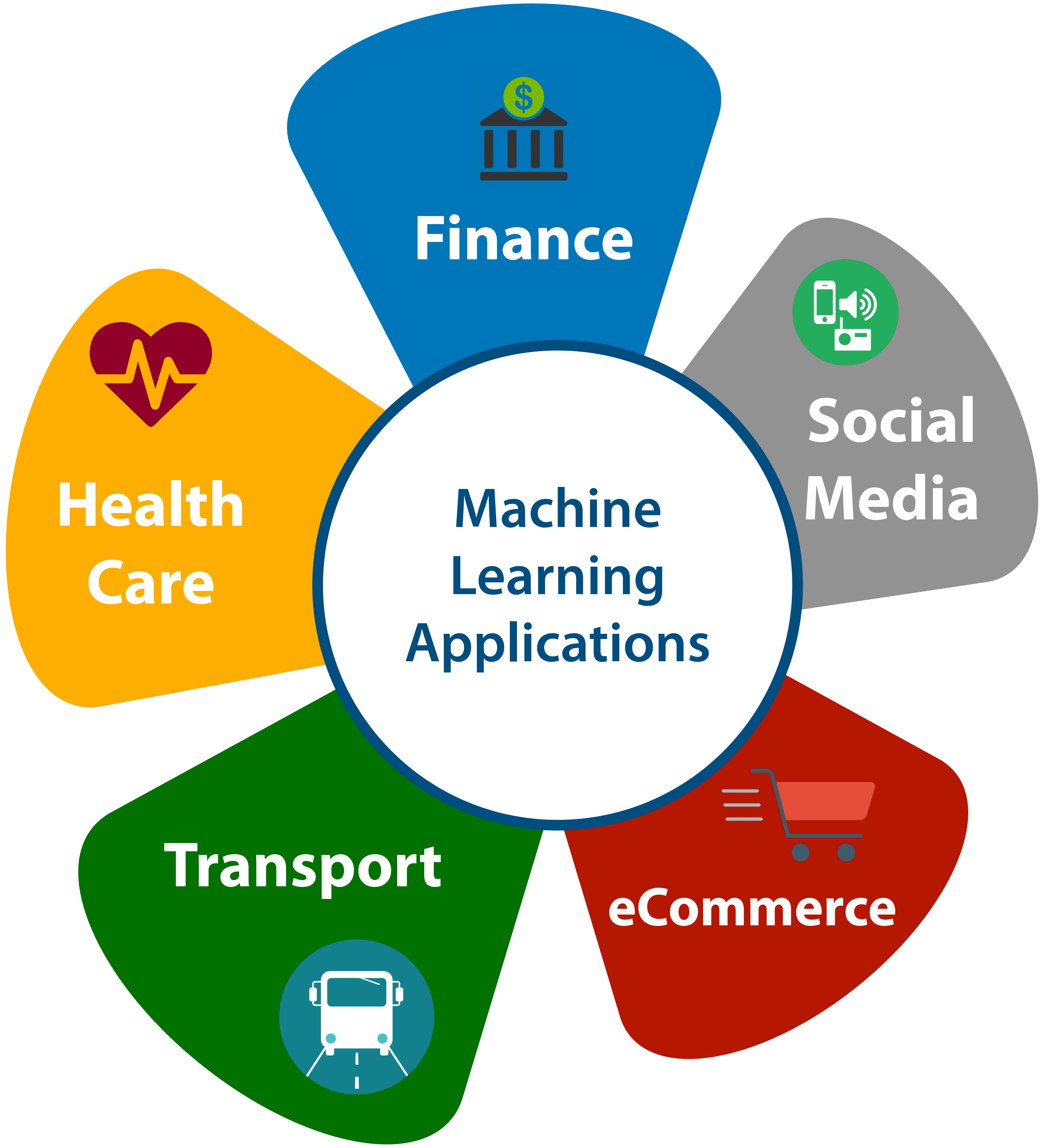
Synthesizing Pareto-Optimal Interpretations for Black-Box Models

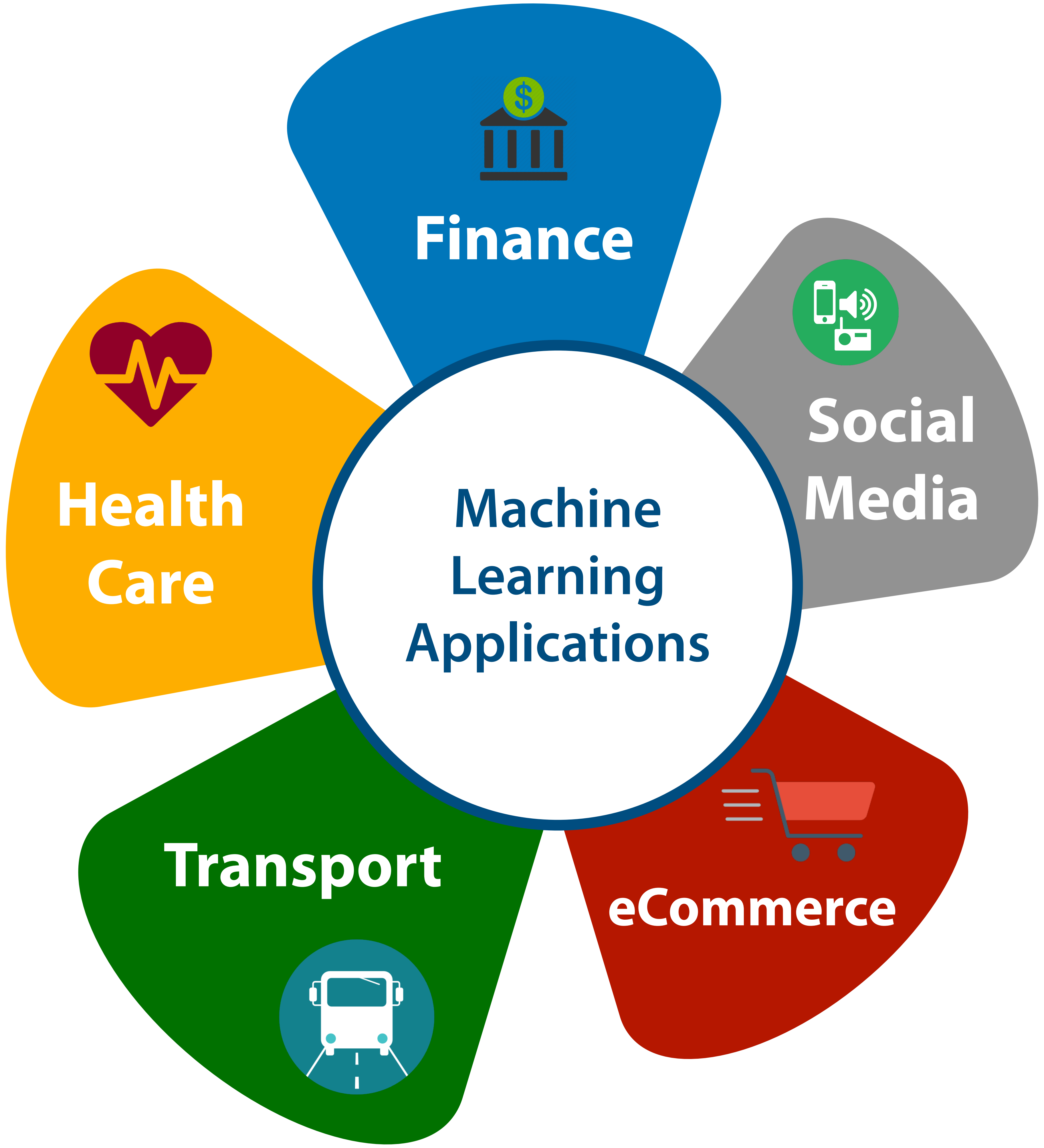
Hazem Torfah, Shetal Shah, Supratik Chakraborty, S. Akshay, Sanjit A. Seshia

@FMCAD 2021

Explaining Machine Learning Components

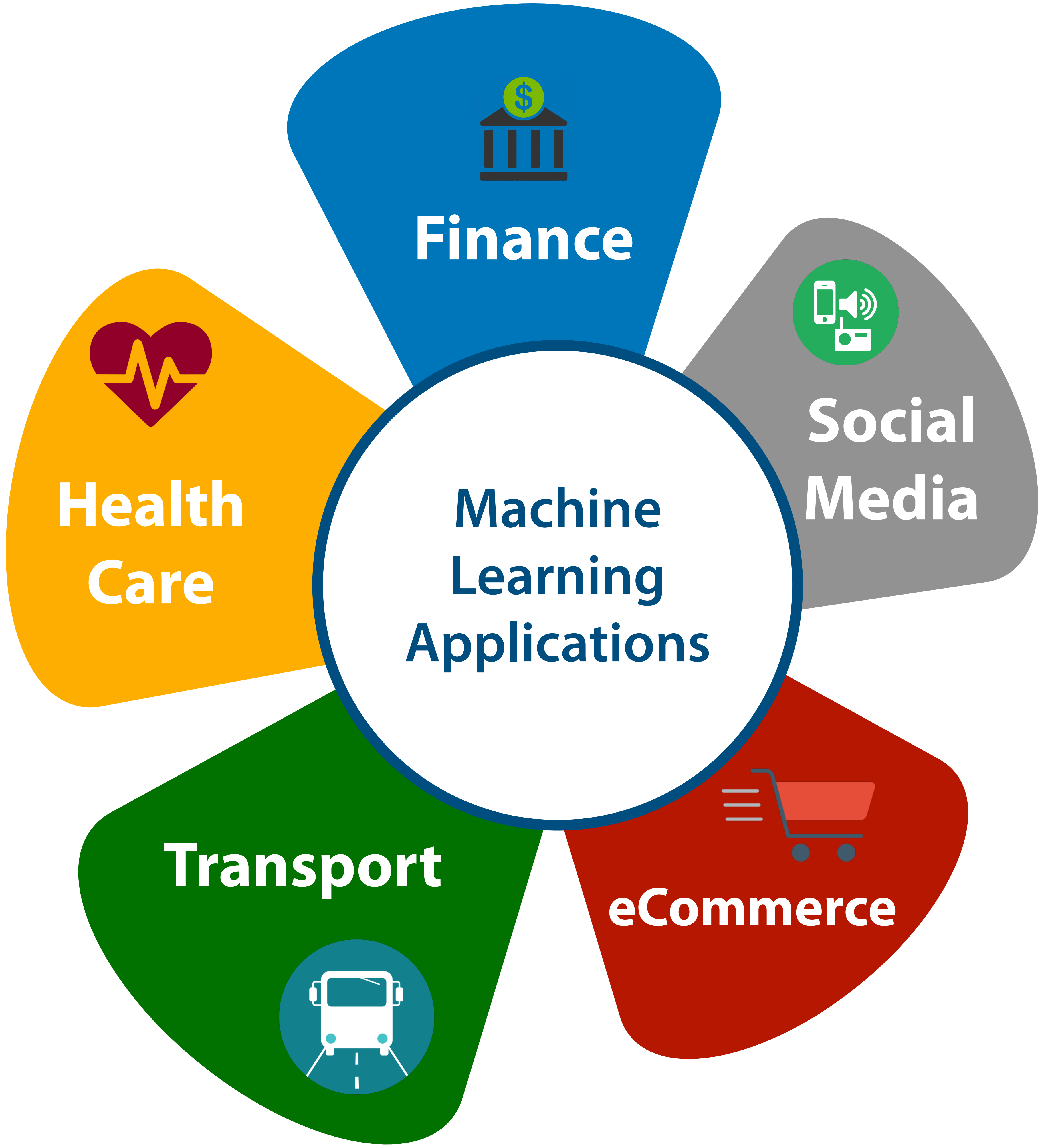


Explaining Machine Learning Components



Machine learning components like DNNs are ***complex*** models that are hard to comprehend

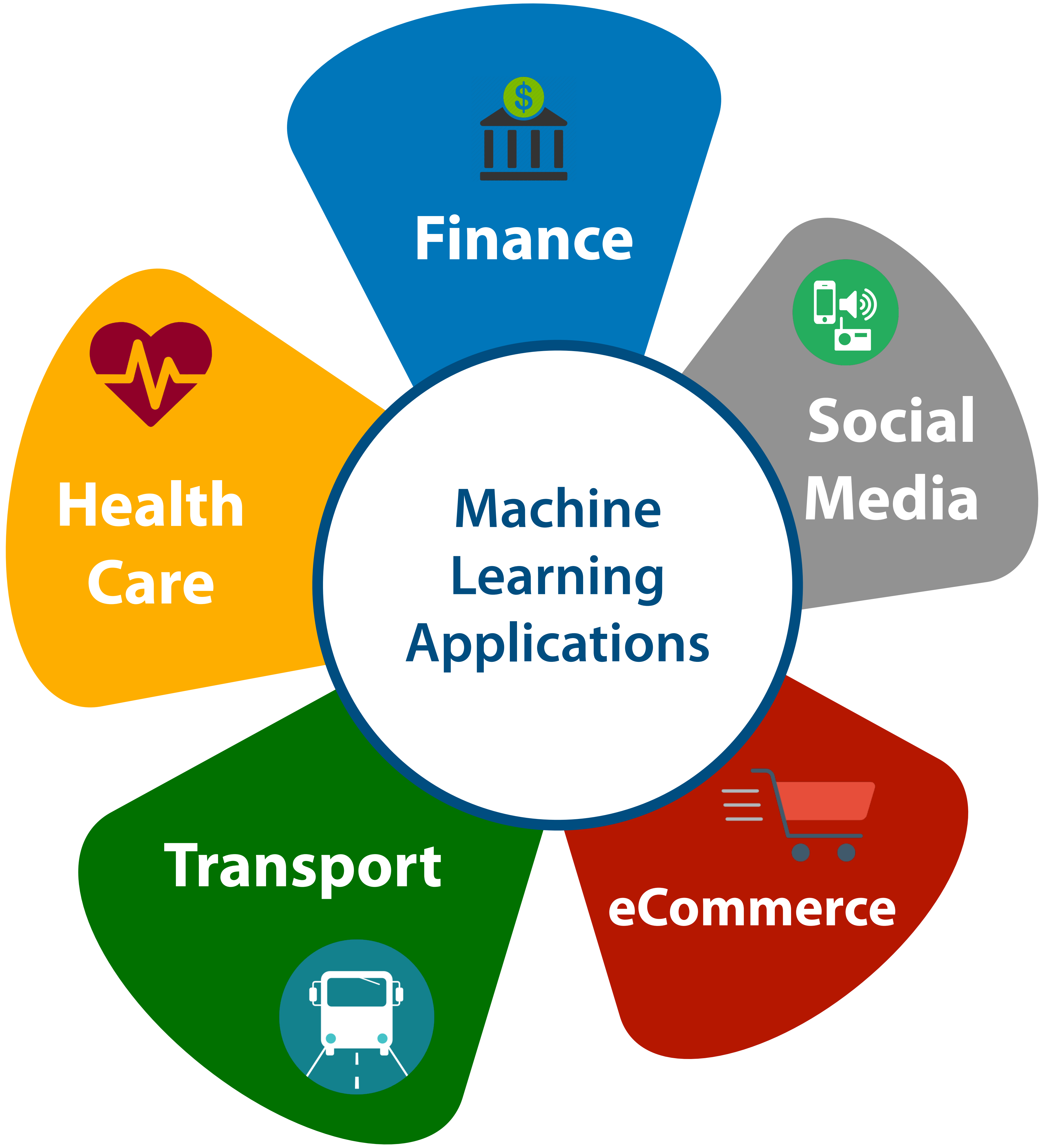
Explaining Machine Learning Components



Machine learning components like DNNs are ***complex*** models that are hard to comprehend

Explaining the behavior of ML components has become a necessity, especially with emerging laws and regulations (e.g. GDPR).

Explaining Machine Learning Components



Machine learning components like DNNs are ***complex*** models that are hard to comprehend

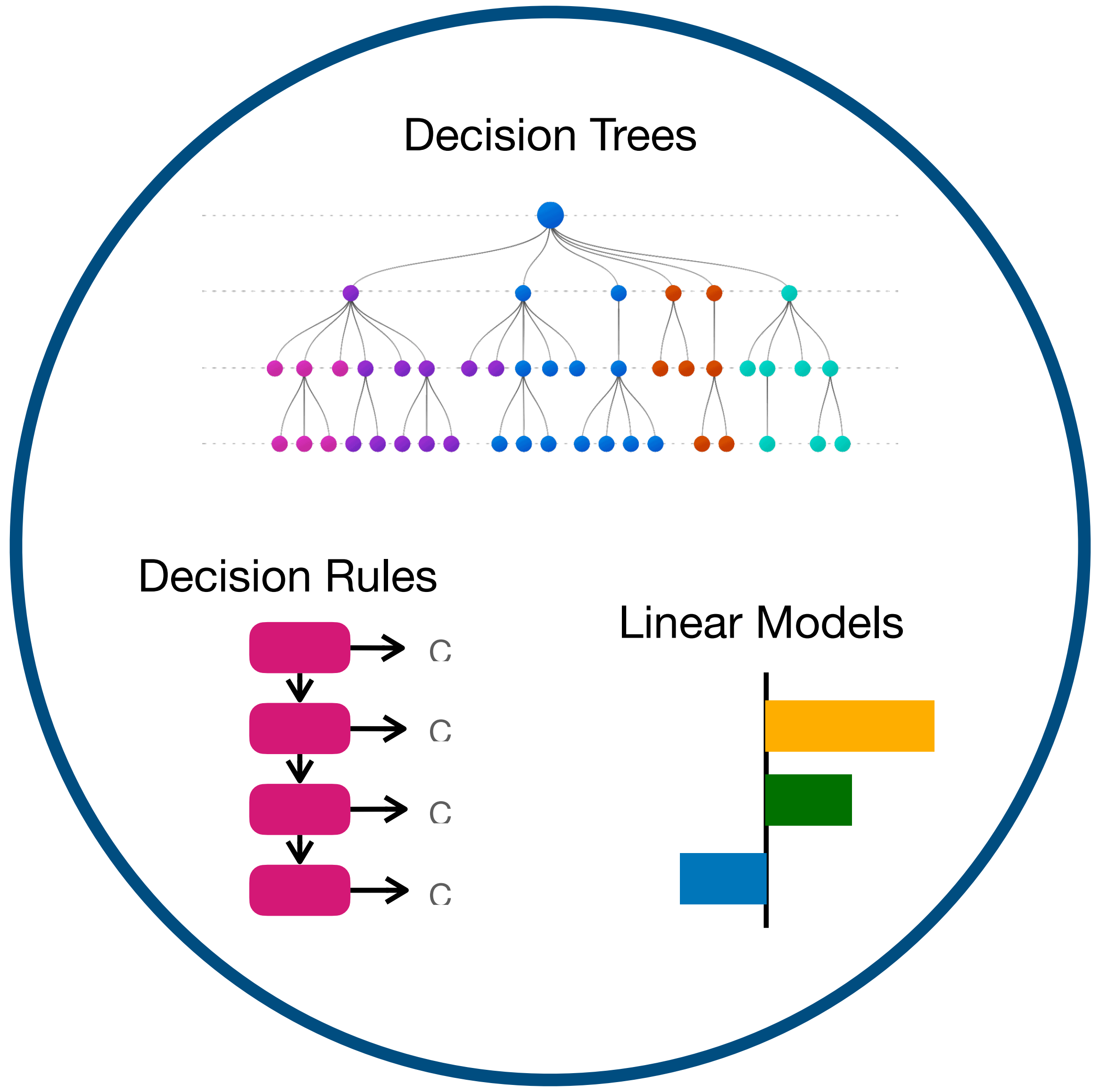
Explaining the behavior of ML components has become a necessity, especially with emerging laws and regulations (e.g. GDPR).

There is an urgent need for tools to ***synthesize "targeted"*** interpretations of ML components, with ***formal guarantees*** on their correctness.

Synthesizing Explainable Models for ML-Components

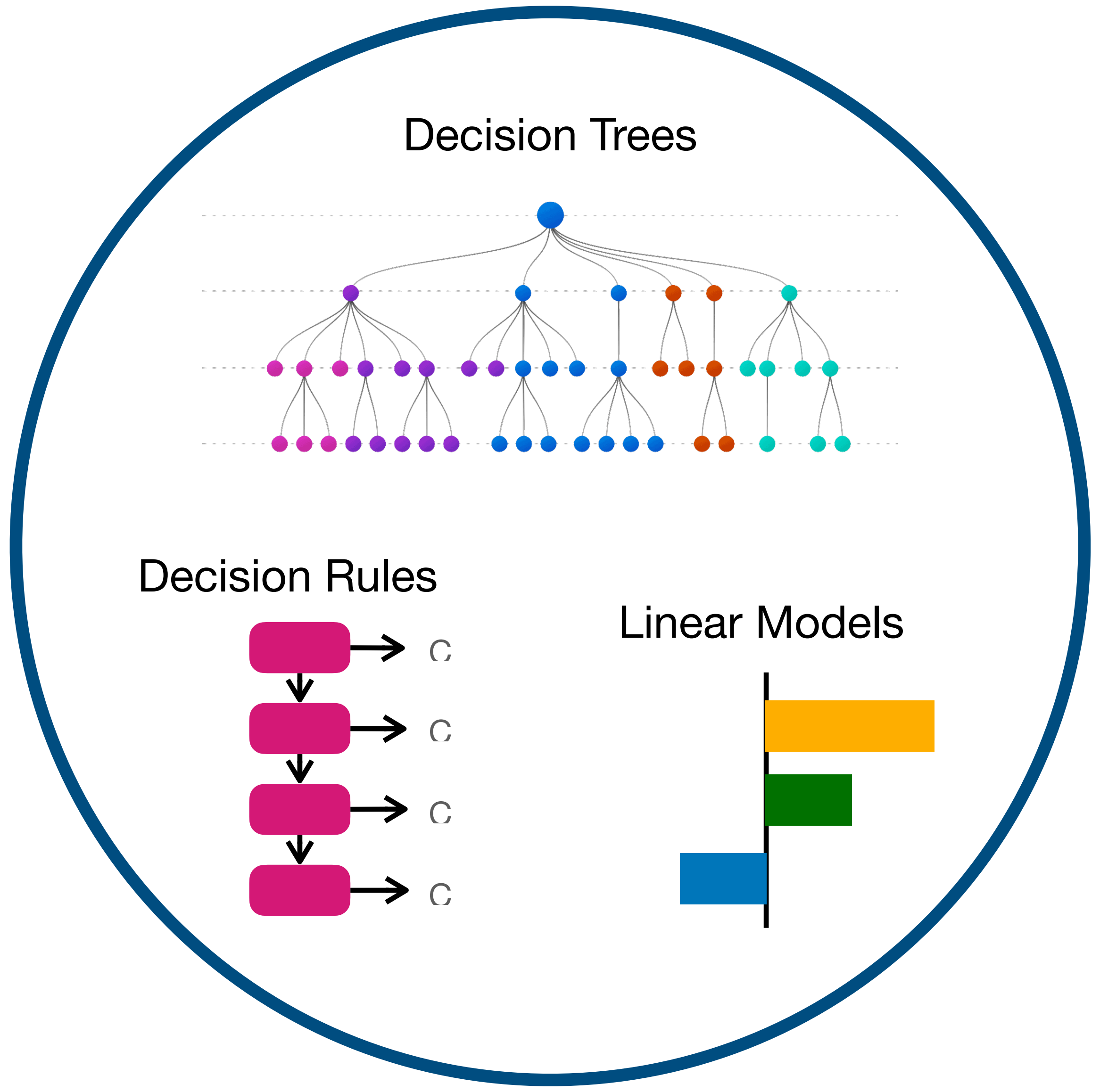
Synthesizing Explainable Models for ML-Components

Synthesis of optimal models



Synthesizing Explainable Models for ML-Components

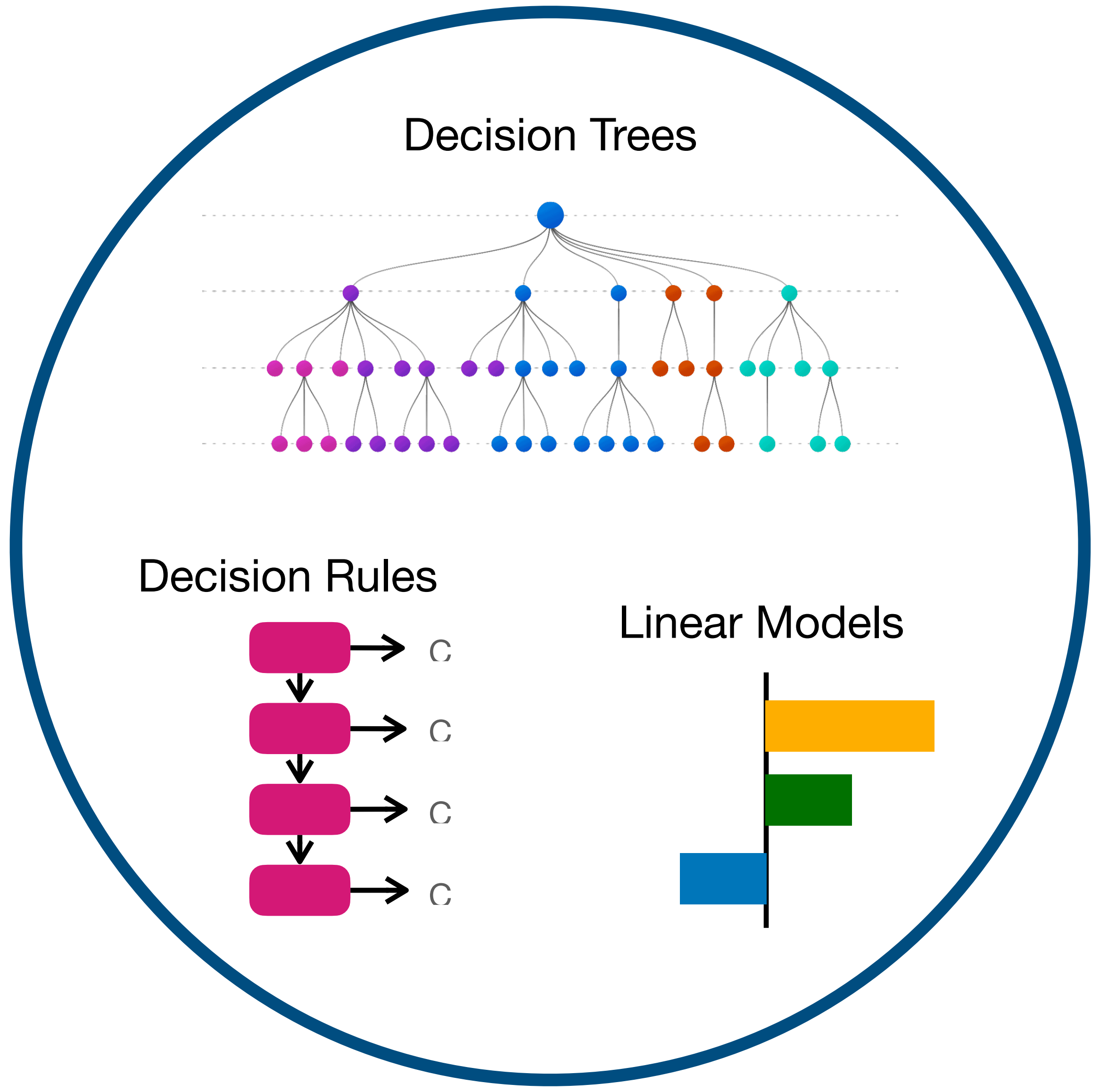
Synthesis of optimal models



- *Verwer and Zhang. Learning Decision Trees with Flexible Constraints and Objectives Using Integer Optimization. Integration of AI and OR Techniques in Constraint Programming. 2017*
- *Verhaeghe et al. Learning Optimal Decision Trees using Constraint Programming. IJCAI 2020*

Synthesizing Explainable Models for ML-Components

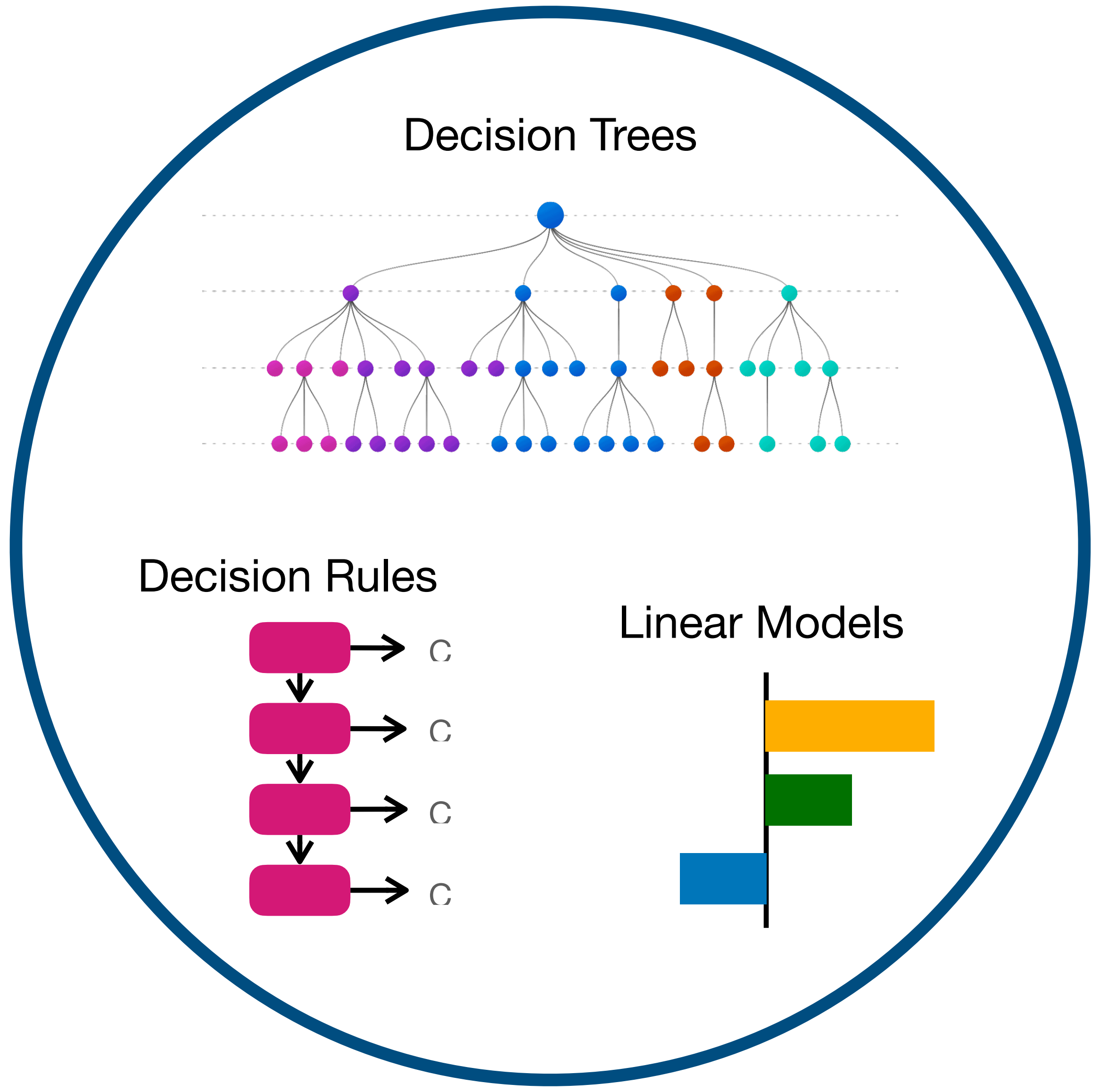
Synthesis of optimal models



- *Verwer and Zhang. Learning Decision Trees with Flexible Constraints and Objectives Using Integer Optimization. Integration of AI and OR Techniques in Constraint Programming. 2017*
- *Verhaeghe et al. Learning Optimal Decision Trees using Constraint Programming. IJCAI 2020*
- *Yu et al. Computing Optimal Decision Sets with SAT. Principles and Practice of Constraint Programming 2020*

Synthesizing Explainable Models for ML-Components

Synthesis of optimal models



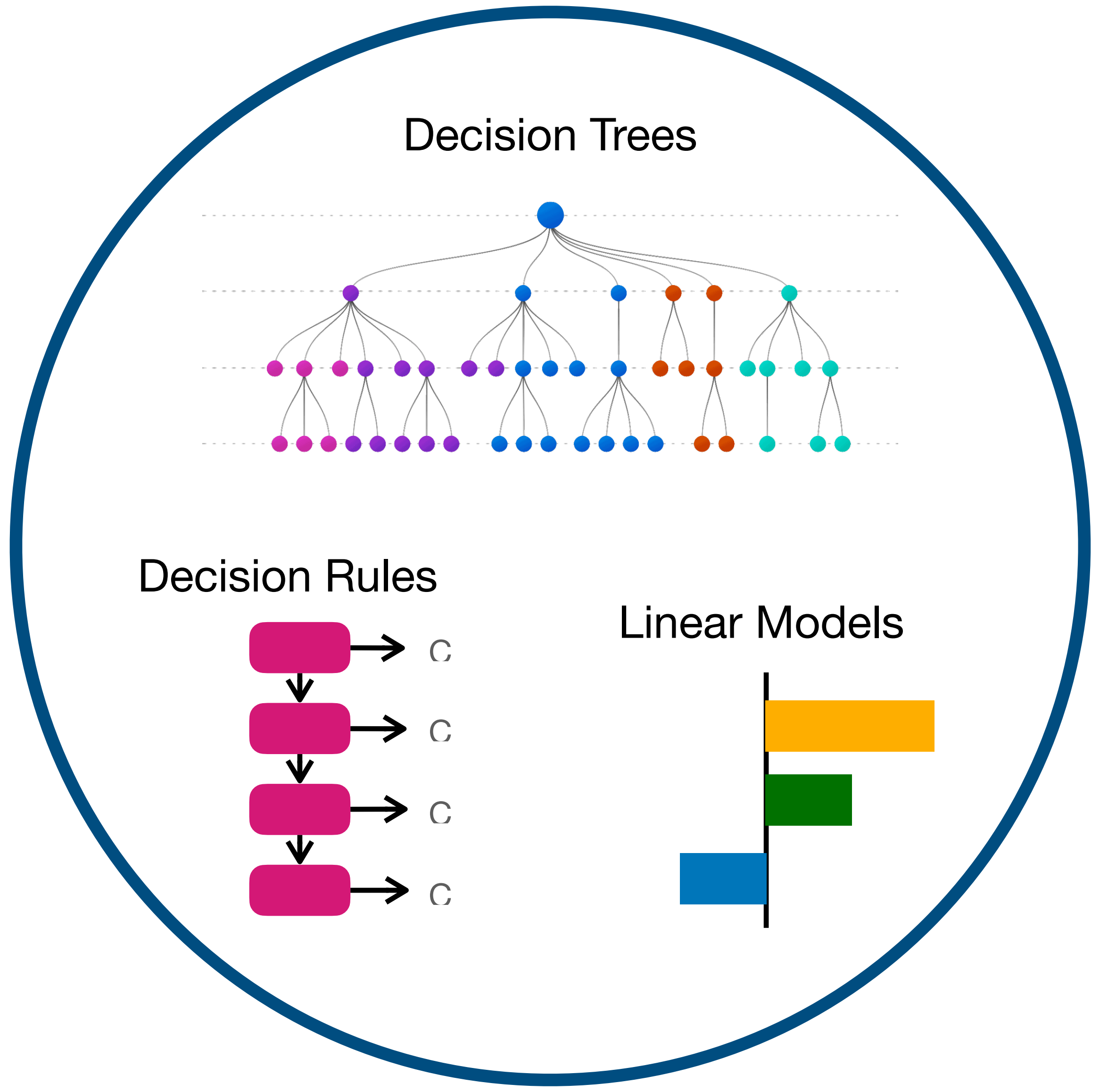
- *Verwer and Zhang. Learning Decision Trees with Flexible Constraints and Objectives Using Integer Optimization. Integration of AI and OR Techniques in Constraint Programming. 2017*
- *Verhaeghe et al. Learning Optimal Decision Trees using Constraint Programming. IJCAI 2020*
- *Yu et al. Computing Optimal Decision Sets with SAT. Principles and Practice of Constraint Programming 2020*

