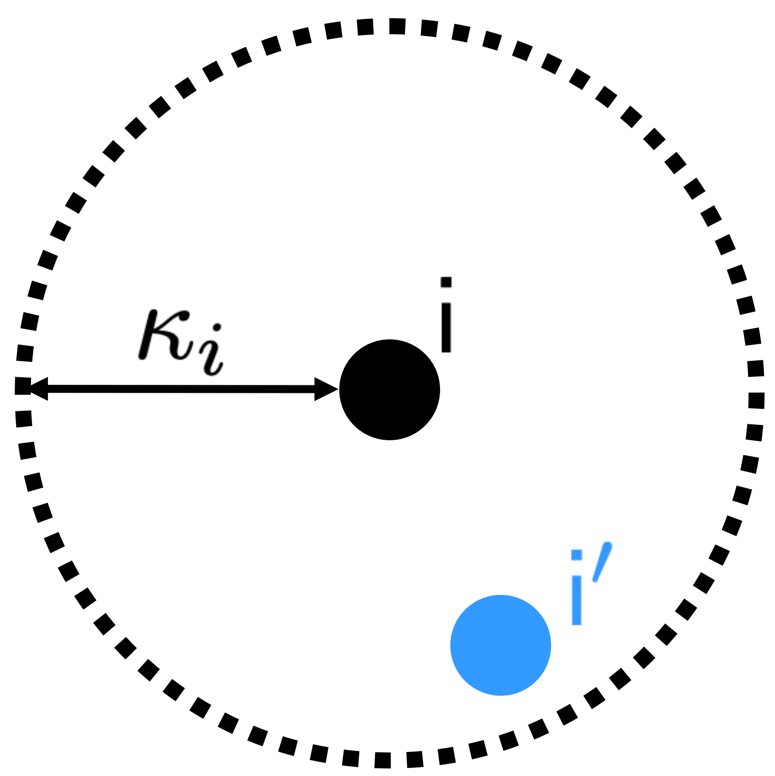
Robust Cleanness

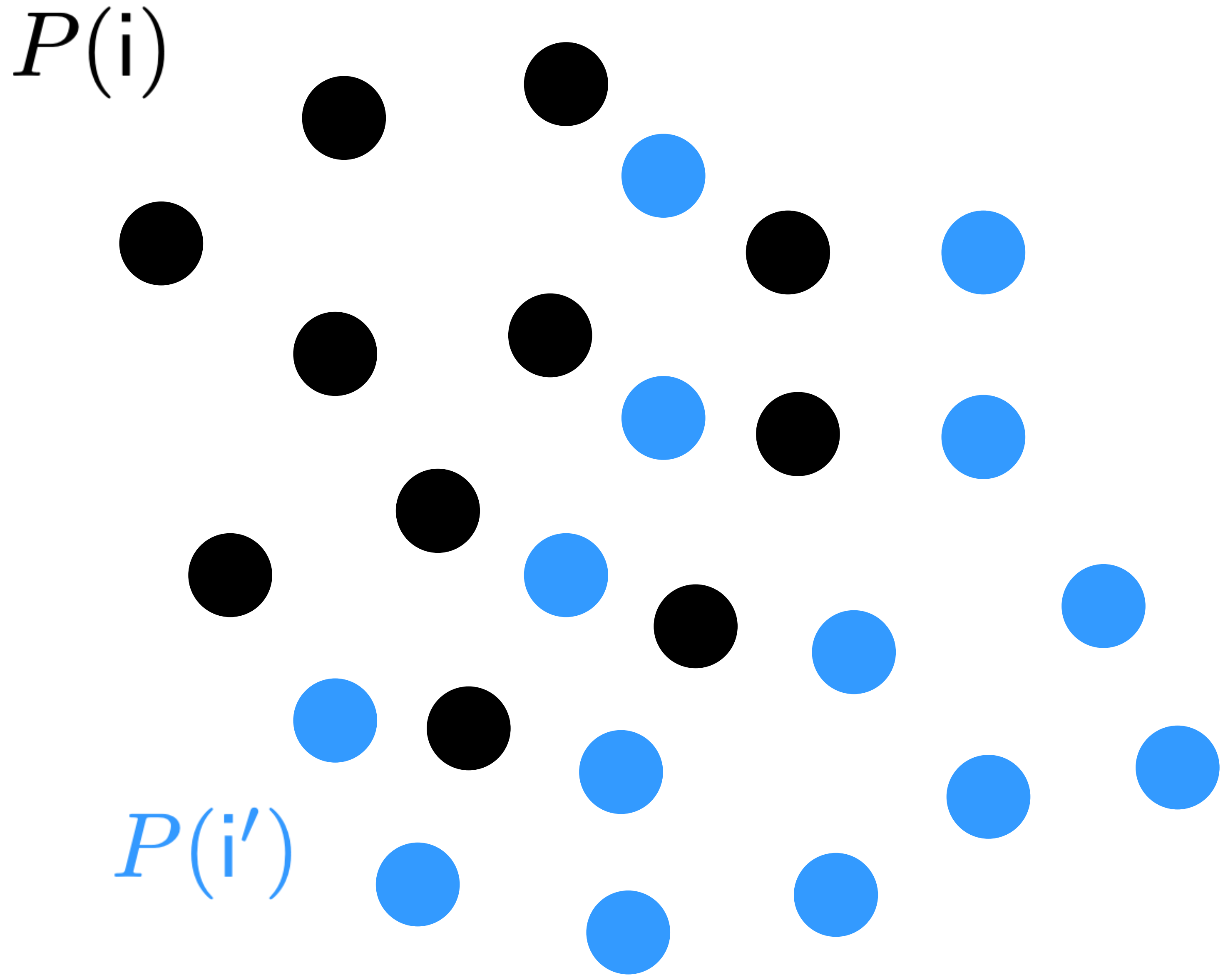
sequential

$$P : \mathsf{In} \to 2^{\mathsf{Out}}$$

nondeterministic

l-robust cleanness + u-robust cleanness ≈ Hausdorff-based robust cleanness





$$\mathcal{H}(d_{\mathsf{Out}})(P(\mathsf{i}), P(\mathsf{i}')) \leq \kappa_o$$

Hausdorff distance

For all $\mathsf{i} \in \mathsf{StdIn}$ and $\mathsf{i}' \in \mathsf{In}$. If $d_{\mathsf{In}}(\mathsf{i}, \mathsf{i}') \leq \kappa_i$, then $\mathcal{H}(d_{\mathsf{Out}})(P(\mathsf{i}), P(\mathsf{i}')) \leq \kappa_o$.

Robust Cleanness in Temporal Logic

robust cleanness in HyperLTL:

$$\forall \pi_1. \forall \pi_2. \exists \pi'_1.\ \mathsf{StdIn}_{\pi_1} \to \Big(\mathsf{G}(\mathsf{i}_{\pi_1} = \mathsf{i}_{\pi'_1}) \wedge \big((\hat{d}_{\mathsf{Out}}(\mathsf{o}_{\pi'_1}, \mathsf{o}_{\pi_2}) \leq \kappa_{\mathsf{o}})\ \mathsf{W}\ (\hat{d}_{\mathsf{In}}(\mathsf{i}_{\pi'_1}, \mathsf{i}_{\pi_2}) > \kappa_{\mathsf{i}})\big)\Big)$$

robust cleanness in Hyper<u>S</u>TL:

$$\forall \pi_1. \forall \pi_2. \exists \pi'_1.\ \mathsf{StdIn}_{\pi_1} > 0 \to \Big(\mathsf{G}(|\mathsf{i}_{\pi_1} - \mathsf{i}_{\pi'_1}| \leq 0) \wedge \big((d_{\mathsf{Out}}(\mathsf{o}_{\pi'_1}, \mathsf{o}_{\pi_2}) - \kappa_{\mathsf{o}} \leq 0)\ \mathsf{W}\ (d_{\mathsf{In}}(\mathsf{i}_{\pi'_1}, \mathsf{i}_{\pi_2}) - \kappa_{\mathsf{i}} > 0)\big)\Big)$$

robust cleanness for <u>finite standard</u> behaviour in STL:

$$\bigwedge_{1 \leq a \leq c} \bigvee_{1 \leq b \leq c} \Big(\mathsf{G}(|\mathsf{i}_a - \mathsf{i}_b| \leq 0) \wedge \big((d_{\mathsf{Out}}(\mathsf{o}_b, \mathsf{o}) - \kappa_{\mathsf{o}} \leq 0)\ \mathsf{W}\ (d_{\mathsf{In}}(\mathsf{i}_b, \mathsf{i}) - \kappa_{\mathsf{i}} > 0)\big)\Big)$$

with self-composition by "copying" standard signals into the trace to be checked:

$$w = (\mathsf{i}, \mathsf{o}) \quad \rightsquigarrow \quad w' = (\mathsf{i}, \mathsf{o}, \mathsf{i}_1, \mathsf{o}_1, \ldots, \mathsf{i}_c, \mathsf{o}_c)$$

Analysis

Cleanness is an observation-based property

$\mathbf{P}(\mathcal{O})$

for, e.g., $\mathcal{O} \subseteq \mathsf{In}^\omega \times \mathsf{Out}^\omega$

White Box

We <u>know a model</u> that defines $\mathcal{O}$

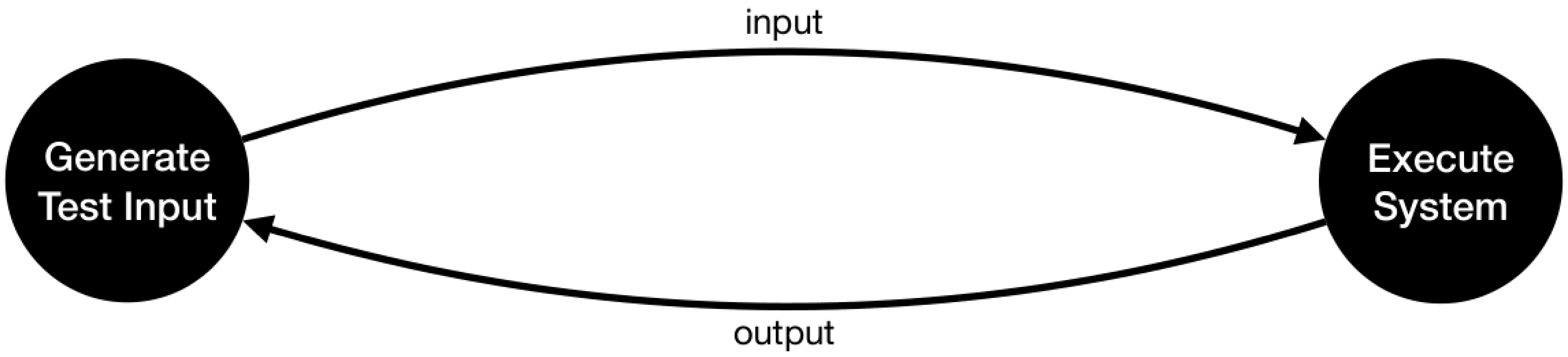
➔ Model-Checking

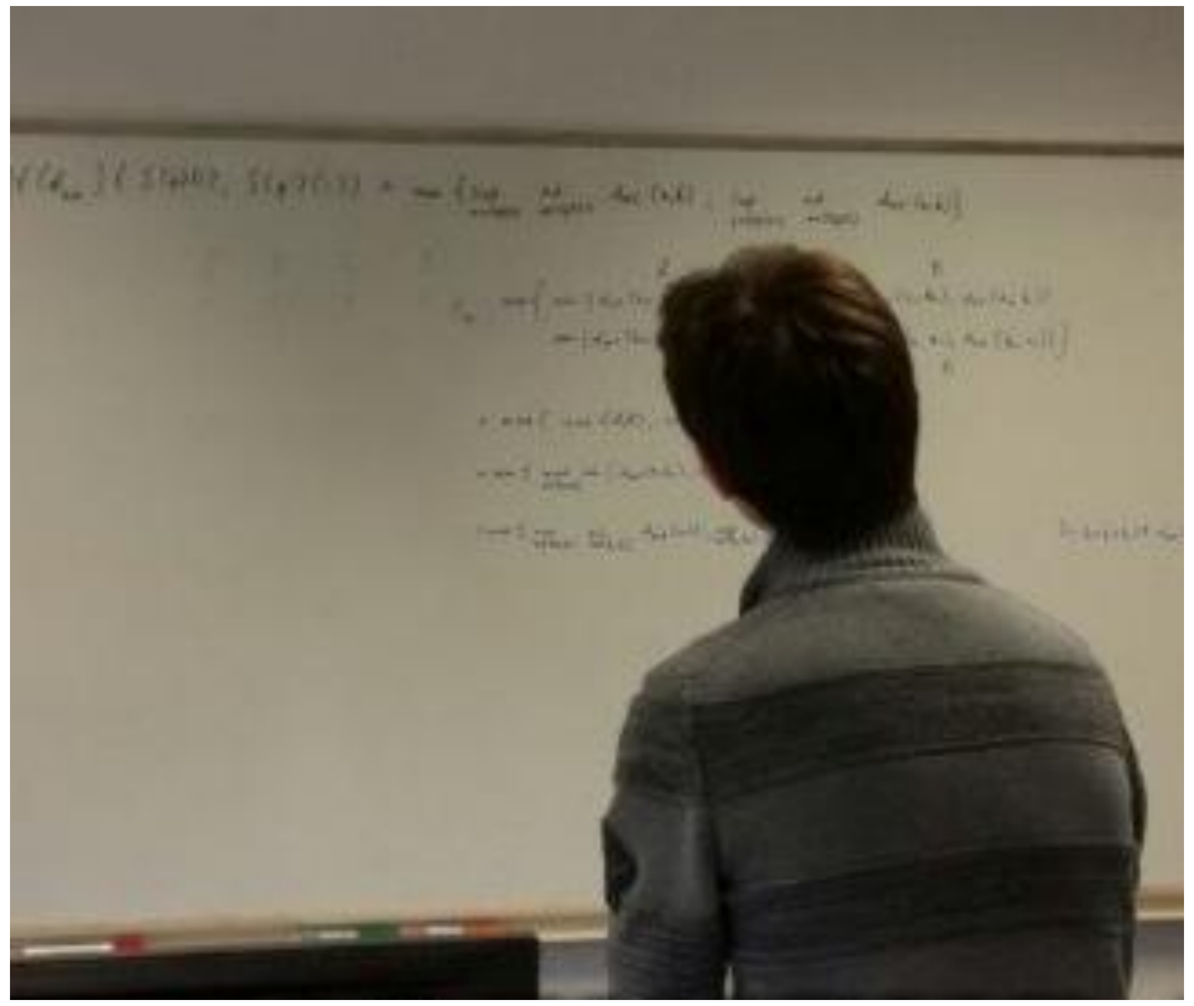
Black Box

We <u>know a subset</u> $\mathcal{O}' \subset \mathcal{O}$ of the system's behaviour

➔ Testing or Monitoring

Testing, classically





1: Invent a test cycle

approx. 1 day per test cycle

2: Fix the car on a chassis dynamometer, attach an emissions measurement device, calibrate it, ...

approx. 1 hr

3: Drive the test cycle

between 30 mins and 1 day for one test cycle

Abbas, Fainekos, Sankaranarayanan, Ivancic, Gupta: Probabilistic temporal logic falsification of cyber-physical systems. ACM Trans. Embed. Comput. Syst. 2013.