⋮

- *Guidotti et al. A Survey of Methods for Explaining Black Box Models. ACM Computing Surveys. 2018*
- *Adadi and Berrada. Peeking inside the black-box: A survey on Explainable Artificial Intelligence. IEEE Access 2018*

Synthesizing Explainable Models for ML-Components

Synthesis of optimal models



- *Verwer and Zhang. Learning Decision Trees with Flexible Constraints and Objectives Using Integer Optimization. Integration of AI and OR Techniques in Constraint Programming. 2017*
- *Verhaeghe et al. Learning Optimal Decision Trees using Constraint Programming. IJCAI 2020*
- *Yu et al. Computing Optimal Decision Sets with SAT. Principles and Practice of Constraint Programming 2020*

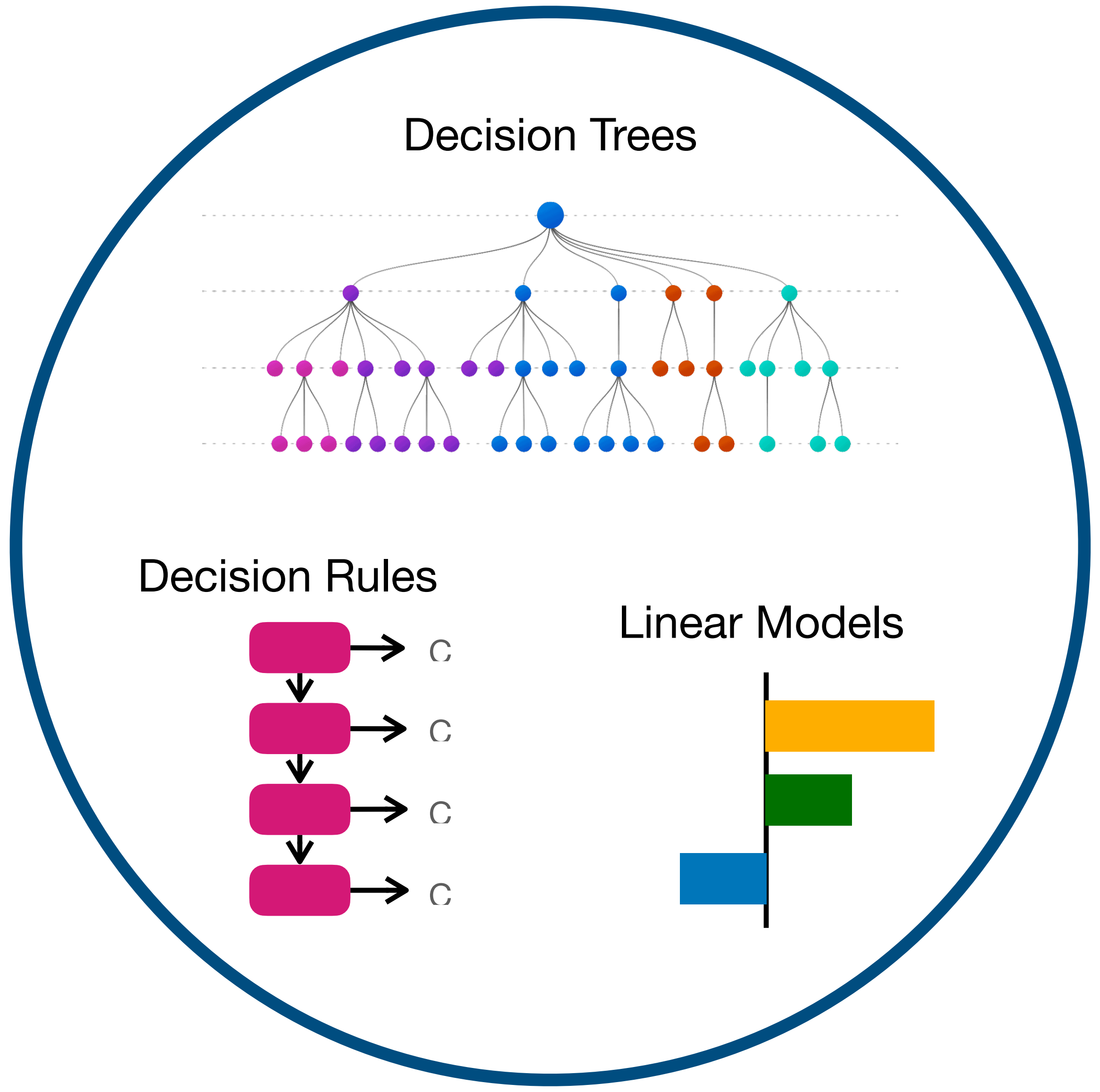
⋮

- *Guidotti et al. A Survey of Methods for Explaining Black Box Models. ACM Computing Surveys. 2018*
- *Adadi and Berrada. Peeking inside the black-box: A survey on Explainable Artificial Intelligence. IEEE Access 2018*

Approaches are based on single-objective formulation of the problem

Synthesizing Explainable Models for ML-Components

Synthesis of optimal models

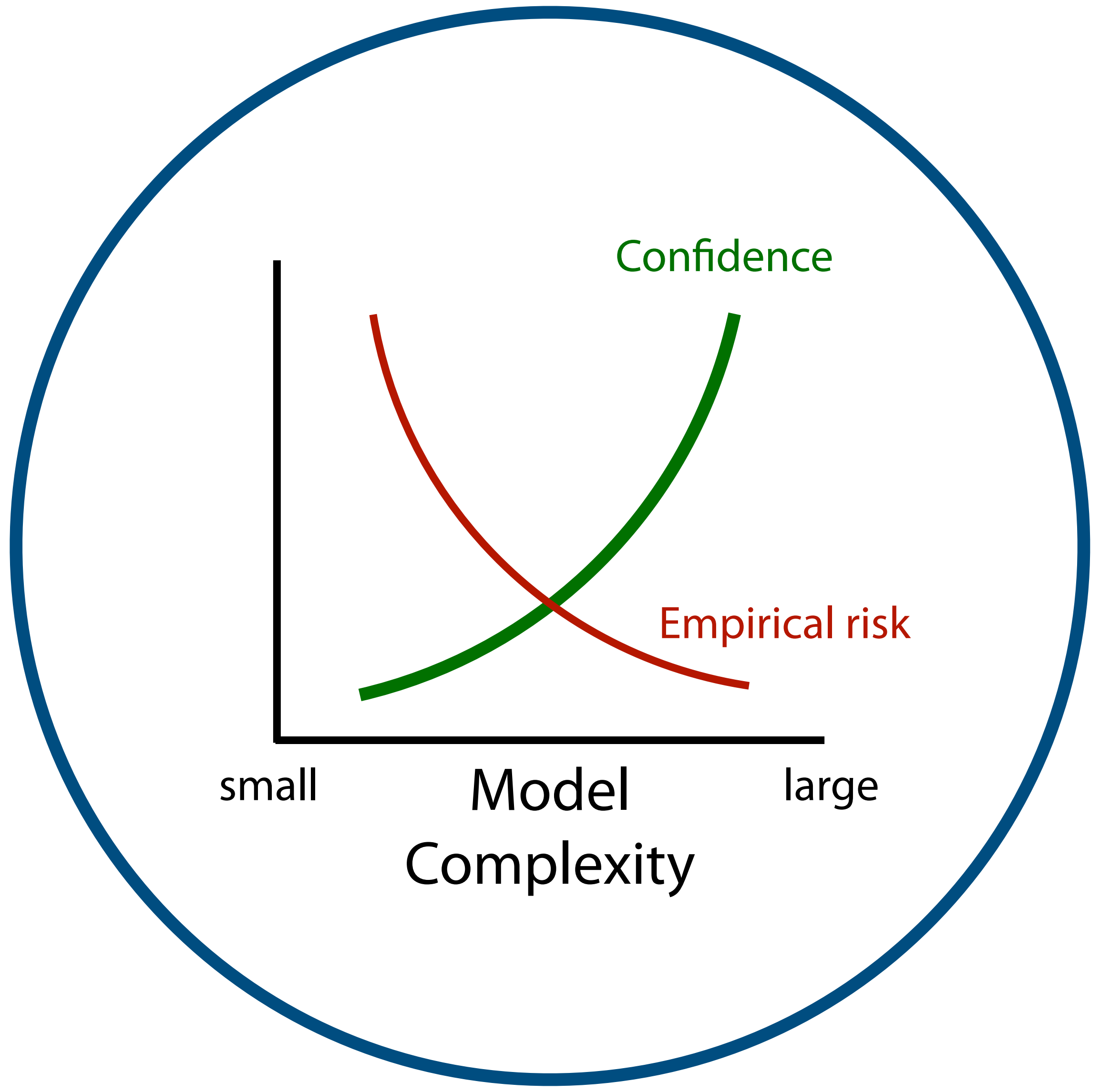


- *Verwer and Zhang. Learning Decision Trees with Flexible Constraints and Objectives Using Integer Optimization. Integration of AI and OR Techniques in Constraint Programming. 2017*
- *Verhaeghe et al. Learning Optimal Decision Trees using Constraint Programming. IJCAI 2020*
- *Yu et al. Computing Optimal Decision Sets with SAT. Principles and Practice of Constraint Programming 2020*

⋮

- *Guidotti et al. A Survey of Methods for Explaining Black Box Models. ACM Computing Surveys. 2018*
- *Adadi and Berrada. Peeking inside the black-box: A survey on Explainable Artificial Intelligence. IEEE Access 2018*

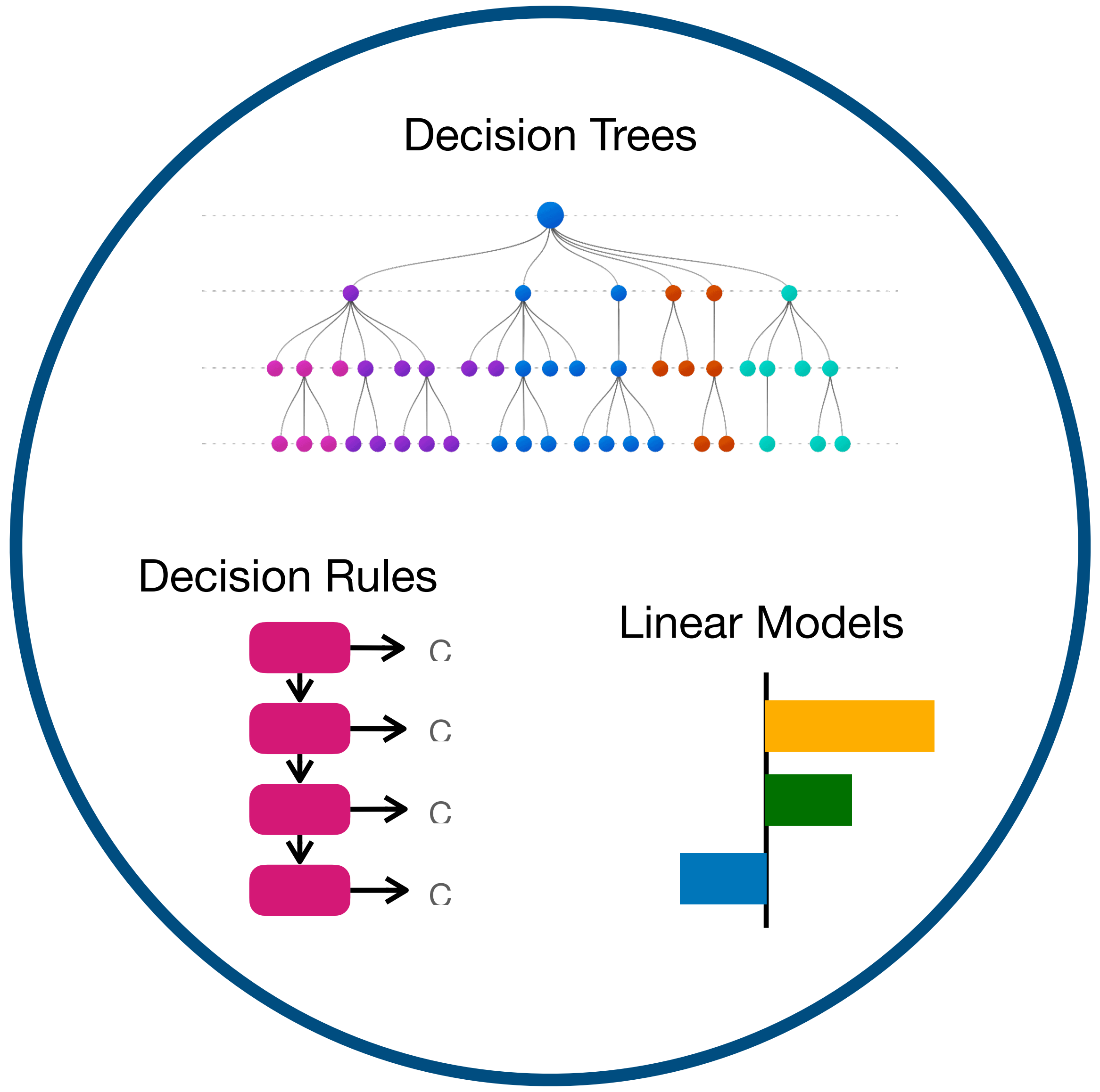
Structural risk minimization



Approaches are based on single-objective formulation of the problem

Synthesizing Explainable Models for ML-Components

Synthesis of optimal models

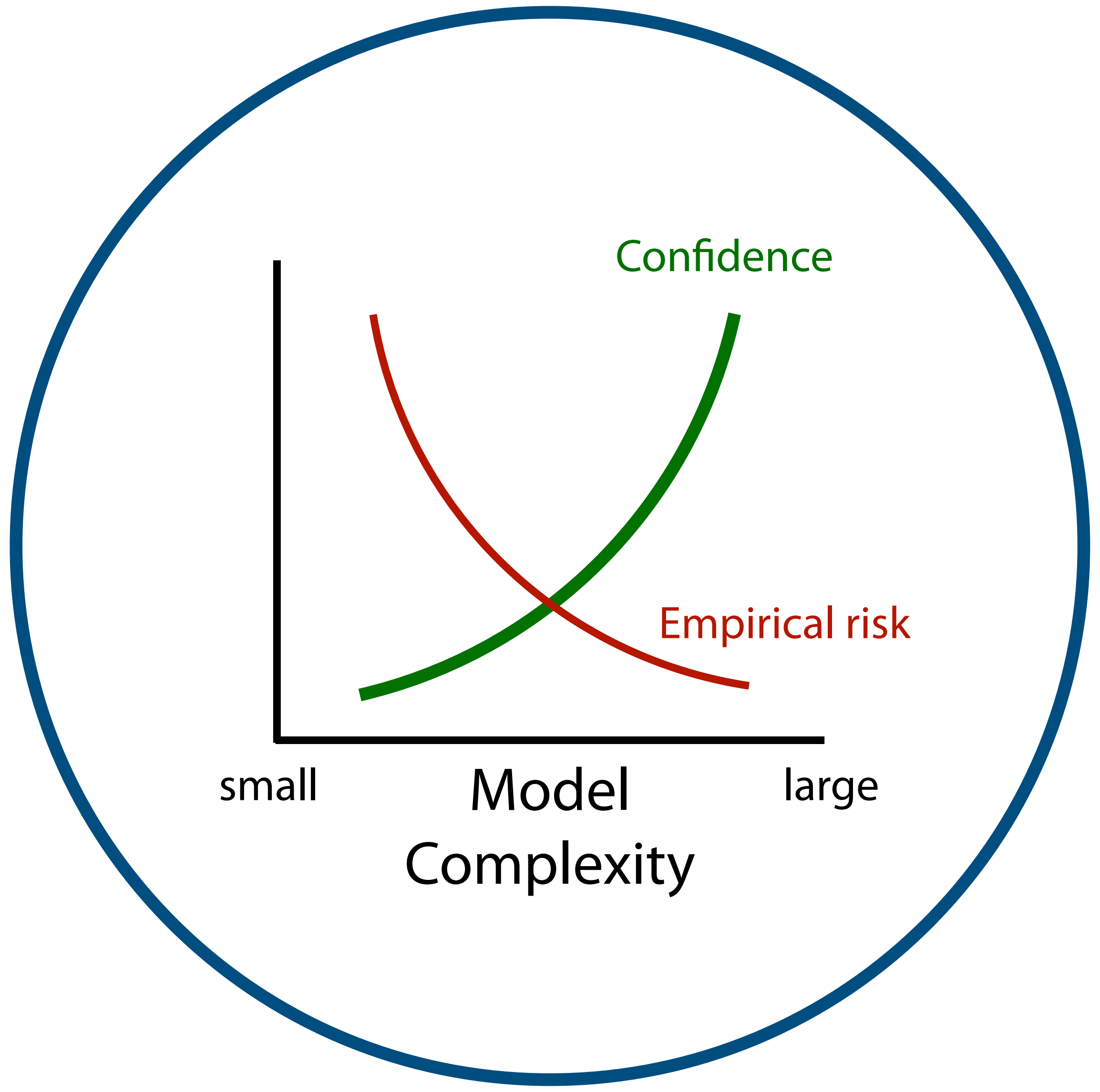


- *Verwer and Zhang. Learning Decision Trees with Flexible Constraints and Objectives Using Integer Optimization. Integration of AI and OR Techniques in Constraint Programming. 2017*
- *Verhaeghe et al. Learning Optimal Decision Trees using Constraint Programming. IJCAI 2020*
- *Yu et al. Computing Optimal Decision Sets with SAT. Principles and Practice of Constraint Programming 2020*

⋮

- *Guidotti et al. A Survey of Methods for Explaining Black Box Models. ACM Computing Surveys. 2018*
- *Adadi and Berrada. Peeking inside the black-box: A survey on Explainable Artificial Intelligence. IEEE Access 2018*

Structural risk minimization

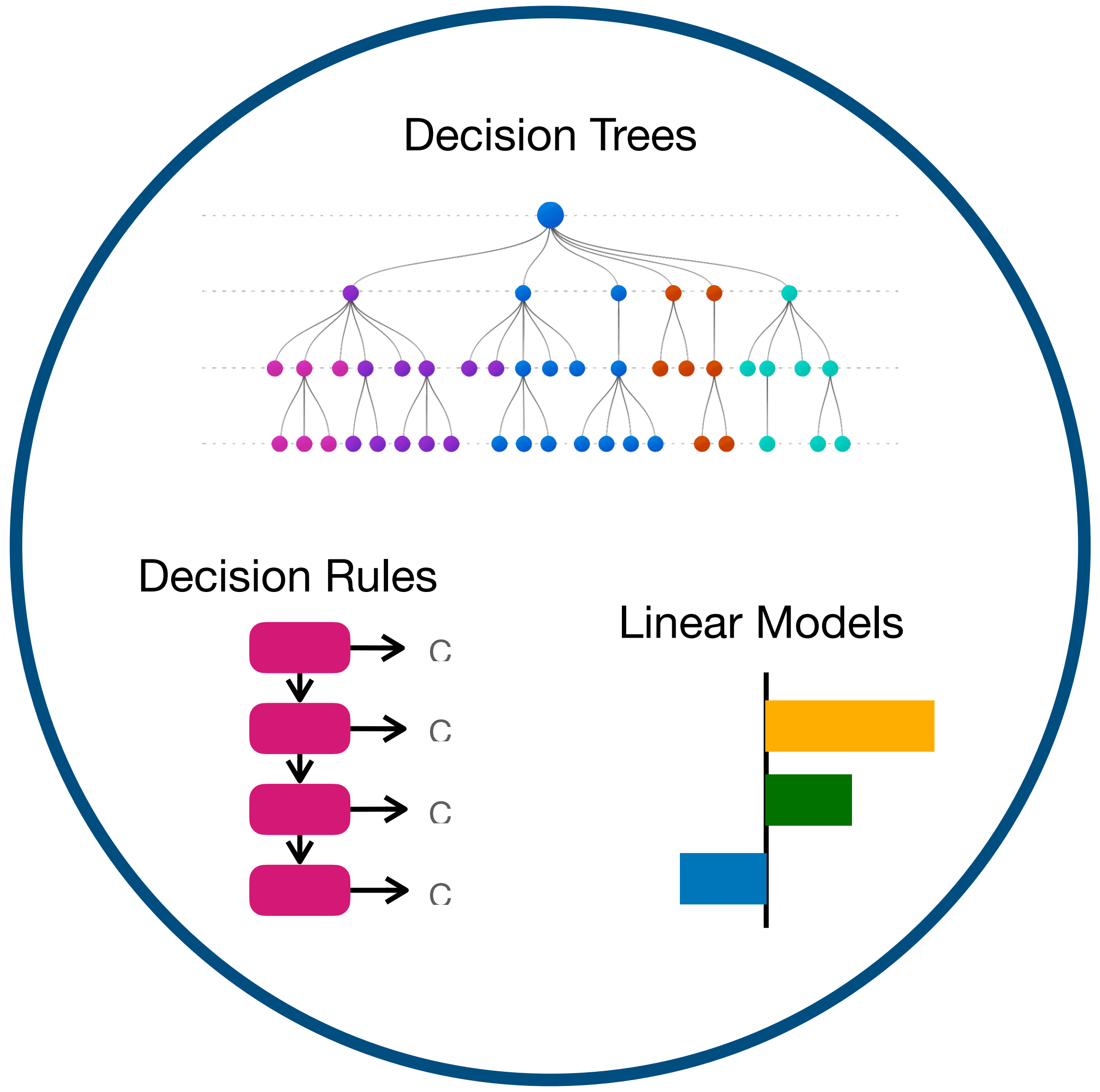


Approaches are based on single-objective formulation of the problem

Interpretation synthesis is an optimization problem with “conflicting" objectives: **correctness** and **explainability**

Synthesizing Explainable Models for ML-Components

Synthesis of optimal models

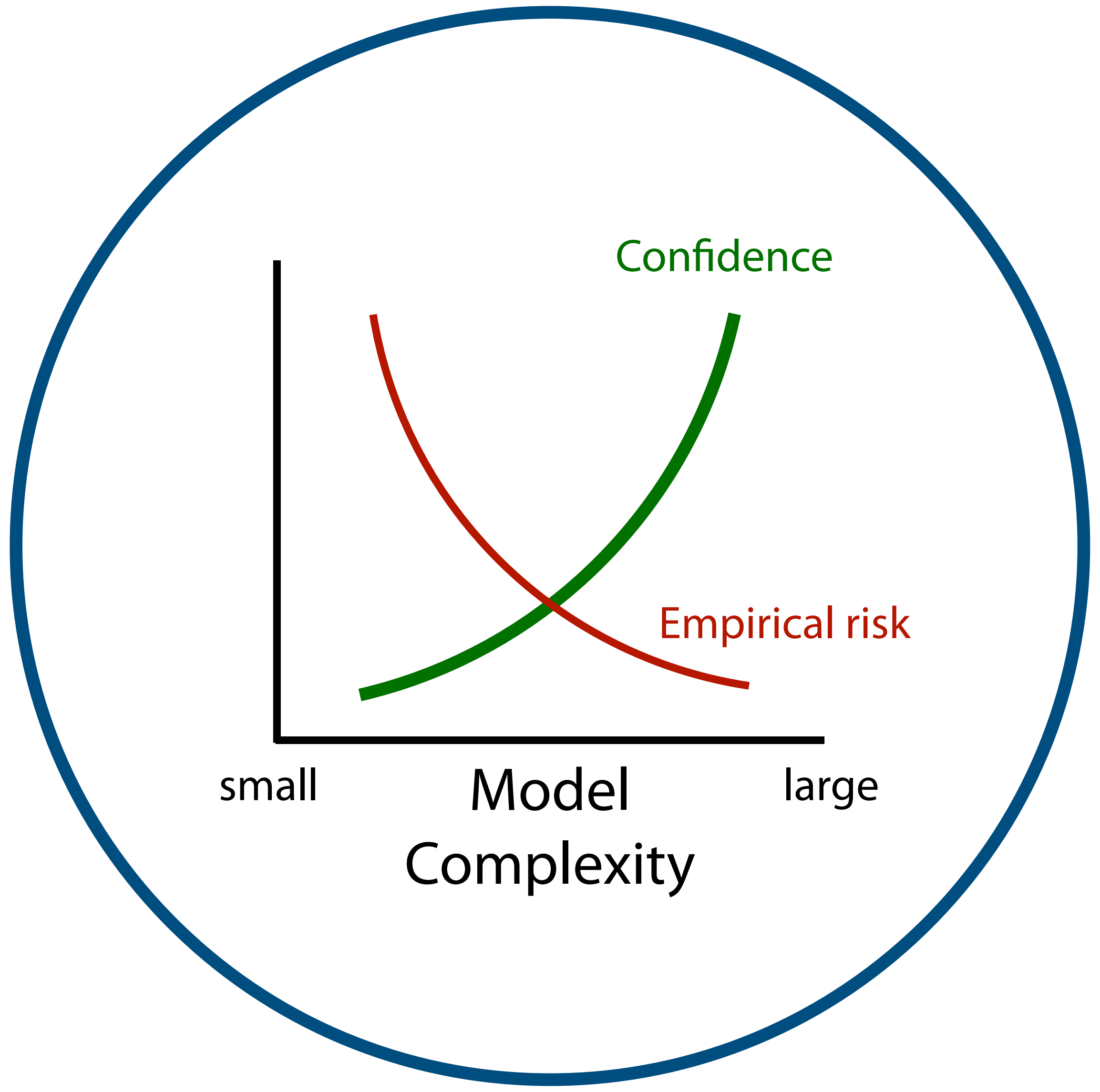


- *Verwer and Zhang. Learning Decision Trees with Flexible Constraints and Objectives Using Integer Optimization. Integration of AI and OR Techniques in Constraint Programming. 2017*
- *Verhaeghe et al. Learning Optimal Decision Trees using Constraint Programming. IJCAI 2020*
- *Yu et al. Computing Optimal Decision Sets with SAT. Principles and Practice of Constraint Programming 2020*

⋮

- *Guidotti et al. A Survey of Methods for Explaining Black Box Models. ACM Computing Surveys. 2018*
- *Adadi and Berrada. Peeking inside the black-box: A survey on Explainable Artificial Intelligence. IEEE Access 2018*

Structural risk minimization



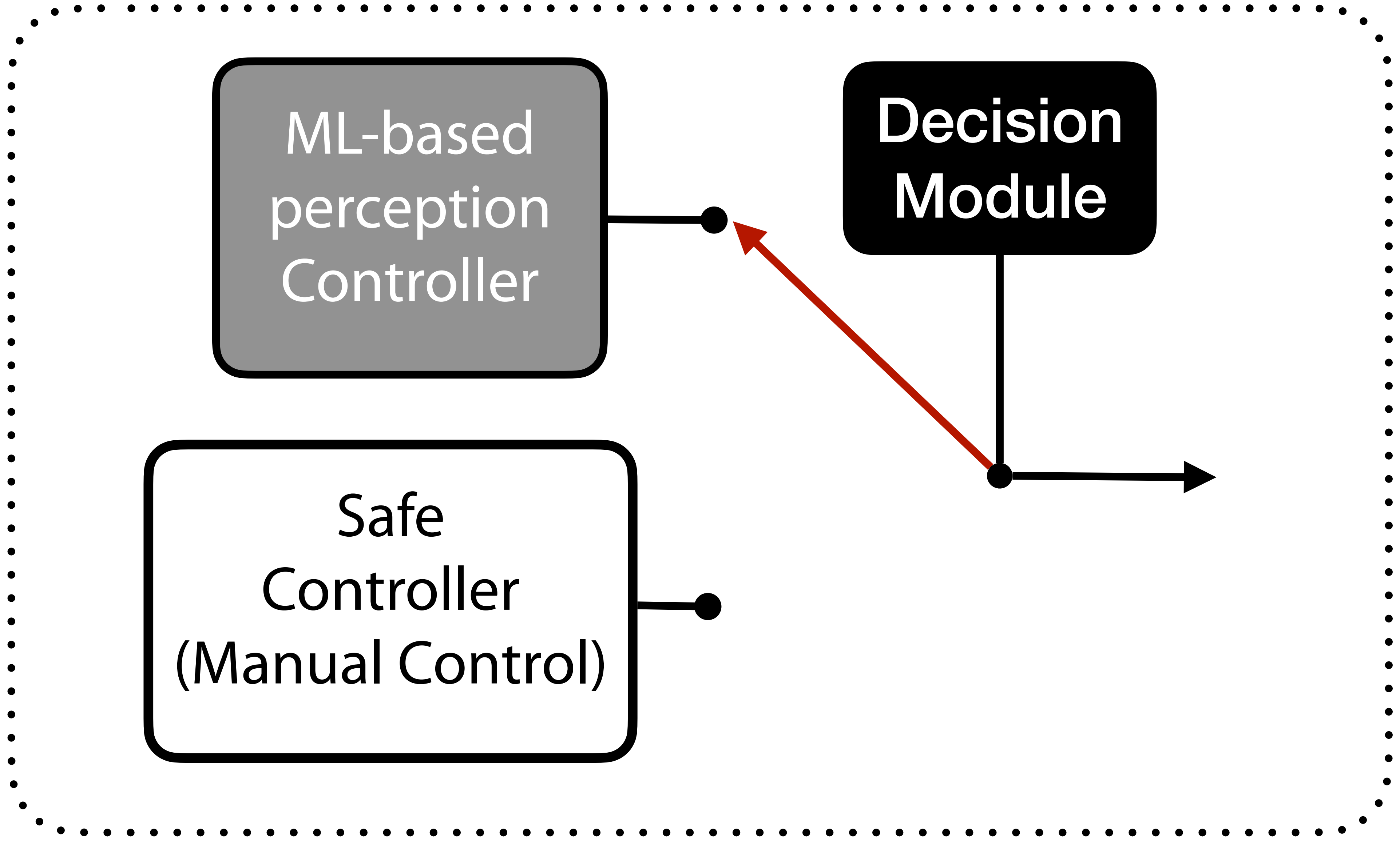
Approaches are based on single-objective formulation of the problem

Interpretation synthesis is an optimization problem with “conflicting" objectives:
correctness and **explainability**
Our goal: exploration of Pareto-optimal interpretations

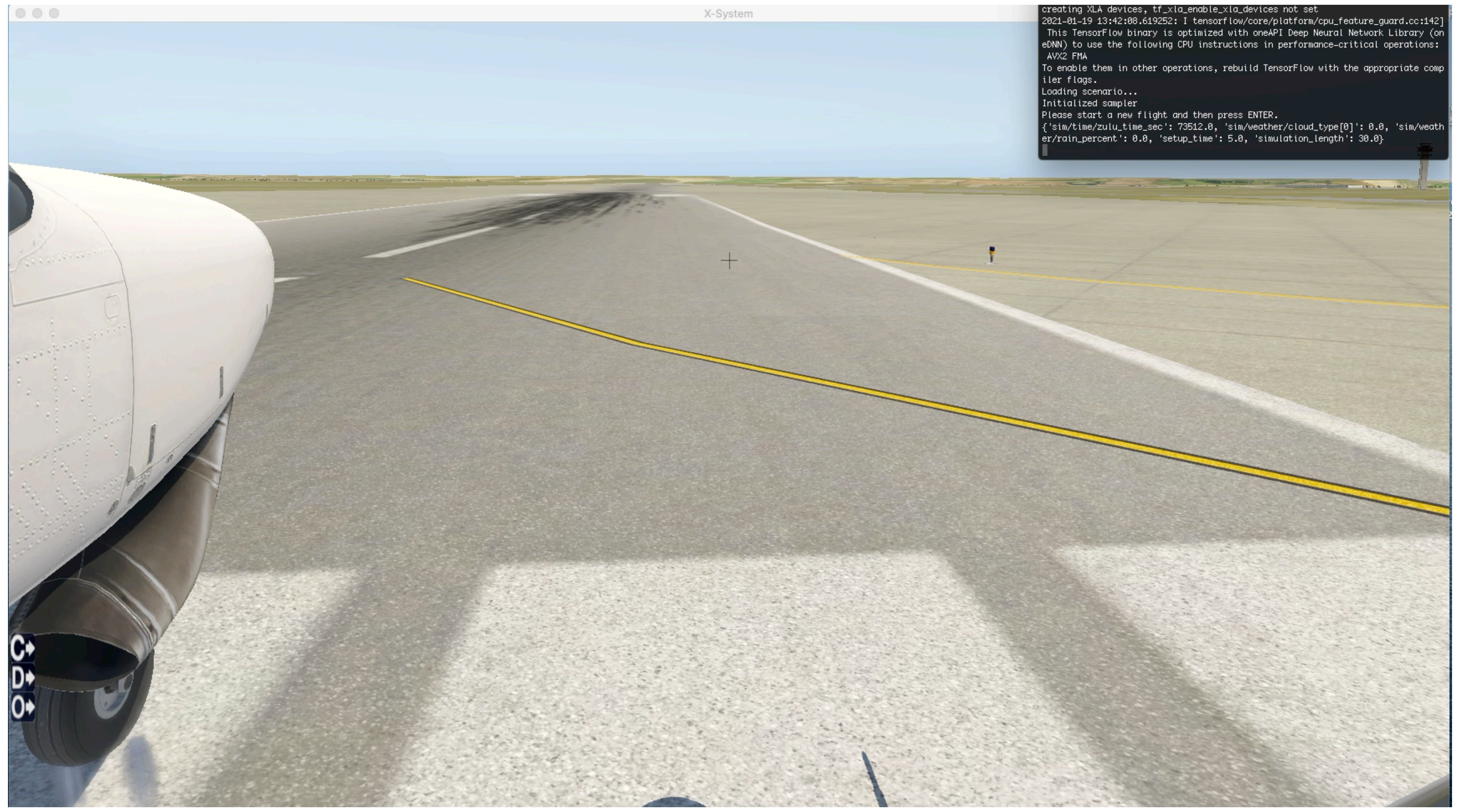
Outline

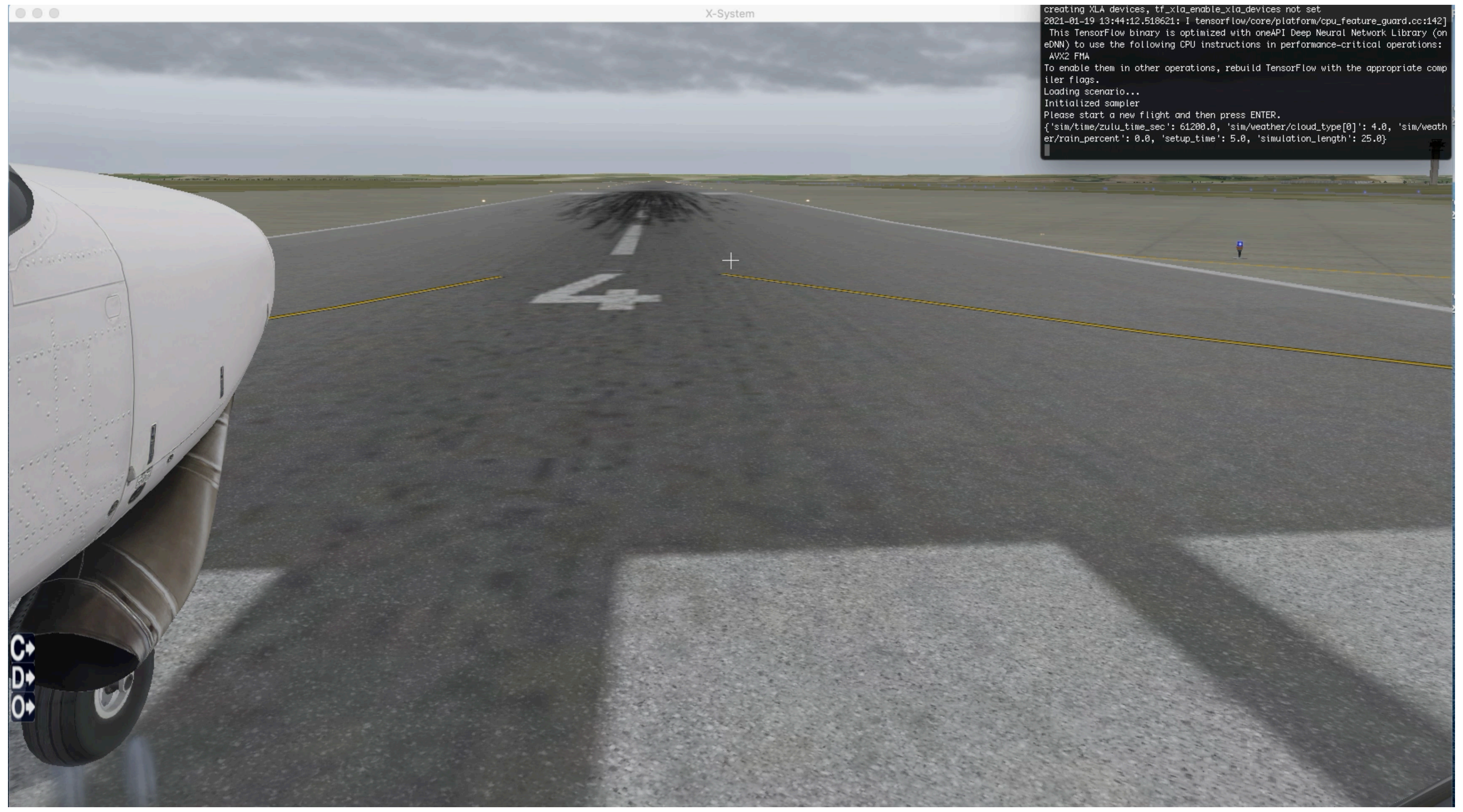
- **Pareto-optimal interpretation synthesis**
  - Example
  - Formal Problem Definition
- **In finite domains: A MaxSAT-based solution**
- **Exploring the Pareto-optimal space of interpretations**
- **Statistical guarantees for black-box models**
- **Experimental results**