Probabilistic Falsification

Temporal Logic e.g., Signal Temporal Logic (STL)

$$\phi ::= \top \mid f > 0 \mid \neg\phi \mid \phi \vee \phi \mid \phi \,\mathcal{U}\, \phi$$

Semantics:

system trace

$w, t \models \phi$ STL formula

time point

$\mathbb{B}$

Quantitative Semantics

Temporal Logic e.g., Signal Temporal Logic (STL)

$$\phi ::= \top \mid f > 0 \mid \neg\phi \mid \phi \vee \phi \mid \phi \,\mathcal{U}\, \phi$$

Semantics:

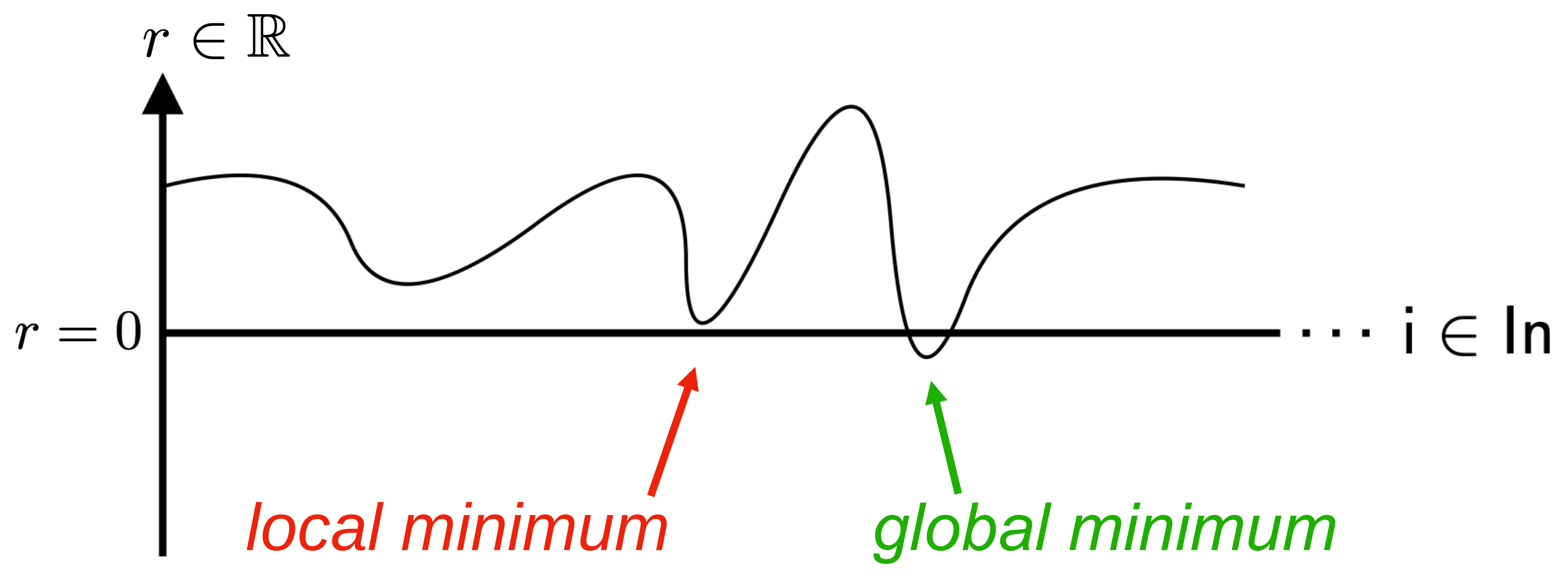
system trace

$\rho(\phi, w, t) = r$ robustness estimate

STL formula

time point

$r \in \mathbb{R}$

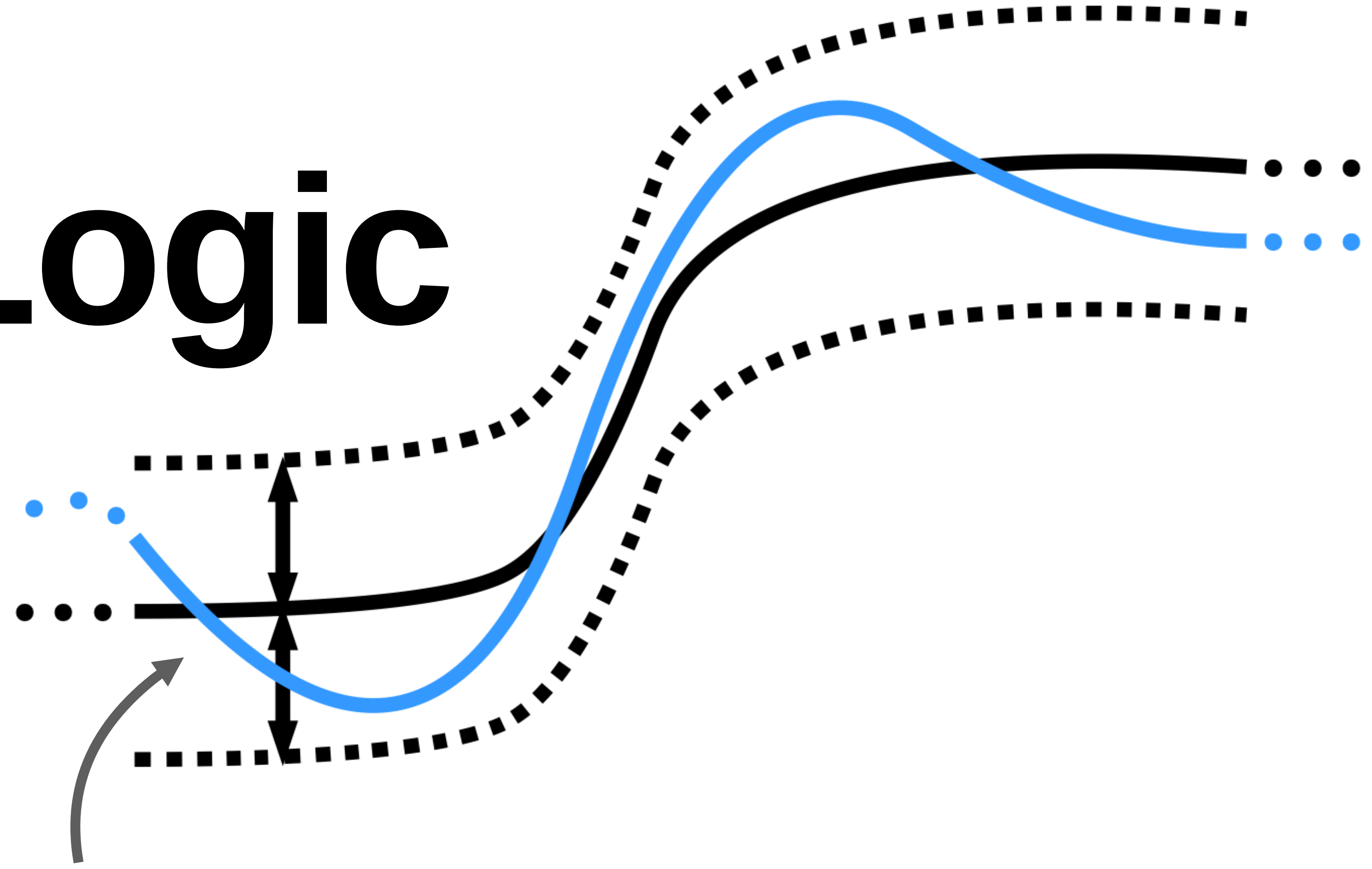


$$\rho(\phi, w, t) > 0 \Rightarrow w, t \models \phi$$

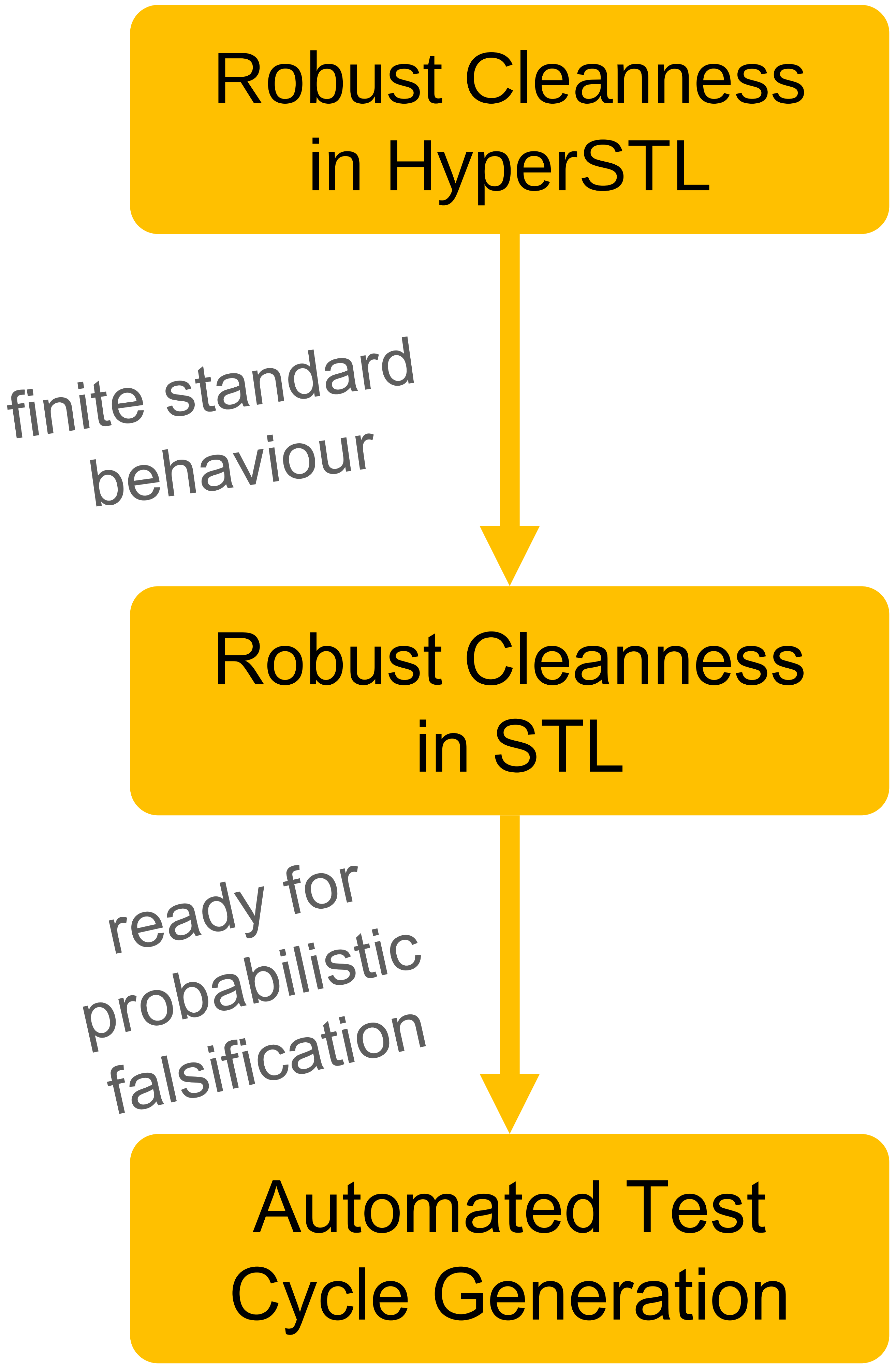
$$\rho(\phi, w, t) < 0 \Rightarrow w, t \not\models \phi$$

Falsification by optimisation: $\text{minimise}_w \; \rho(\phi, w, 0)$

Robust Cleanness in Temporal Logic



reasoning about two traces simultaneously



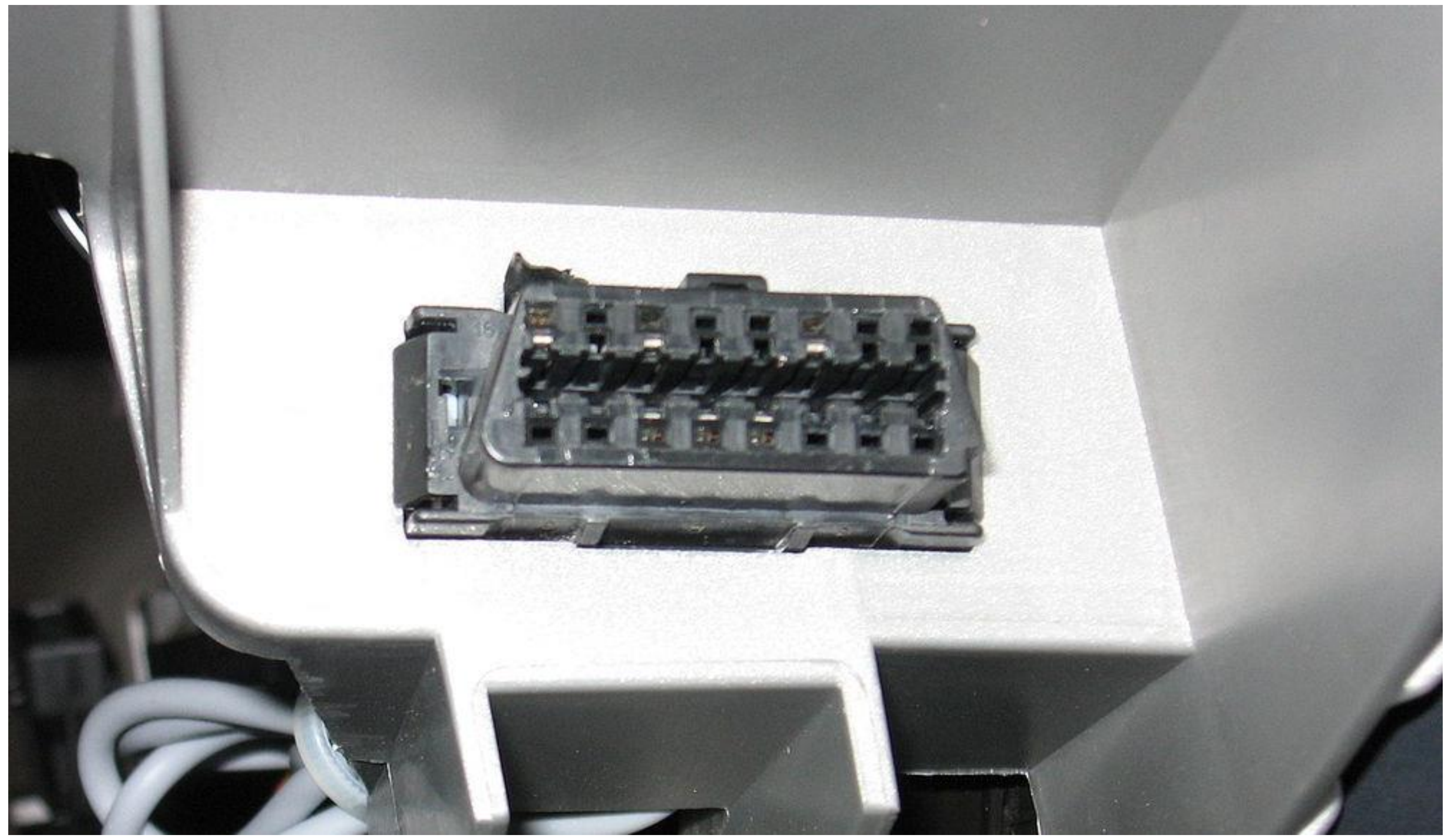
Algorithm 2.1 Monte-Carlo falsification

Input: w: Initial trace, $\mathcal{R}$: Robustness function, PS: Proposal Scheme

Output: $w \in \mathsf{M}$

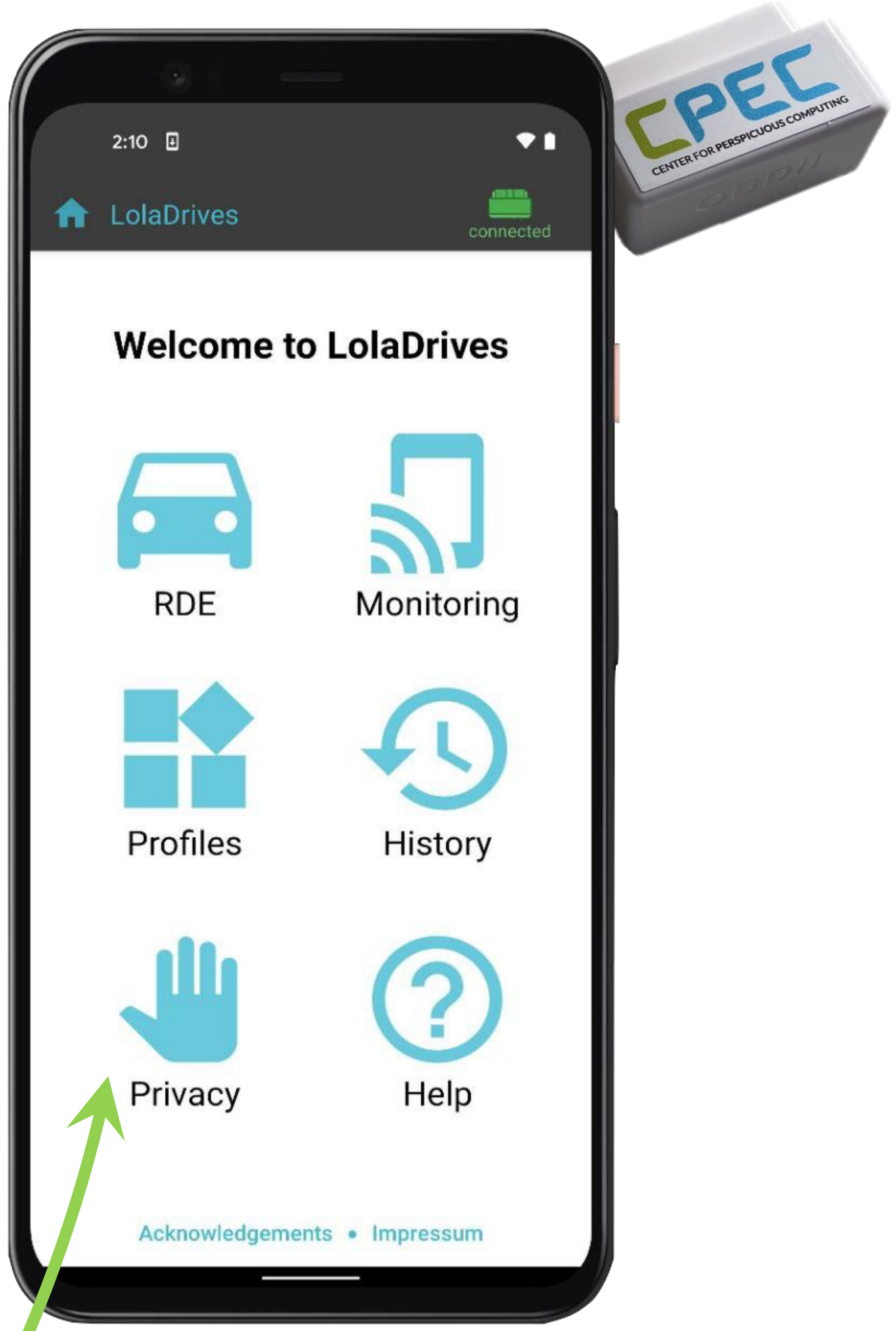
```
while R(w) > 0 do
    w' ← PS(w)
    α ← exp(−β(R(w') − R(w)))
    r ← UniformRandomReal(0, 1)
    if r ≤ α then
        w ← w'
    end if
end while
```

LolaDrives



On-Board Diagnostics (OBD)

OBD ↔ Bluetooth Adapter



Smartphone

LolaDrives App

➔ Originally for Real Driving Emissions Tests

➔ Can replace the external NOx emissions measurement device

Prediction of emission behaviour

speed

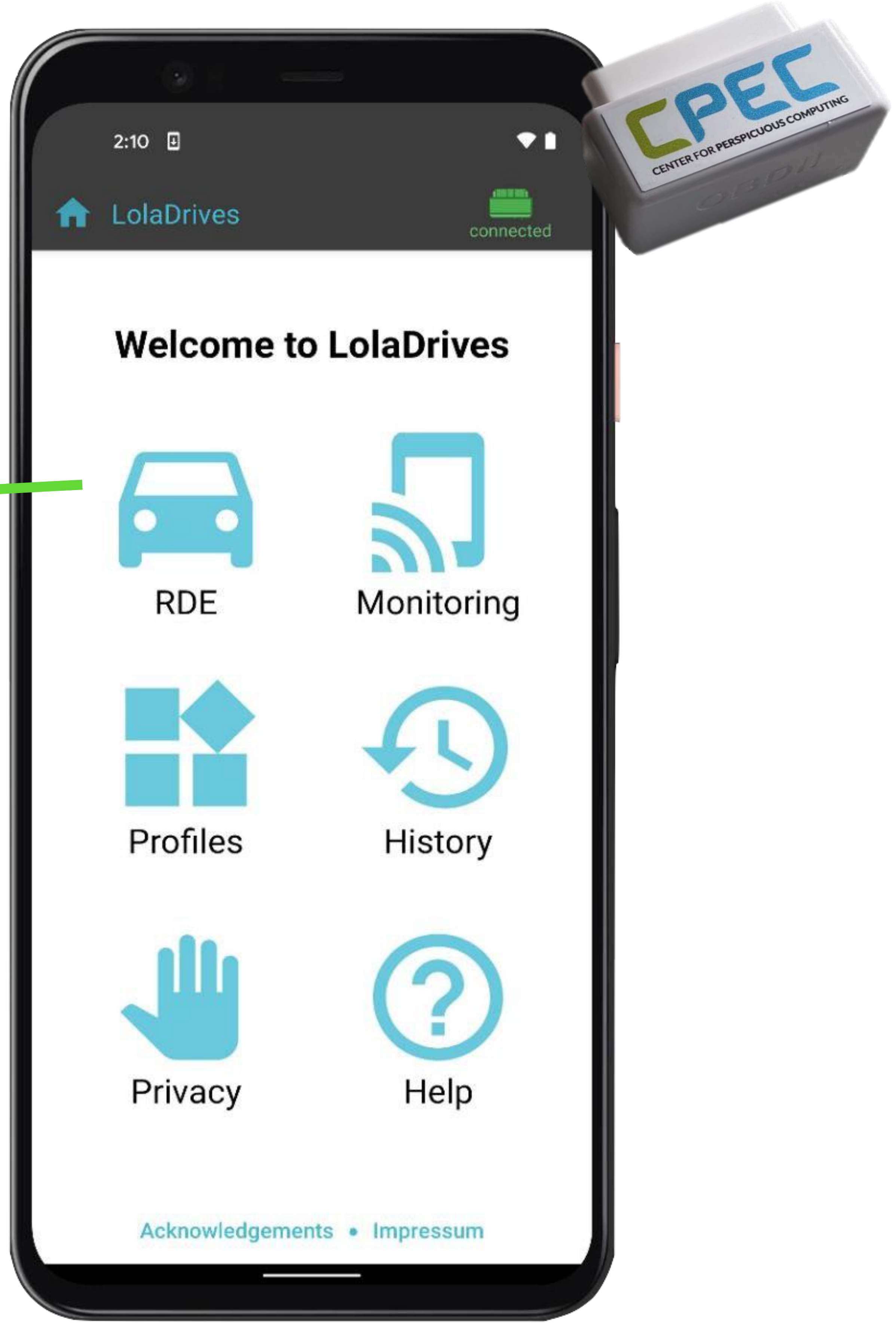
acceleration

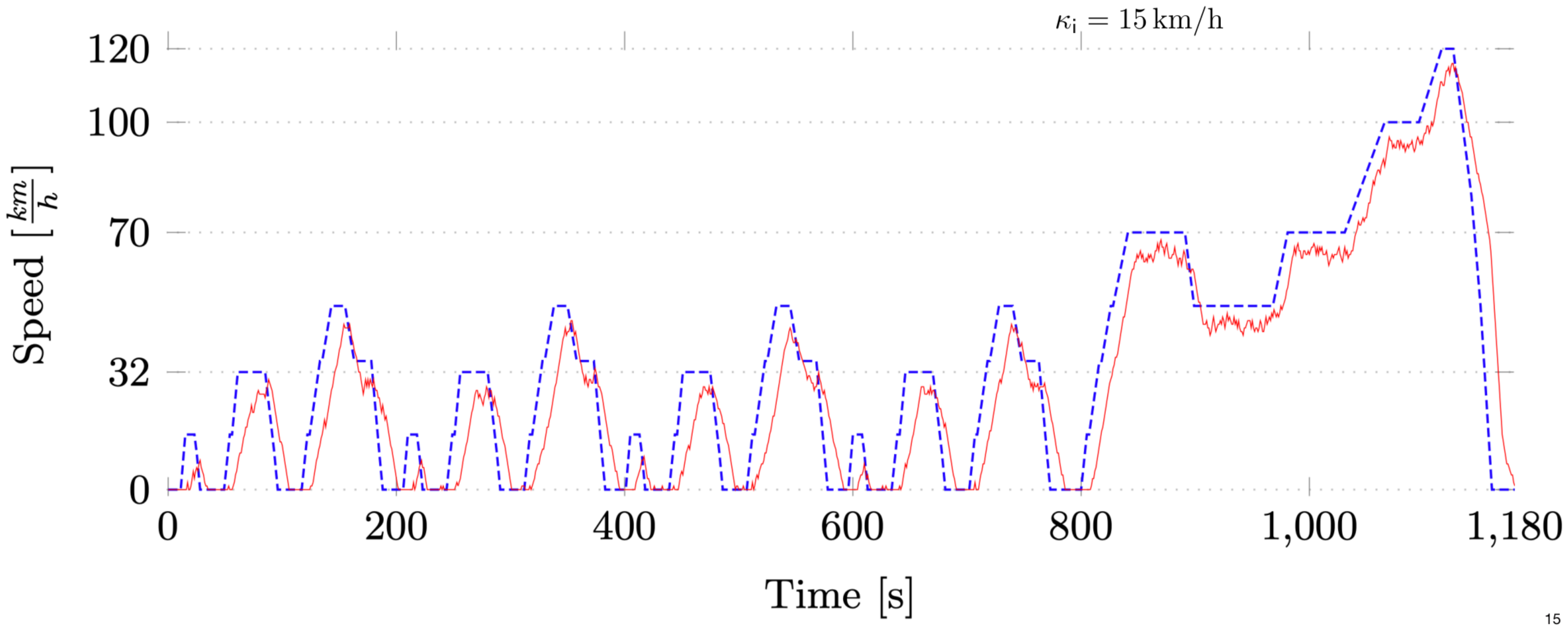
NOx value

real driving emissions data

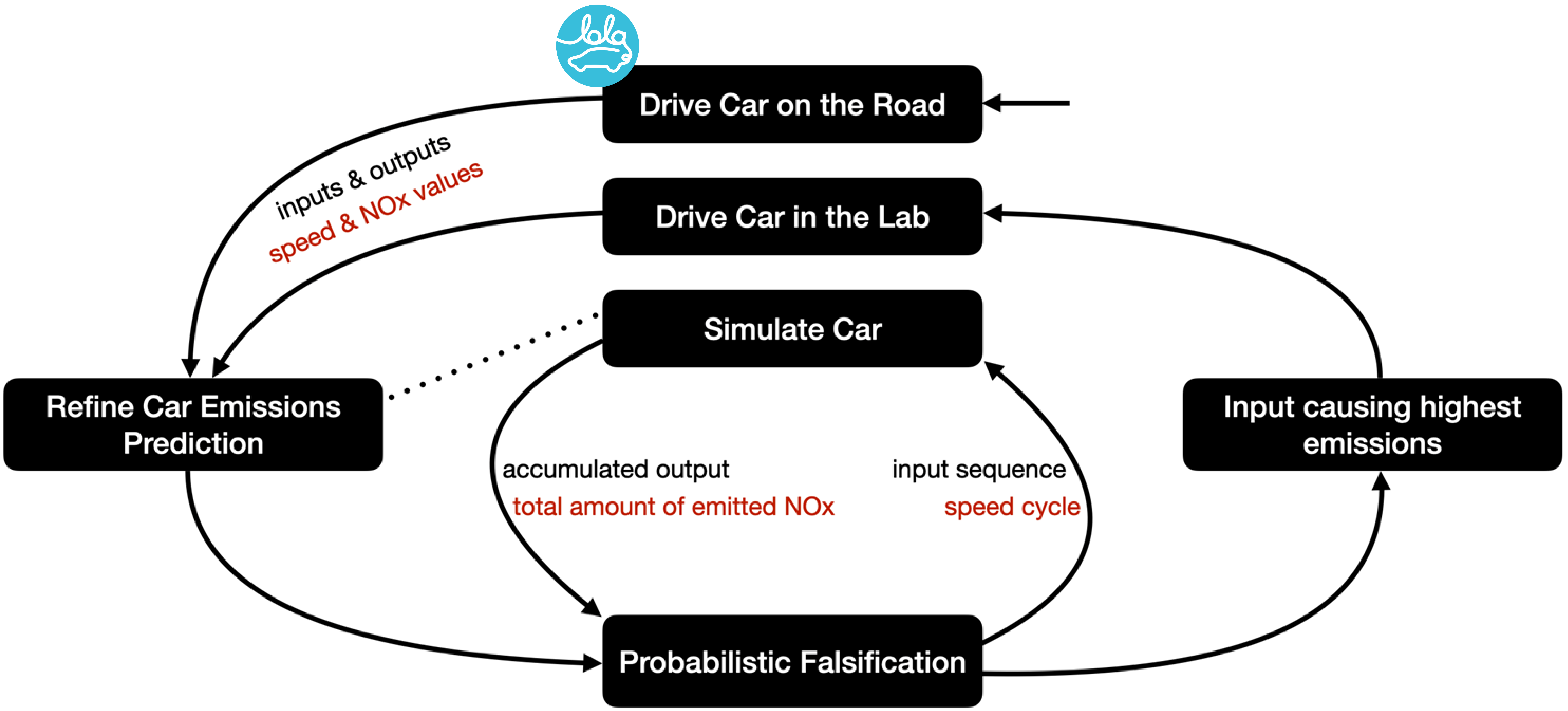
$$\mathcal{P}(v, a) = \text{average}[n \mid (v, a, n) \in \mathcal{D}]$$

Binning of pairs of speed and acceleration





An Integrated Testing Approach



A synthesised test input

Audi A6 Avant (2020)

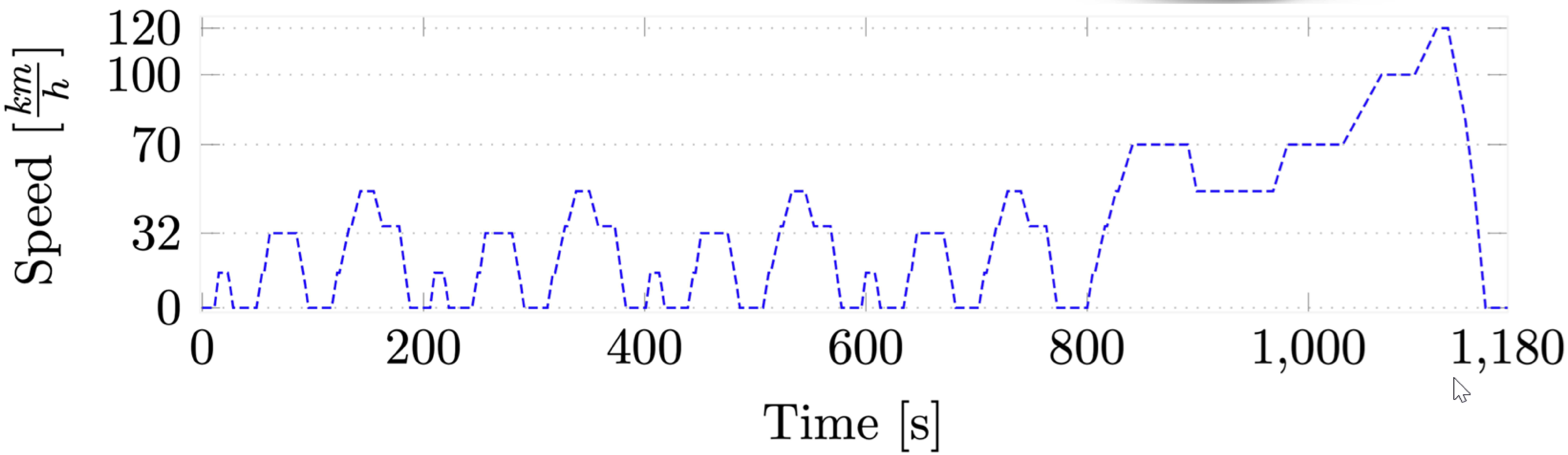
NEDC emissions (NOx): 86 mg/km

Generated emissions (NOx): 182 mg/km

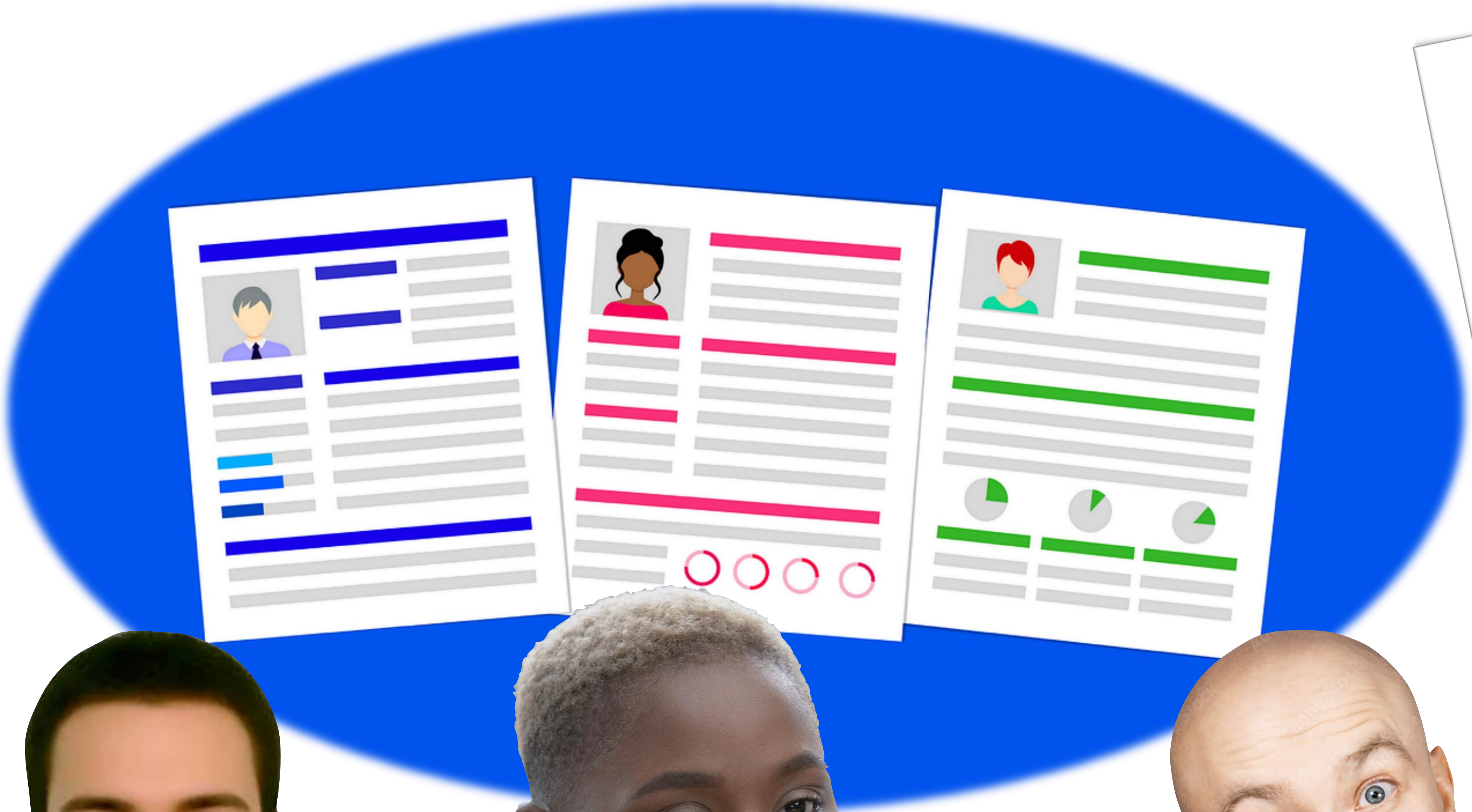
$\kappa_{\mathrm{i}} = 15\,\mathrm{km/h}$

Example – Software Doping

New European Driving Cycle (NEDC):