Balancing between Correctness and Explainability

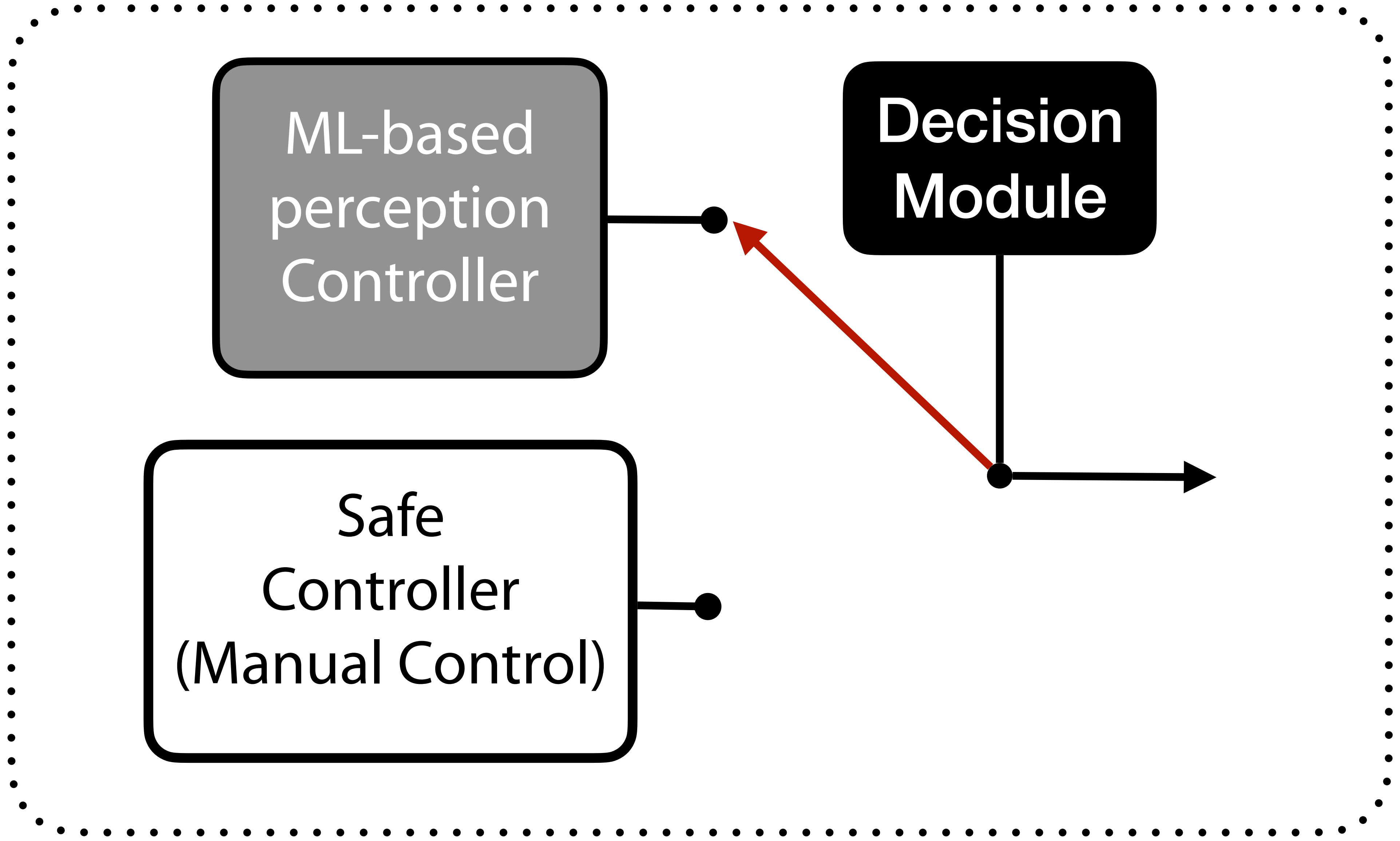


AutoTAXI

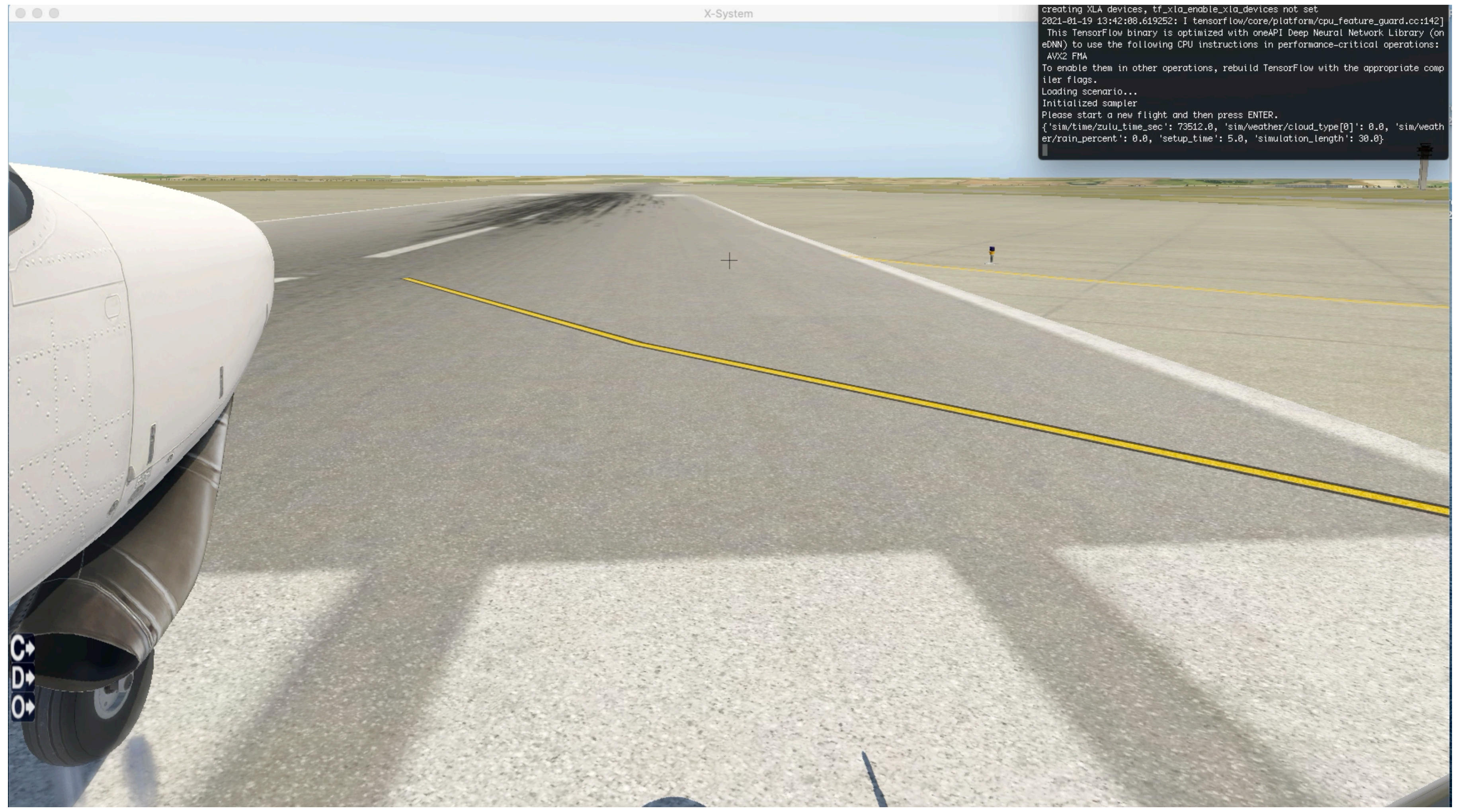


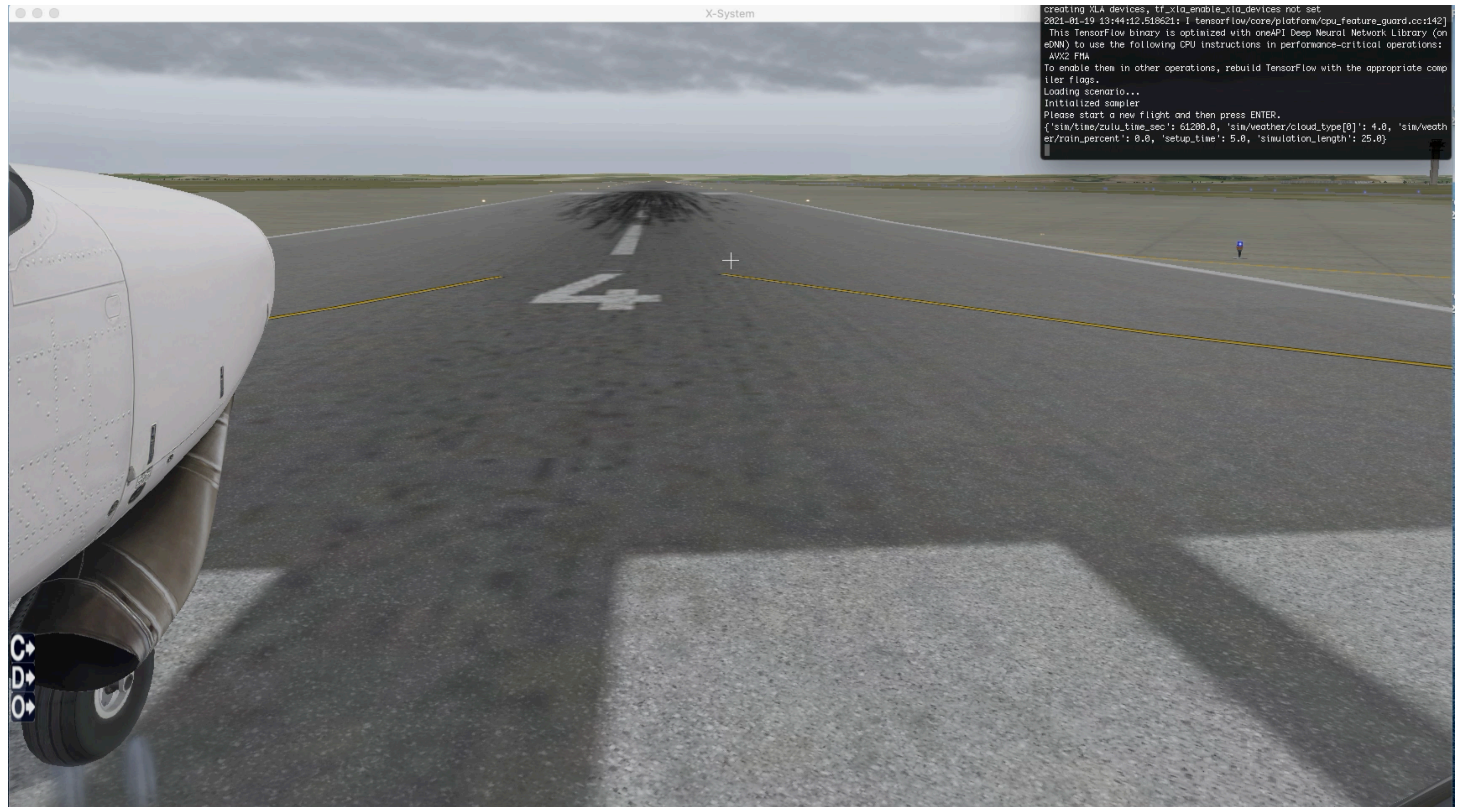


Balancing between Correctness and Explainability



AutoTAXI

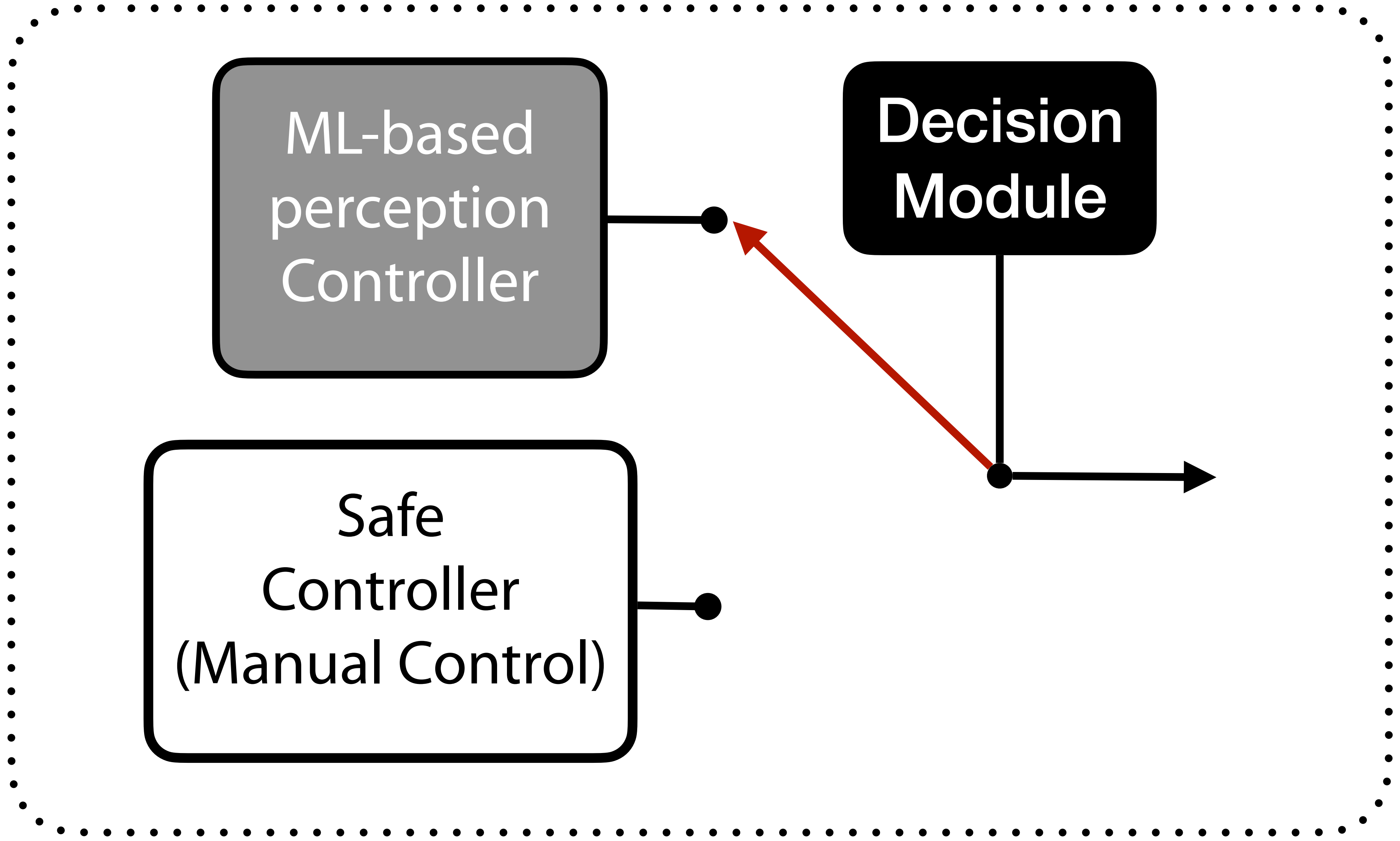




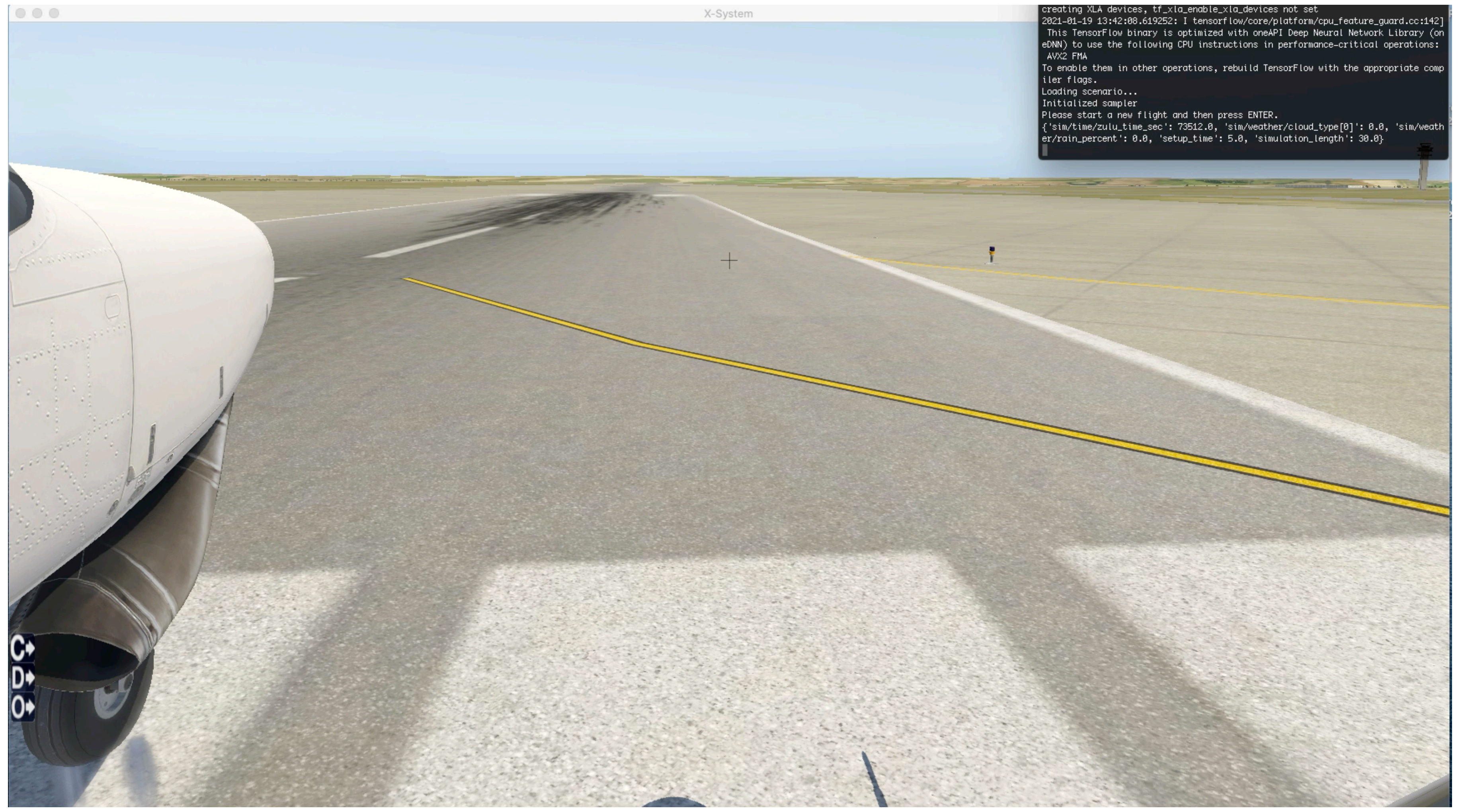
Decision Module decides to trust ML-component based on:

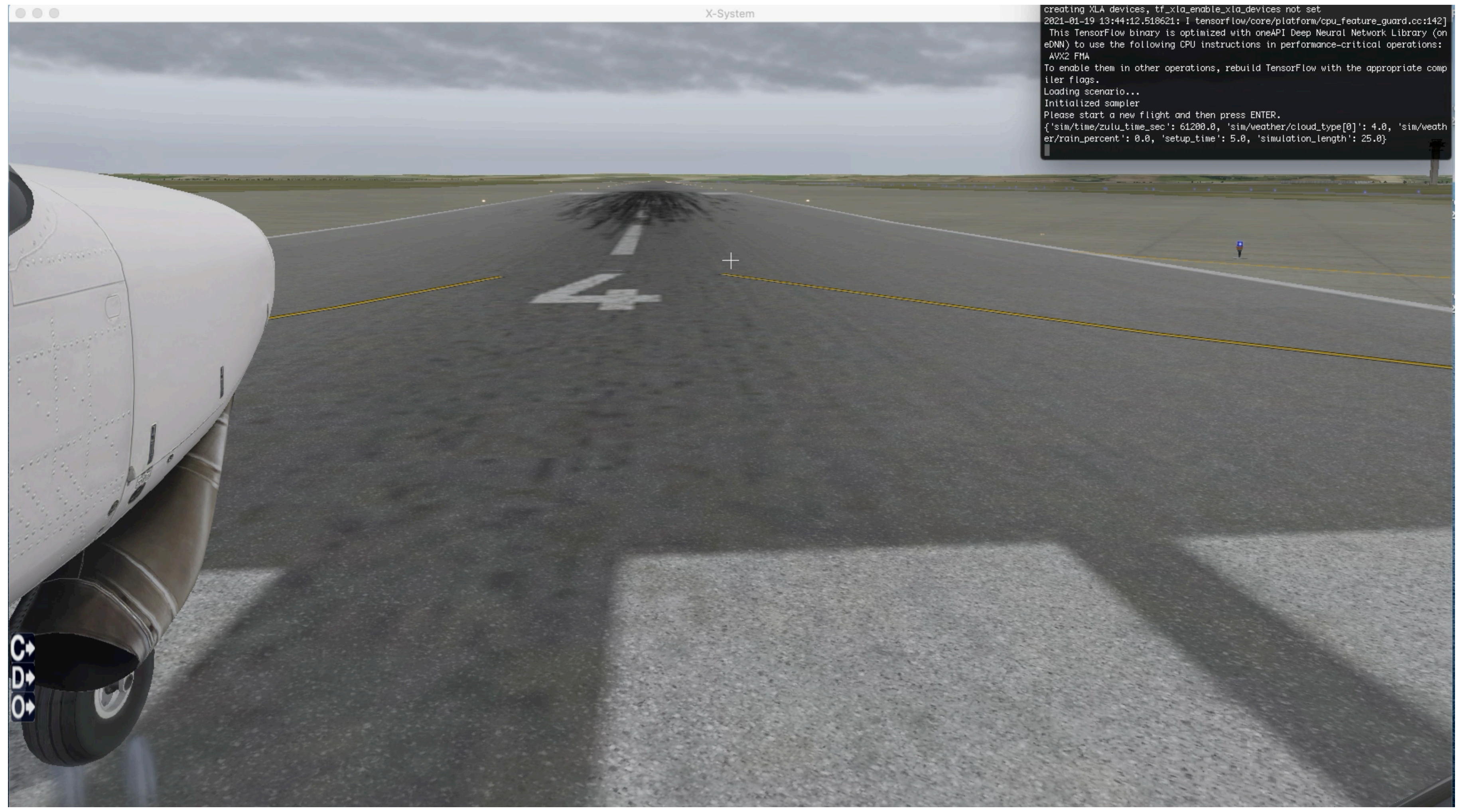
- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

Balancing between Correctness and Explainability



AutoTAXI

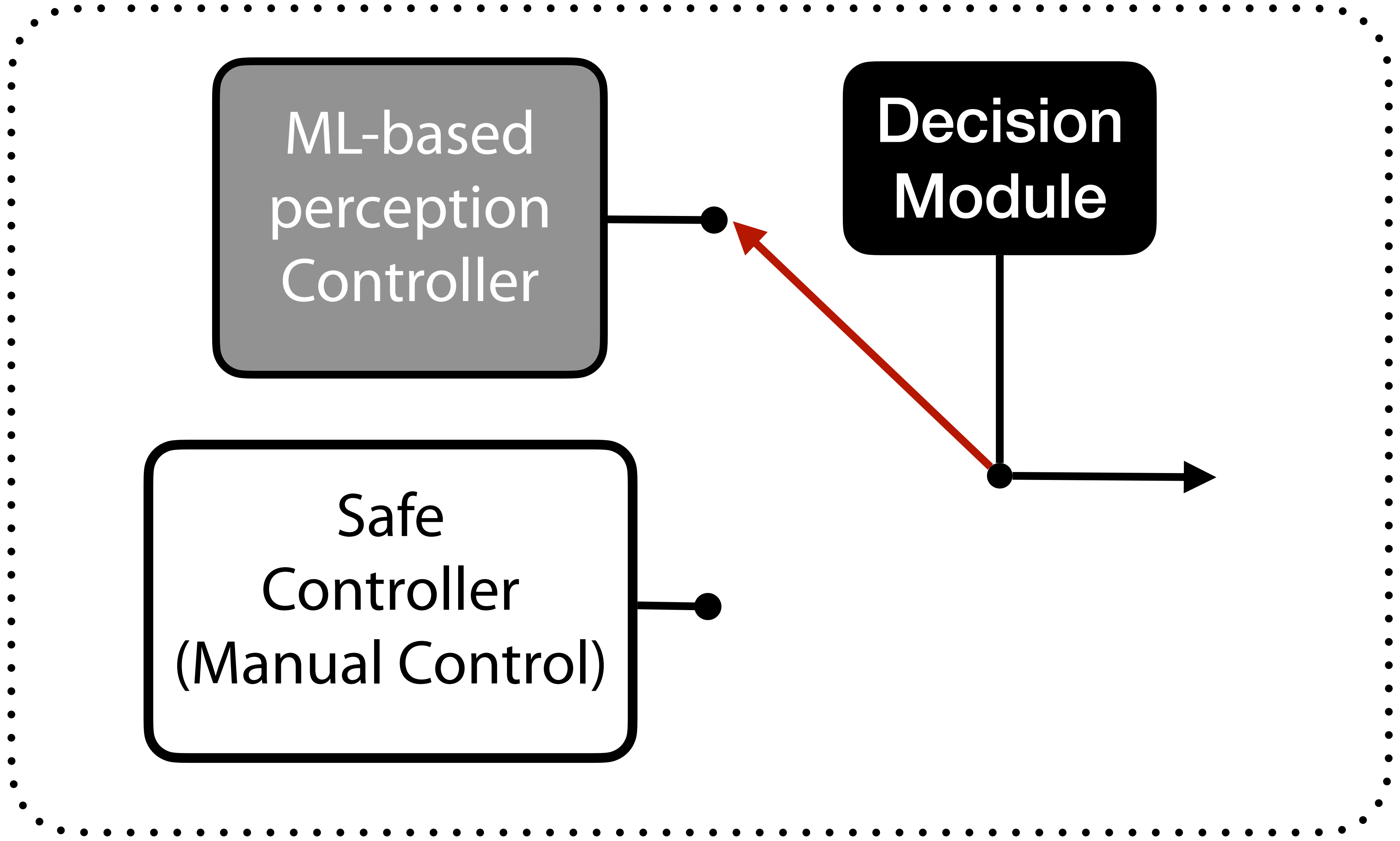




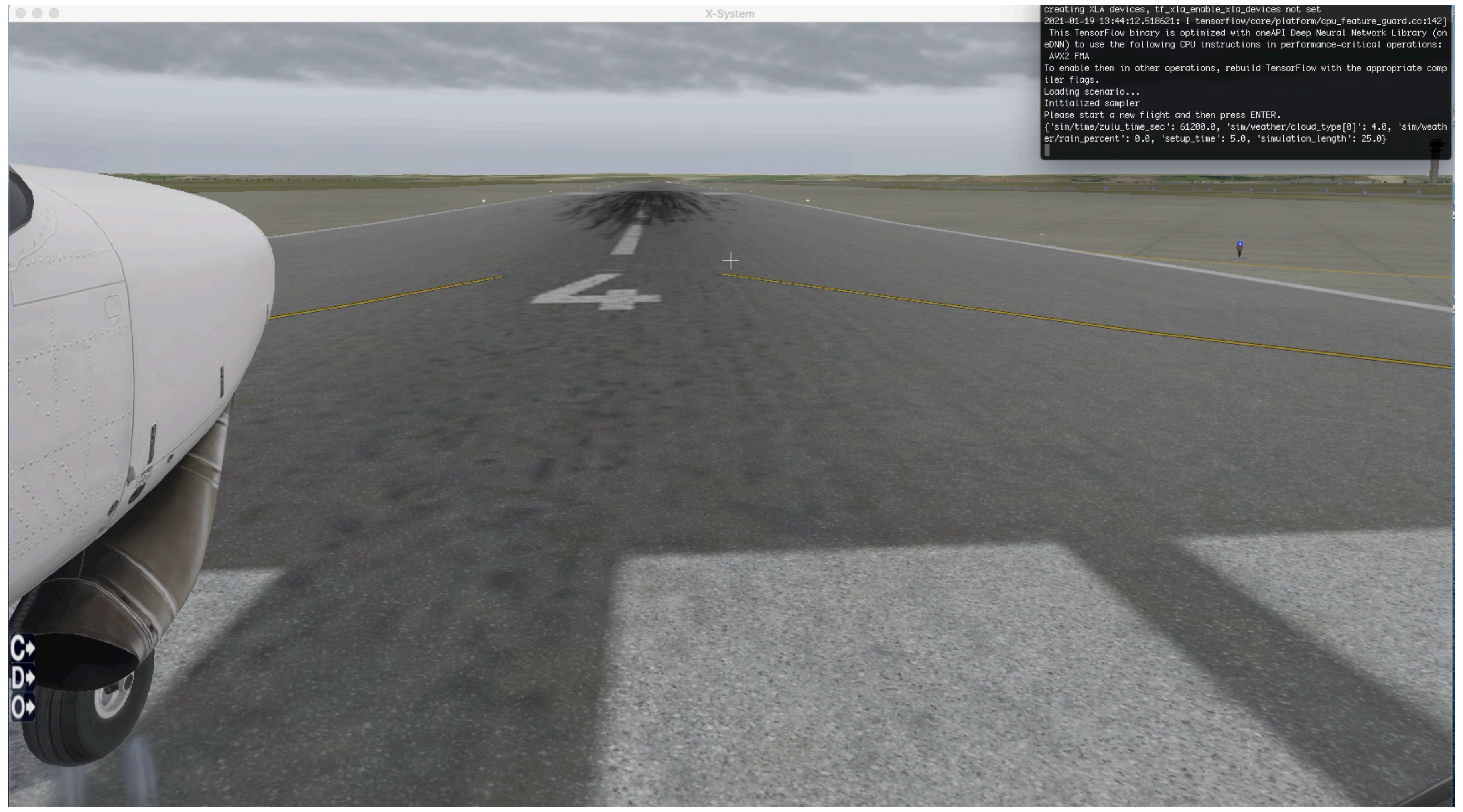
Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

Balancing between Correctness and Explainability



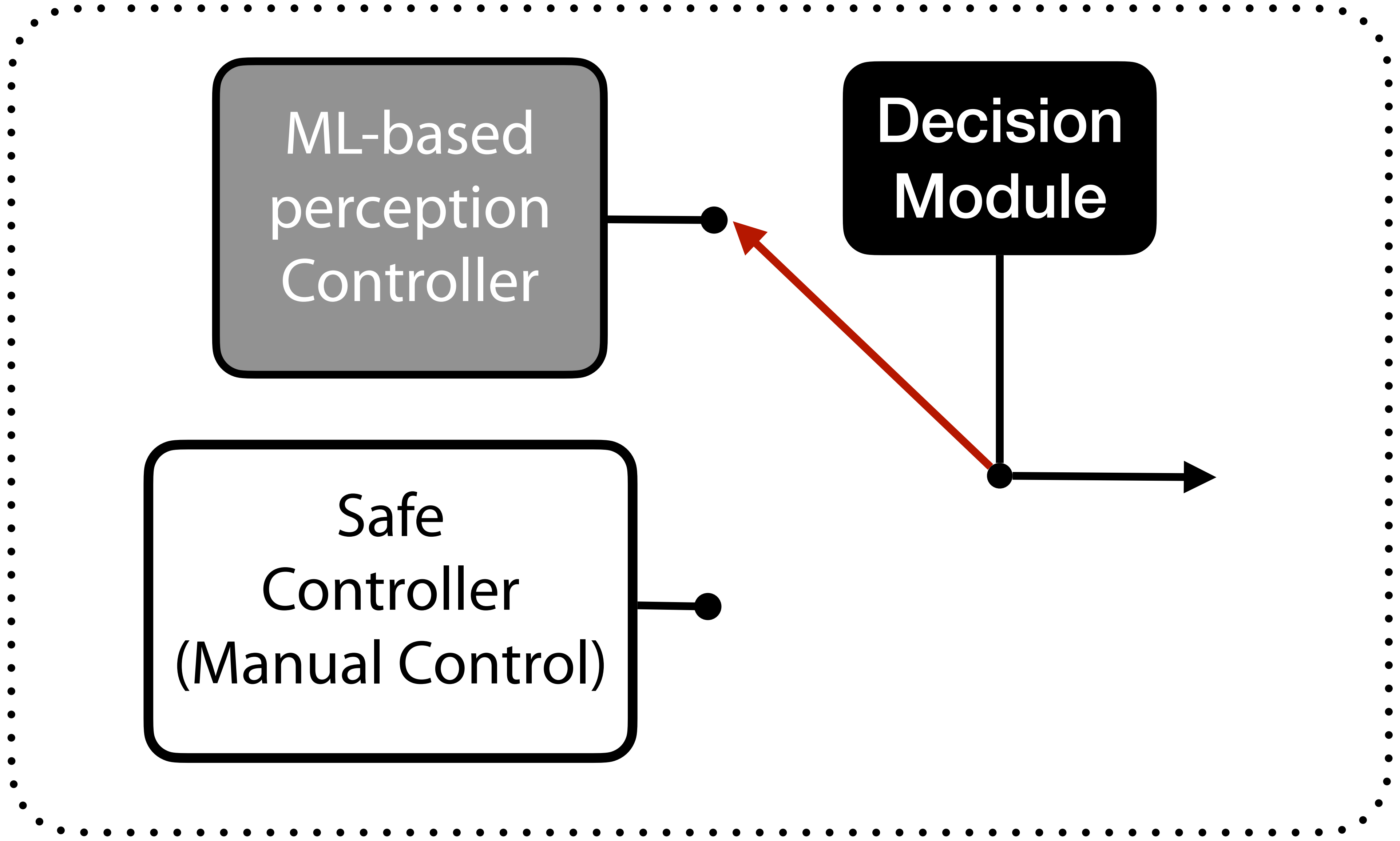
AutoTAXI



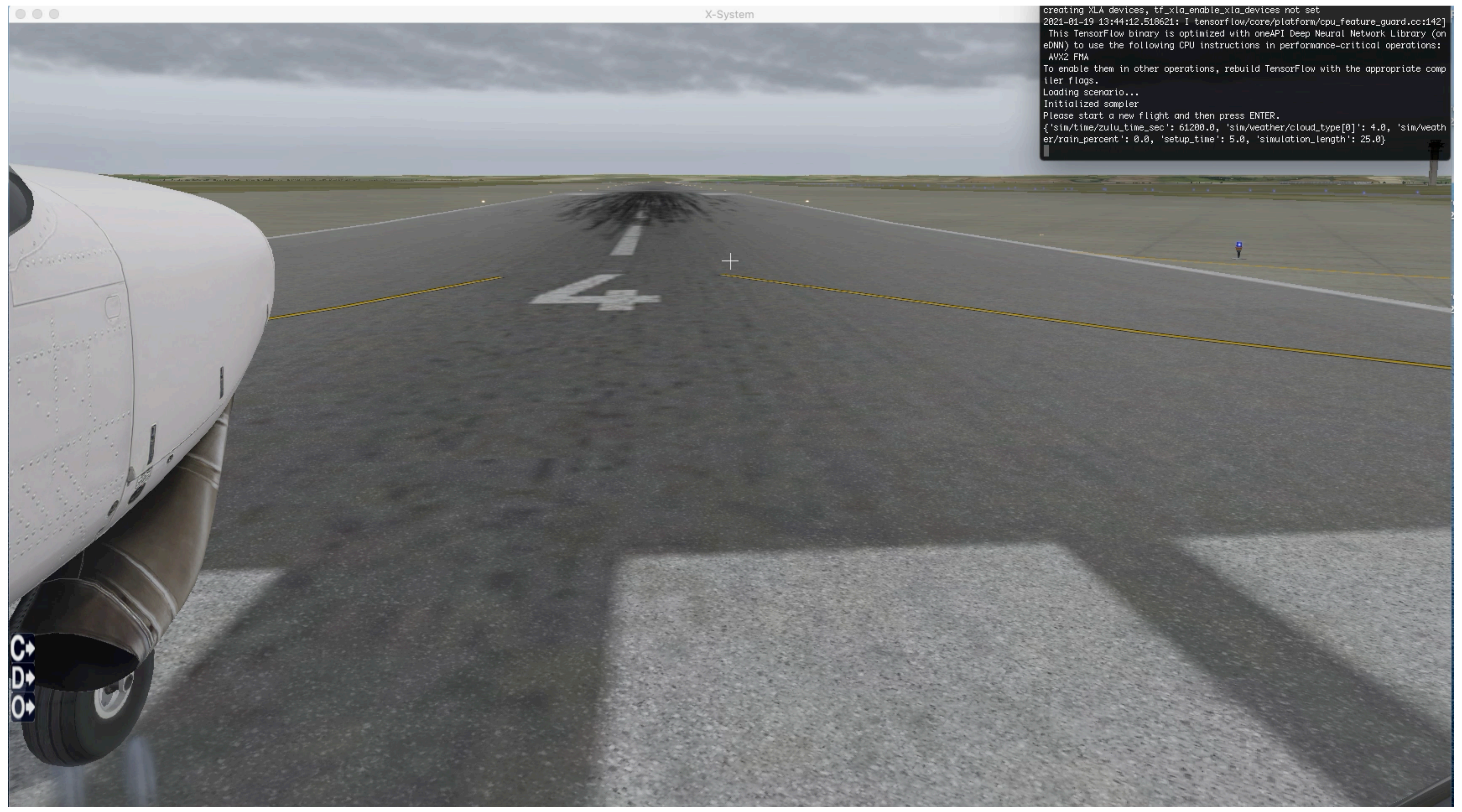
Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

Balancing between Correctness and Explainability



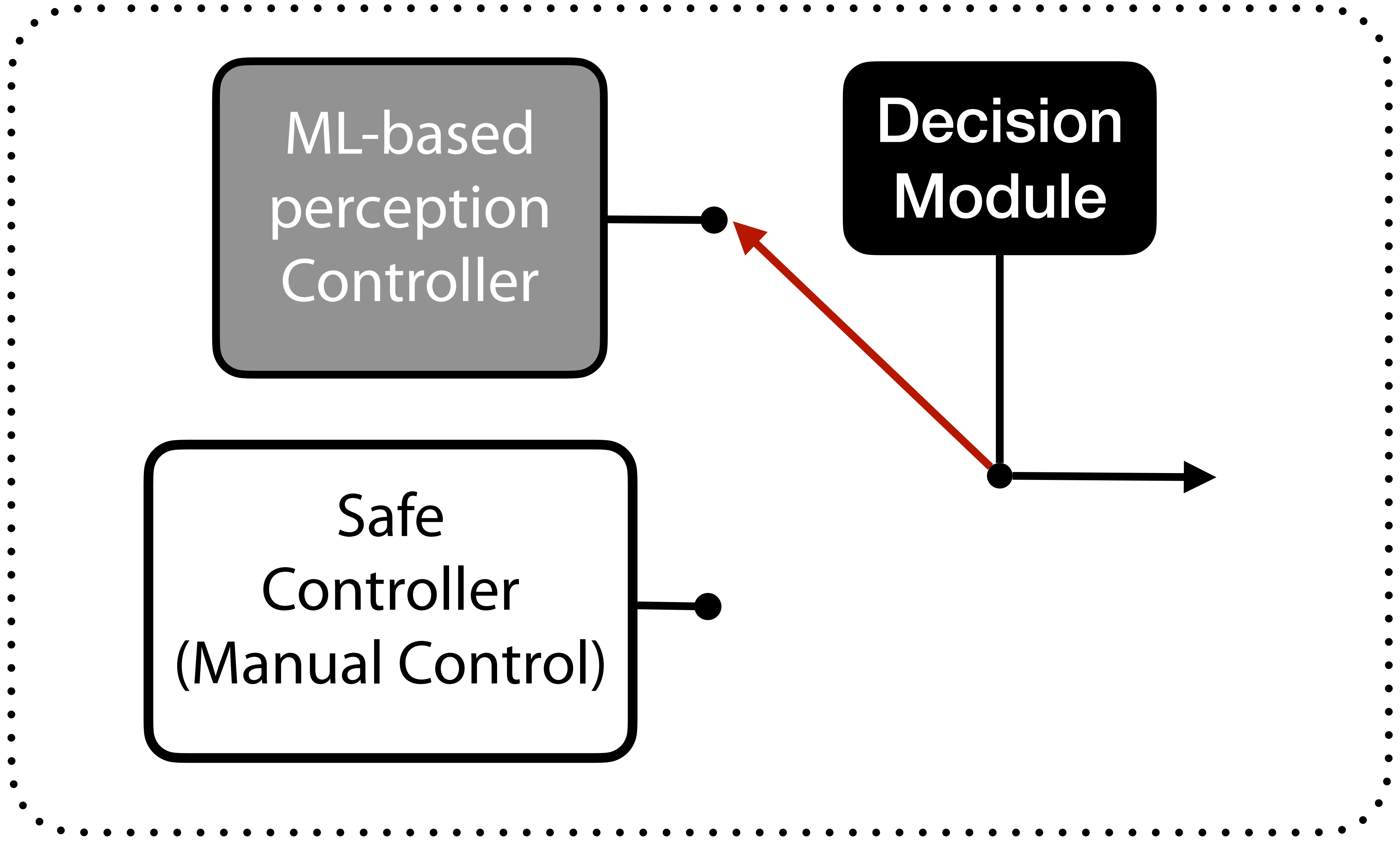
AutoTAXI



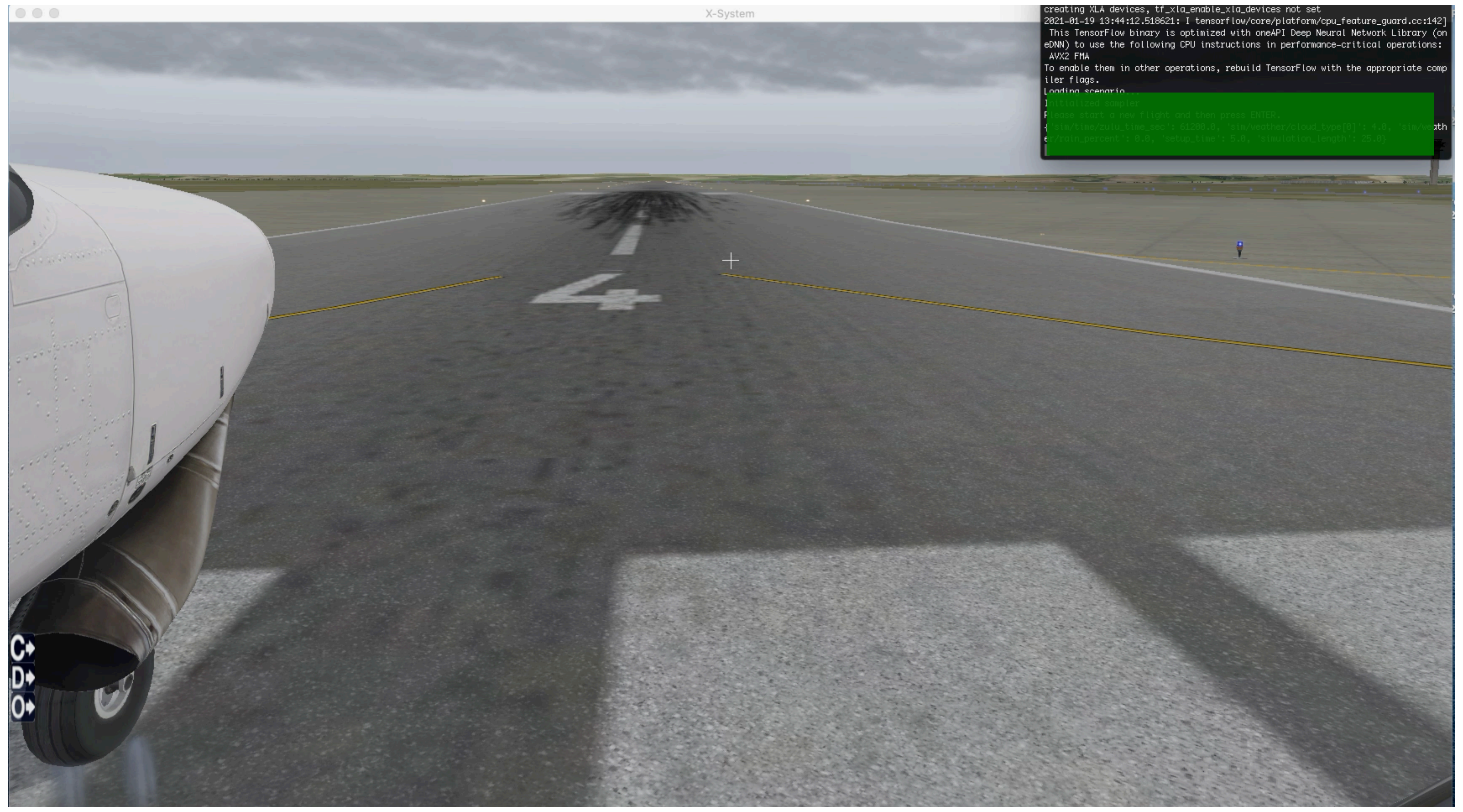
Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

Balancing between Correctness and Explainability



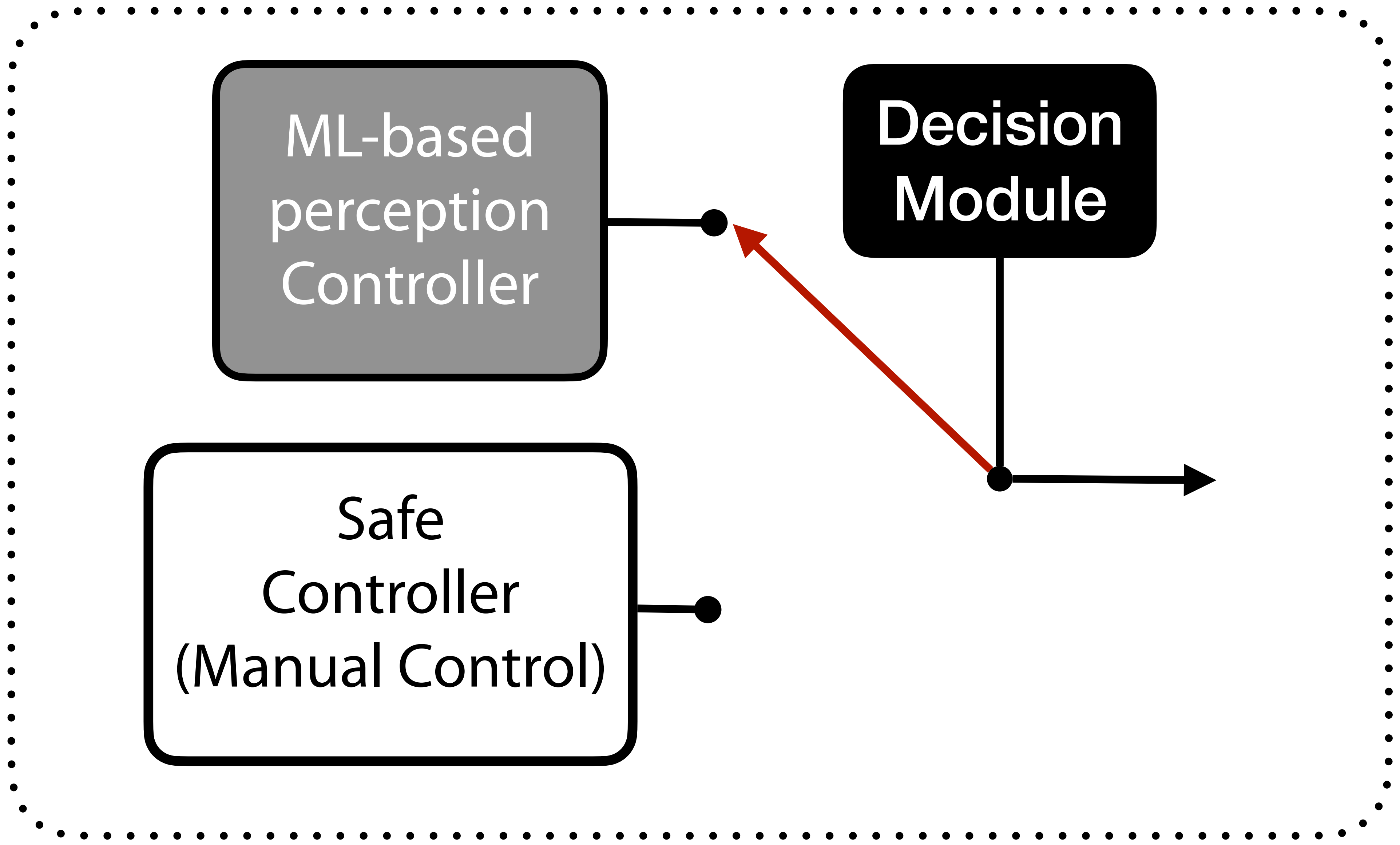
AutoTAXI



Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

Balancing between Correctness and Explainability

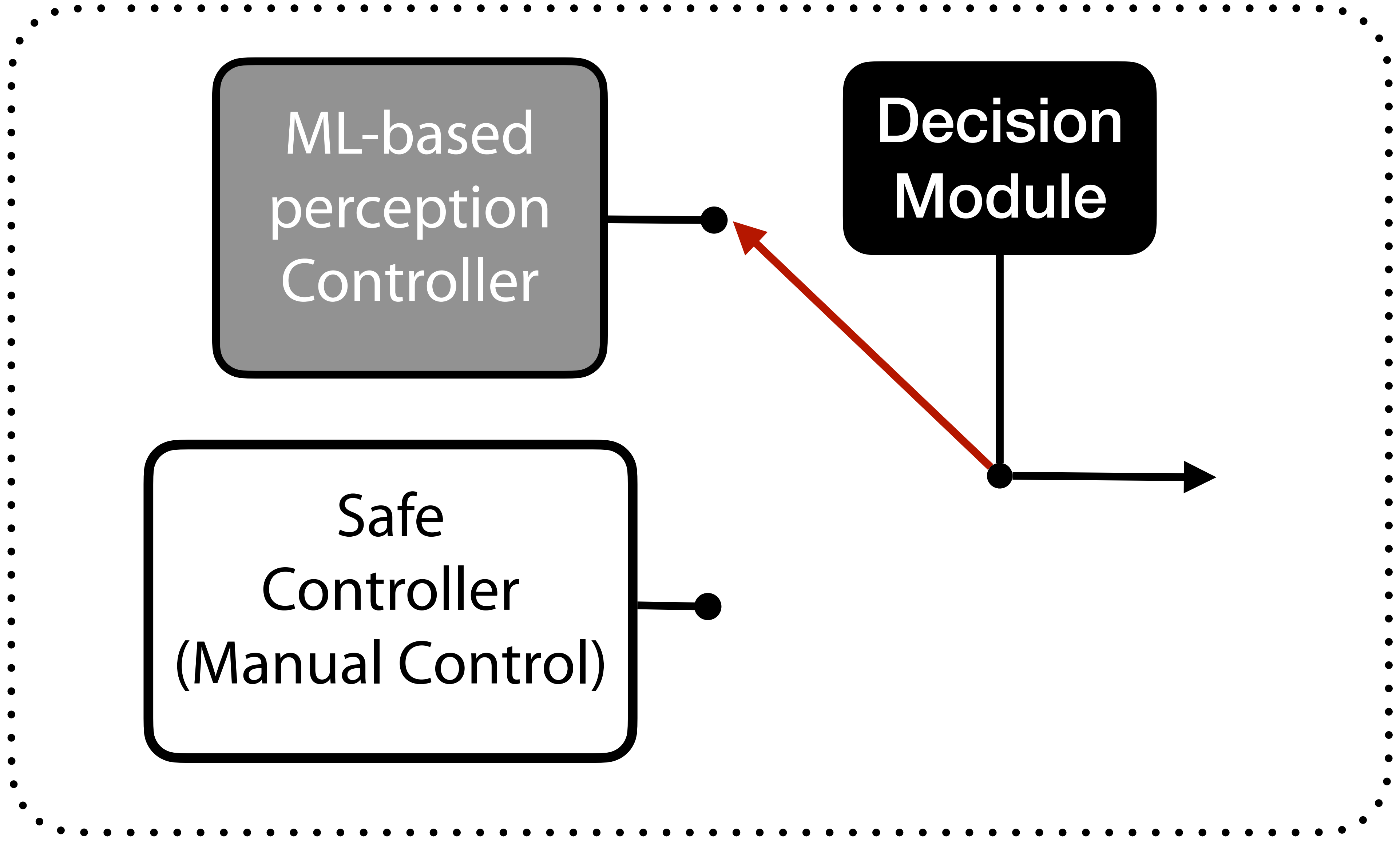


AutoTAXI

Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

Balancing between Correctness and Explainability



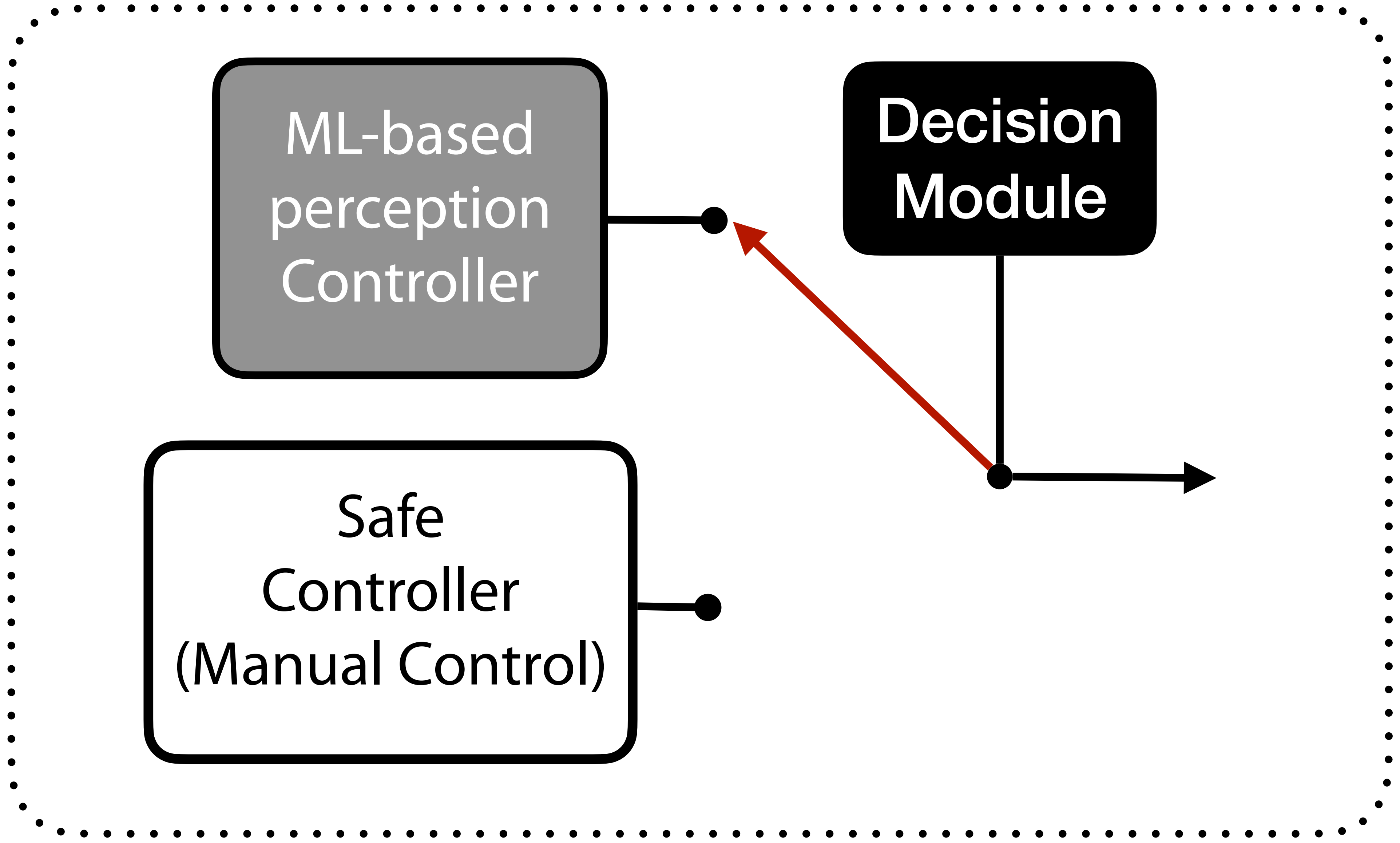
AutoTAXI

Class of interpretations: Decision diagrams

Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

Balancing between Correctness and Explainability



AutoTAXI

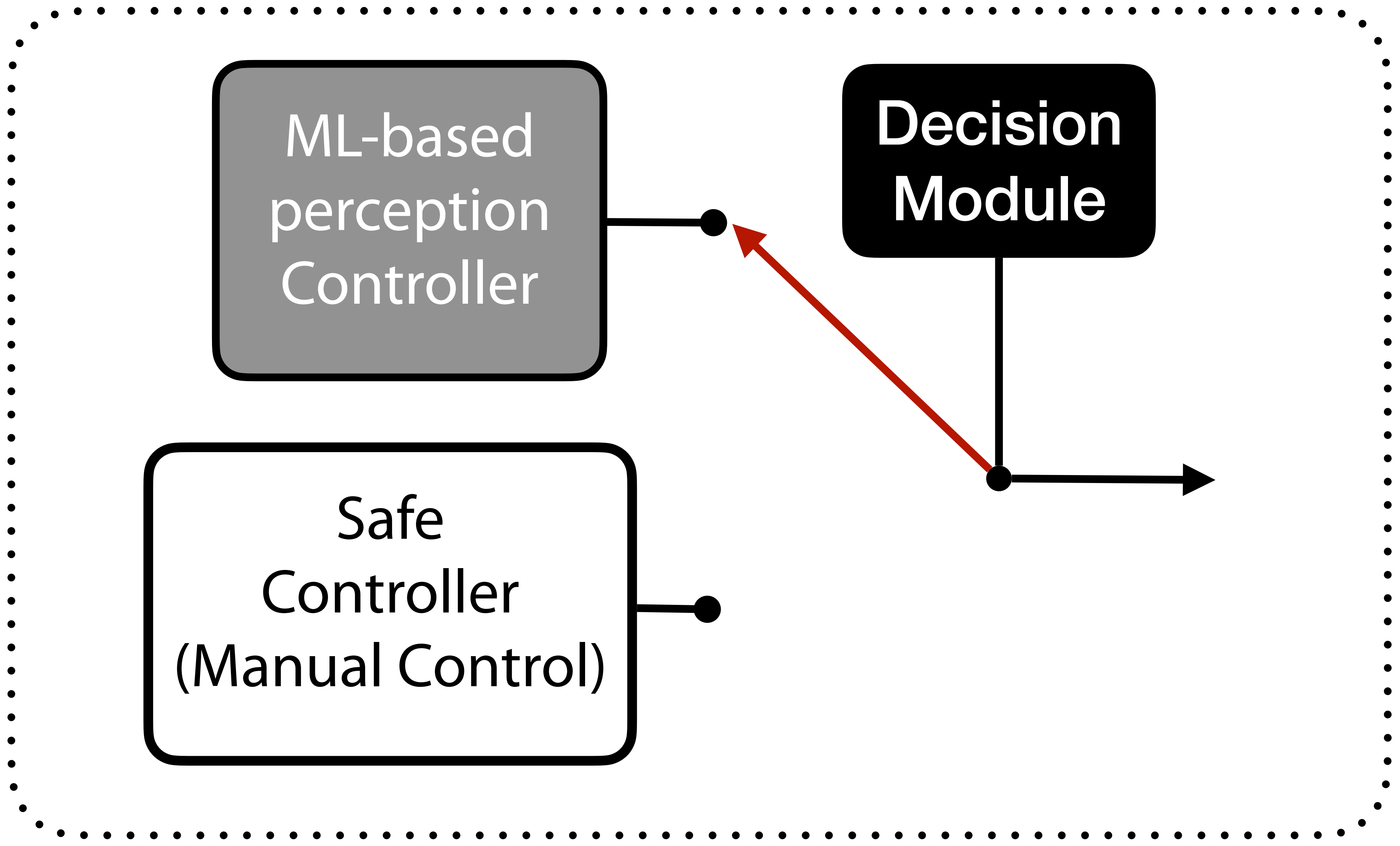
Class of interpretations: Decision diagrams

Predicates: Clouds (1)
Rain (1)
Initial position (2)
Time of day (4)

Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

Balancing between Correctness and Explainability



AutoTAXI

Class of interpretations: Decision diagrams

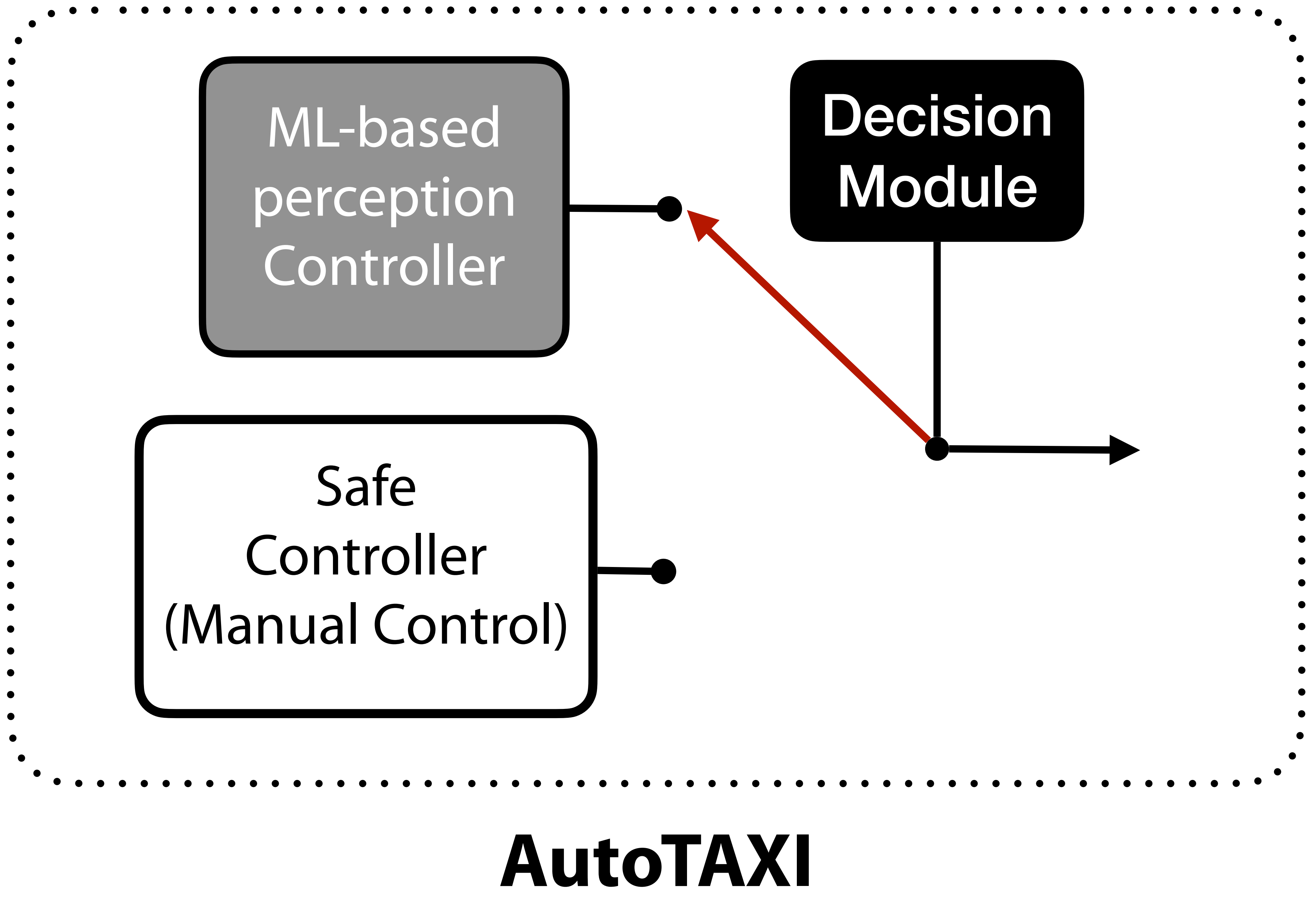
Predicates: Clouds (1)
Rain (1)
Initial position (2)
Time of day (4)

Explainability: score based on number of nodes and used predicates

Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

Balancing between Correctness and Explainability



Class of interpretations: Decision diagrams

Predicates: Clouds (1)
Rain (1)
Initial position (2)
Time of day (4)

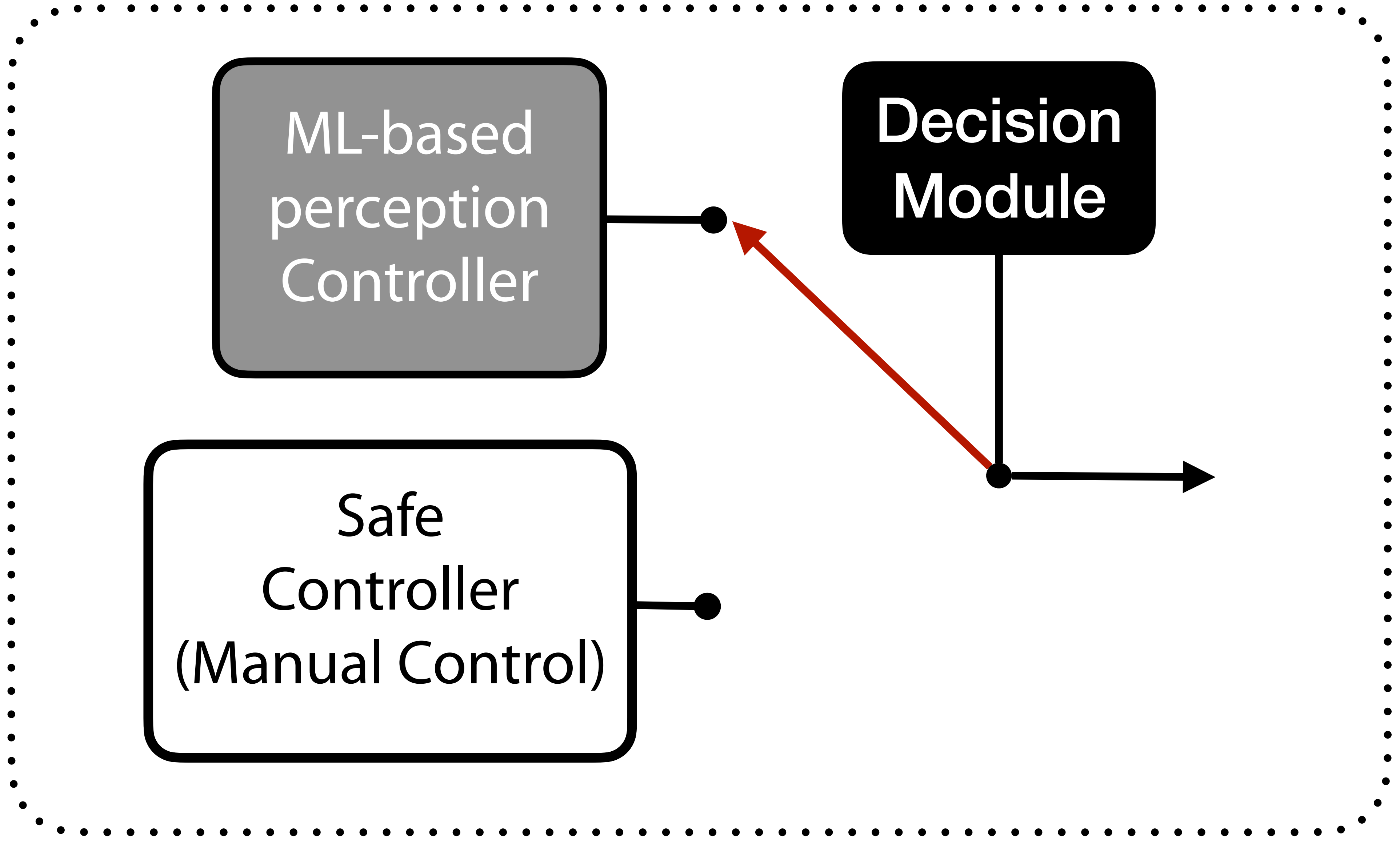
Explainability: score based on number of nodes and used predicates