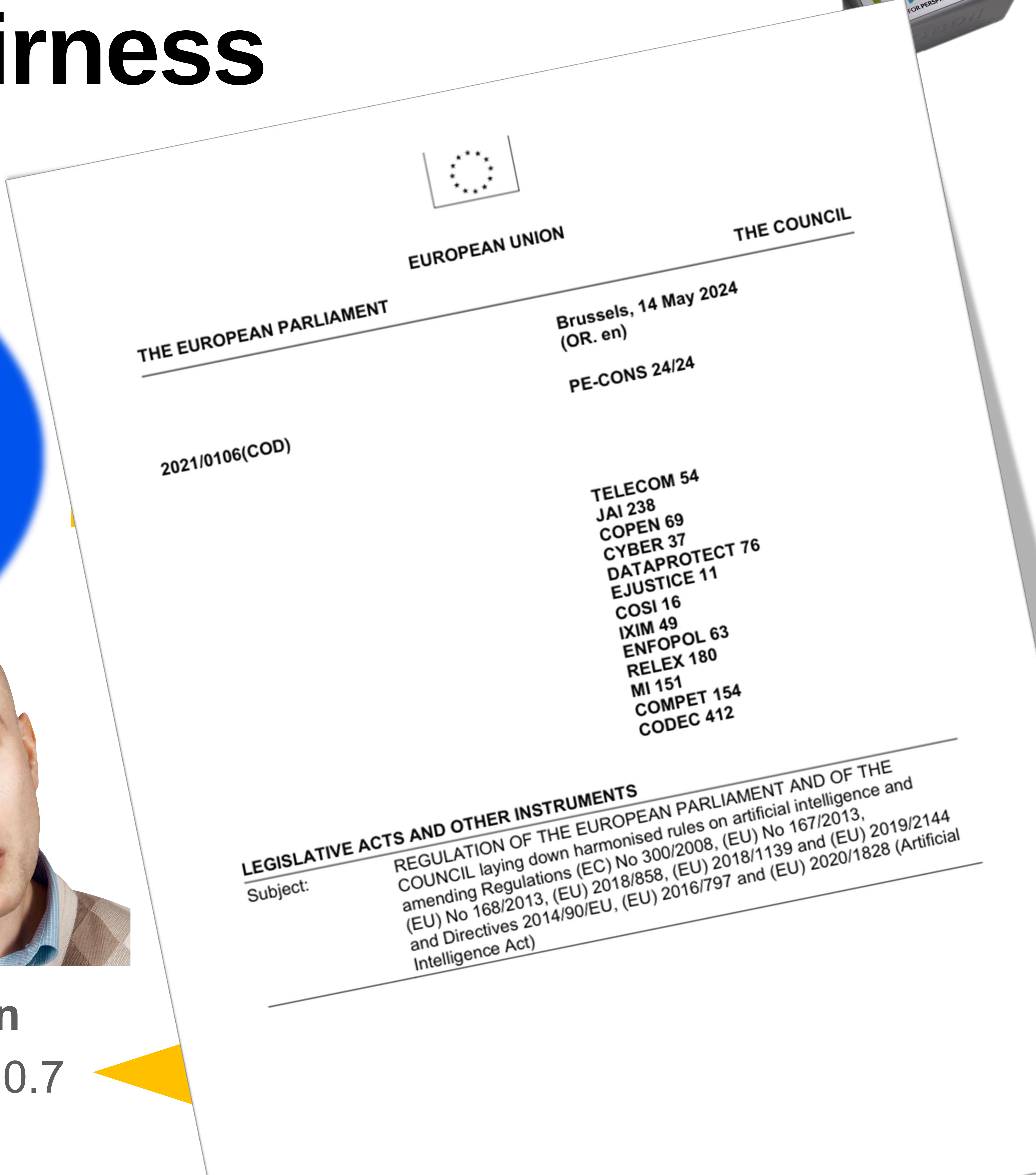
Example – Individual Fairness



Eugene
Score: 0.9

Alexa
Score: 0.5

John
Score: 0.7



EUROPEAN UNION

THE EUROPEAN PARLIAMENT

THE COUNCIL

Brussels, 14 May 2024
(OR. en)

2021/0106(COD)

PE-CONS 24/24

TELECOM 54
JAI 238
COPEN 69
CYBER 37
DATAPROTECT 76
EJUSTICE 11
COSI 16
IXIM 49
ENFOPOL 63
RELEX 180
MI 151
COMPET 154
CODEC 412

LEGISLATIVE ACTS AND OTHER INSTRUMENTS

Subject: REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act)

AI Act

Regulates the use of AI in Europe.

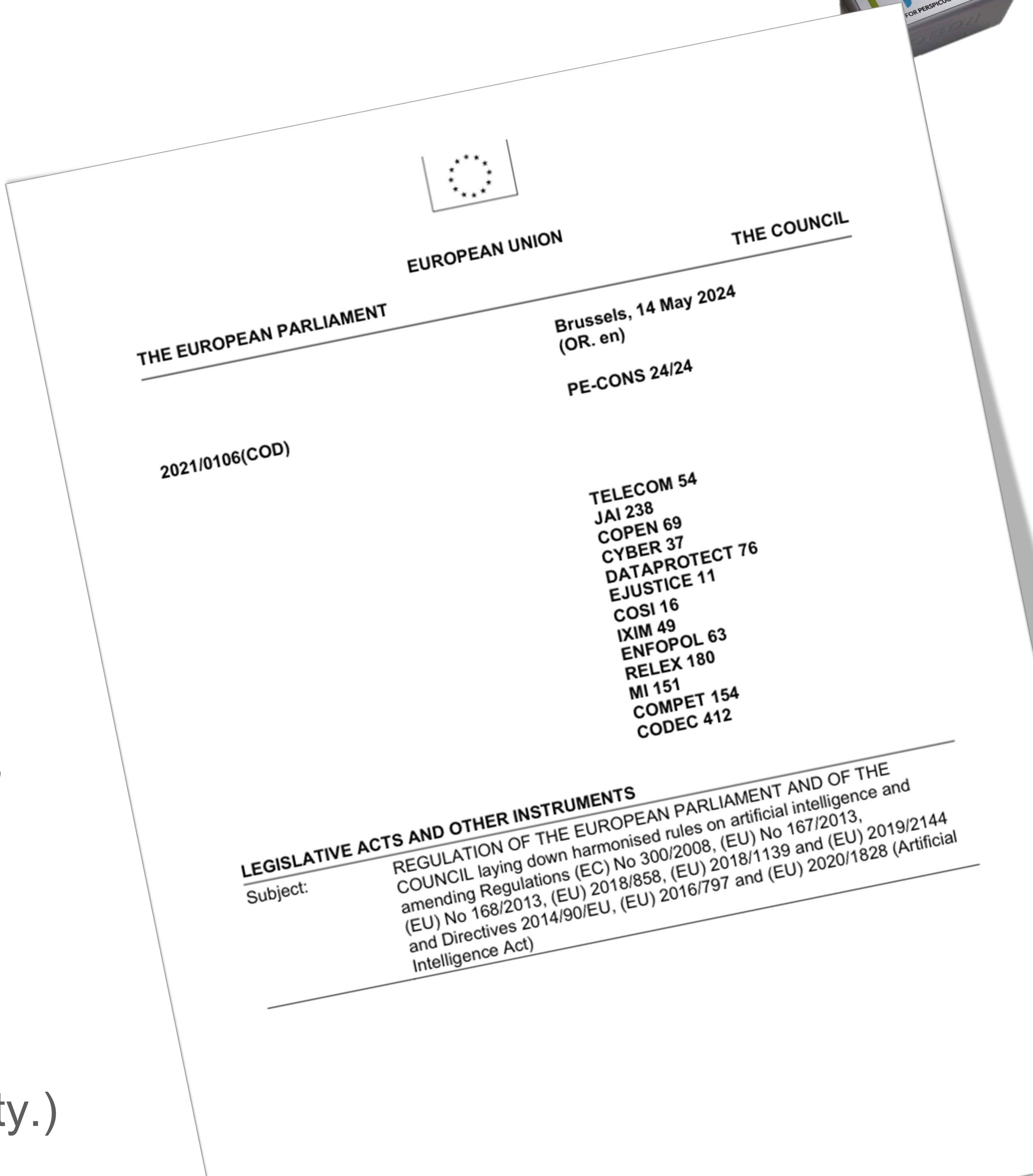
Final signature on June 13, 2024.

Publication expected soon:

Official Journal of the European Union

Is about “risks” and about “AI systems”.

(Spin is inherited from regulatory texts on product safety.)



EUROPEAN UNION

THE EUROPEAN PARLIAMENT

THE COUNCIL

Brussels, 14 May 2024
(OR. en)

PE-CONS 24/24

2021/0106(COD)

TELECOM 54
JAI 238
COPEN 69
CYBER 37
DATAPROTECT 76
EJUSTICE 11
COSI 16
IXIM 49
ENFOPOL 63
RELEX 180
MI 151
COMPET 154
CODEC 412

LEGISLATIVE ACTS AND OTHER INSTRUMENTS

Subject: REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act)

AI System?

An AI system can infer how to generate outputs from inputs or data.

predictions, content, recommendations, or decisions which can influence physical and virtual environments

Inference by

- machine learning approaches
  that learn from data how to achieve certain objectives, or
- logic- and knowledge-based approaches
  that derive from encoded knowledge or
  from symbolic representation of the task to be solved.

AI systems have some degree of independence of actions from human involvement
and of capabilities to operate without human intervention.

AI System?

An AI system can infer how to generate outputs from inputs or data.

predictions, content, recommendations, or decisions which can influence physical and virtual environments

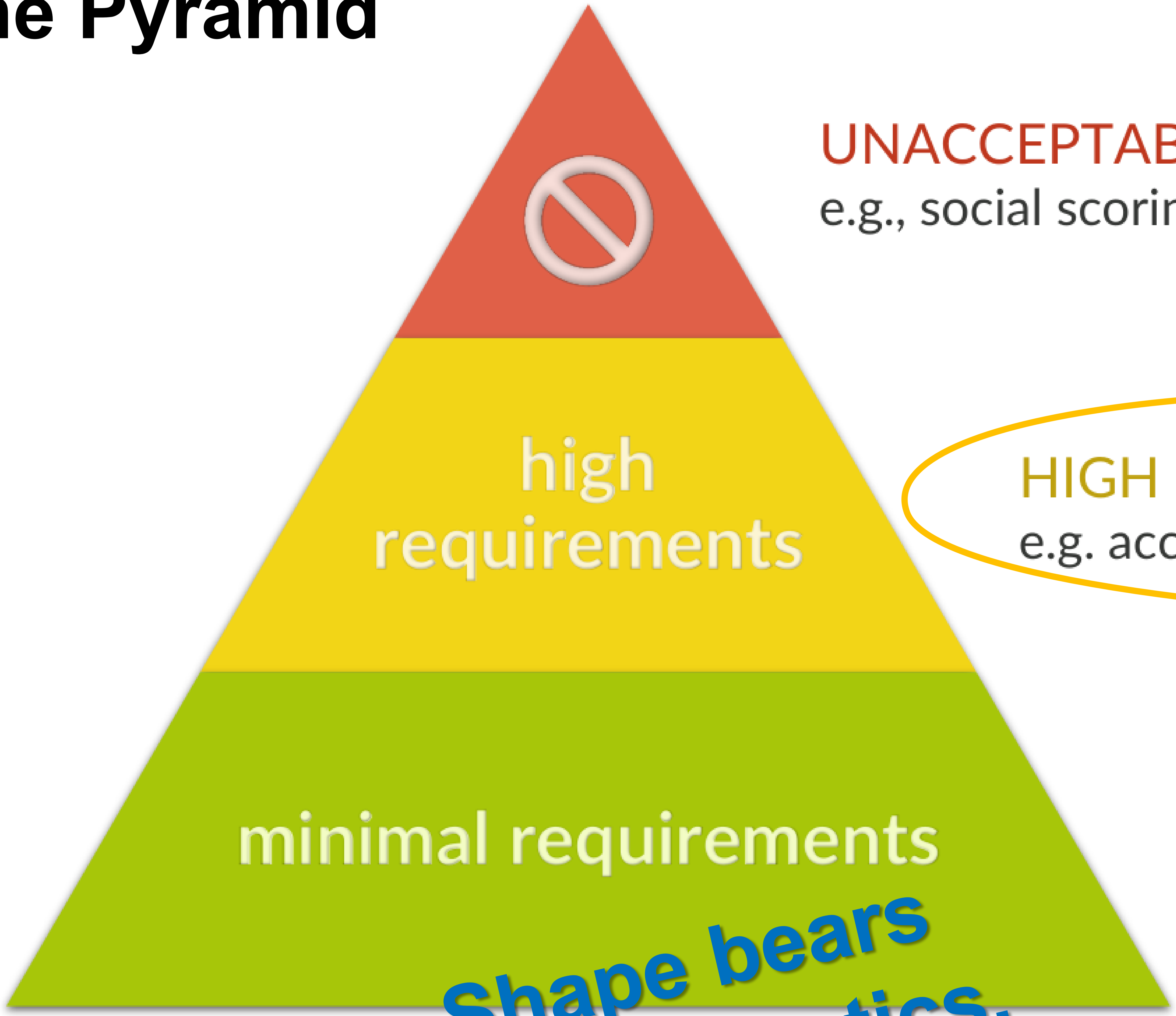
Inference by

- machine learning approaches
  that learn from data how to achieve certain objectives, or
- logic- and knowledge-based approaches
  that derive from encoded knowledge or
  from symbolic representation of the task to be solved.

AI systems have some degree of independence of actions from human involvement
and of capabilities to operate without human intervention.

AI Risks

The Pyramid



Shape bears no semantics.

UNACCEPTABLE RISK
e.g., social scoring, certain facial recognition

HIGH RISK
e.g. access to education, hiring, immigration

MINIMAL RISK
e.g. spam filters, video games

AI System? High Risk?

~~AI~~

- A compiler for a high-level programming language regardless of its (potentially excessive) complexity, used to compile the code to run an airbag controller.

high risk

AI

- A purely logic-based system that can infer how to decide whether the airbag inside some car has to ignite.

high risk

~~AI~~

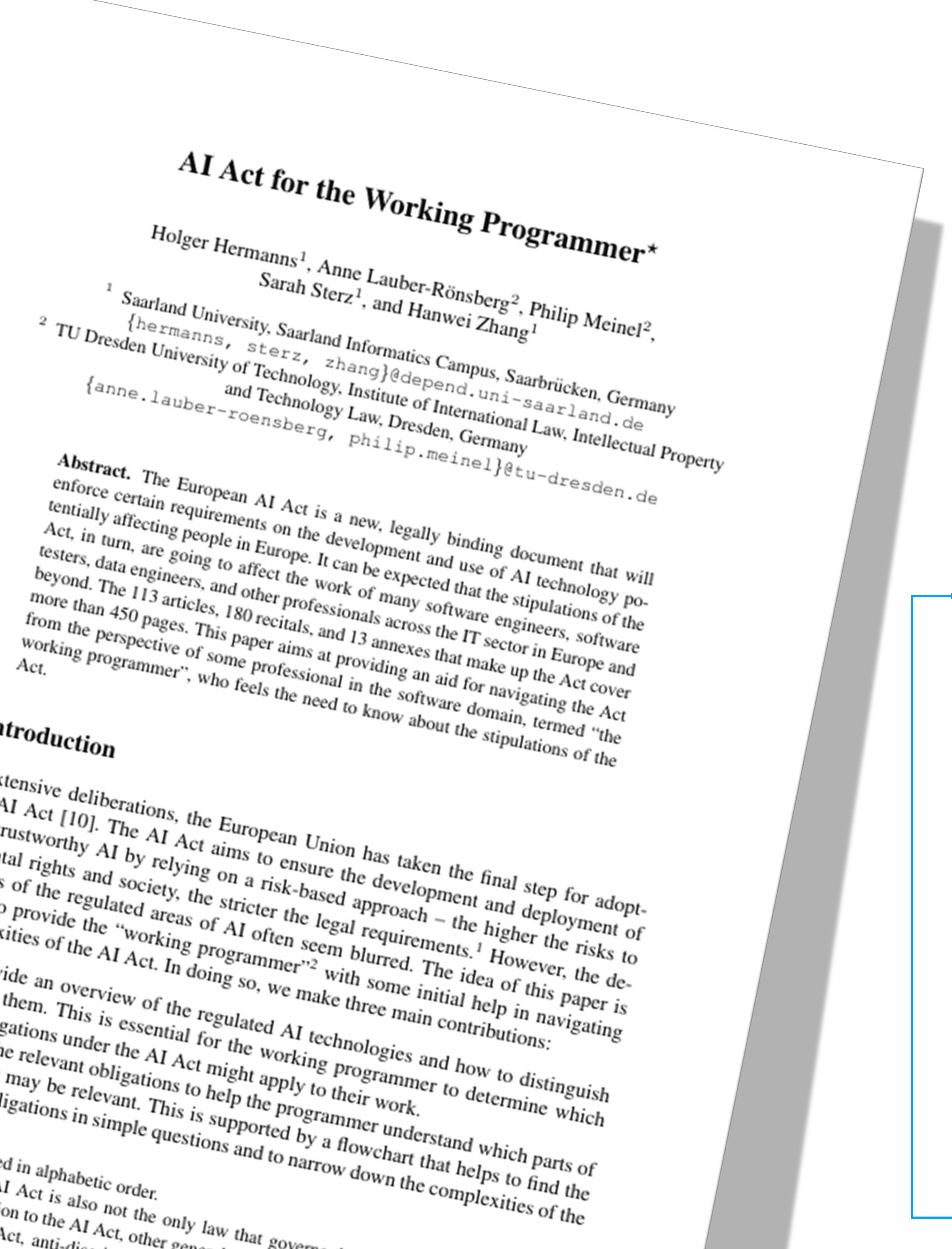
- A purely logic-based system that can infer whether the airbag inside some car has to ignite.

high risk

AI

- A system where machine learning from past accident characteristics has been used to infer how to decide whether the airbag inside some car has to ignite.

high risk



AI Act for the Working Programmer*

Holger Hermanns[1], Anne Lauber-Rönsberg[2], Philip Meinel[2], Sarah Sterz[1], and Hanwei Zhang[1]