Correctness: Prediction accuracy w.r.t. the given sample set

Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

Balancing between Correctness and Explainability



AutoTAXI

Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

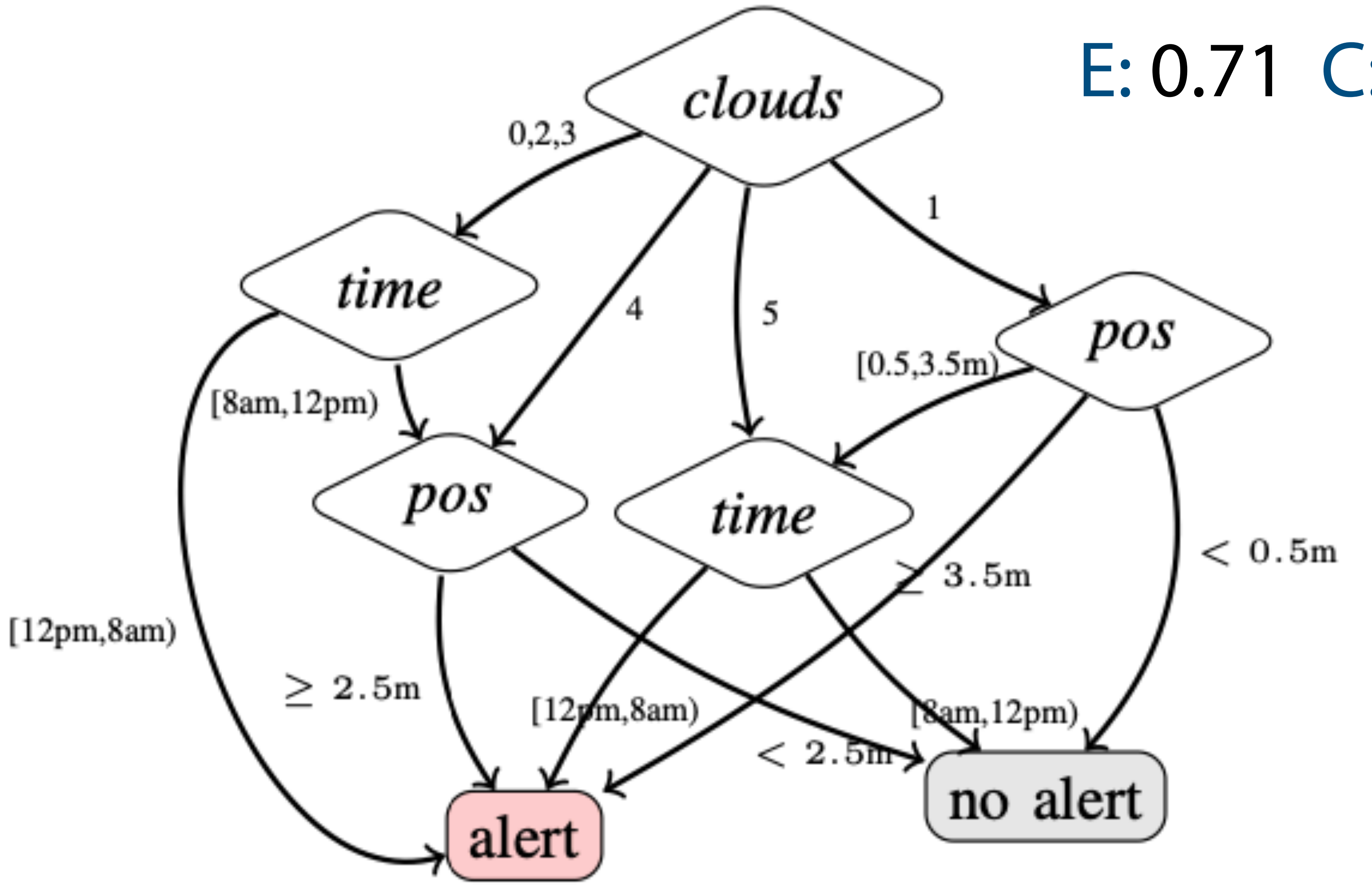
Class of interpretations: Decision diagrams

Predicates: Clouds (1)
Rain (1)
Initial position (2)
Time of day (4)

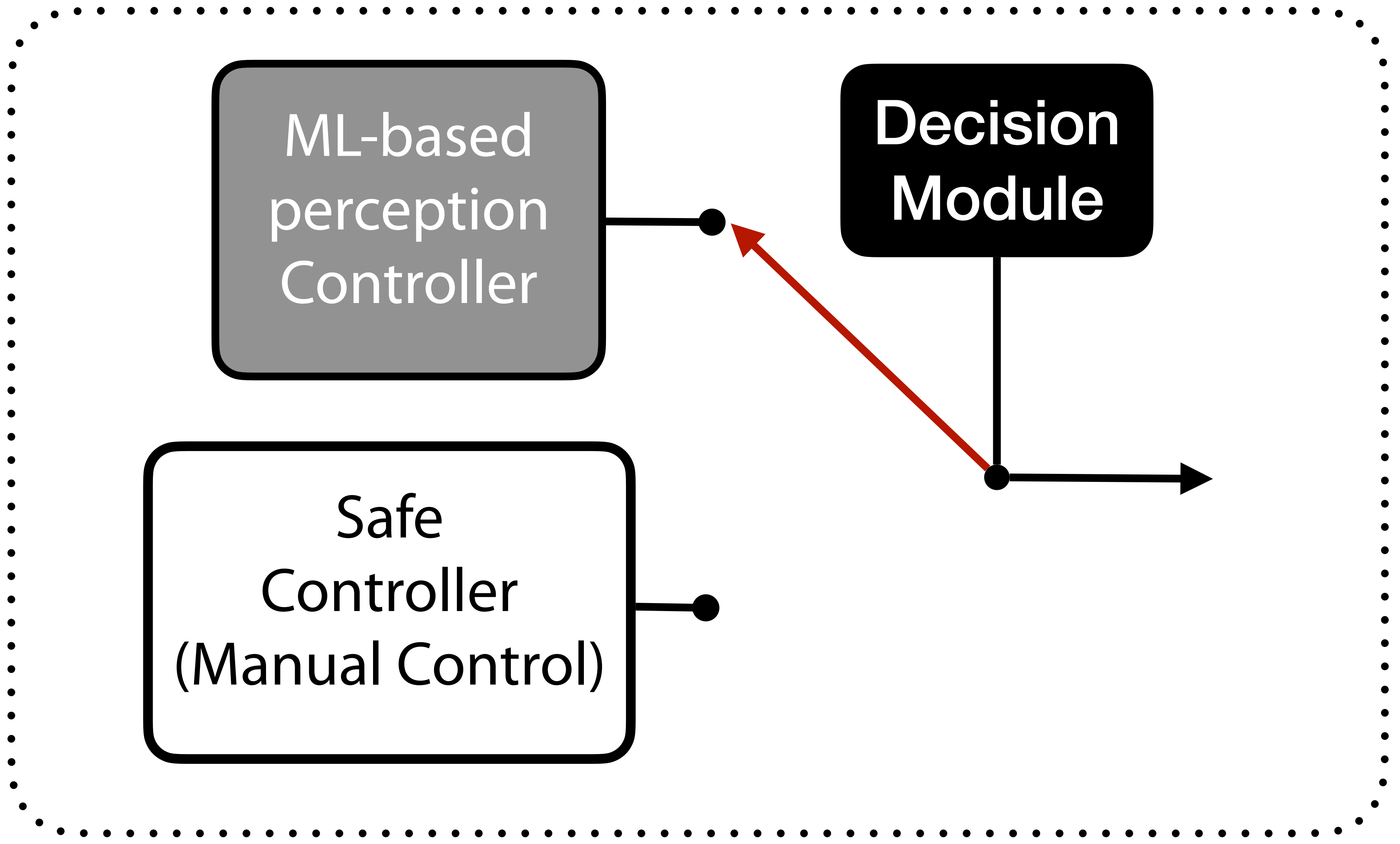
Explainability: score based on number of nodes and used predicates

Correctness: Prediction accuracy w.r.t. the given sample set

E: 0.71 C: 0.94



Balancing between Correctness and Explainability



AutoTAXI

Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

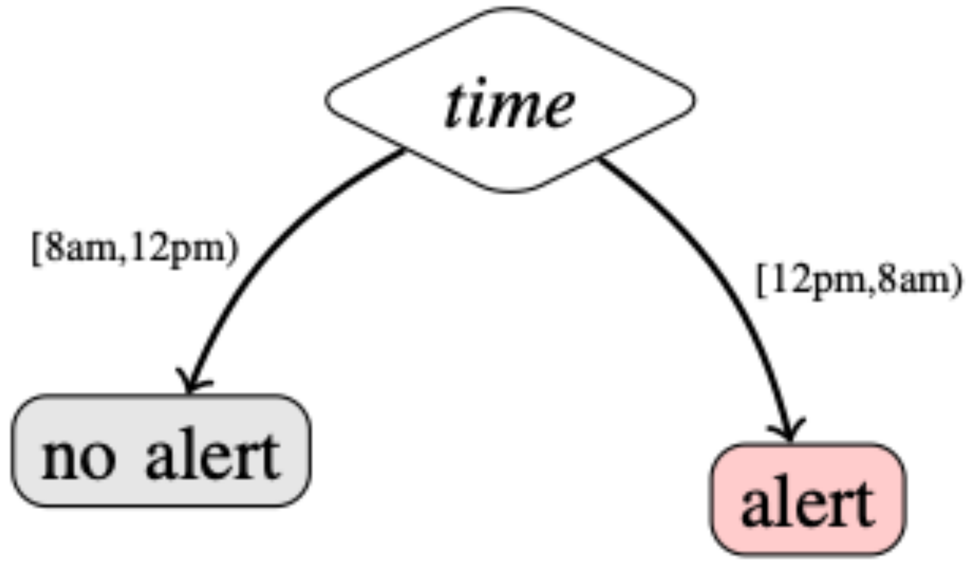
Class of interpretations: Decision diagrams

Predicates: Clouds (1)
Rain (1)
Initial position (2)
Time of day (4)

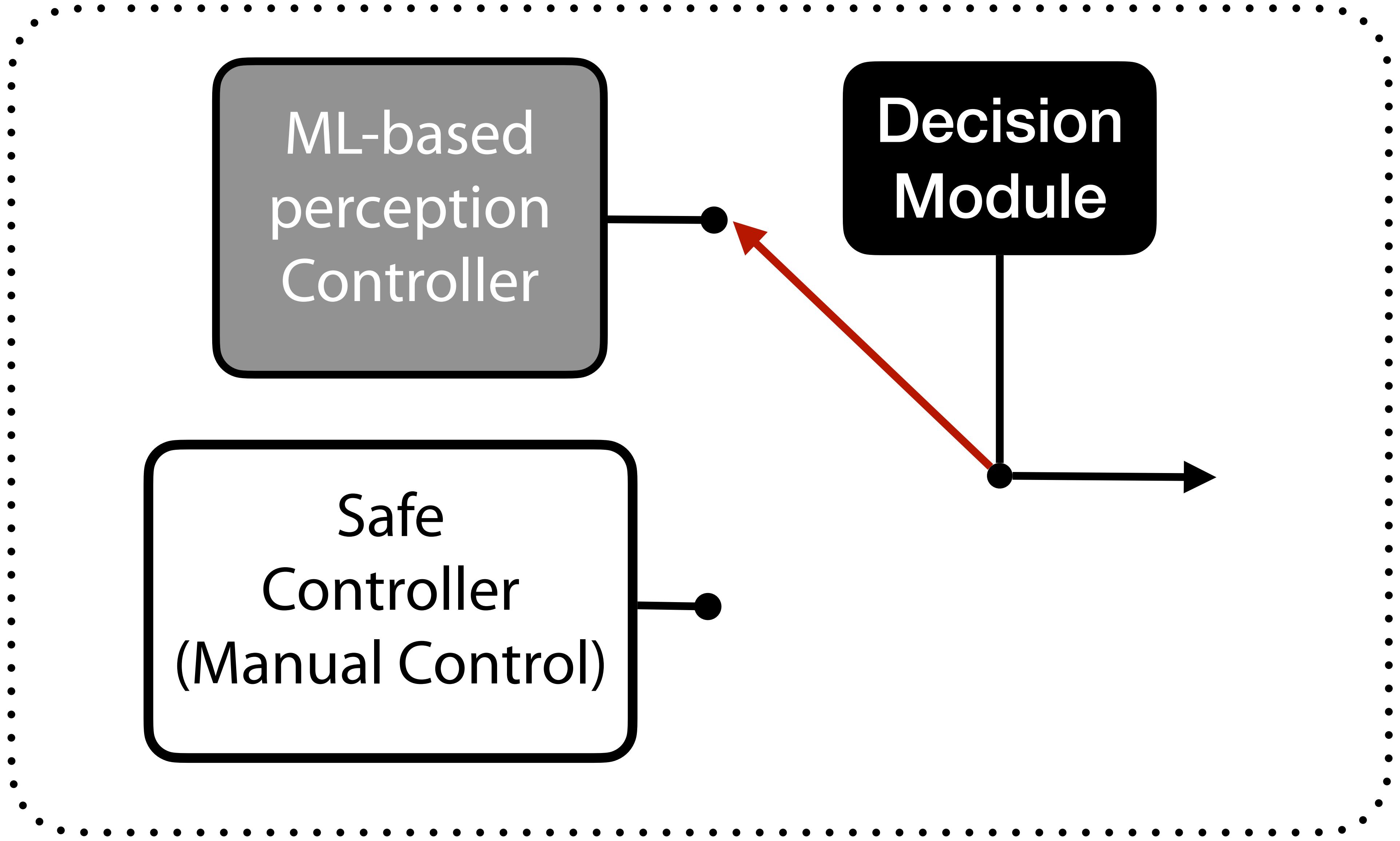
Explainability: score based on number of nodes and used predicates

Correctness: Prediction accuracy w.r.t. the given sample set

E: 0.95 C: 0.61



Balancing between Correctness and Explainability



AutoTAXI

Decision Module decides to trust ML-component based on:

- Weather conditions: clouds, rain
- Time of day
- Initial configuration: initial positioning, initial heading

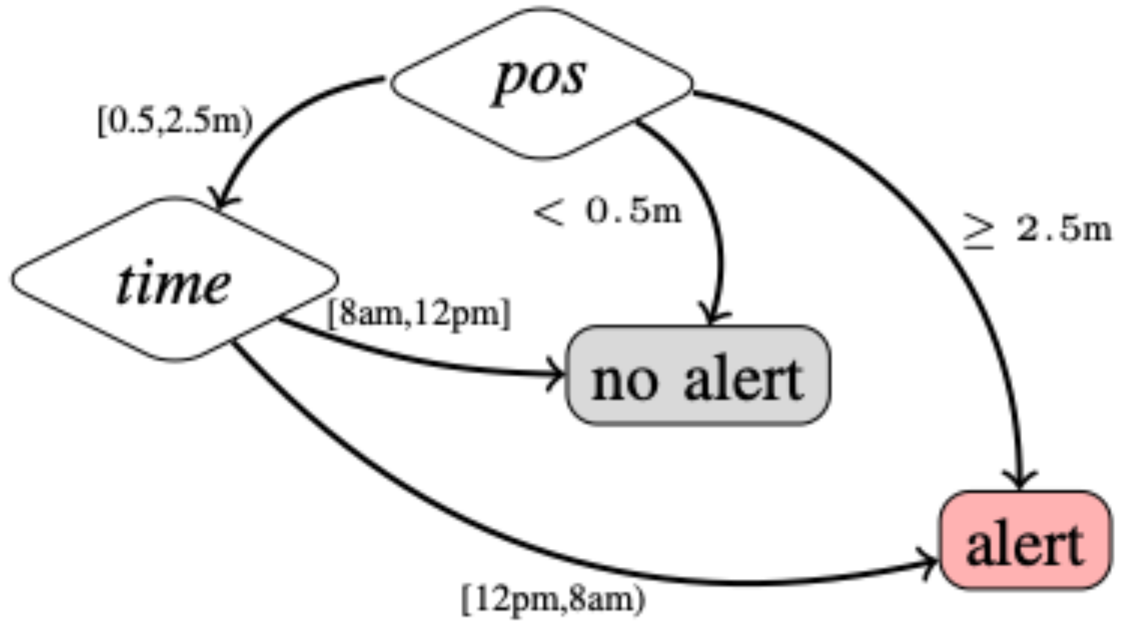
Class of interpretations: Decision diagrams

Predicates: Clouds (1)
Rain (1)
Initial position (2)
Time of day (4)

Explainability: score based on number of nodes and used predicates

Correctness: Prediction accuracy w.r.t. the given sample set

E: 0.89 C: 0.90



Pareto-optimal Interpretation Synthesis

Pareto-optimal
Synthesis

Pareto-optimal Interpretation Synthesis

Syntactic class of interpretations:
Decision trees, decision rules, …

$$\mathcal{E} : (\mathcal{I} \rightarrow \mathcal{O})$$

Pareto-optimal
Synthesis

Pareto-optimal Interpretation Synthesis

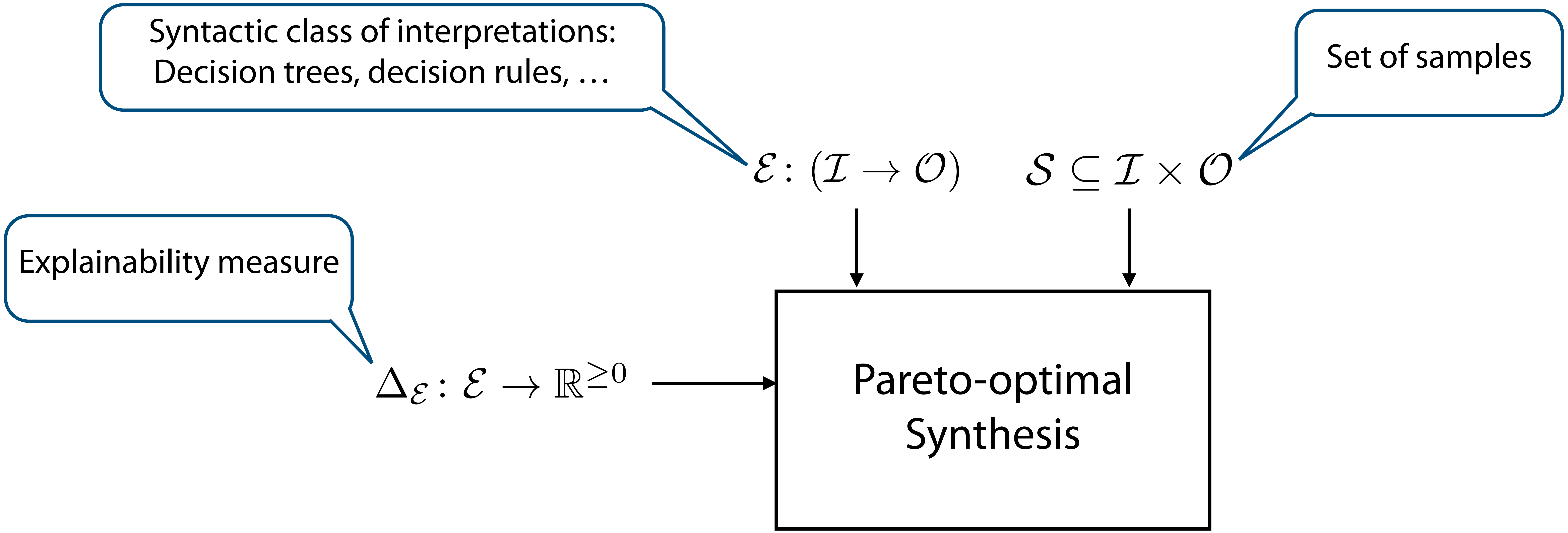
Syntactic class of interpretations:
Decision trees, decision rules, …

Set of samples

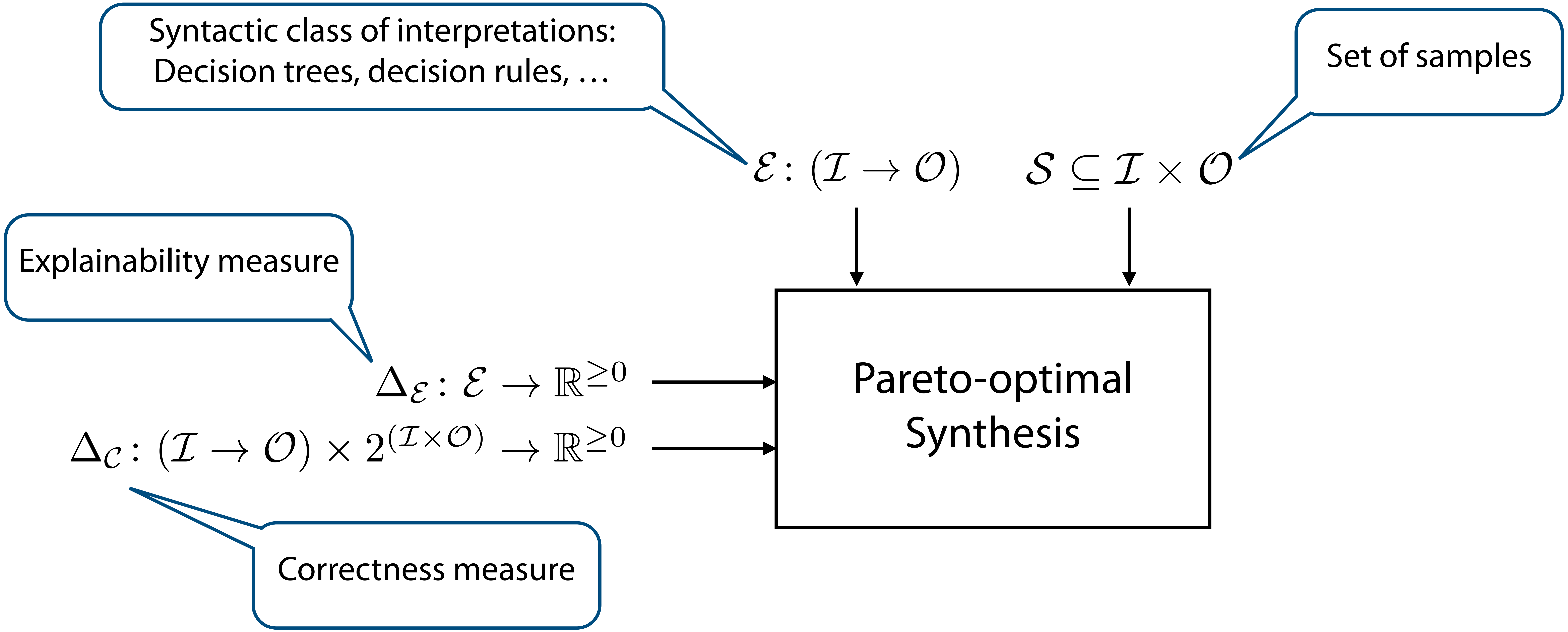
$\mathcal{E} : (\mathcal{I} \to \mathcal{O})$ $\qquad$ $\mathcal{S} \subseteq \mathcal{I} \times \mathcal{O}$

Pareto-optimal
Synthesis

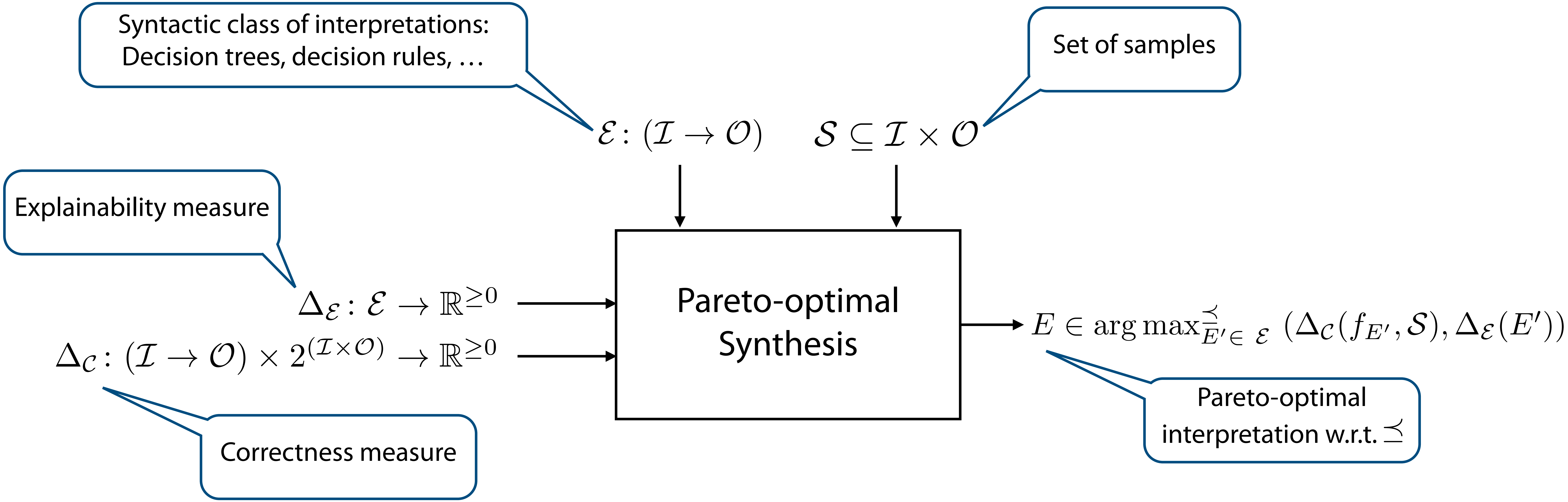
Pareto-optimal Interpretation Synthesis



Pareto-optimal Interpretation Synthesis



Pareto-optimal Interpretation Synthesis



Synthesis via weighted MaxSAT

Synthesis via weighted MaxSAT

Recap weighted MaxSAT

Given a boolean formula $\varphi = \bigwedge_{i=1}^{m} C_i$ and a weight function $w\colon \{C_1, \dots C_m\} \to \mathbb{R}^{\geq 0}$, the weighted MaxSAT problem is to find an assignment σ which maximizes:

$$\sum_{\{C_i \mid \ \sigma \models C_i\}} w(C_i)$$

Synthesis via weighted MaxSAT

Recap weighted MaxSAT

Given a boolean formula $\varphi = \bigwedge_{i=1}^{m} C_i$ and a weight function $w \colon \{C_1, \dots C_m\} \to \mathbb{R}^{\geq 0}$, the weighted MaxSAT problem is to find an assignment σ which maximizes:

$$\sum_{\{C_i \mid \sigma \models C_i\}} w(C_i)$$

Encoding of interpretation synthesis in weighted MaxSat:

Synthesis via weighted MaxSAT

Recap weighted MaxSAT

Given a boolean formula $\varphi = \bigwedge_{i=1}^{m} C_i$ and a weight function $w\colon \{C_1, \ldots C_m\} \to \mathbb{R}^{\geq 0}$, the weighted MaxSAT problem is to find an assignment σ which maximizes:

$$\sum_{\{C_i \mid \ \sigma \models C_i\}} w(C_i)$$

Encoding of interpretation synthesis in weighted MaxSat:

$$\phi_{\mathcal{E}} \wedge \phi_{\mathcal{S}} \wedge \phi_{\Delta_{\mathcal{C}}} \wedge \phi_{\Delta_{\mathcal{E}}}$$

Synthesis via weighted MaxSAT

Recap weighted MaxSAT

Given a boolean formula $\varphi = \bigwedge_{i=1}^{m} C_i$ and a weight function $w\colon \{C_1, \ldots C_m\} \to \mathbb{R}^{\geq 0}$, the weighted MaxSAT problem is to find an assignment σ which maximizes:

$$\sum_{\{C_i \mid \ \sigma \models C_i\}} w(C_i)$$

Encoding of interpretation synthesis in weighted MaxSat:

$$\phi_{\mathcal{E}} \wedge \phi_{\mathcal{S}} \wedge \phi_{\Delta_{\mathcal{C}}} \wedge \phi_{\Delta_{\mathcal{E}}}$$

Syntactic class:

- Symbolic encoding of decision trees, diagrams,…

Synthesis via weighted MaxSAT

Recap weighted MaxSAT

Given a boolean formula $\varphi = \bigwedge_{i=1}^{m} C_i$ and a weight function $w\colon \{C_1, \ldots C_m\} \to \mathbb{R}^{\geq 0}$, the weighted MaxSAT problem is to find an assignment σ which maximizes:

$$\sum_{\{C_i \mid \ \sigma \models C_i\}} w(C_i)$$

Encoding of interpretation synthesis in weighted MaxSat:

$$\phi_{\mathcal{E}} \wedge \phi_{\mathcal{S}} \wedge \phi_{\Delta_{\mathcal{C}}} \wedge \phi_{\Delta_{\mathcal{E}}}$$

Syntactic class:

- Symbolic encoding of decision trees, diagrams,…

Samples:

- Uses variables $m_{(i,o)}$ for each sample (i, o)
- $m_{(i,o)}$ is true iff interpretation satisfying $\phi_{\mathcal{E}}$ produces o on i

Synthesis via weighted MaxSAT

Recap weighted MaxSAT

Given a boolean formula $\varphi = \bigwedge_{i=1}^{m} C_i$ and a weight function $w \colon \{C_1, \ldots C_m\} \to \mathbb{R}^{\geq 0}$, the weighted MaxSAT problem is to find an assignment σ which maximizes:

$$\sum_{\{C_i \mid \sigma \models C_i\}} w(C_i)$$

Encoding of interpretation synthesis in weighted MaxSat:

$$\phi_{\mathcal{E}} \wedge \phi_{\mathcal{S}} \wedge \phi_{\Delta_{\mathcal{C}}} \wedge \phi_{\Delta_{\mathcal{E}}}$$

Syntactic class:

- Symbolic encoding of decision trees, diagrams,…

Samples:

- Uses variables $m_{(i,o)}$ for each sample (i, o)
- $m_{(i,o)}$ is true iff interpretation satisfying $\phi_{\mathcal{E}}$ produces o on i

Correctness measure:

- Add unit clause for each sample $m_{(i,o)}$

Synthesis via weighted MaxSAT

Recap weighted MaxSAT

Given a boolean formula $\varphi = \bigwedge_{i=1}^{m} C_i$ and a weight function $w \colon \{C_1, \ldots C_m\} \to \mathbb{R}^{\geq 0}$, the weighted MaxSAT problem is to find an assignment σ which maximizes:

$$\sum_{\{C_i \mid \ \sigma \models C_i\}} w(C_i)$$

Encoding of interpretation synthesis in weighted MaxSat:

$$\phi_{\mathcal{E}} \wedge \phi_{\mathcal{S}} \wedge \phi_{\Delta_{\mathcal{C}}} \wedge \phi_{\Delta_{\mathcal{E}}}$$

Syntactic class:

- Symbolic encoding of decision trees, diagrams,…

Samples:

- Uses variables $m_{(i,o)}$ for each sample (i,o)
- $m_{(i,o)}$ is true iff interpretation satisfying $\phi_{\mathcal{E}}$ produces o on i

Correctness measure:

- Add unit clause for each sample $m_{(i,o)}$

Explainability measure:

- Add unit clause for each syntactic structure: e.g. predicate used, node used, …

Synthesis via weighted MaxSAT

Recap weighted MaxSAT

Given a boolean formula $\varphi = \bigwedge_{i=1}^{m} C_i$ and a weight function $w\colon \{C_1, \ldots C_m\} \to \mathbb{R}^{\geq 0}$,
the weighted MaxSAT problem is to find an assignment σ which maximizes:

$$\sum_{\{C_i \mid \sigma \models C_i\}} w(C_i)$$

Encoding of interpretation synthesis in weighted MaxSat:

$$\phi_{\mathcal{E}} \wedge \phi_{\mathcal{S}} \wedge \phi_{\Delta_{\mathcal{C}}} \wedge \phi_{\Delta_{\mathcal{E}}}$$

Syntactic class:

- Symbolic encoding of decision trees, diagrams,…

Samples:

- Uses variables $m_{(i,o)}$ for each sample (i, o)
- $m_{(i,o)}$ is true iff interpretation satisfying $\phi_{\mathcal{E}}$ produces o on i

Correctness measure:

- Add unit clause for each sample $m_{(i,o)}$

Explainability measure:

- Add unit clause for each syntactic structure: e.g. predicate used, node used, …

Assign appropriate weights to unit clause

Synthesis via weighted MaxSAT

Recap weighted MaxSAT

Given a boolean formula $\varphi = \bigwedge_{i=1}^{m} C_i$ and a weight function $w\colon \{C_1, \ldots C_m\} \to \mathbb{R}^{\geq 0}$, the weighted MaxSAT problem is to find an assignment σ which maximizes:

$$\sum_{\{C_i | \ \sigma \models C_i\}} w(C_i)$$

Encoding of interpretation synthesis in weighted MaxSat:

$$\phi_{\mathcal{E}} \wedge \phi_{\mathcal{S}} \wedge \phi_{\Delta_{\mathcal{C}}} \wedge \phi_{\Delta_{\mathcal{E}}}$$

Syntactic class:

- Symbolic encoding of decision trees, diagrams,…

Samples:

- Uses variables $m_{(i,o)}$ for each sample (i, o)
- $m_{(i,o)}$ is true iff interpretation satisfying $\phi_{\mathcal{E}}$ produces o on i

Correctness measure:

- Add unit clause for each sample $m_{(i,o)}$

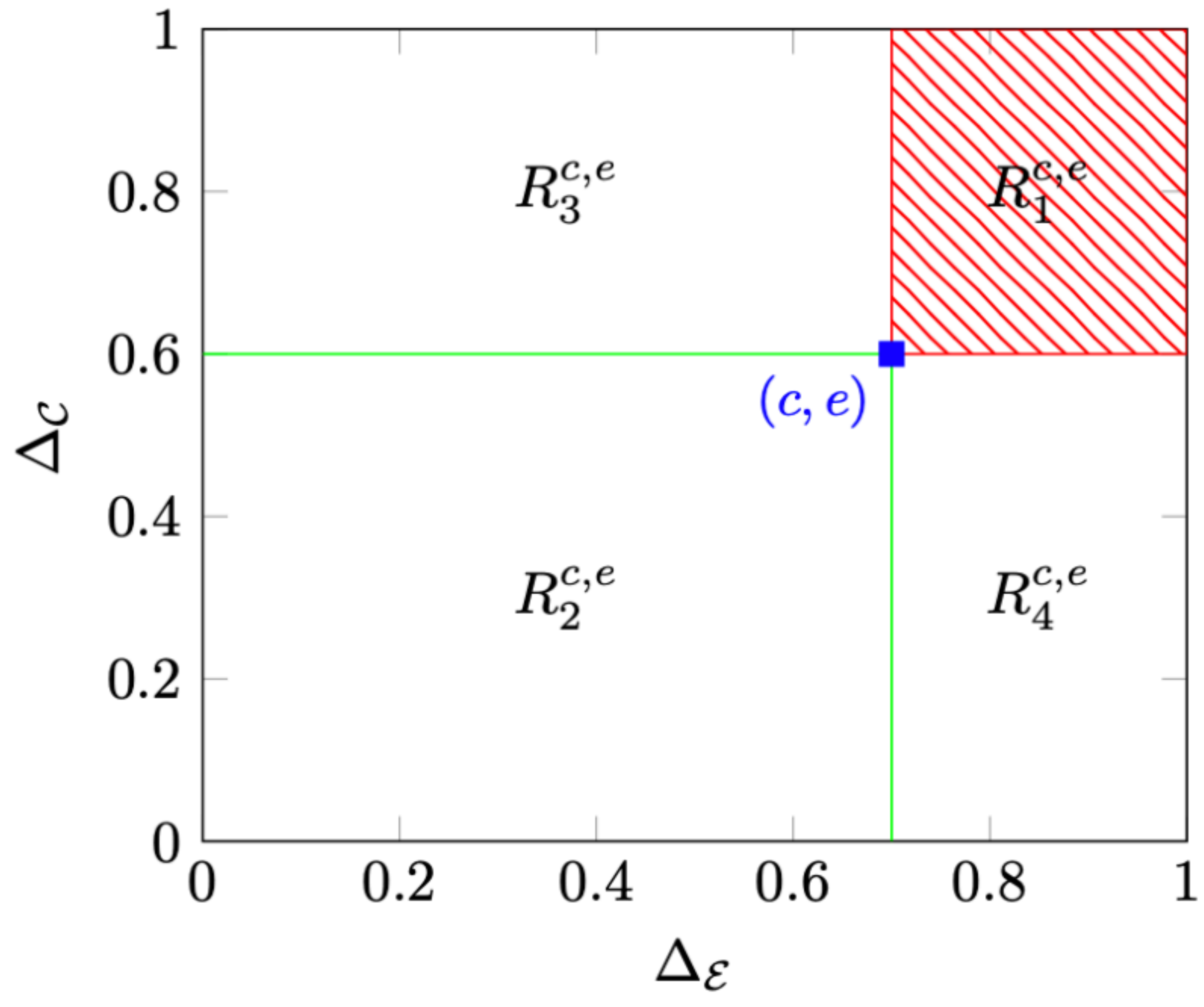
Explainability measure:

- Add unit clause for each syntactic structure: e.g. predicate used, node used, …

Assign appropriate weights to unit clause

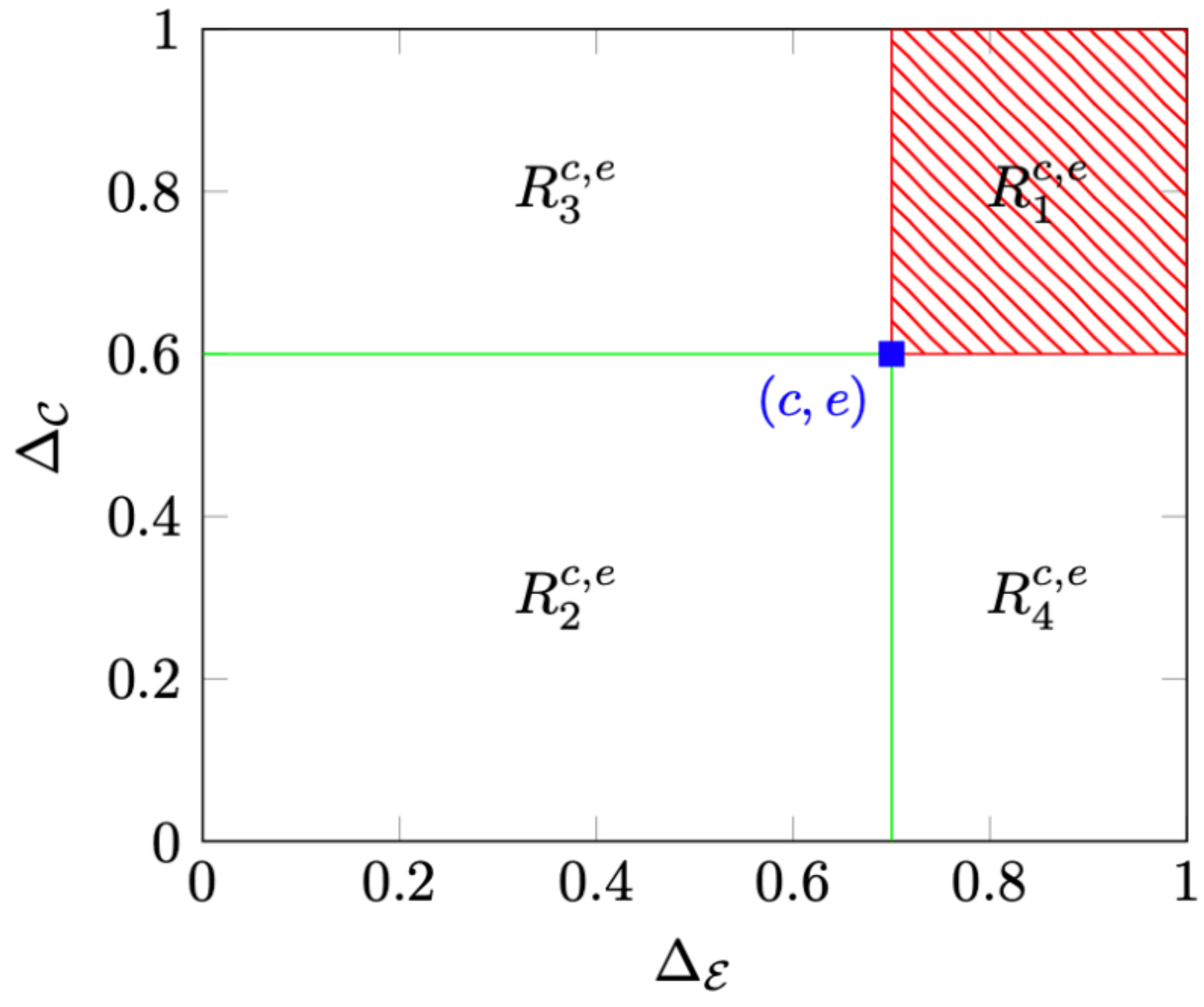
Outcome: Pareto-optimal interpretation with maximum sum of correctness and explainability score

Exploration of Pareto-Optimal Space



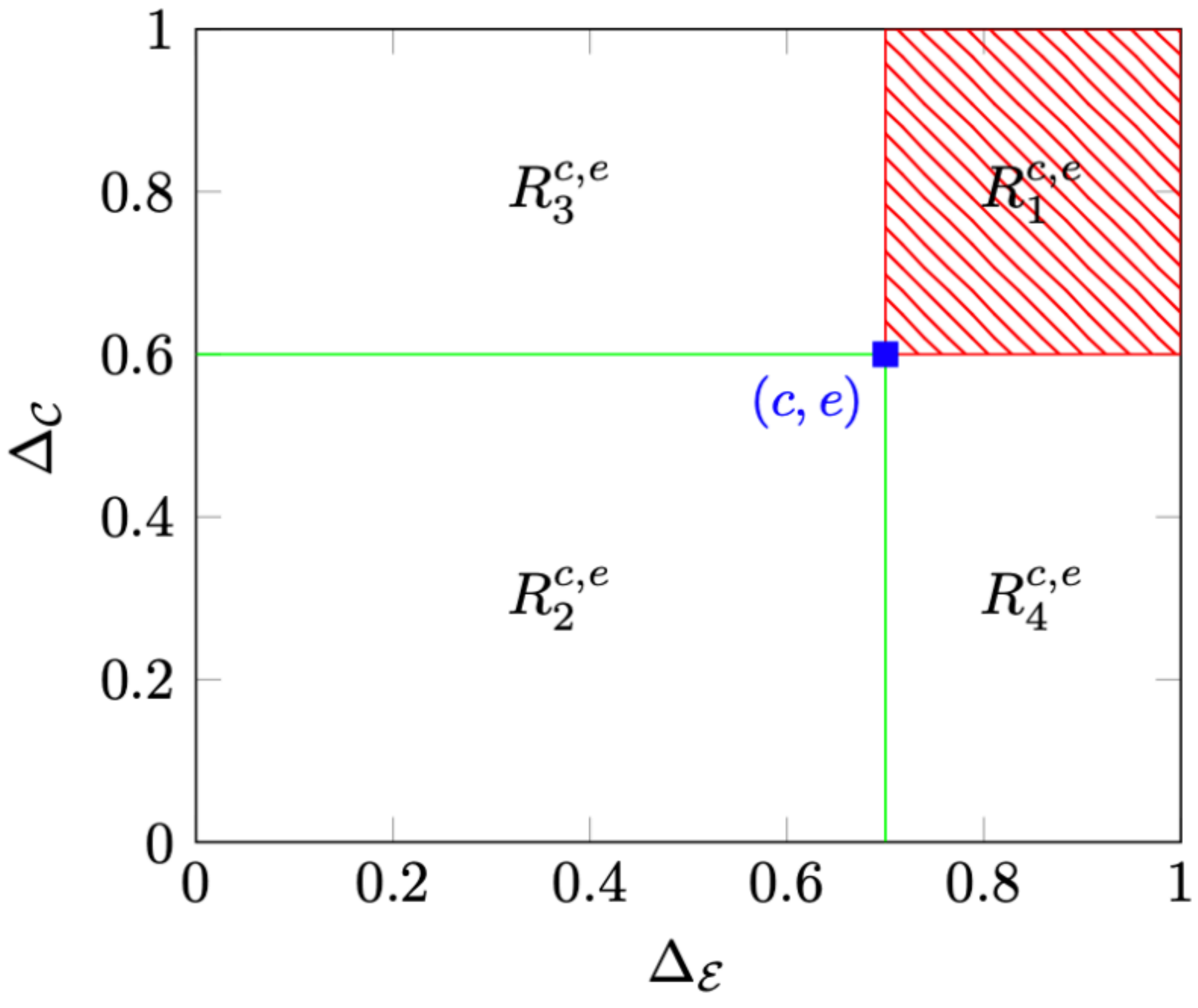
Exploration of Pareto-Optimal Space

- Synthesize initial Pareto-optimal interpretation



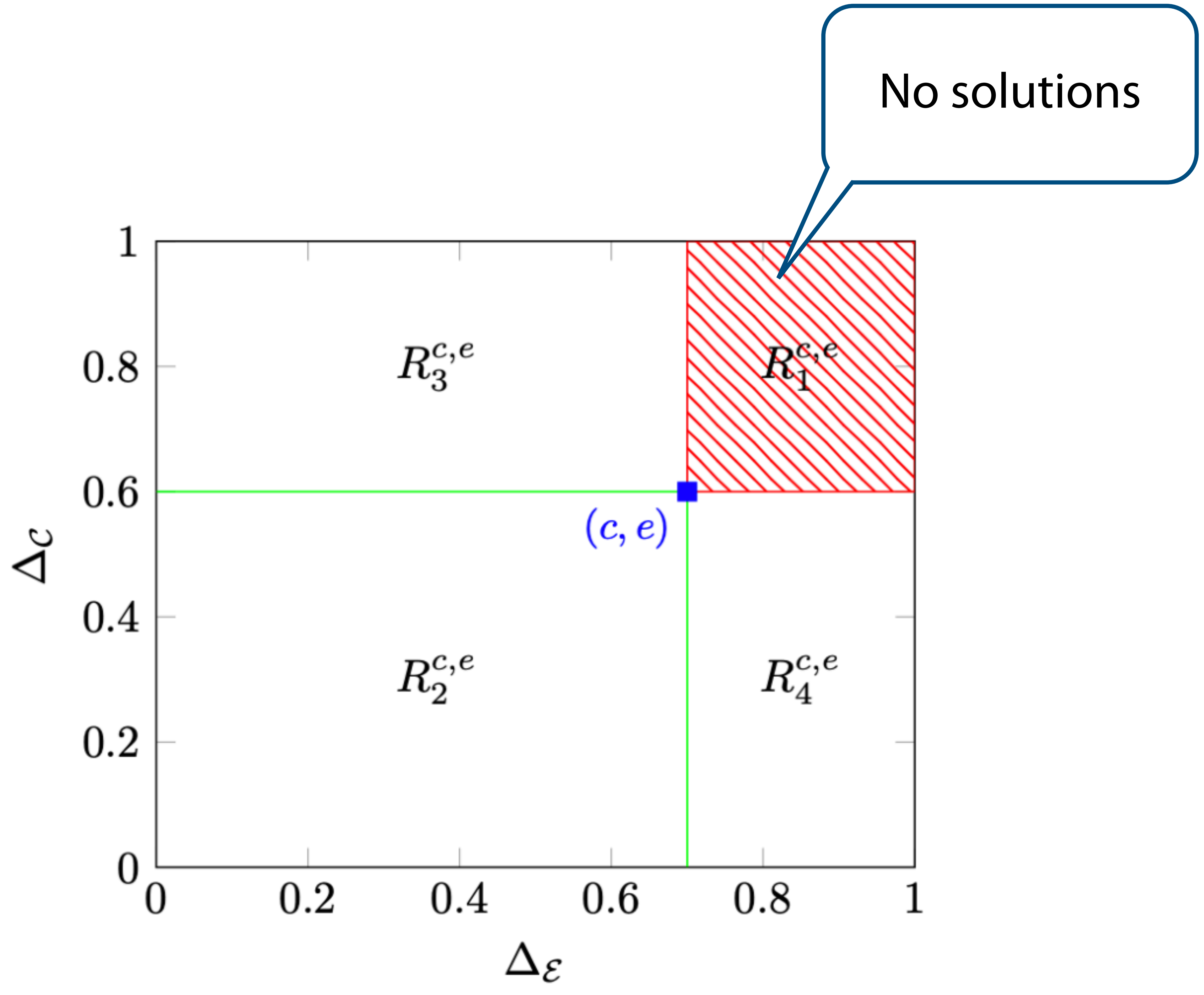
Exploration of Pareto-Optimal Space

- Synthesize initial Pareto-optimal interpretation
- Every PO-interpretation splits space into four regions



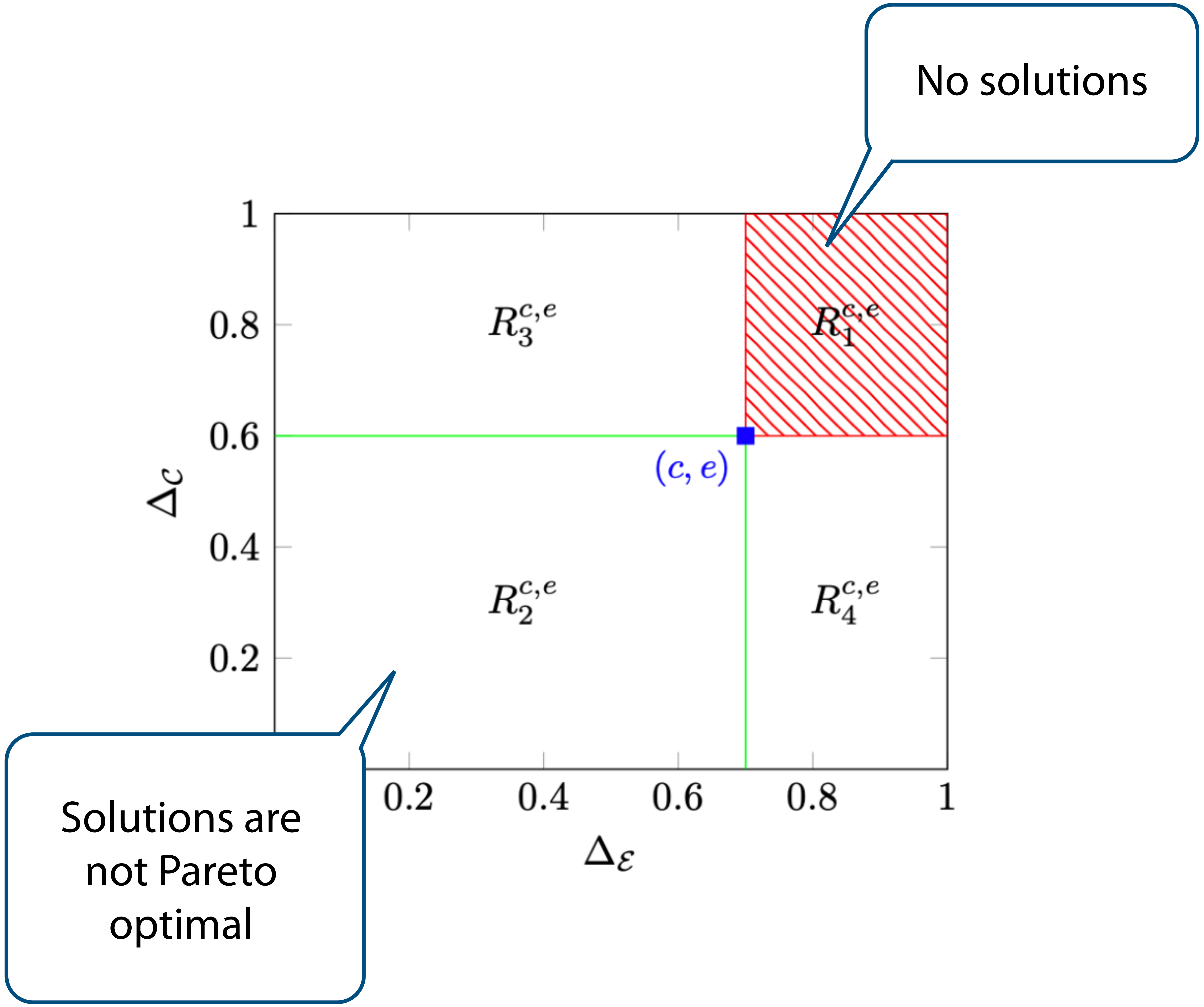
Exploration of Pareto-Optimal Space

- Synthesize initial Pareto-optimal interpretation
- Every PO-interpretation splits space into four regions



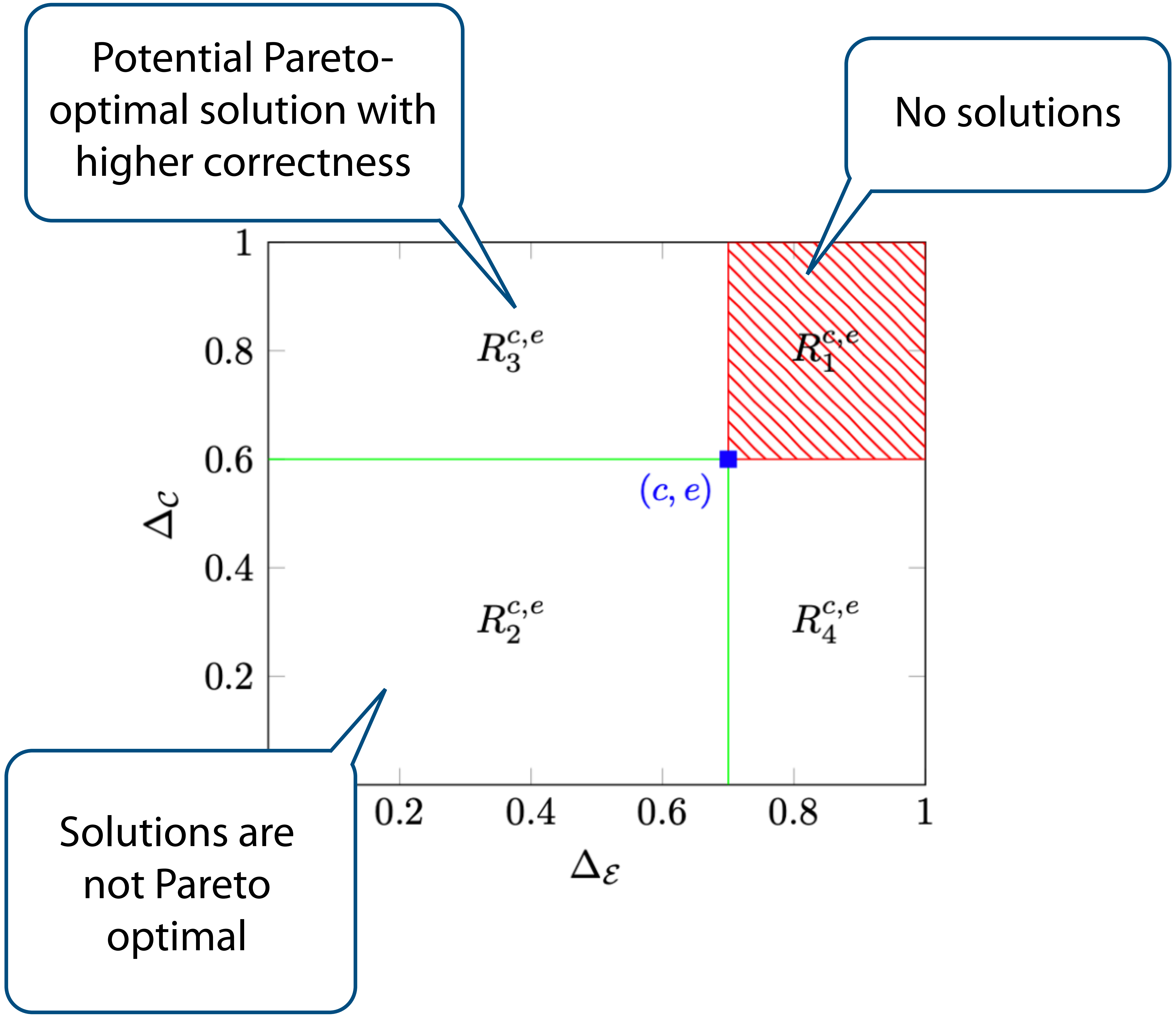
Exploration of Pareto-Optimal Space

- Synthesize initial Pareto-optimal interpretation
- Every PO-interpretation splits space into four regions



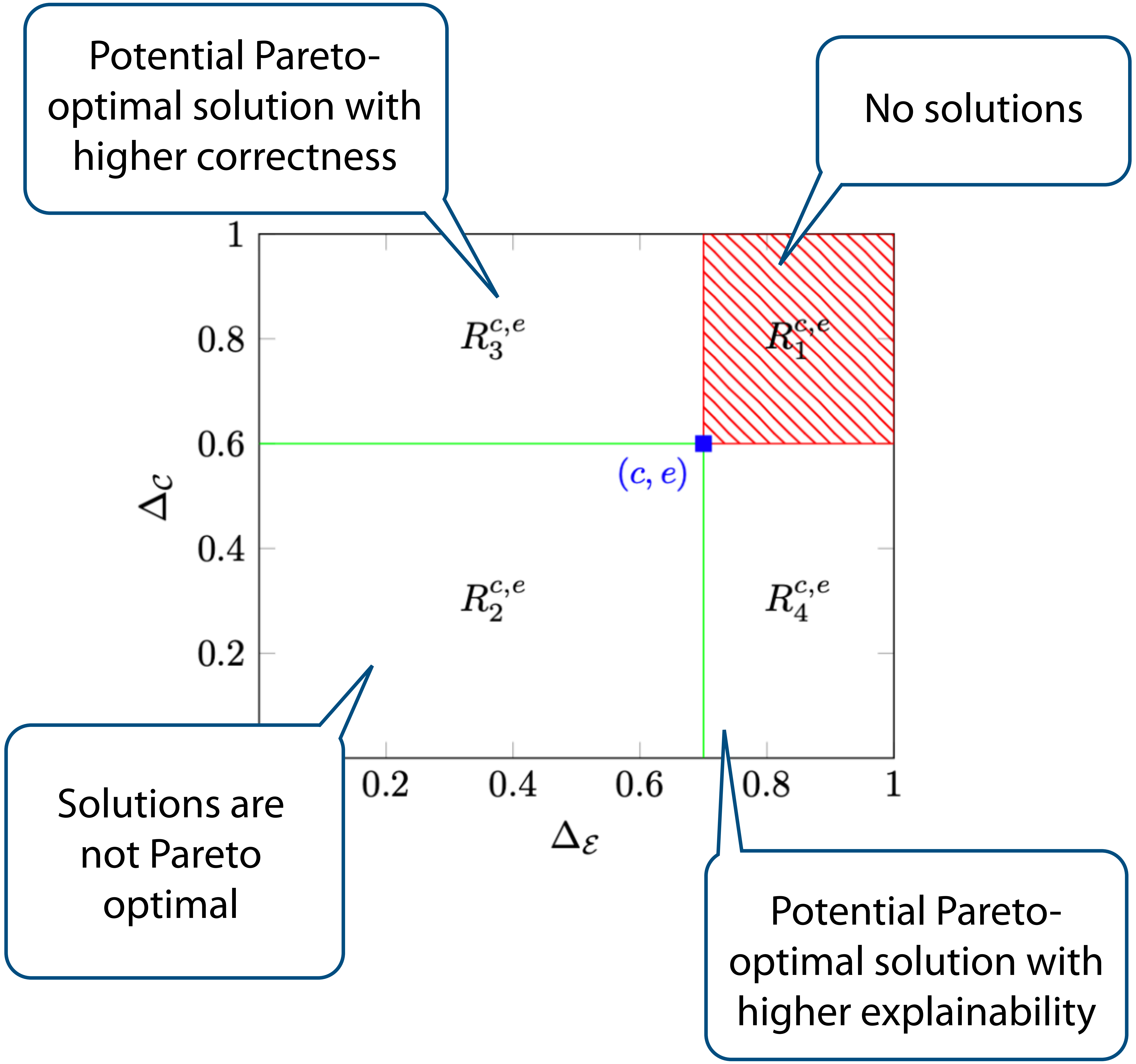
Exploration of Pareto-Optimal Space

- Synthesize initial Pareto-optimal interpretation
- Every PO-interpretation splits space into four regions



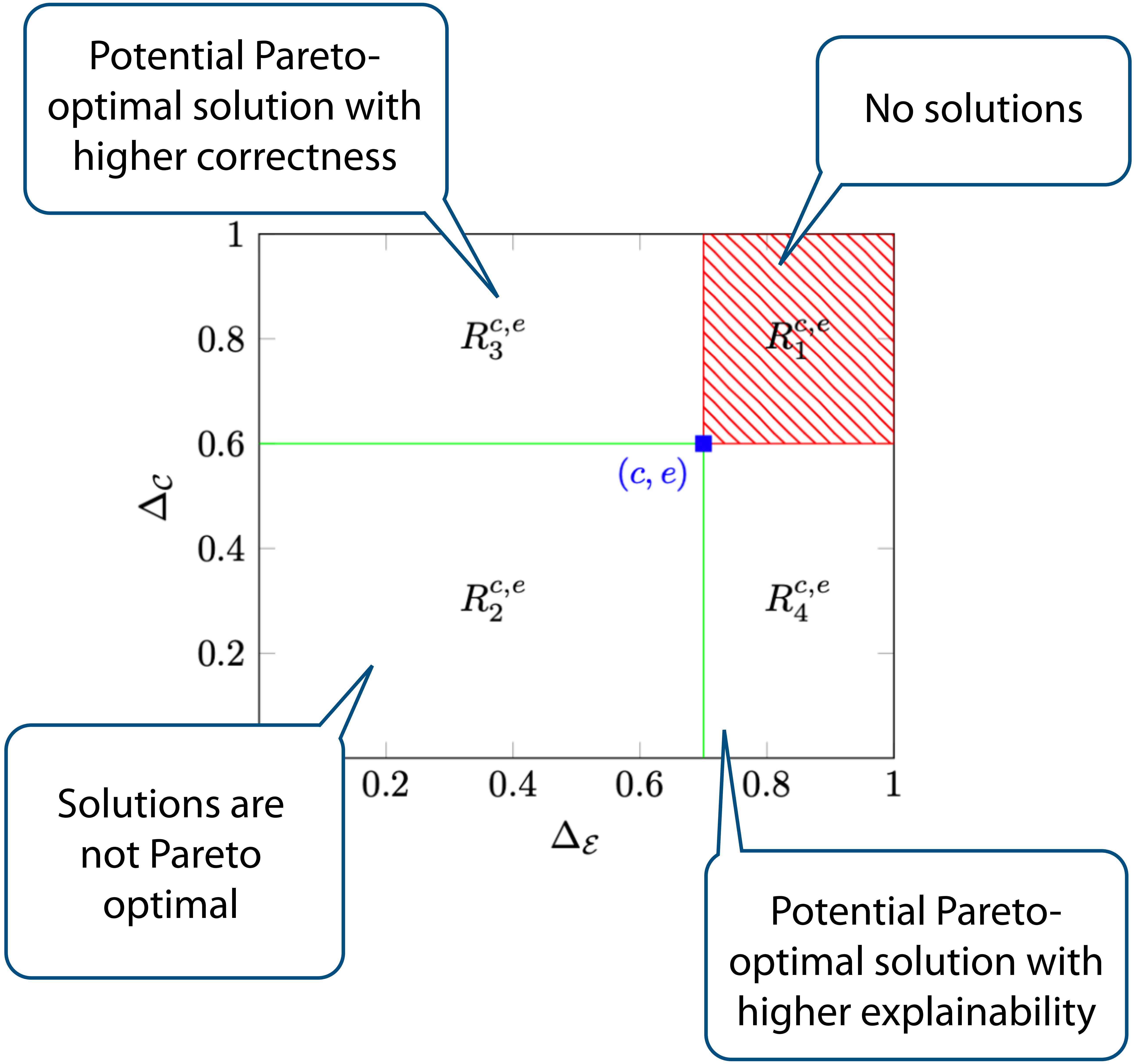
Exploration of Pareto-Optimal Space

- Synthesize initial Pareto-optimal interpretation
- Every PO-interpretation splits space into four regions



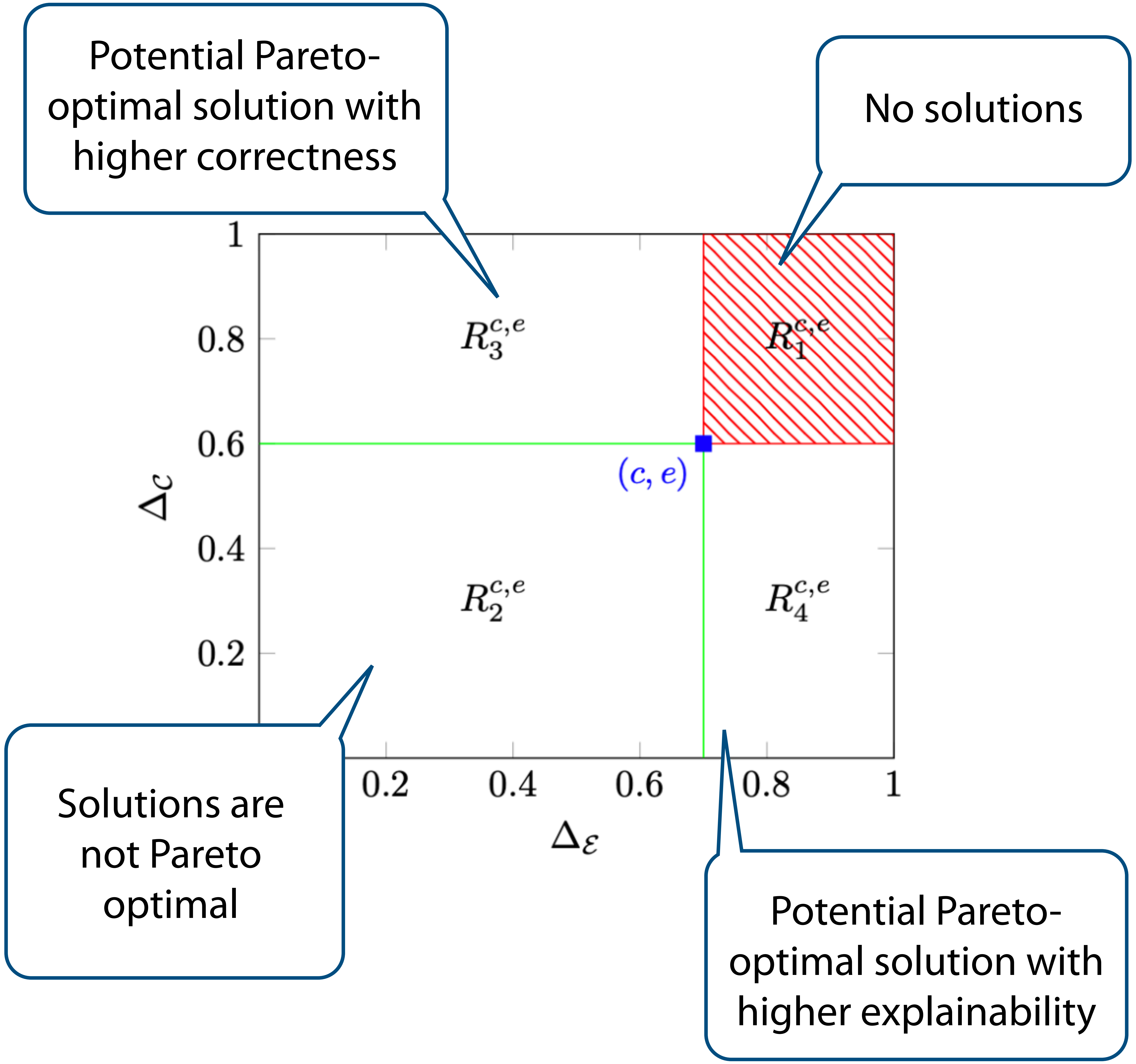
Exploration of Pareto-Optimal Space

- Synthesize initial Pareto-optimal interpretation
- Every PO-interpretation splits space into four regions
- Continue search in regions 3 and 4



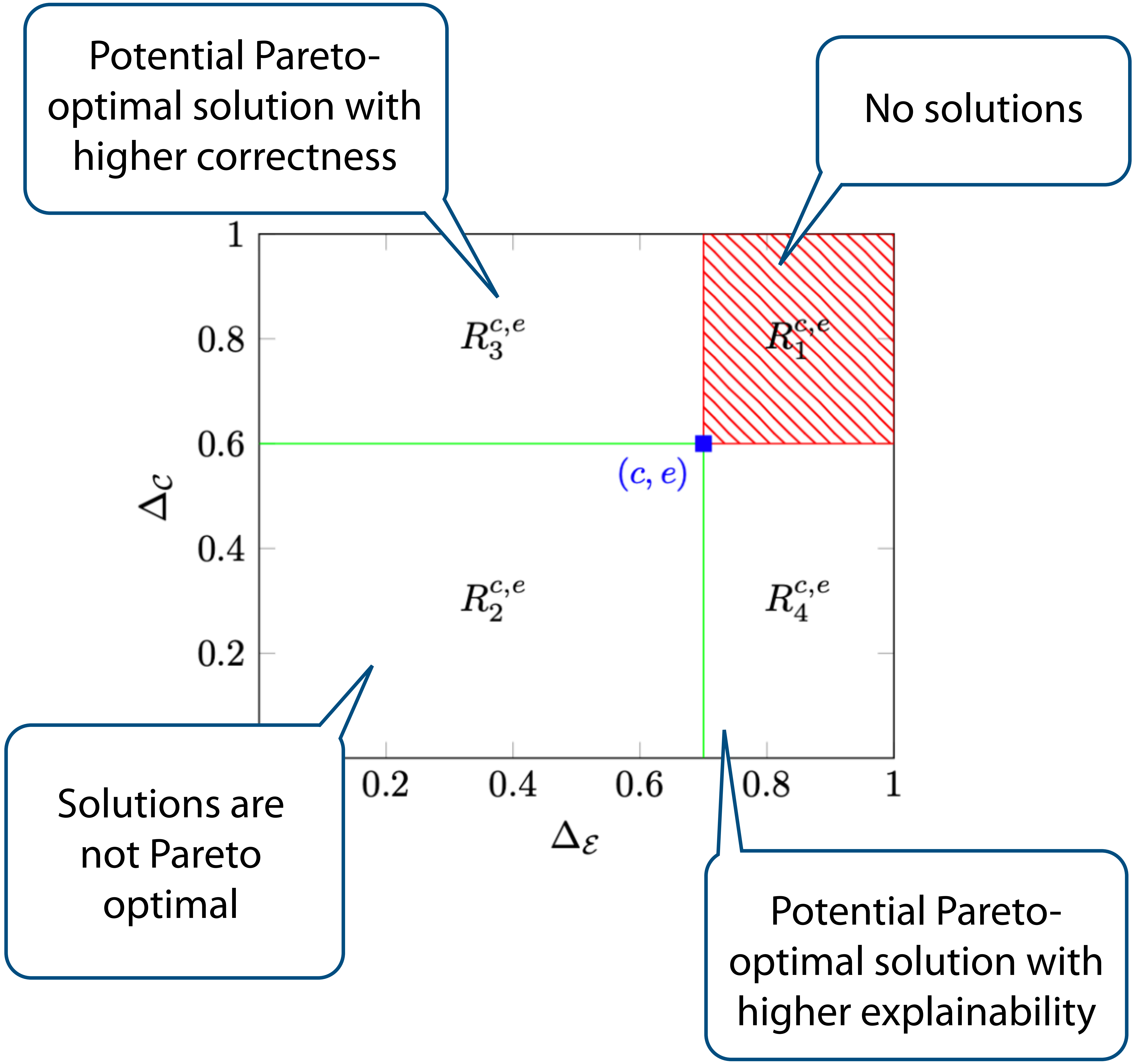
Exploration of Pareto-Optimal Space

- **Synthesize initial Pareto-optimal interpretation**
- **Every PO-interpretation splits space into four regions**
- **Continue search in regions 3 and 4**
    - can be done by setting upper and lower bounds on explainability measure



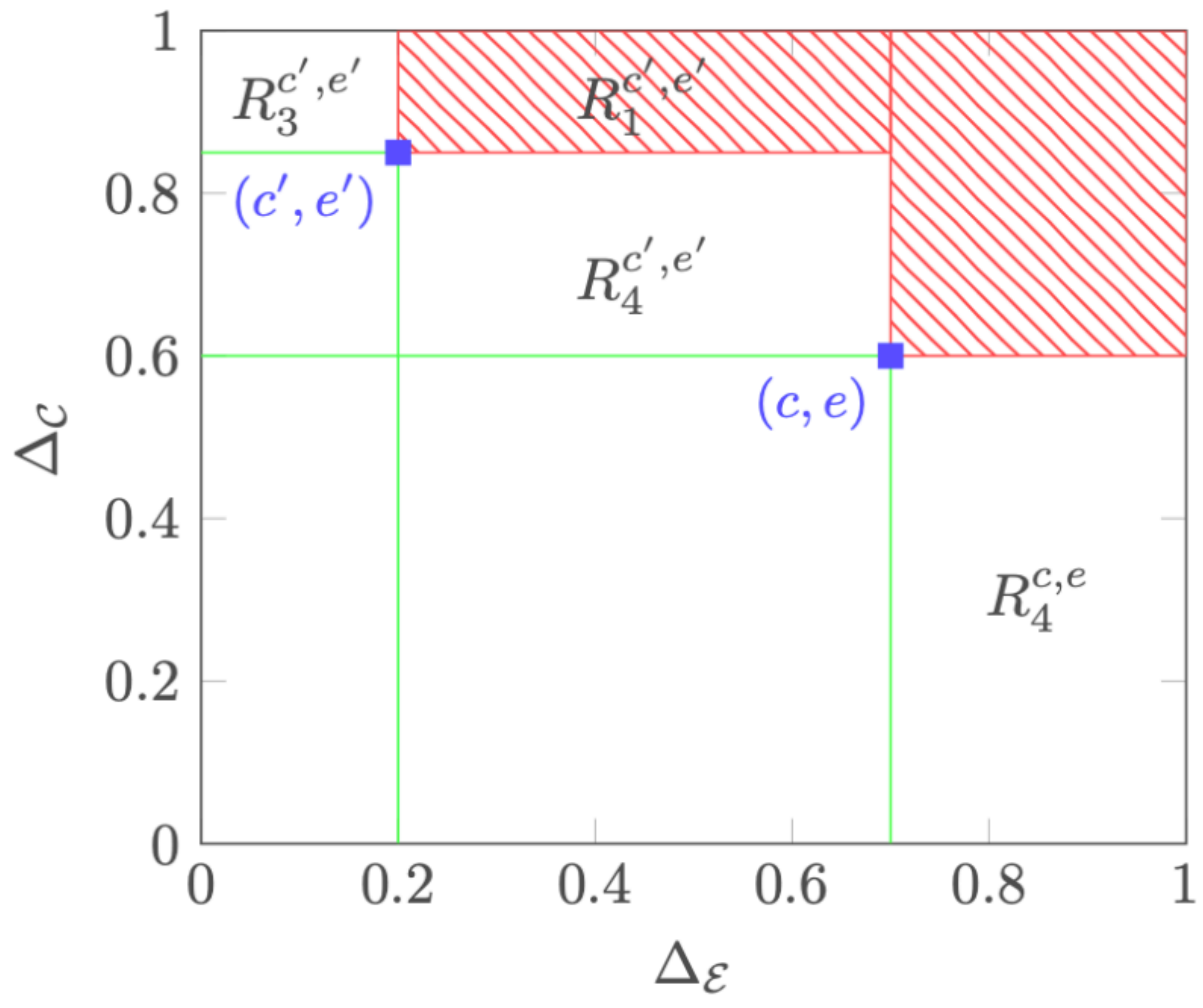
Exploration of Pareto-Optimal Space

- **Synthesize initial Pareto-optimal interpretation**
- **Every PO-interpretation splits space into four regions**
- **Continue search in regions 3 and 4**
  - can be done by setting upper and lower bounds on explainability measure



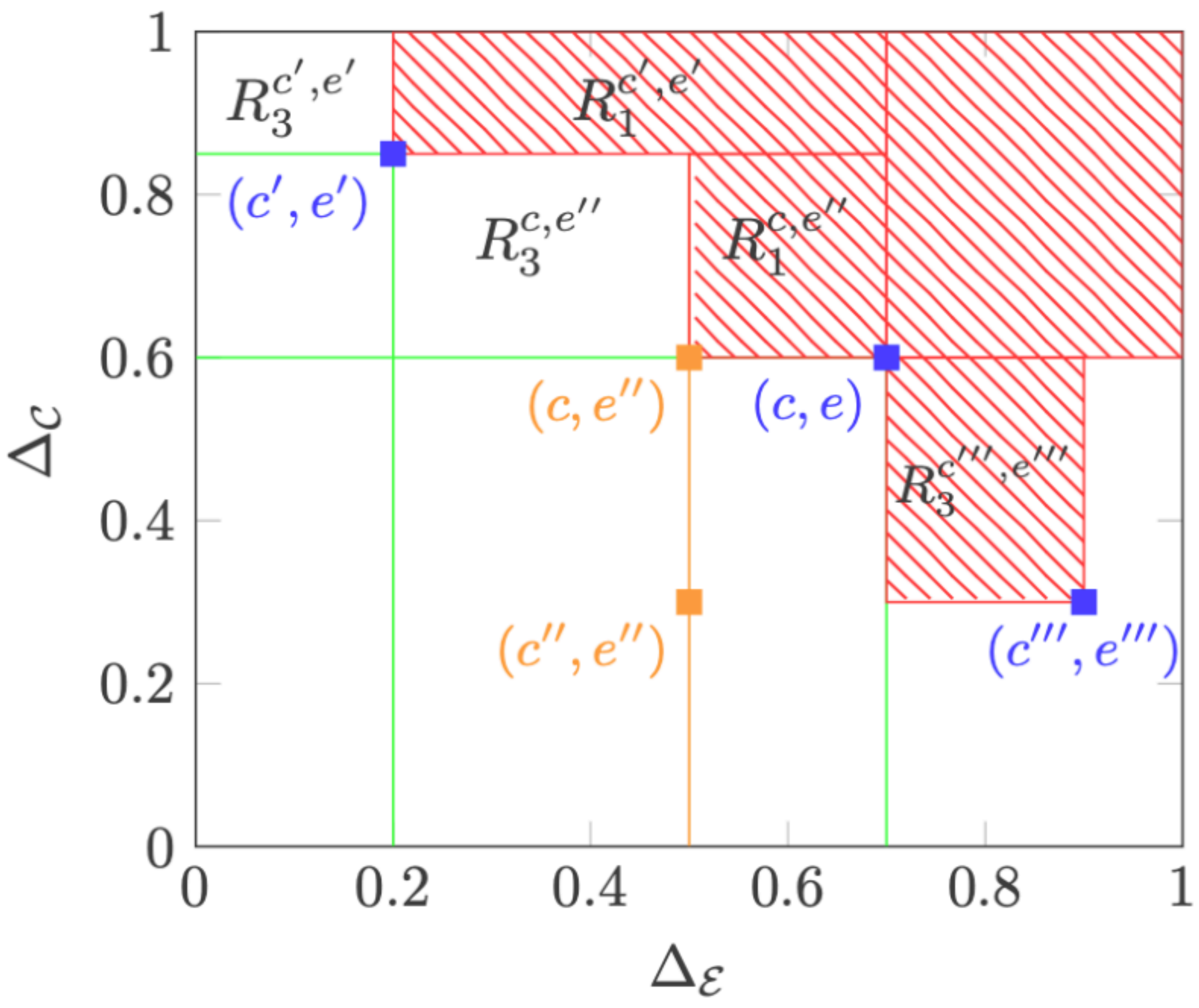
Exploration of Pareto-Optimal Space

- Synthesize initial Pareto-optimal interpretation
- Every PO-interpretation splits space into four regions
- Continue search in regions 3 and 4:
    - can be done by setting upper and lower bounds on explainability measure
    - if correctness measure higher than previous measure, then new PO-interpretation found



Exploration of Pareto-Optimal Space

- **Synthesize initial Pareto-optimal interpretation**
- **Every PO-interpretation splits space into four regions**
- **Continue search in regions 3 and 4:**
  - can be done by setting upper and lower bounds on explainability measure
  - if correctness measure higher than previous measure, then new PO-interpretation found
  - otherwise, repeat process with new explainability threshold



Statistical Guarantees for Black-Box Models

Set of samples

$\mathcal{E} : (\mathcal{I} \to \mathcal{O})$

$\mathcal{S} \subseteq \mathcal{I} \times \mathcal{O}$

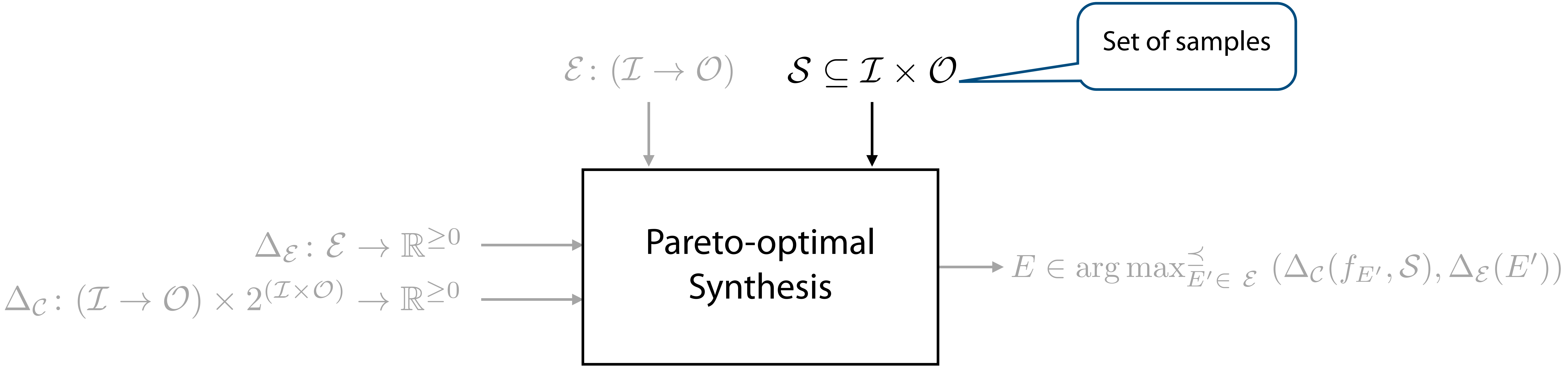
$\Delta_{\mathcal{E}} : \mathcal{E} \to \mathbb{R}^{\geq 0}$

$\Delta_{\mathcal{C}} : (\mathcal{I} \to \mathcal{O}) \times 2^{(\mathcal{I} \times \mathcal{O})} \to \mathbb{R}^{\geq 0}$

Pareto-optimal Synthesis

$E \in \arg\max^{\preceq}_{E' \in \mathcal{E}} \left(\Delta_{\mathcal{C}}(f_{E'}, \mathcal{S}), \Delta_{\mathcal{E}}(E')\right)$

Statistical Guarantees for Black-Box Models



- **Obtaining an exhaustive set** of samples is often **not practical**

Statistical Guarantees for Black-Box Models

Set of samples

$\mathcal{E} : (\mathcal{I} \to \mathcal{O})$ $\quad$ $\mathcal{S} \subseteq \mathcal{I} \times \mathcal{O}$

$\Delta_{\mathcal{E}} : \mathcal{E} \to \mathbb{R}^{\geq 0}$

$\Delta_{\mathcal{C}} : (\mathcal{I} \to \mathcal{O}) \times 2^{(\mathcal{I} \times \mathcal{O})} \to \mathbb{R}^{\geq 0}$

Pareto-optimal Synthesis

$E \in \arg\max^{\preceq}_{E' \in \mathcal{E}} \left(\Delta_{\mathcal{C}}(f_{E'}, \mathcal{S}), \Delta_{\mathcal{E}}(E')\right)$

- **Obtaining an exhaustive set** of samples is often **not practical**
- **How large must the set of samples be** to get an interpretation that does not overfit the set of samples (with a certain probability)?

Statistical Guarantees for Black-Box Models

$\mathcal{E} : (\mathcal{I} \to \mathcal{O})$

$\mathcal{S} \subseteq \mathcal{I} \times \mathcal{O}$

Set of samples

$\Delta_{\mathcal{E}} : \mathcal{E} \to \mathbb{R}^{\geq 0}$

$\Delta_{\mathcal{C}} : (\mathcal{I} \to \mathcal{O}) \times 2^{(\mathcal{I} \times \mathcal{O})} \to \mathbb{R}^{\geq 0}$

Pareto-optimal Synthesis

$E \in \arg\max^{\preceq}_{E' \in \mathcal{E}} \left(\Delta_{\mathcal{C}}(f_{E'}, \mathcal{S}), \Delta_{\mathcal{E}}(E')\right)$

- **Obtaining an exhaustive set** of samples is often **not practical**
- **How large must the set of samples be** to get an interpretation that does not overfit the set of samples (with a certain probability)?
- Answer: ***Probably Approximately Correct (PAC) Learnability***

Statistical Guarantees for Black-Box Models

PAC Learnability

A class of interpretations $\mathcal{E}$ is PAC-learnable with respect to the set of samples $\mathcal{S}$ and a loss function ℓ, if there exists a function $m_{\mathcal{E}}: (0,1)^2 \to \mathbb{N}$ and an algorithm such that:

Statistical Guarantees for Black-Box Models

PAC Learnability

A class of interpretations $\mathcal{E}$ is PAC-learnable with respect to the set of samples $\mathcal{S}$ and a loss function ℓ, if there exists a function $m_{\mathcal{E}}: (0,1)^2 \to \mathbb{N}$ and an algorithm such that:

- for every $\delta, \epsilon \in (0,1)$ and distribution D over $\mathcal{S}$,

Statistical Guarantees for Black-Box Models

PAC Learnability

A class of interpretations $\mathcal{E}$ is PAC-learnable with respect to the set of samples $\mathcal{S}$ and a loss function ℓ, if there exists a function $m_{\mathcal{E}} : (0,1)^2 \to \mathbb{N}$ and an algorithm such that:

- for every $\delta, \epsilon \in (0,1)$ and distribution D over $\mathcal{S}$,
- when running the algorithm on $m \geq m_{\mathcal{E}}(\epsilon, \delta)$ i.i.d. samples generated according to D,

Statistical Guarantees for Black-Box Models

PAC Learnability

A class of interpretations $\mathcal{E}$ is PAC-learnable with respect to the set of samples $\mathcal{S}$ and a loss function ℓ, if there exists a function $m_{\mathcal{E}}: (0,1)^2 \rightarrow \mathbb{N}$ and an algorithm such that:

- for every $\delta, \epsilon \in (0,1)$ and distribution D over $\mathcal{S}$,
- when running the algorithm on $m \geq m_{\mathcal{E}}(\epsilon, \delta)$ i.i.d. samples generated according to D,
- the algorithm returns an interpretation E s.t. $Pr(|L_D(E) - \min_{E' \in \mathcal{E}} L_D(E')| \leq \epsilon) \geq 1 - \delta$

  where $L_D(E) = \mathbb{E}_{z \sim D}[\ell(E, z)]$

Statistical Guarantees for Black-Box Models

PAC Learnability

A class of interpretations $\mathcal{E}$ is PAC-learnable with respect to the set of samples $\mathcal{S}$ and a loss function ℓ, if there exists a function $m_{\mathcal{E}}: (0,1)^2 \rightarrow \mathbb{N}$ and an algorithm such that:

- for every $\delta, \epsilon \in (0,1)$ and distribution D over $\mathcal{S}$,
- when running the algorithm on $m \geq m_{\mathcal{E}}(\epsilon, \delta)$ i.i.d. samples generated according to D,
- the algorithm returns an interpretation E s.t. $Pr(|L_D(E) - \min_{E' \in \mathcal{E}} L_D(E')| \leq \epsilon) \geq 1 - \delta$

  where $L_D(E) = \mathbb{E}_{z \sim D}[\ell(E, z)]$

- Every finite class of interpretations is PAC-learnable

Statistical Guarantees for Black-Box Models

PAC Learnability

A class of interpretations $\mathcal{E}$ is PAC-learnable with respect to the set of samples $\mathcal{S}$ and a loss function ℓ, if there exists a function $m_{\mathcal{E}}: (0,1)^2 \rightarrow \mathbb{N}$ and an algorithm such that:

- for every $\delta, \epsilon \in (0,1)$ and distribution D over $\mathcal{S}$,
- when running the algorithm on $m \geq m_{\mathcal{E}}(\epsilon, \delta)$ i.i.d. samples generated according to D,
- the algorithm returns an interpretation E s.t. $Pr(|L_D(E) - \min_{E' \in \mathcal{E}} L_D(E')| \leq \epsilon) \geq 1 - \delta$

  where $L_D(E) = \mathbb{E}_{z \sim D}[\ell(E, z)]$

- Every finite class of interpretations is PAC-learnable
- Our MaxSAT-based algorithm satisfies PAC-learnability since it minimizes $\frac{\sum_{z \in \mathcal{S}} \ell(E,z)}{|\mathcal{S}|}$

Statistical Guarantees for Black-Box Models

PAC Learnability

A class of interpretations $\mathcal{E}$ is PAC-learnable with respect to the set of samples $\mathcal{S}$ and a loss function ℓ, if there exists a function $m_{\mathcal{E}}: (0,1)^2 \to \mathbb{N}$ and an algorithm such that:

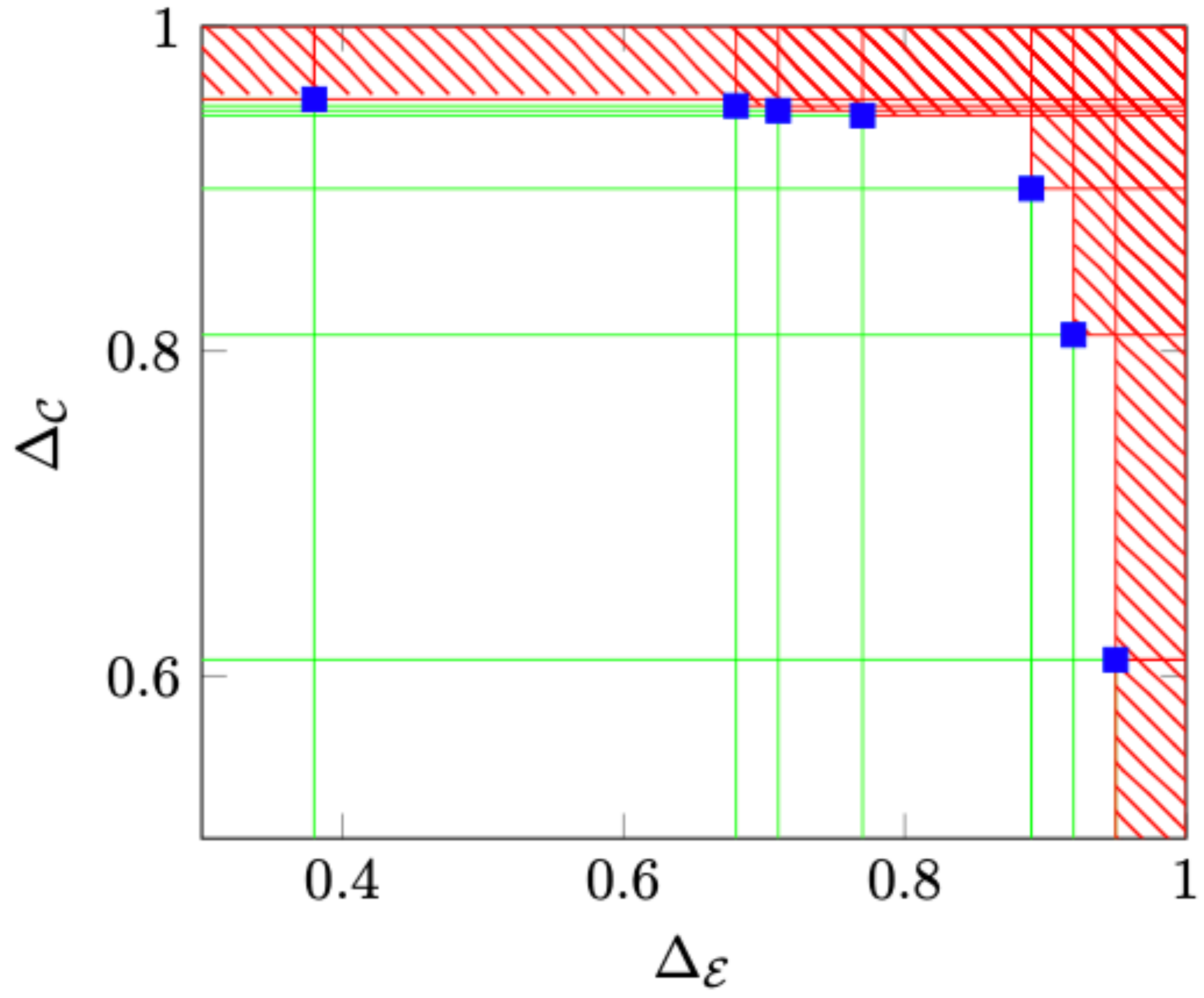
- for every $\delta, \epsilon \in (0,1)$ and distribution D over $\mathcal{S}$,
- when running the algorithm on $m \geq m_{\mathcal{E}}(\epsilon, \delta)$ i.i.d. samples generated according to D,
- the algorithm returns an interpretation E s.t. $Pr(|L_D(E) - \min_{E' \in \mathcal{E}} L_D(E')| \leq \epsilon) \geq 1 - \delta$

  where $L_D(E) = \mathbb{E}_{z \sim D}[\ell(E, z)]$

- Every finite class of interpretations is PAC-learnable
- Our MaxSAT-based algorithm satisfies PAC-learnability since it minimizes $\frac{\sum_{z \in \mathcal{S}} \ell(E,z)}{|\mathcal{S}|}$
- The number of samples can be determined in terms of $\delta, \epsilon, |\mathcal{E}|$