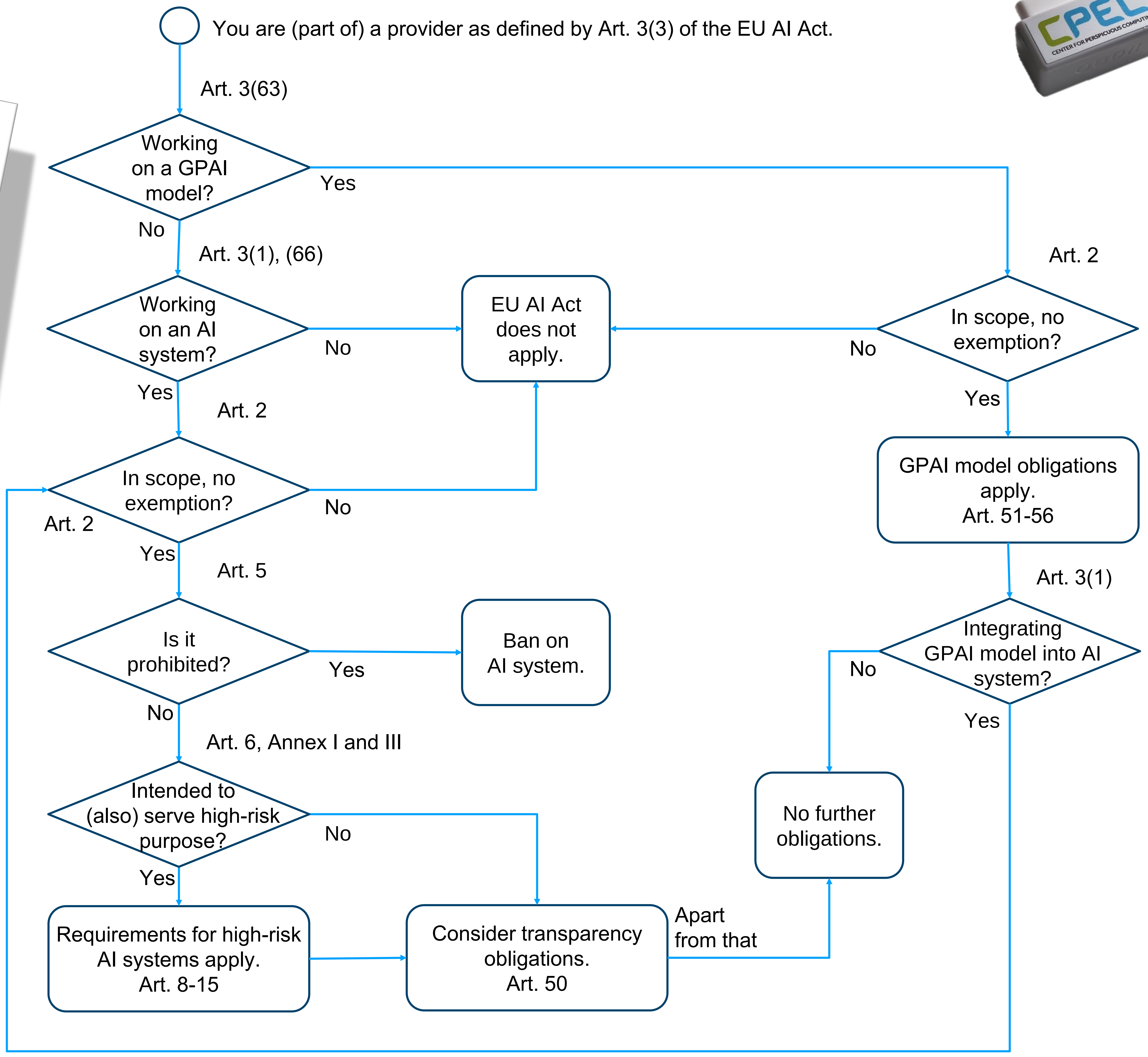
[1] Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
{hermanns, sterz, zhang}@depend.uni-saarland.de
[2] TU Dresden University of Technology, Institute of International Law, Intellectual Property and Technology Law, Dresden, Germany
{anne.lauber-roensberg, philip.meinel}@tu-dresden.de

Abstract. The European AI Act is a new, legally binding document that will enforce certain requirements on the development and use of AI technology potentially affecting people in Europe. It can be expected that the stipulations of the Act, in turn, are going to affect the work of many software engineers, software testers, data engineers, and other professionals across the IT sector in Europe and beyond. The 113 articles, 180 recitals, and 13 annexes that make up the Act cover more than 450 pages. This paper aims at providing an aid for navigating the Act from the perspective of some professional in the software domain, termed "the working programmer", who feels the need to know about the stipulations of the Act.

...troduction

...tensive deliberations, the European Union has taken the final step for adopt-
...AI Act [10]. The AI Act aims to ensure the development and deployment of
...rustworthy AI by relying on a risk-based approach – the higher the risks to
...tal rights and society, the stricter the legal requirements.[1] However, the de-
...s of the regulated areas of AI often seem blurred. The idea of this paper is
...o provide the "working programmer"[2] with some initial help in navigating
...xities of the AI Act. In doing so, we make three main contributions:
...ide an overview of the regulated AI technologies and how to distinguish
...them. This is essential for the working programmer to determine which
...gations under the AI Act might apply to their work.
...e relevant obligations to help the programmer understand which parts of
... may be relevant. This is supported by a flowchart that helps to find the
...ligations in simple questions and to narrow down the complexities of the

...d in alphabetic order.
...I Act is also not the only law that govern...
...on to the AI Act, other gen...
...Act, anti-di...



AI Act for the Working Programmer: High Risk

STEP 1: Development of a high-risk AI system

→ **STEP 2:** Conformity assessment and compliance with AI requirements (some systems: notified body involved)

→ **STEP 3:** Registration of stand-alone AI systems in EU database

→ **STEP 4:** Signing of conformity declaration + CE marking

→ **STEP 5:** Substantial changes? Back to STEP 2

“Risks for health, safety and fundamental rights of persons.“

AI Act for the Working Programmer

STEP 2:

Conformity assessment and compliance with AI requirements

(some systems: notified body involved)

Art 9: Risk management

Art 10: Data and data governance

Art 11: Technical documentation

Art 12: Record keeping

Art 13: Transparency and provision of information to users

Art 14: Human oversight

Art 15: Accuracy, robustness and cybersecurity

Human Oversight: Article 14

For the purpose of implementing paragraphs 1, 2 and 3, the high-risk AI system shall be provided to the deployer in such a way that natural persons to whom human oversight is assigned are enabled, as appropriate and proportionate:

(a) to properly understand the relevant capacities and limitations of the high-risk AI system and be able to duly monitor its operation, including in view of detecting and addressing anomalies, dysfunctions and unexpected performance;

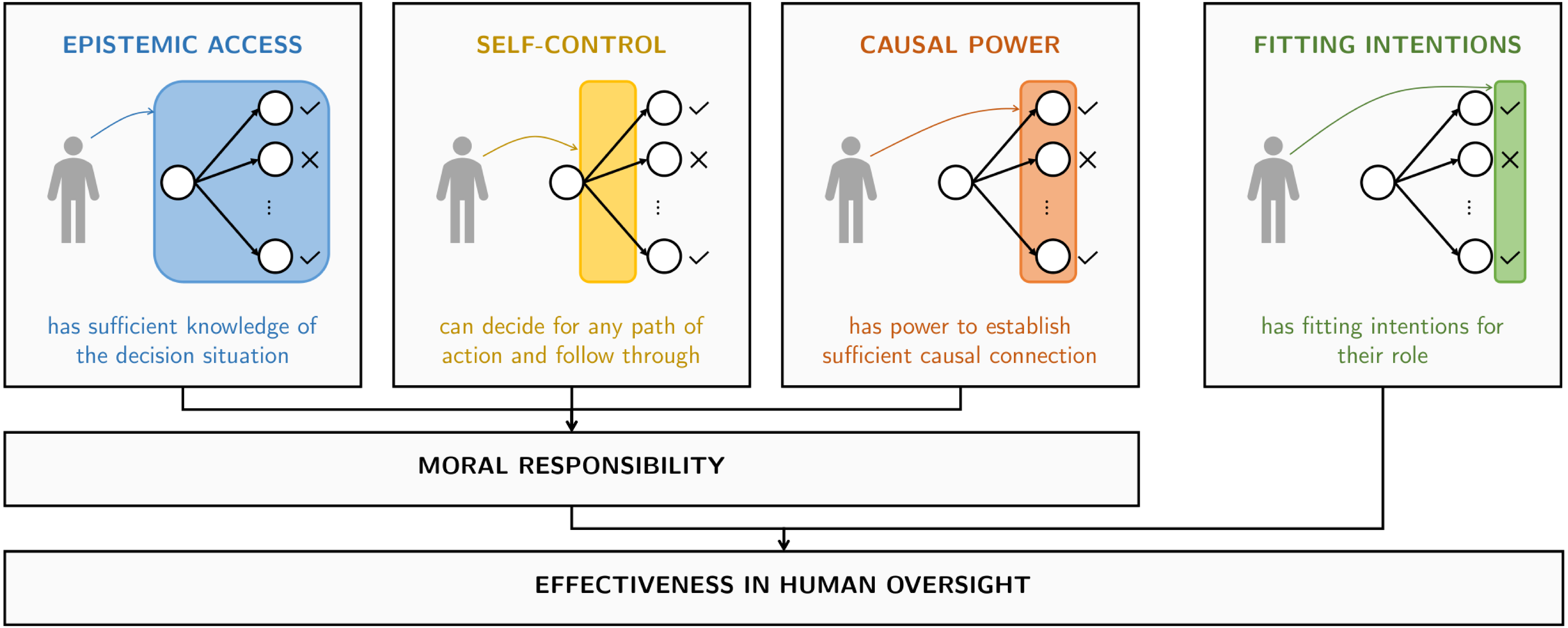
(b) to remain aware of the possible tendency of automatically relying or over-relying on the output produced by a high-risk AI system (automation bias), in particular for high-risk AI systems used to provide information or recommendations for decisions to be taken by natural persons;

(c) to correctly interpret the high-risk AI system's output, taking into account, for example, the interpretation tools and methods available;

(d) to decide, in any particular situation, not to use the high-risk AI system or to otherwise disregard, override or reverse the output of the high-risk AI system;

(e) to intervene in the operation of the high-risk AI system or interrupt the system through a 'stop' button or a similar procedure that allows the system to come to a halt in a safe state.

Effective Human Oversight



Article 14: Human Oversight

The high-risk AI system shall be designed in such a way that the OP is enabled to…

a) Understand the limits and capacities of the system and duly monitor its operation

b) „Remain aware“ of the automation bias

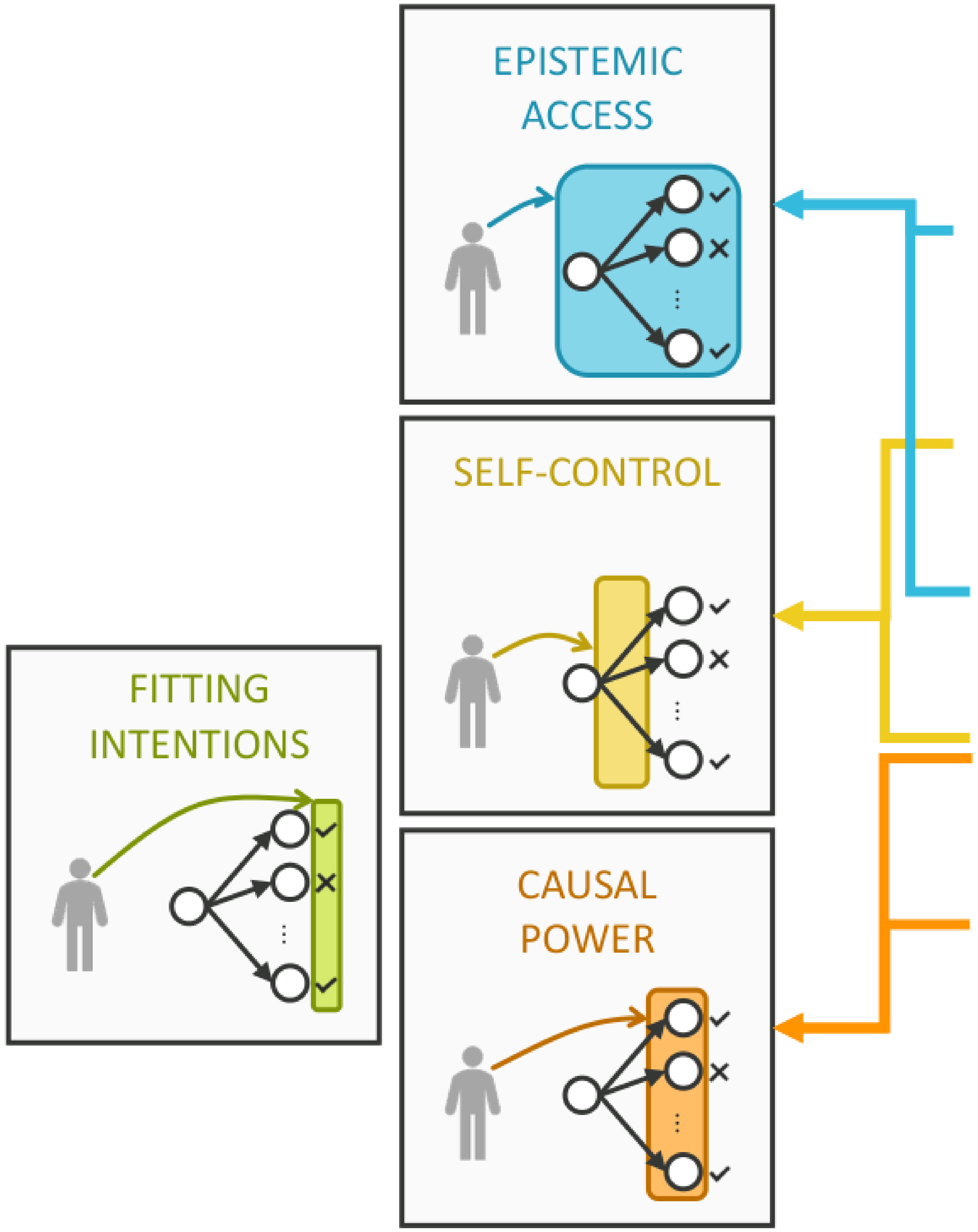
c) Correctly interpret the system‘s output

d) Decide not to use the system or disregard, override or reverse its decisions

e) Intervene or interrupt the system, e.g. through a „stop“ button

As appropriate and proportionate to the circumstances…

Article 14: Human Oversight



The high-risk AI system shall be designed in such a way that the OP is enabled to…

a) Understand the limits and capacities of the system and duly monitor its operation

b) „Remain aware“ of the automation bias

c) Correctly interpret the system‘s output

d) Decide not to use the system or disregard, override or reverse its decisions

e) Intervene or interrupt the system, e.g. through a „stop“ button

As appropriate and proportionate to the circumstances…

Facilitators and Inhibitors of Effectiveness

● ○

	intervention options	system adaptability	system understandability	interpretability of in- and outputs	preselection of outputs to review	overseer training	domain expertise	conscientiousness	exhaustion	motivation	automation bias	adequate job design	role conflicts	independent thinking	accountability	time pressure
	technical design					individual factors						environment				
causal power	●	●				●	●									○
epistemic access		●	●	●	●	●	●	●	○	●	○	●		●	●	○
self-control						●		●	○	●	○	●			●	
fitting intentions						●		●	○	●	○	●	○		●/○	

Facilitators and Inhibitors of Effectiveness

● ○

	technical design						individual factors					environment				
	intervention options	system adaptability	system understandability	interpretability of in- and outputs	preselection of outputs to review	overseer training	domain expertise	conscientiousness	exhaustion	motivation	automation bias	adequate job design	role conflicts	independent thinking	accountability	time pressure
causal power	●	●				●	●									○
epistemic access		●	●	●	●	●	●	●	○	●	○	●		●	●	○
self-control						●		●	○	●	○	●			●	
fitting intentions						●		●	○	●	○	●	○		●/○	

Technical Aspects of Effectiveness

FACTORS

- intervention options
- system adaptability
- system understandability
- interpretability of in- and outputs
- preselection of outputs to review
- overseer training

DESIGN CHOICES

- peripherals
- XAI and interpretability
- simulators
- runtime monitors
- model choice
- …
- methods for taking over manual control
- parameter tuning
- model properties/cards

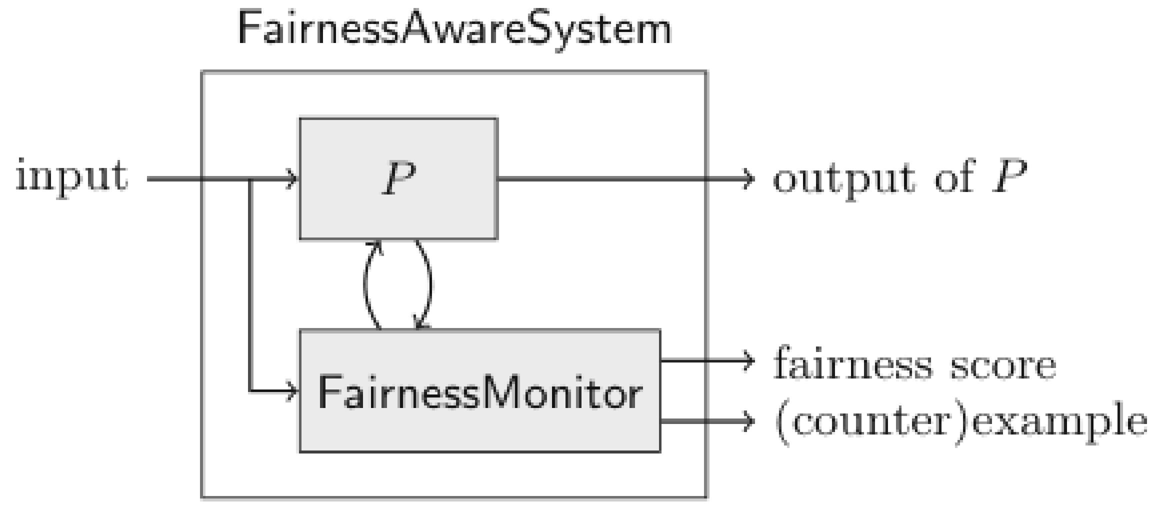
Technical Aspects of Effectiveness

FACTORS

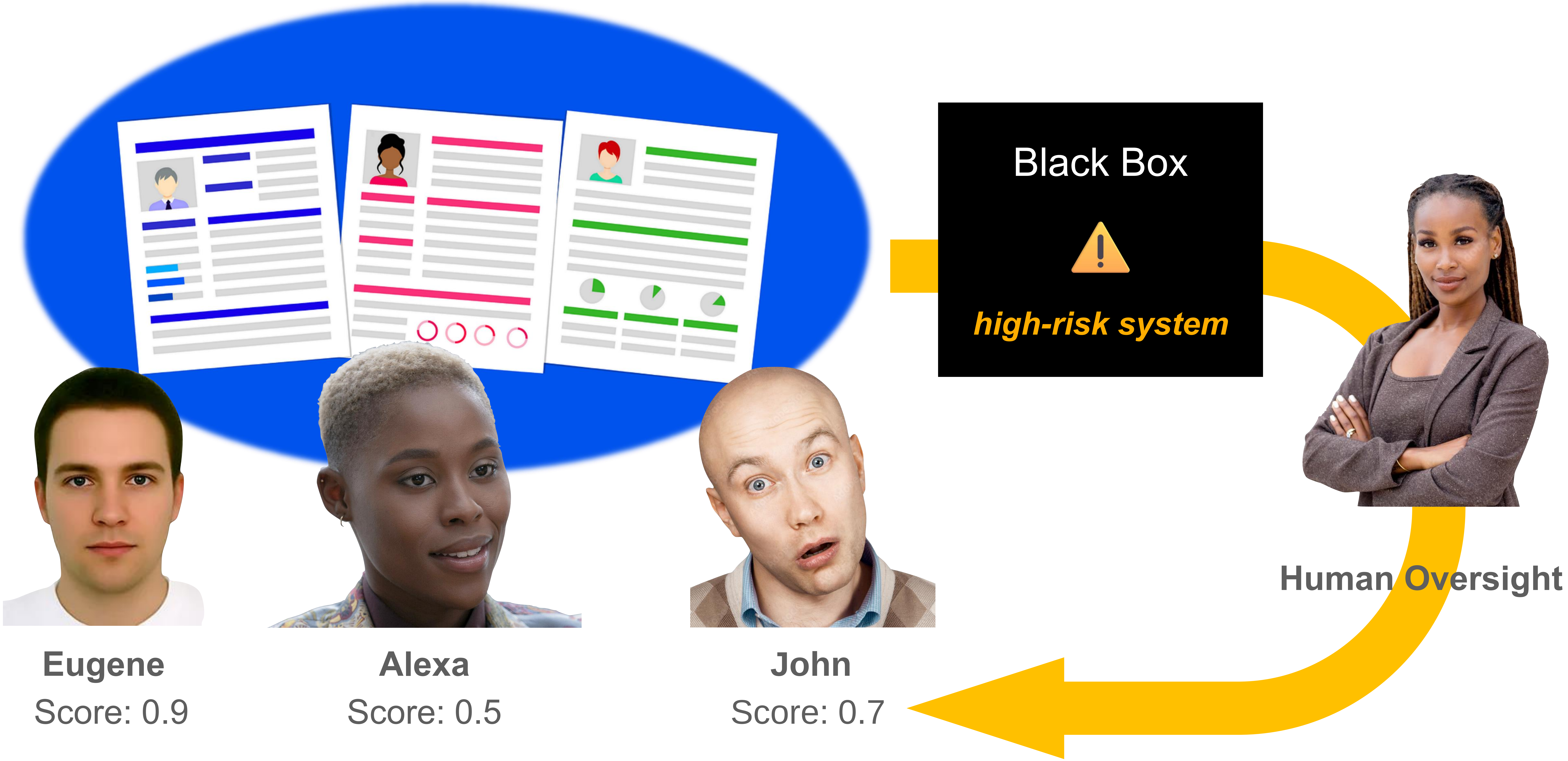
- intervention options
- system adaptability
- system understandability
- interpretability of in- and outputs
- preselection of outputs to review
- overseer training

DESIGN CHOICES

- peripherals
- XAI and interpretability
- simulators
- **runtime monitors**
- model choice
- ...
- methods for taking over manual control
- parameter tuning
- model properties/cards



Example – Individual Fairness



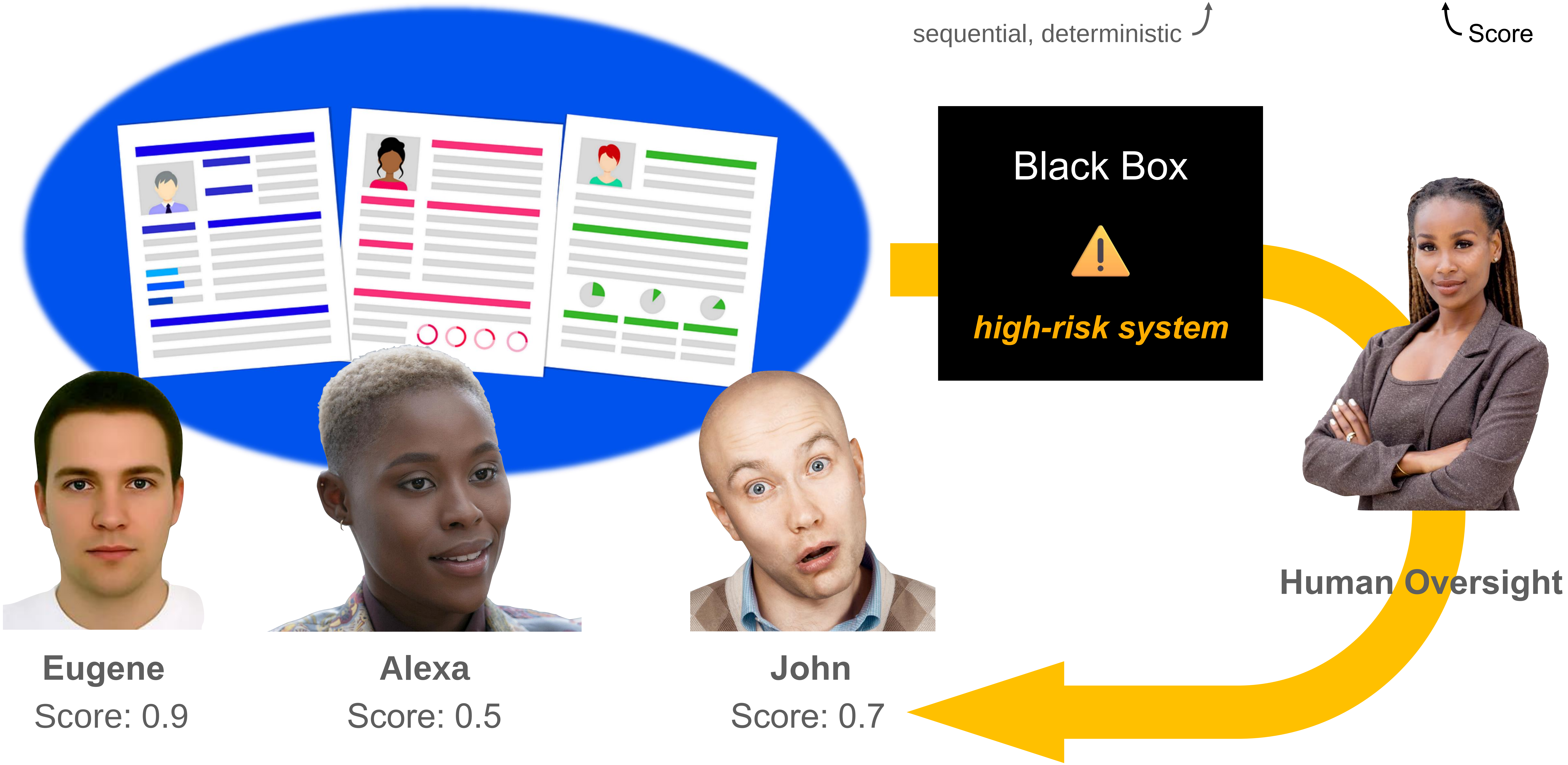
Example – Individual Fairness

Data about a human

$$P : \mathsf{In} \to \mathsf{Out}$$

sequential, deterministic

Score



Robust Cleanness

$P : \mathsf{In} \to \mathsf{Out}$

sequential,
deterministic

distance function for inputs, $(\mathsf{In} \times \mathsf{In}) \to \overline{\mathbb{R}}_{\geq 0}$

distance function for outputs, $(\mathsf{Out} \times \mathsf{Out}) \to \overline{\mathbb{R}}_{\geq 0}$

Contract $\mathcal{C} = \langle \mathsf{StdIn}, d_{\mathsf{In}}, d_{\mathsf{Out}}, \kappa_i, \kappa_o \rangle$

standard inputs

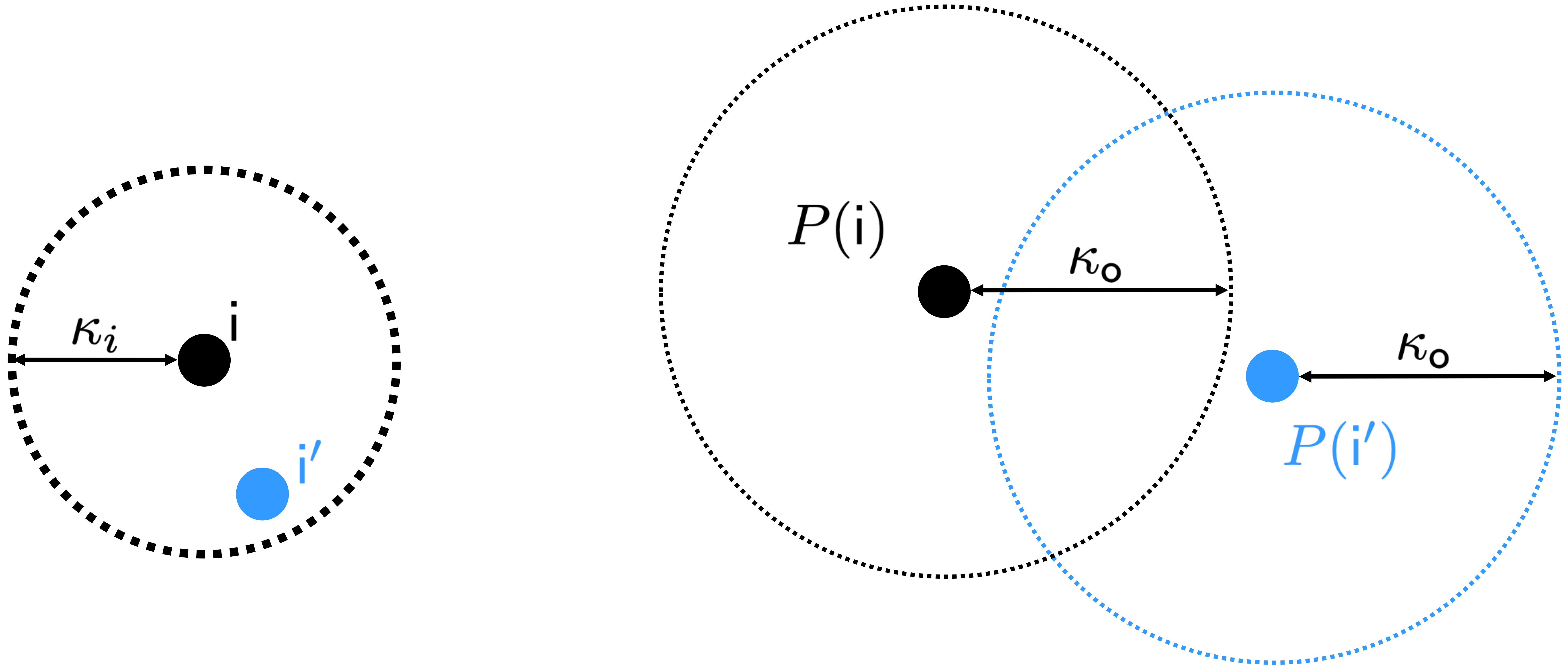
threshold for output distance

threshold for input distance

$\mathsf{StdIn} \subseteq \mathsf{In}$

$\mathsf{i} \in \mathsf{StdIn}$

$\mathsf{i}' \in \mathsf{In}$



For all $\mathsf{i} \in \mathsf{StdIn}$ and $\mathsf{i}' \in \mathsf{In}$. If $d_{\mathsf{In}}(\mathsf{i}, \mathsf{i}') \leq \kappa_i$, then $d_{\mathsf{Out}}(P(\mathsf{i}), P(\mathsf{i}')) \leq \kappa_o$.

Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R. Fairness through awareness. ITCS 2012.

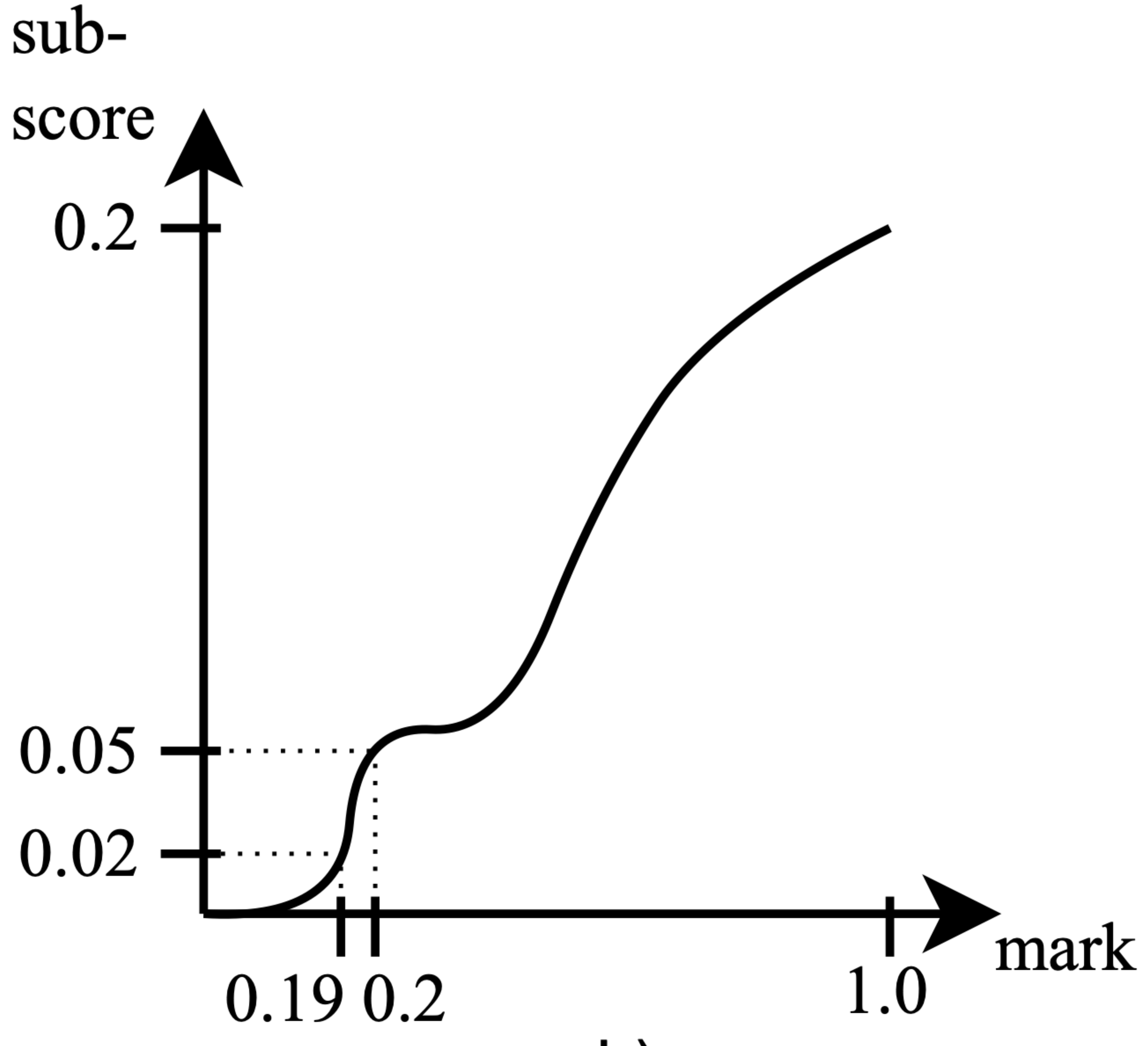
Baseline: Lipschitz-Fairness

$P : \mathsf{In} \to \mathsf{Out}$

sequential,
deterministic

For all $\mathsf{i}_1, \mathsf{i}_2 \in \mathsf{In},\ d_{\mathsf{Out}}(\mathsf{P}(\mathsf{i}_1), P(\mathsf{i}_2)) \leq L \cdot d_{\mathsf{In}}(\mathsf{i}_1, \mathsf{i}_2)$

- d_{In} and d_{Out} related by a constant *L*
- ranges over all input pairs
- monitorability is problematic



b)

Individual Fairness

$P : \mathsf{In} \to \mathsf{Out}$

sequential,
deterministic

$\mathcal{I} \subseteq \mathsf{In}$

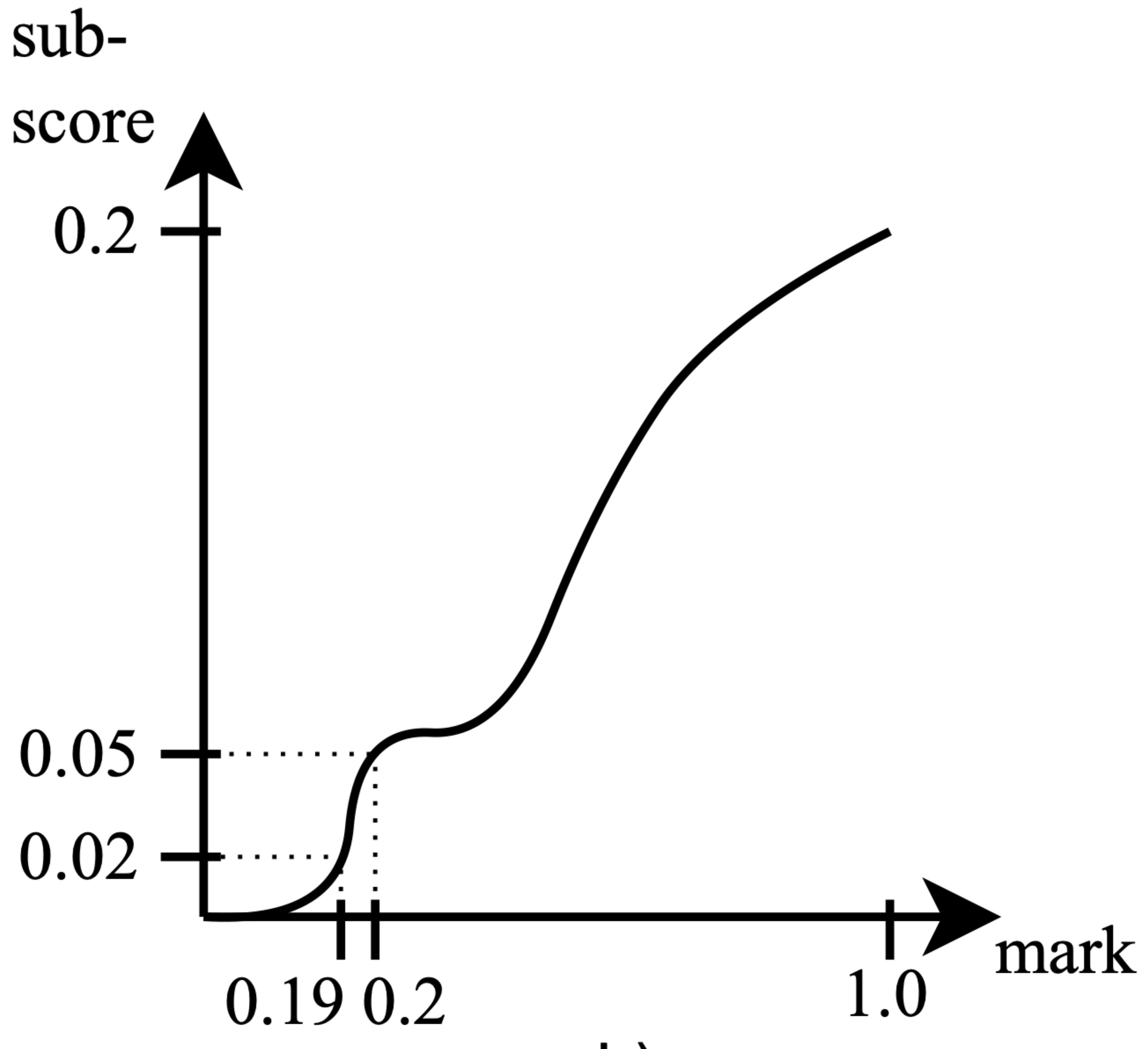
$\in \mathcal{I}$

For all $\mathsf{i}_1, \mathsf{i}_2 \in \mathsf{In},\ d_{\mathsf{Out}}(\mathsf{P}(\mathsf{i}_1), P(\mathsf{i}_2)) \leq \cancel{L \cdot d_{\mathsf{In}}(\mathsf{i}_1, \mathsf{i}_2)}$

$f(d_{\mathsf{In}}(\mathsf{i}, \mathsf{i}'))$

… assuming a Fairness Contract $\mathcal{F} = \langle d_{\mathsf{In}}, d_{\mathsf{Out}}, f \rangle$

- d_{In} and d_{Out} related by means of a function f



b)

Individual Fairness

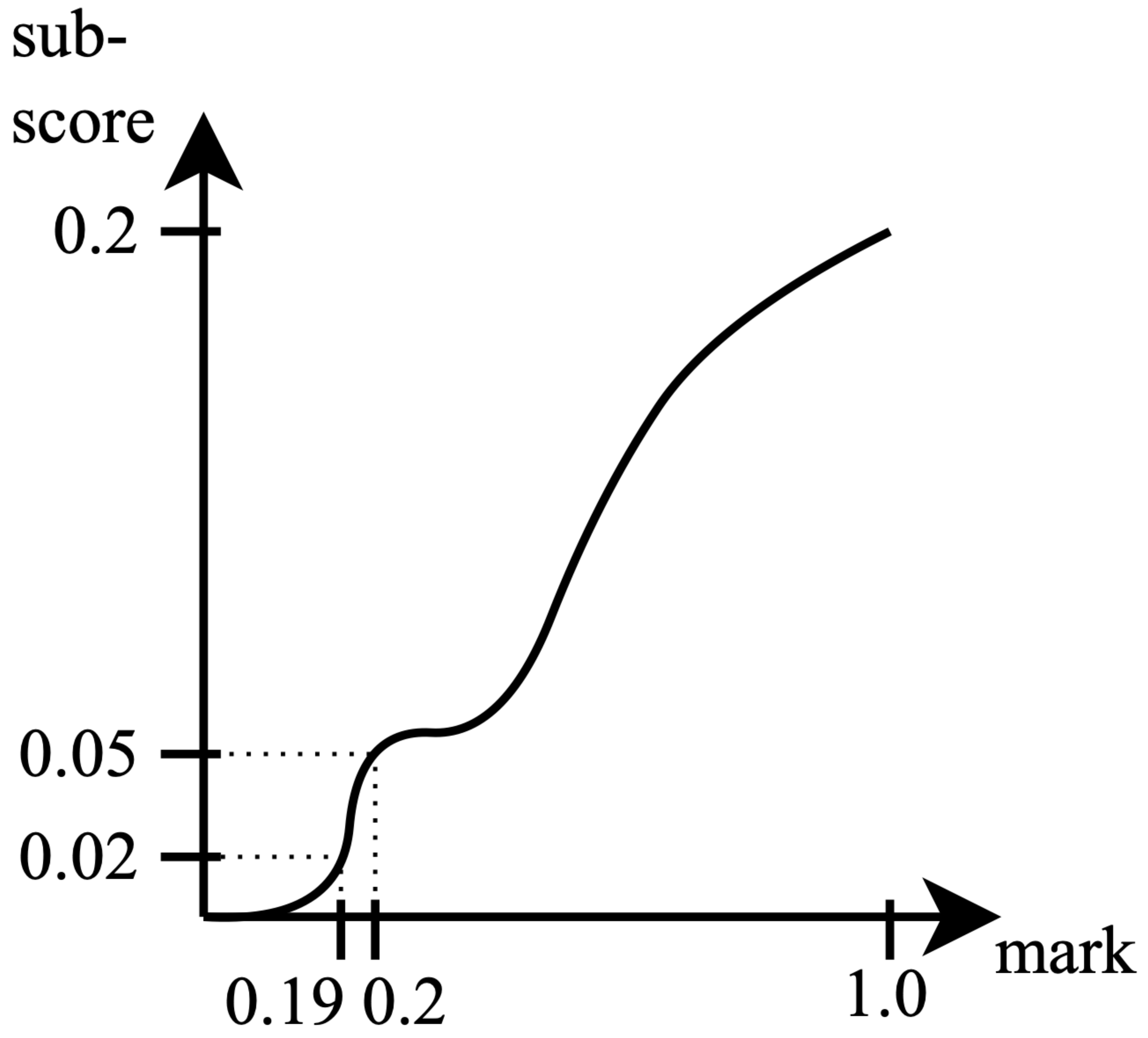
$P : \mathsf{In} \to \mathsf{Out}$

sequential,
deterministic

For all $\mathsf{i}_1 \in \mathcal{I}, \mathsf{i}_2 \in \mathsf{In}$, $d_{\mathsf{Out}}(\mathsf{P}(\mathsf{i}_1), P(\mathsf{i}_2)) \leq f(d_{\mathsf{In}}(\mathsf{i}, \mathsf{i}'))$

… assuming a Fairness Contract $\mathcal{F} = \langle d_{\mathsf{In}}, d_{\mathsf{Out}}, f \rangle$

- d_{In} and d_{Out} related by means of a function f
- distinction of actual vs. synthetic inputs
- monitorable, if $\mathcal{I}$ is finite



b)

Fairness Aware AI System

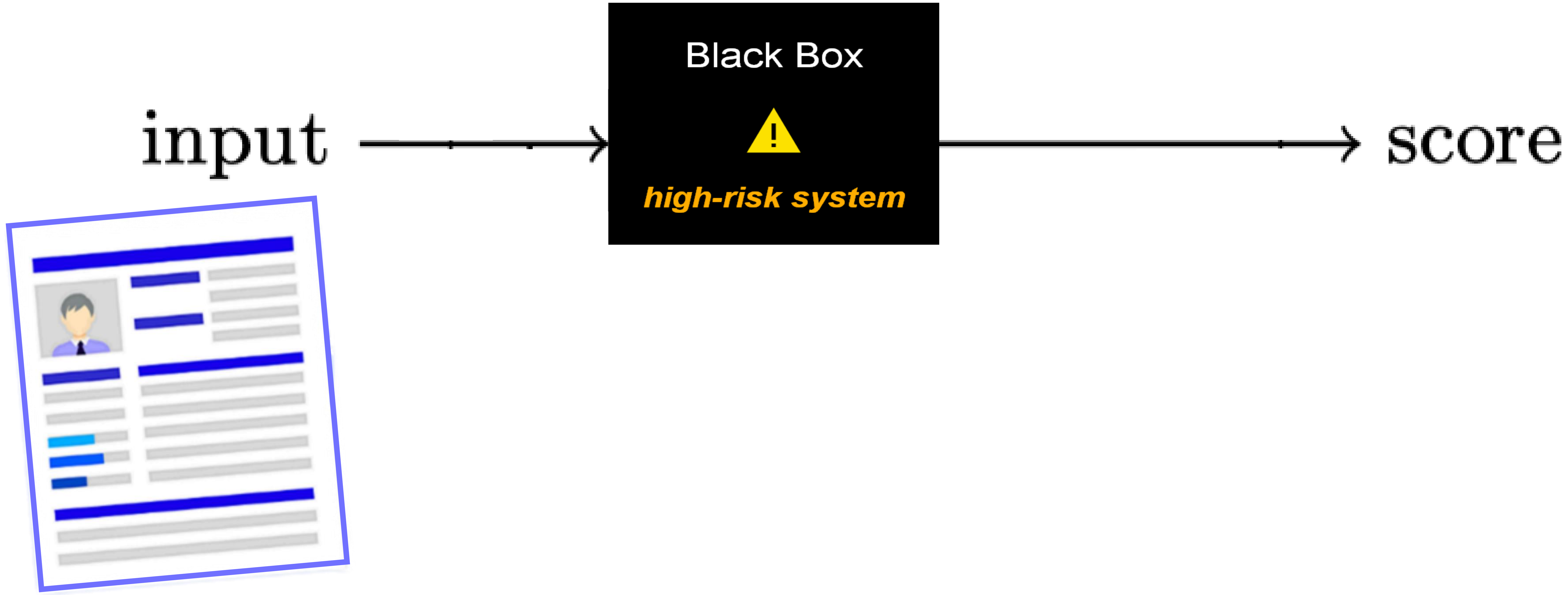
Data about a human

$$P : \mathsf{In} \rightarrow \mathsf{Out}$$

sequential, deterministic

Score

For all $\mathsf{i}_1 \in \mathcal{I}, \mathsf{i}_2 \in \mathsf{In}$, $d_{\mathsf{Out}}(\mathsf{P}(\mathsf{i}_1), P(\mathsf{i}_2)) \leq f(d_{\mathsf{In}}(\mathsf{i}, \mathsf{i}'))$



Fairness Aware AI System

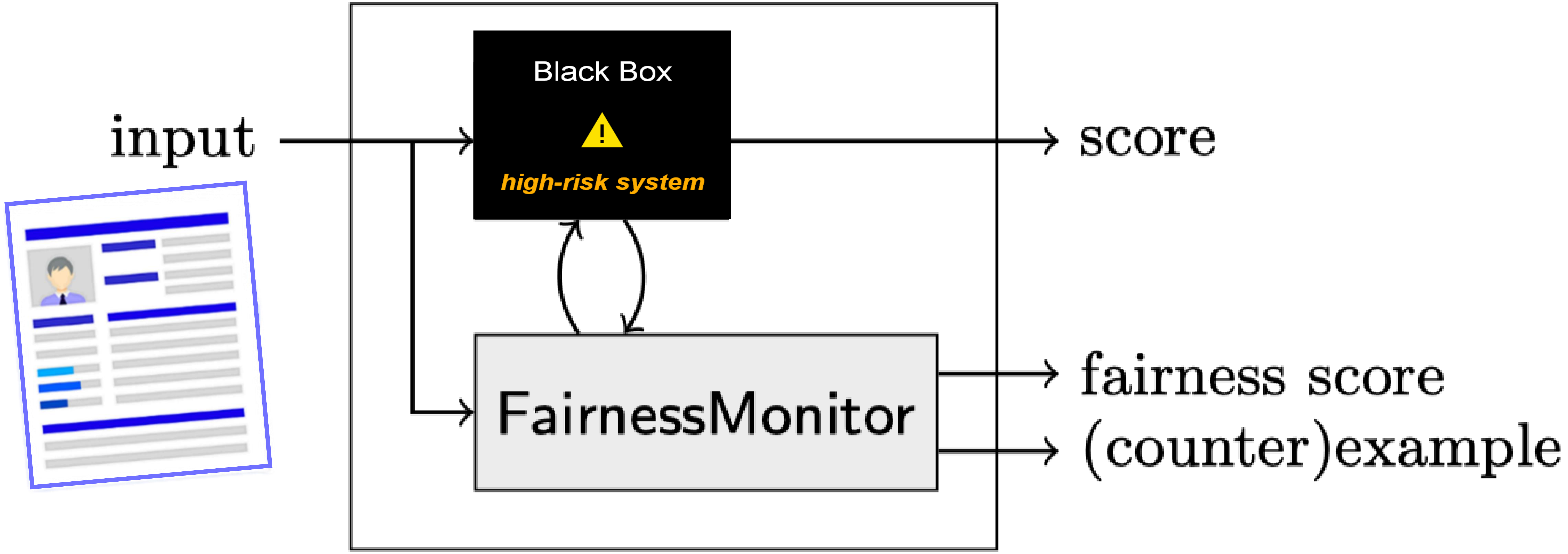
For all $\mathsf{i}_1 \in \mathcal{I}, \mathsf{i}_2 \in \mathsf{In}$, $d_{\mathsf{Out}}(\mathsf{P}(\mathsf{i}_1), P(\mathsf{i}_2)) \leq f(d_{\mathsf{In}}(\mathsf{i}, \mathsf{i}'))$

Data about a human

$$P : \mathsf{In} \to \mathsf{Out}$$

sequential,
deterministic

Score



Fairness Monitoring

For all $\mathsf{i}_1 \in \mathcal{I}, \mathsf{i}_2 \in \mathsf{In}$, $d_{\mathsf{Out}}(\mathsf{P}(\mathsf{i}_1), P(\mathsf{i}_2)) \leq f(d_{\mathsf{In}}(\mathsf{i}, \mathsf{i}'))$

Algorithm 2.1 Monte-Carlo falsification

Input: w: Initial trace, $\mathcal{R}$: Robustness function, PS: Proposal Scheme

Output: $w \in \mathsf{M}$

```
1: while R(w) > 0 do
2:     w' ← PS(w)
3:     α ← exp(−β(R(w') − R(w)))
4:     r ← UniformRandomReal(0, 1)
5:     if r ≤ α then
6:         w ← w'
7:     end if
8: end while
```

Fairness score – Robustness estimate

$$F(\mathsf{i_a}, \mathsf{i_s}) := f(d_{\mathsf{In}}(\mathsf{i_a}, \mathsf{i_s})) - d_{\mathsf{Out}}(\mathsf{P}(\mathsf{i_a}), \mathsf{P}(\mathsf{i_s}))$$

$$F(\mathcal{I}, \mathsf{i_s}) := \min\{F(\mathsf{i_a}, \mathsf{i_s}) \mid \mathsf{i_a} \in \mathcal{I}\}$$

$$\mathcal{R}_{\mathcal{I}}(\mathsf{i_s}) := F(\mathcal{I}, \mathsf{i_s})$$

Fairness Monitoring

Algorithm 2 FairnessMonitor,
with ξ-min $S = (\xi, \mathsf{i}_1, \mathsf{i}_2)$ only if $(\xi, \mathsf{i}_1, \mathsf{i}_2) \in S$ and for all $(\xi', \mathsf{i}_1', \mathsf{i}_2') \in S$, $\xi' \geq \xi$

Falsification Parameters: PS: Proposal scheme, β: Temperature parameter
Input: System $P : \mathsf{In} \rightarrow \mathsf{Out}$, Fairness contract $\mathcal{F} = \langle d_{\mathsf{In}}, d_{\mathsf{Out}}, f \rangle$, and set of actual inputs $\mathcal{I}$
Output: A minimal fairness score triple from $\mathbb{R} \times \mathcal{I} \times \mathsf{In}$.

```
i_s ← any input i_a ∈ I
(ξ, i_min, i_s) ← ξ-min{(F(i_a, i_s), i_a, i_s) | i_a ∈ I}
(ξ_min, i_1, i_2) ← (ξ, i_min, i_s)
while not timeout do
    i'_s ← PS(i_s, P(i_s))
    (ξ', i'_min, i'_s) ← ξ-min{(F(i_a, i'_s), i_a, i'_s) | i_a ∈ I}
    (ξ_min, i_1, i_2) ← ξ-min{(ξ_min, i_1, i_2), (ξ', i'_min, i'_s)}
    α ← exp(−β(ξ' − ξ))
    r ← UniformRandomReal(0, 1)
    if r ≤ α then
        i_s ← i'_s
        ξ ← ξ'
    end if
end while
return (ξ_min, i_1, i_2)
```

Algorithm 2.1 Monte-Carlo falsification
Input: w: Initial trace, $\mathcal{R}$: Robustness function, PS: Proposal Scheme
Output: $w \in \mathsf{M}$

```
while R(w) > 0 do
    w' ← PS(w)
    α ← exp(−β(R(w') − R(w)))
    r ← UniformRandomReal(0, 1)
    if r ≤ α then
        w ← w'
    end if
end while
```

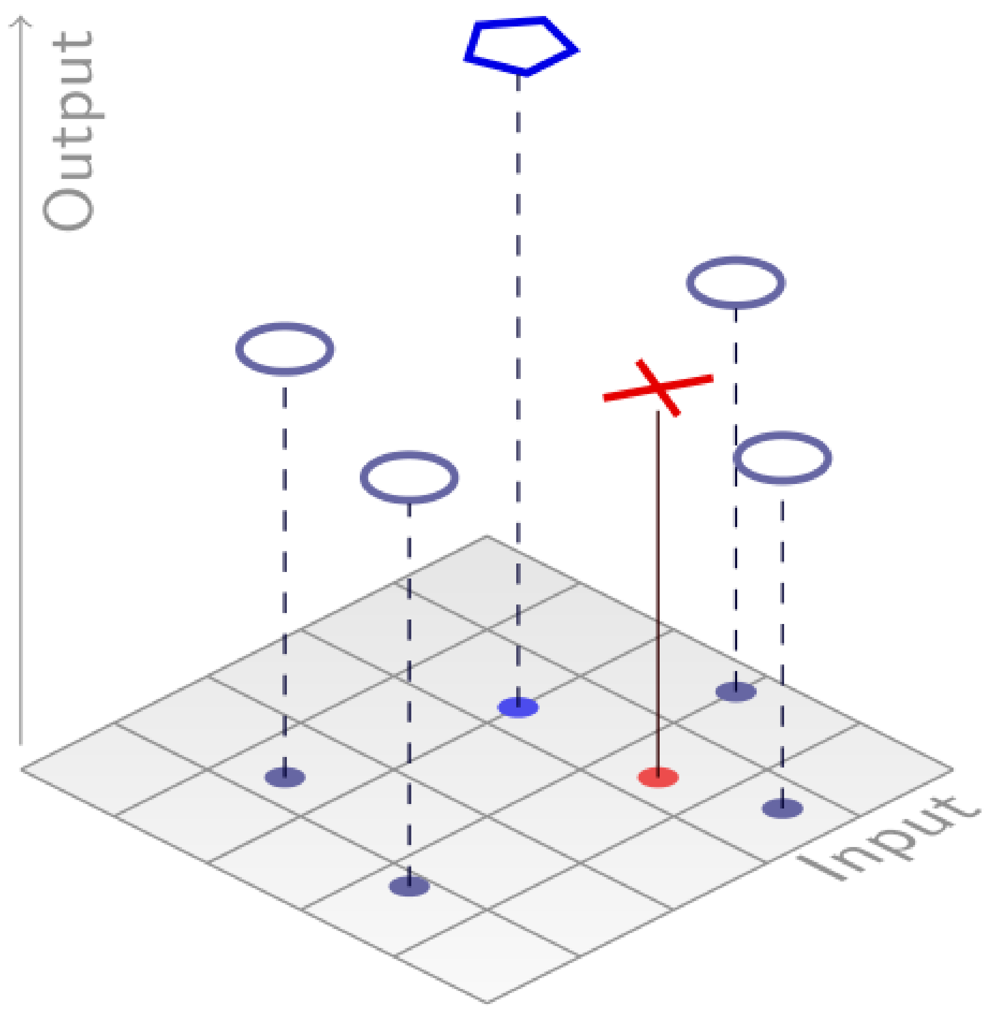
Fairness score – Robustness estimate

$$F(\mathsf{i}_a, \mathsf{i}_s) := f(d_{\mathsf{In}}(\mathsf{i}_a, \mathsf{i}_s)) - d_{\mathsf{Out}}(\mathsf{P}(\mathsf{i}_a), \mathsf{P}(\mathsf{i}_s))$$

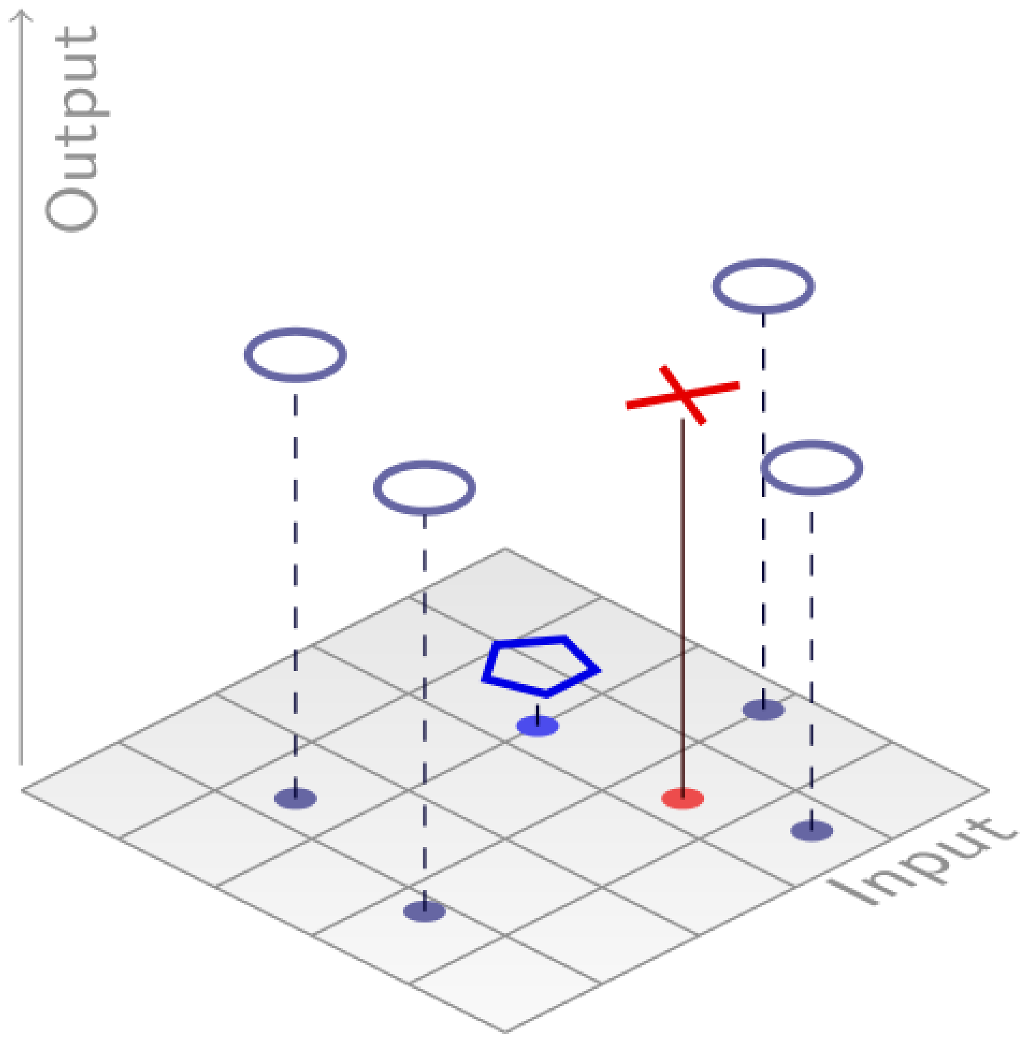
$$F(\mathcal{I}, \mathsf{i}_s) := \min\{F(\mathsf{i}_a, \mathsf{i}_s) \mid \mathsf{i}_a \in \mathcal{I}\}$$

$$\mathcal{R}_{\mathcal{I}}(\mathsf{i}_s) := F(\mathcal{I}, \mathsf{i}_s)$$

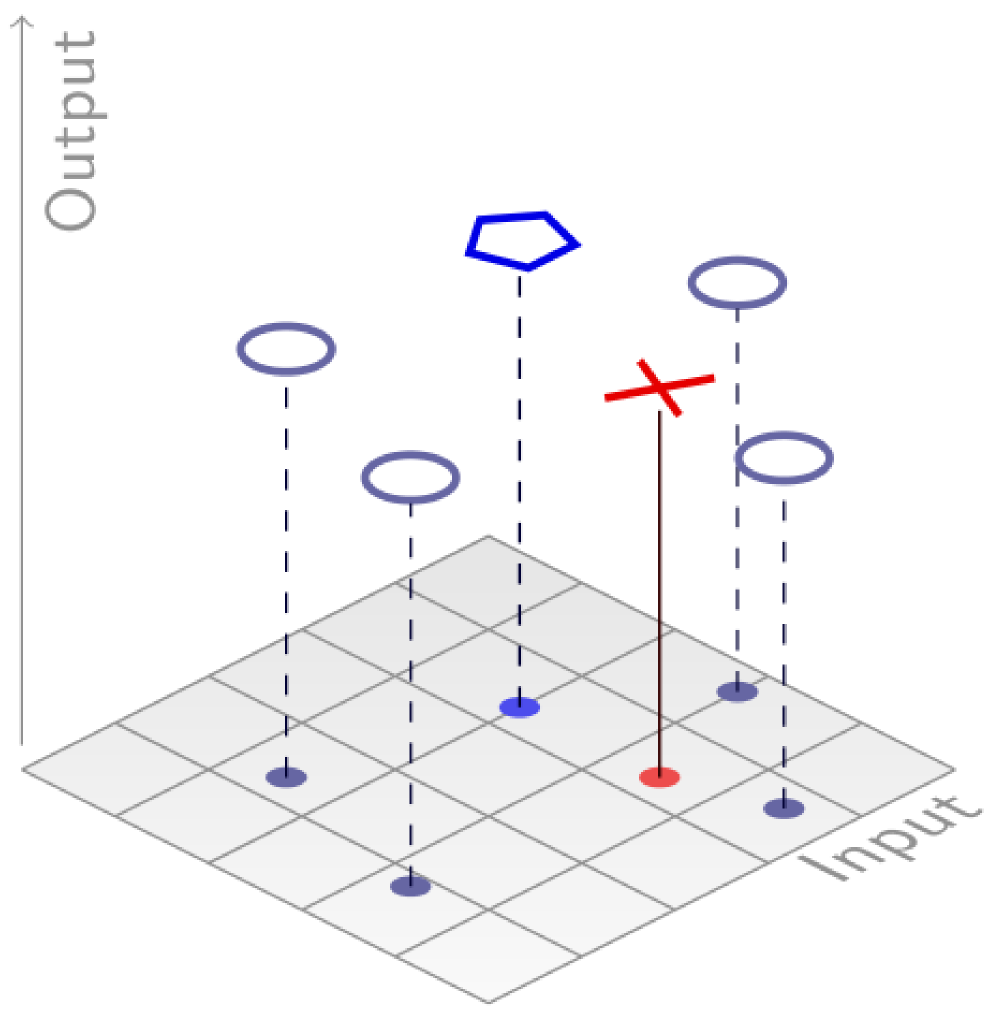
Cases of Unfairness



Individual scores worse than synthetic counterpart.



Individual scores better than synthetic counterpart.



No unfairness detected.

In Practice

Eugene

Score: 0.9

Alexa

Rainbow University

Score: 0.5

John

Poor Grades

Trump University

Score: 0.7

The score should be greater than 0.5...

Very similar to Eugene

Score: 0.75

Snow University

Score: 0.6

Same Poor Grades

Saarland University

Score: 0.4

Formal Methods in System Design
https://doi.org/10.1007/s10703-024-00445-2

Check for updates

Software doping analysis for human oversight

Sebastian Biewer[1] · Kevin Baum[1,2,3] · Sarah Sterz[1] · Holger Hermanns[1] · Sven Hetmank[4] · Markus Langer[5] · Anne Lauber-Rönsberg[4] · Franz Lehr[4]

Received: 22 December 2022 / Accepted: 11 January 2024

Abstract
This article introduces a frame
ware can pose. Concretely, th
and discrimination in high-ris
software that contains surrepti
A prominent example of soft
were found in millions of cars
The first part of this article co
established probabilistic falsi
for identifying undesired effe
systems in diesel cars but also
or discriminating way. We de
make better informed and m
oversight, which will be a cer

https://doi.org/10.1145/3630106.3659051

On the Quest for Effectiveness in Human Oversight: Interdisciplinary Perspectives

Sarah Sterz
Dependable Systems and Software,
Saarland University
Saarland Informatics Campus,
Saarbrücken, Germany
sterz@depend.uni-saarland.de

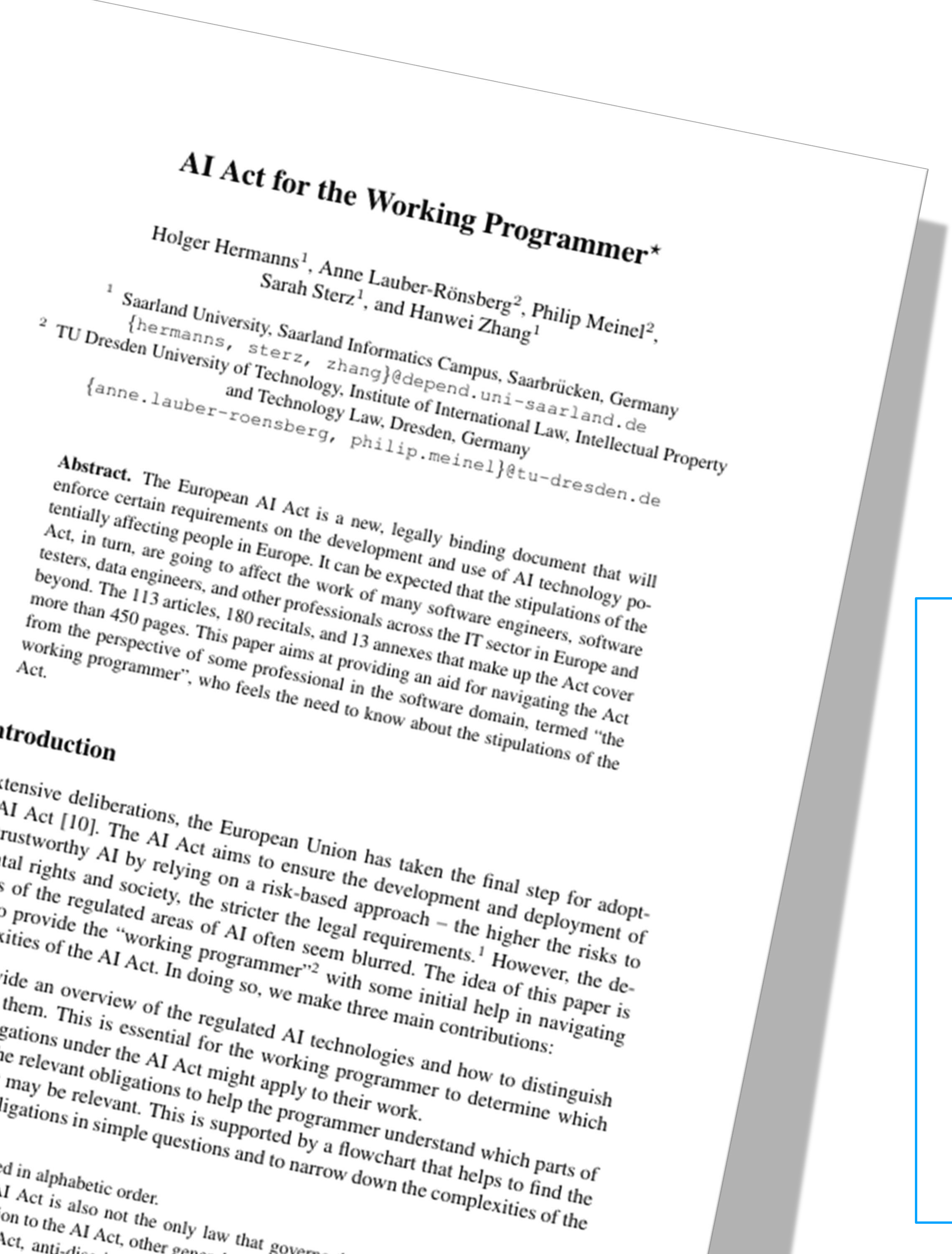
Kevin Baum
Neuro-Mechanistic Modeling,
German Research Center for Artificial
Intelligence (DFKI)
Saarbrücken, Germany
kevin.baum@dfki.de

Sebastian Biewer
Dependable Systems and Software,
Saarland University
Saarland Informatics Campus,
Saarbrücken, Germany
biewer@depend.uni-saarland.de

Holger Hermanns
Dependable Systems and Software,
Saarland University
Saarland Informatics Campus,
Saarbrücken, Germany
...@cs.uni-saarland.de

Anne Lauber-Rönsberg
IRGET, Faculty of Humanities and
Social Science, TU Dresden
Dresden, Germany
anne.lauber-roensberg@tu-dresden.de

Philip Meinel
IRGET, Faculty of Humanities and
Social Science, TU Dresden
Dresden, Germany
philip.meinel@tu-dresden.de



AI Act for the Working Programmer*

Holger Hermanns[1], Anne Lauber-Rönsberg[2], Philip Meinel[2], Sarah Sterz[1], and Hanwei Zhang[1]

[1] Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
{hermanns, sterz, zhang}@depend.uni-saarland.de
[2] TU Dresden University of Technology, Institute of International Law, Intellectual Property and Technology Law, Dresden, Germany
{anne.lauber-roensberg, philip.meinel}@tu-dresden.de

Abstract. The European AI Act is a new, legally binding document that will enforce certain requirements on the development and use of AI technology potentially affecting people in Europe. It can be expected that the stipulations of the Act, in turn, are going to affect the work of many software engineers, software testers, data engineers, and other professionals across the IT sector in Europe and beyond. The 113 articles, 180 recitals, and 13 annexes that make up the Act cover more than 450 pages. This paper aims at providing an aid for navigating the Act from the perspective of some professional in the software domain, termed "the working programmer", who feels the need to know about the stipulations of the Act.

ntroduction

tensive deliberations, the European Union has taken the final step for adopt-
AI Act [10]. The AI Act aims to ensure the development and deployment of
rustworthy AI by relying on a risk-based approach – the higher the risks to
tal rights and society, the stricter the legal requirements.[1] However, the de-
s of the regulated areas of AI often seem blurred. The idea of this paper is
o provide the "working programmer"[2] with some initial help in navigating
kities of the AI Act. In doing so, we make three main contributions:

ide an overview of the regulated AI technologies and how to distinguish
them. This is essential for the working programmer to determine which
gations under the AI Act might apply to their work.
he relevant obligations to help the programmer understand which parts of
may be relevant. This is supported by a flowchart that helps to find the
ligations in simple questions and to narrow down the complexities of the

ed in alphabetic order.
I Act is also not the only law that govern
on to the AI Act, other gen