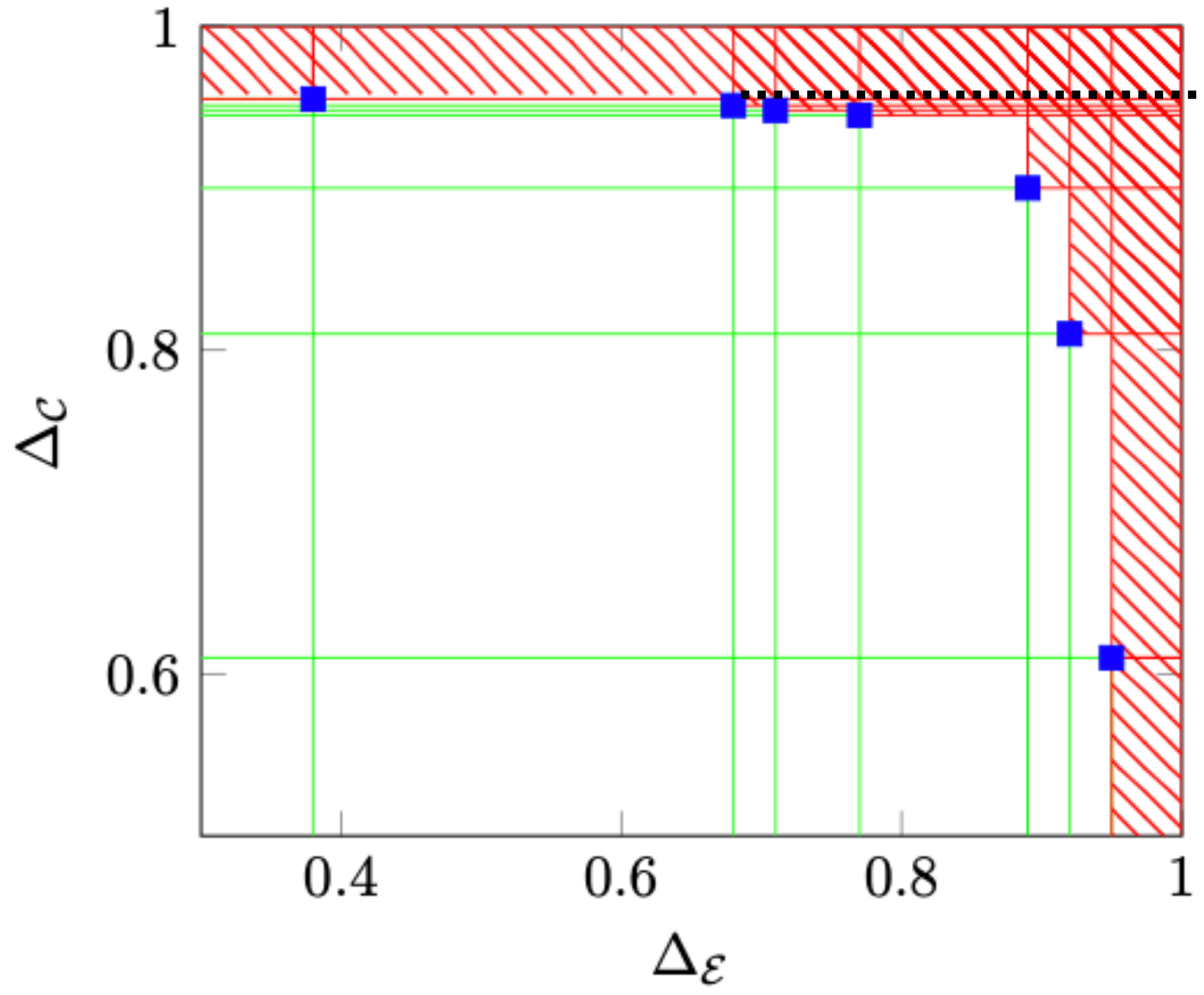
Experimental Results

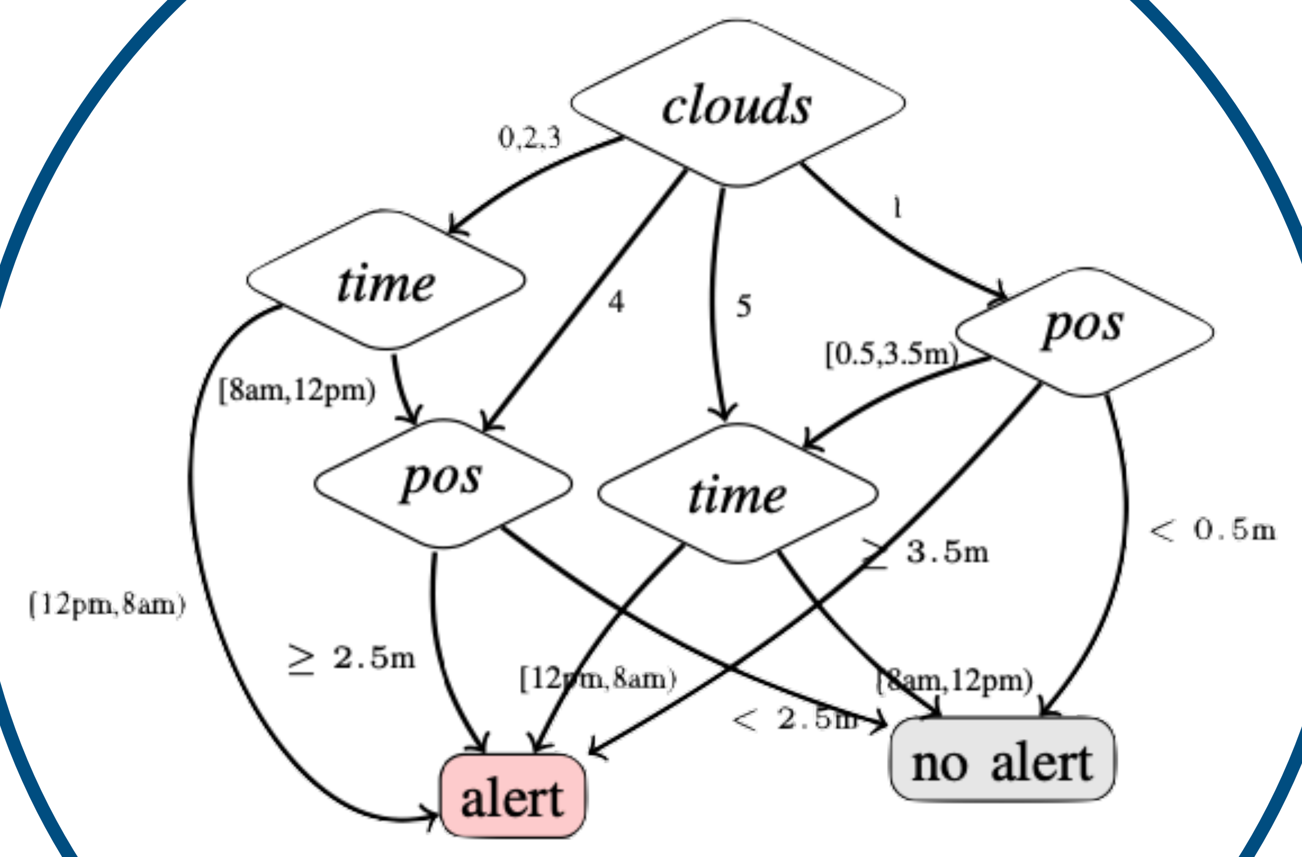
AutoTAXI



Experimental Results

AutoTAXI

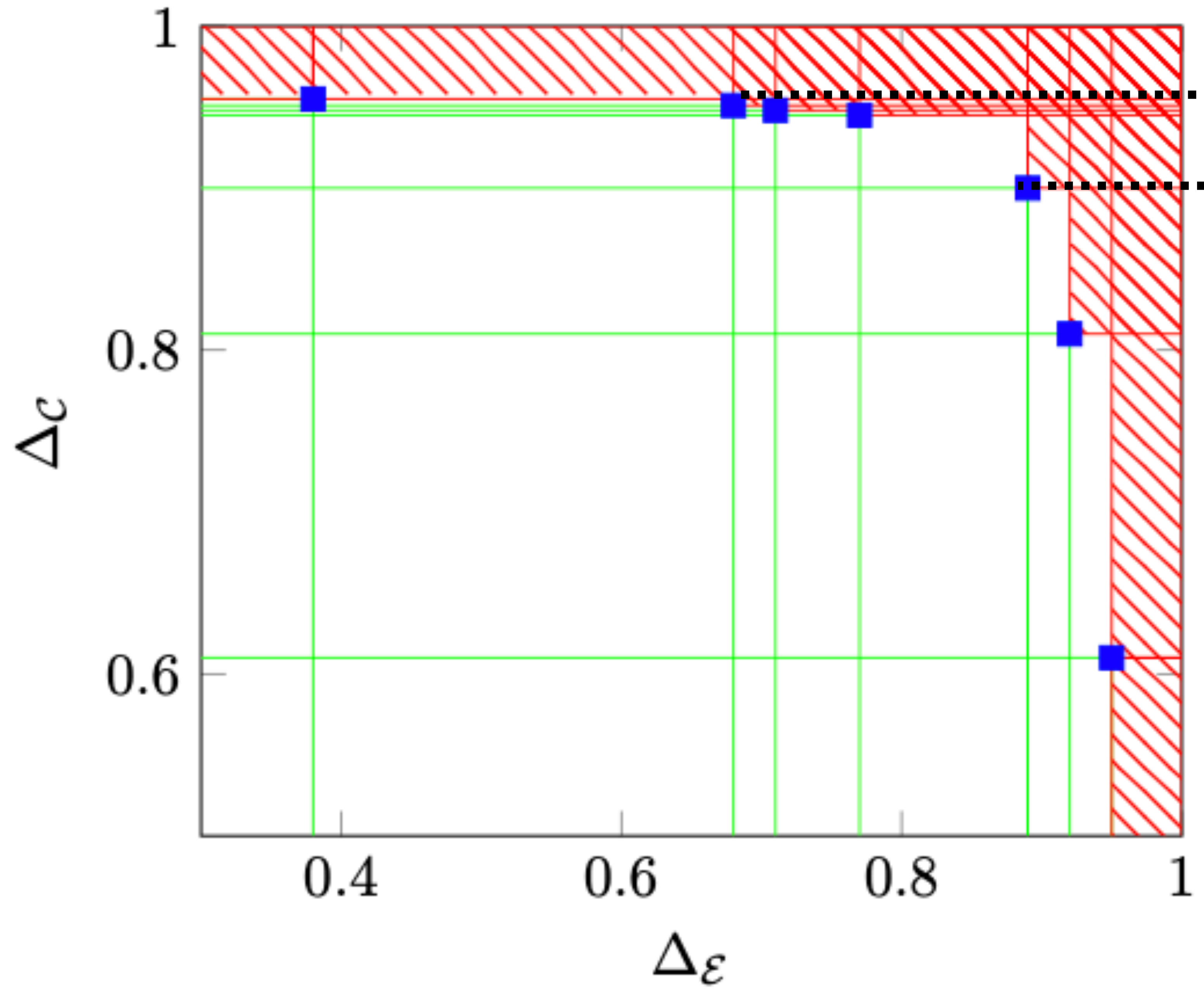


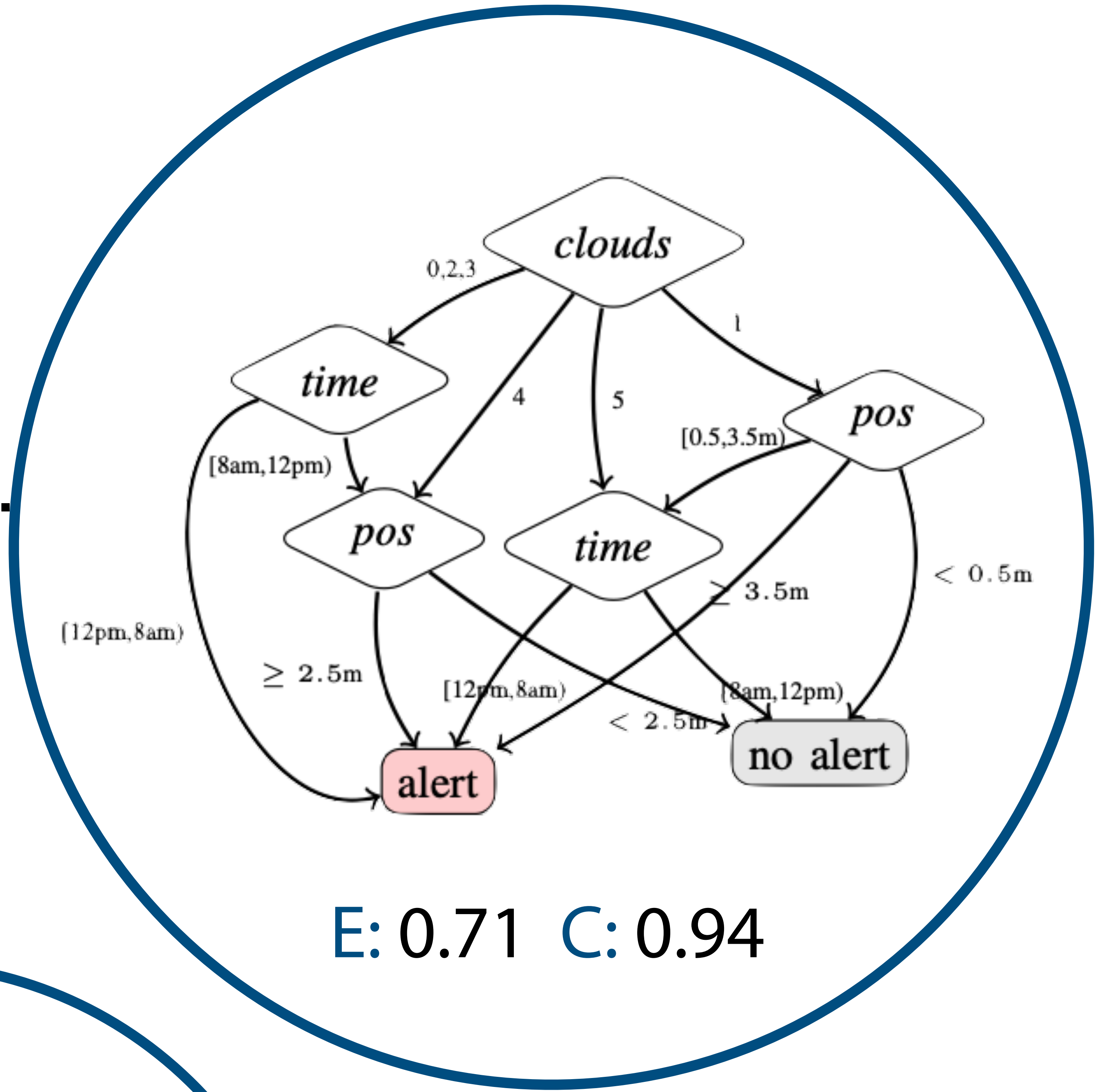


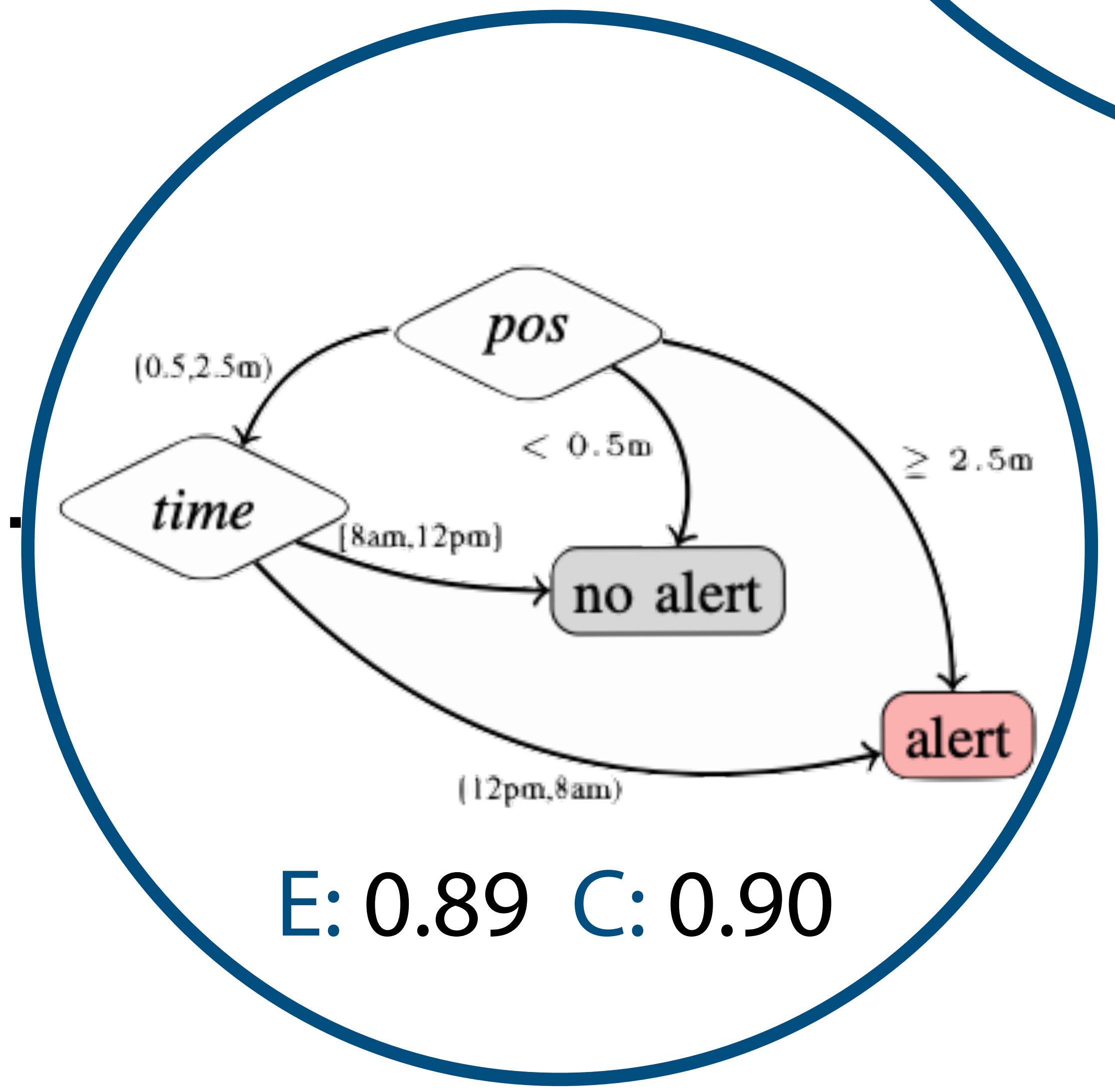
E: 0.71 C: 0.94

Experimental Results

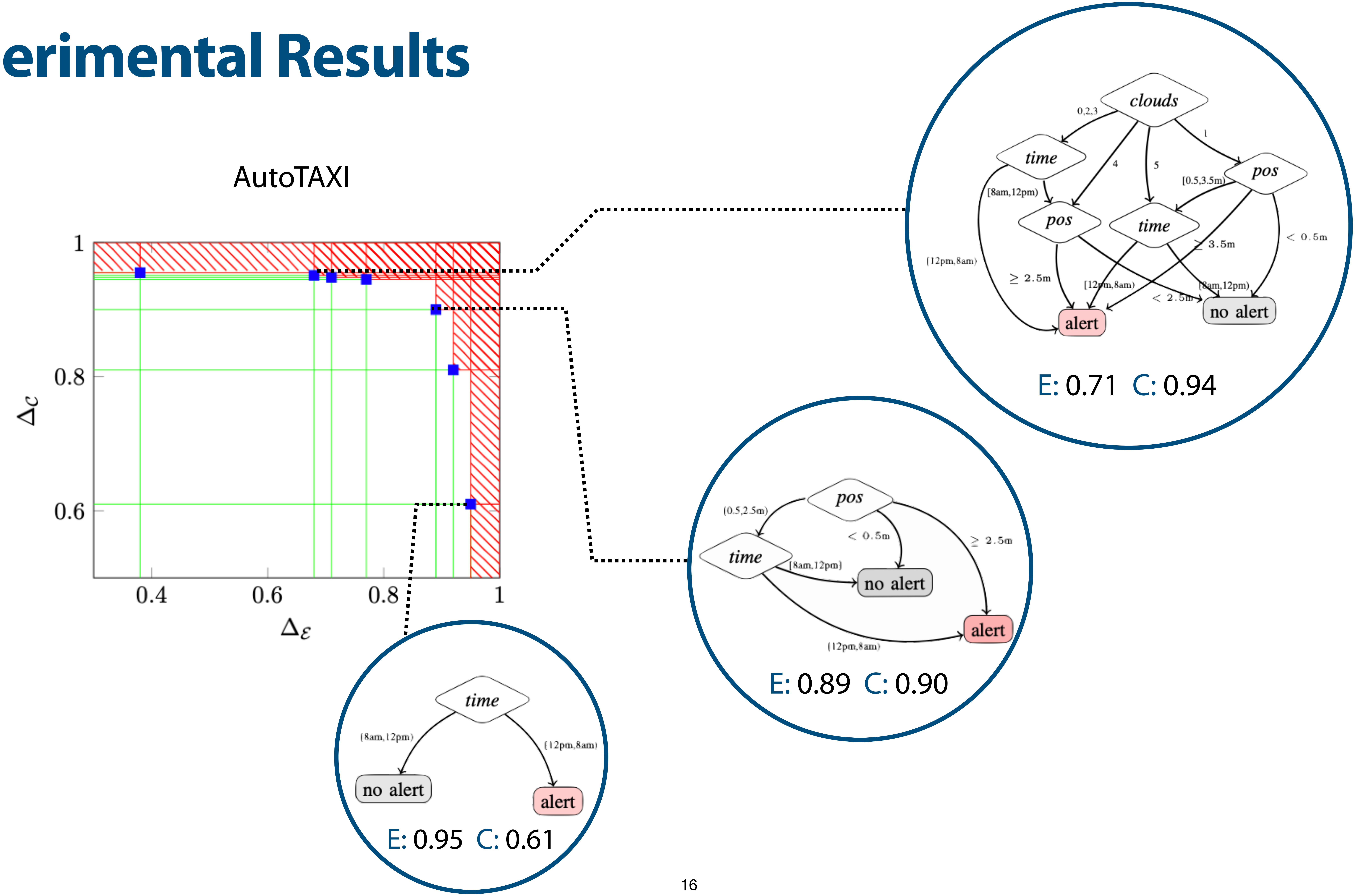
AutoTAXI





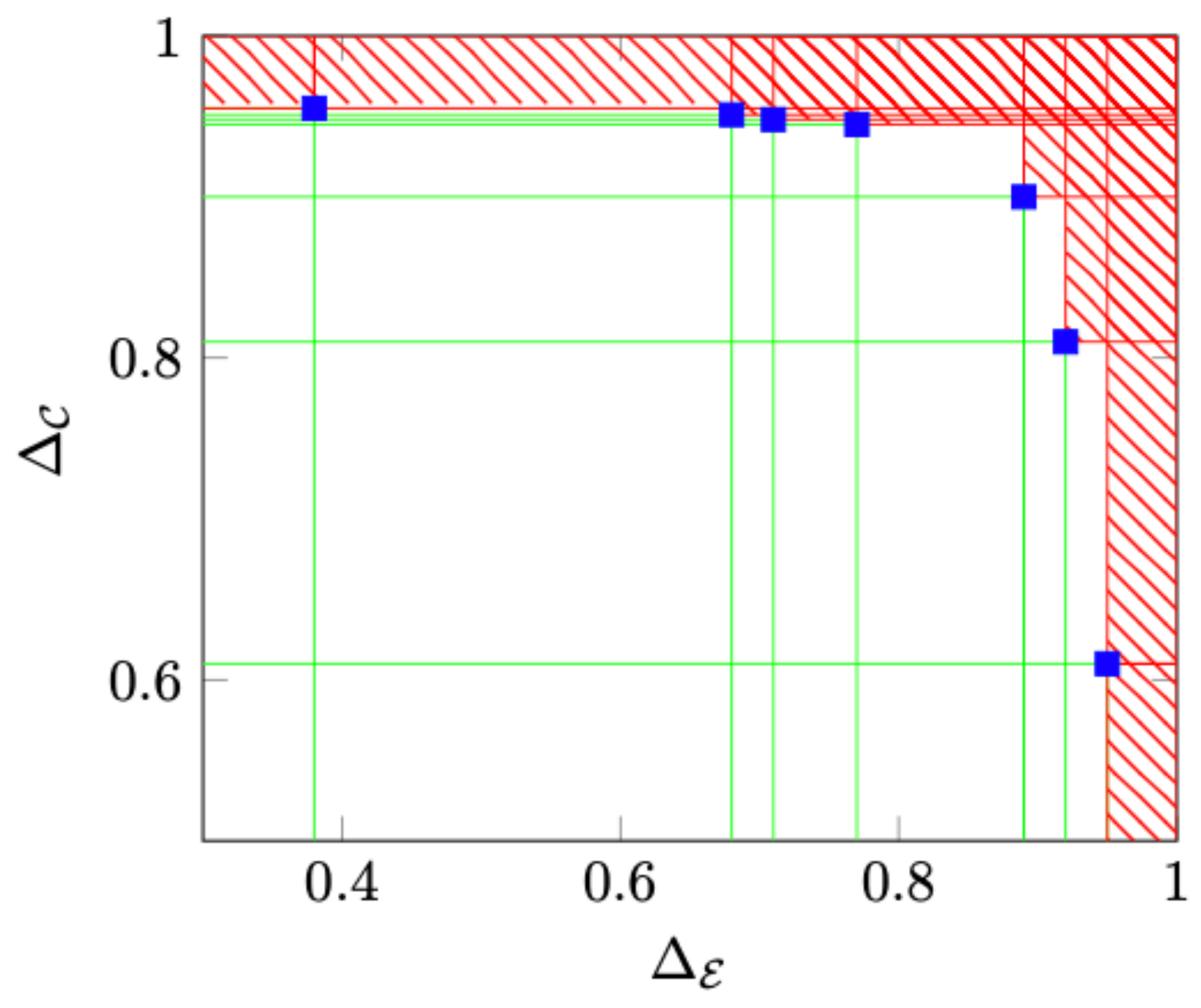


Experimental Results

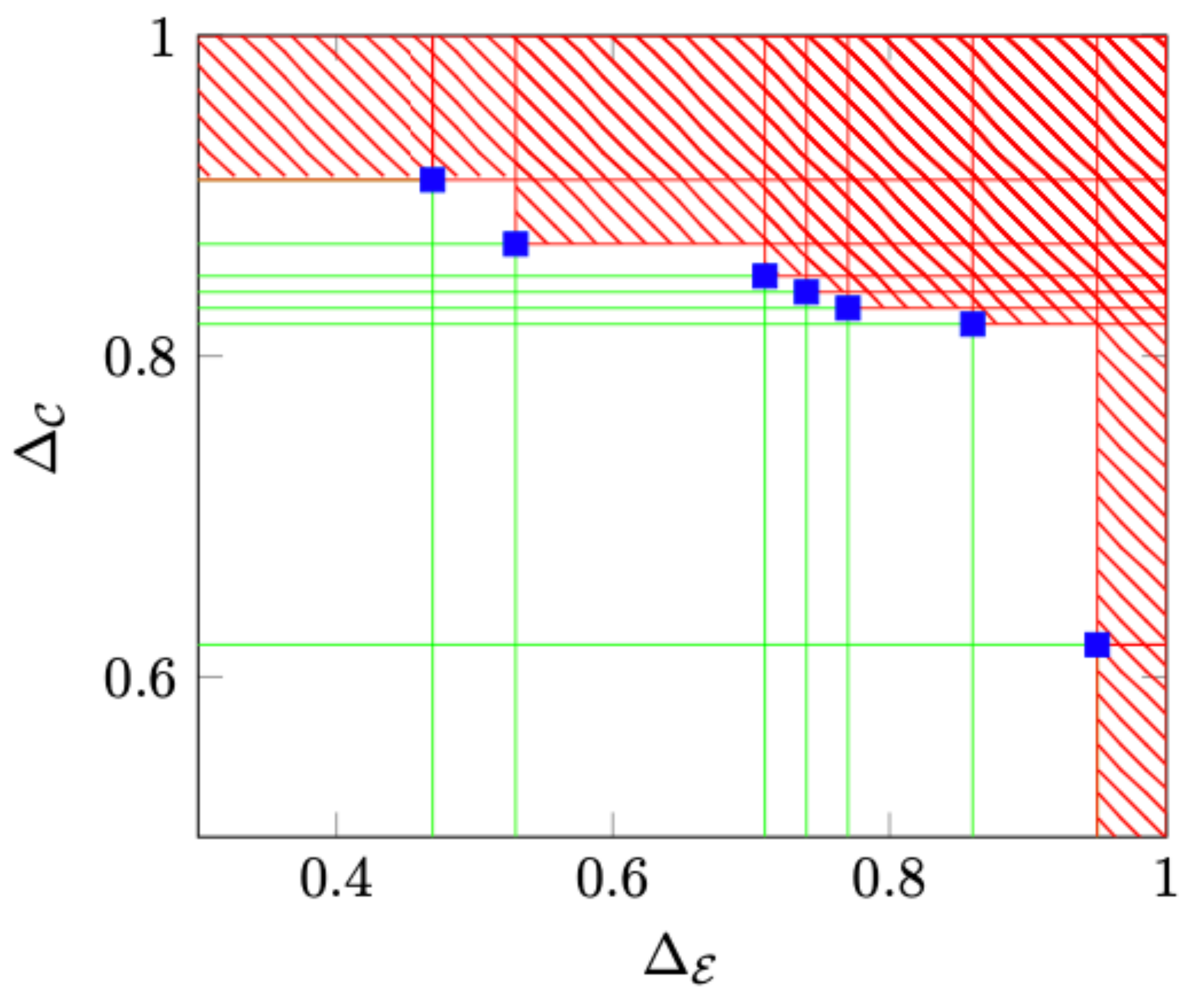


Experimental Results

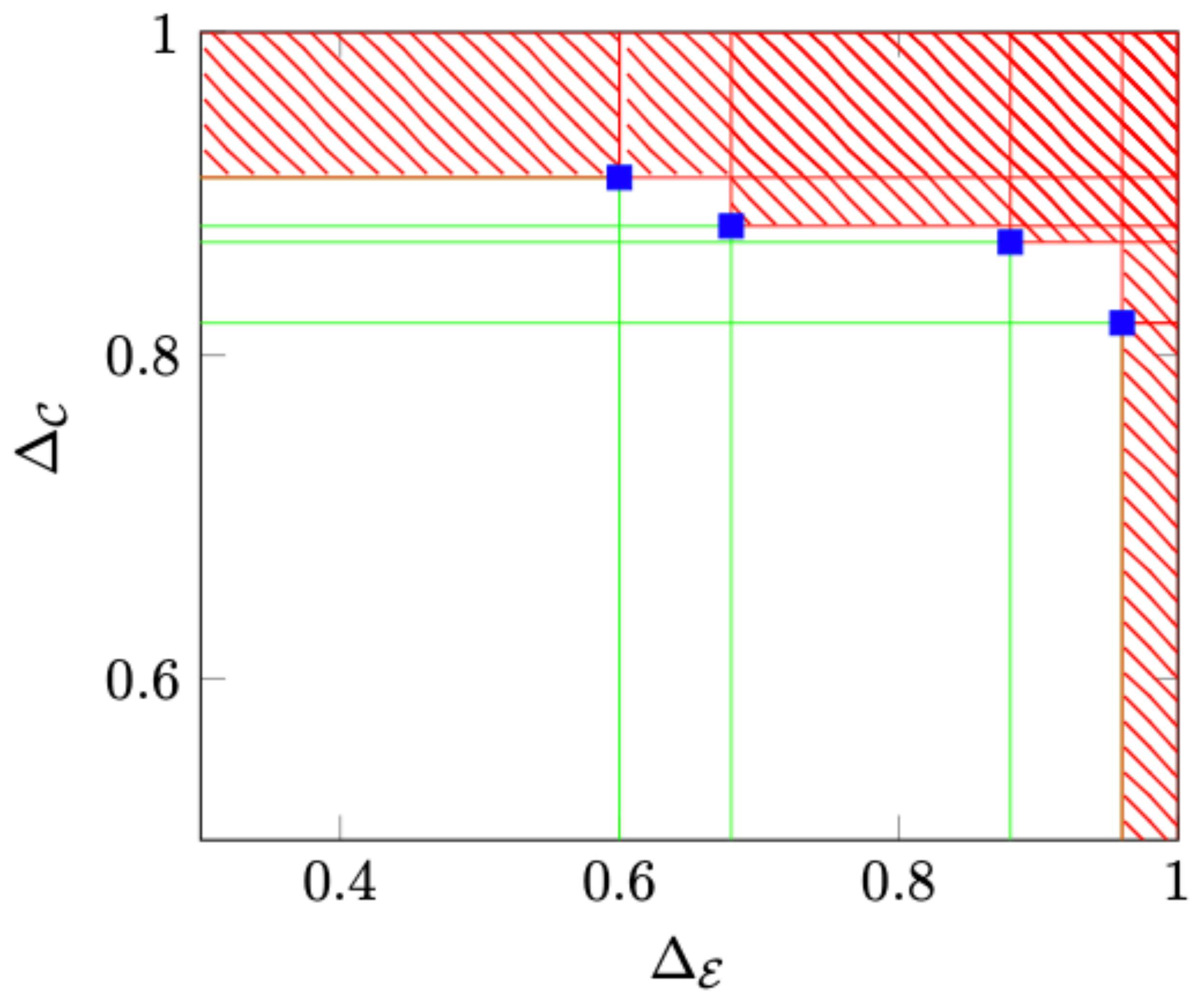
AutoTAXI



Bank Loan

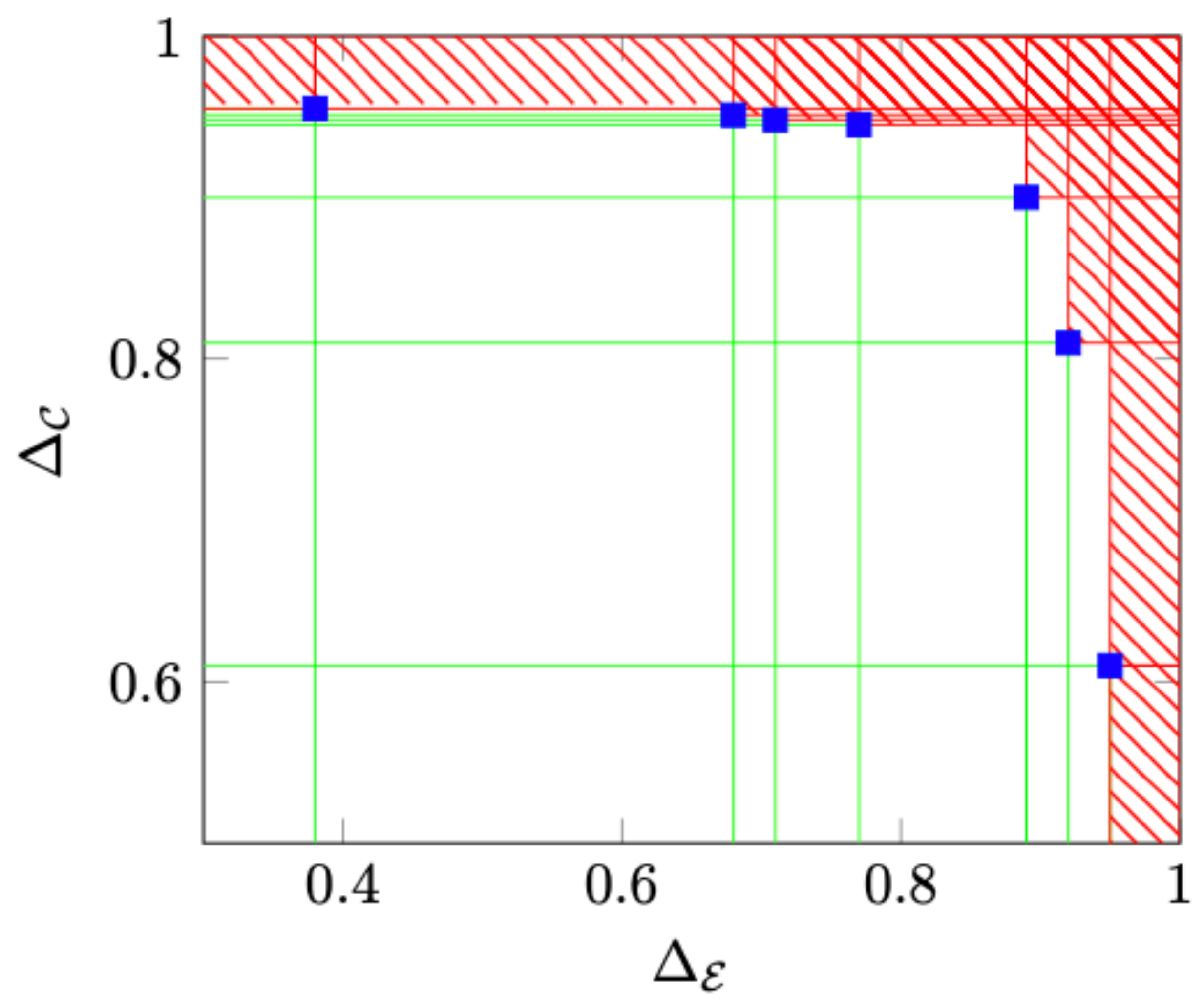


Theorem Prover

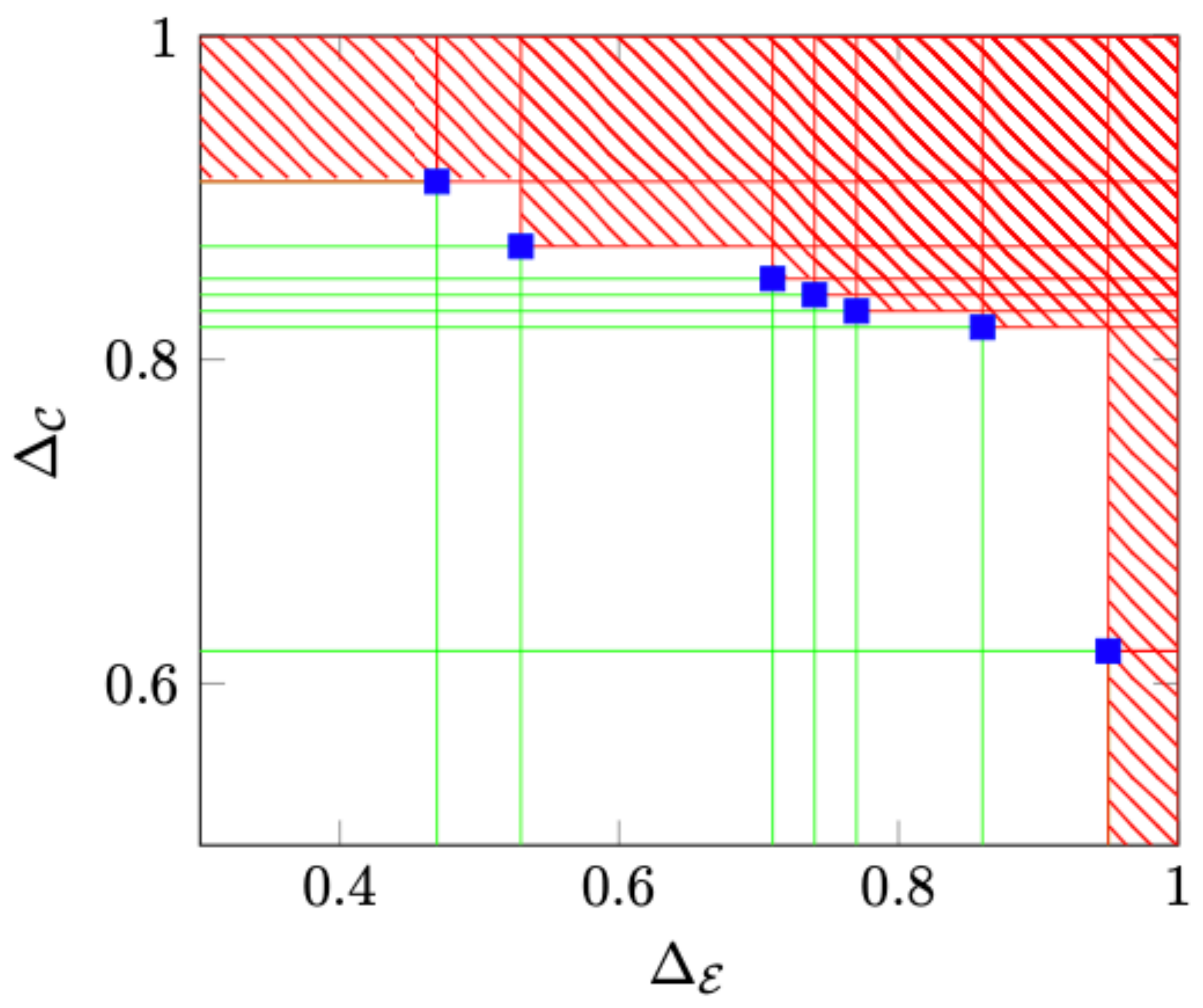


Experimental Results

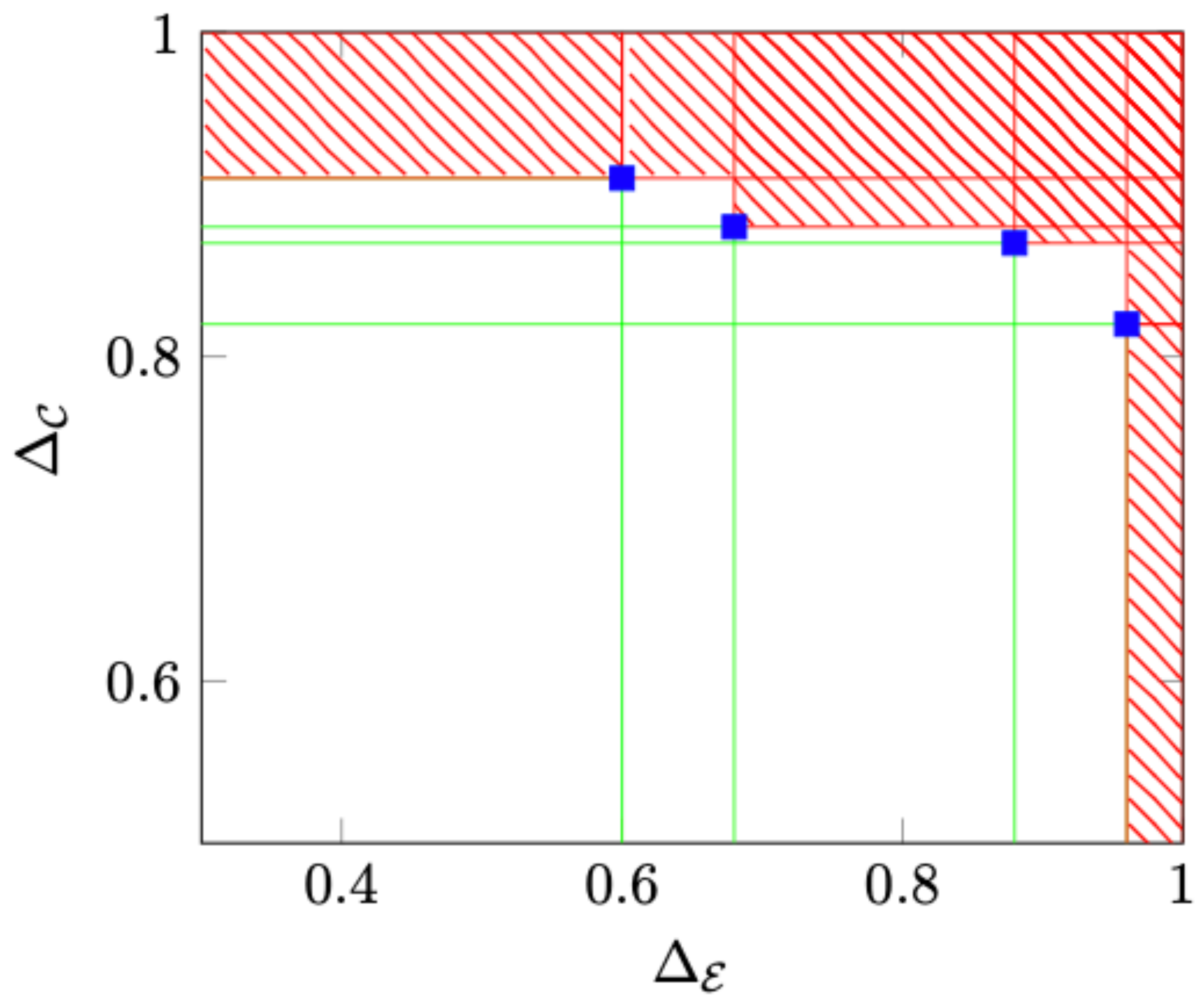
AutoTAXI



Bank Loan



Theorem Prover



Bench mark	δ,ϵ	$\lvert\mathcal{S}\rvert$	Explored (PO, TNP)	min time (s)	max time (s)	median time (s)	unsat time (s)
Auto TAXI (3)	0.05, 0.05	333	7, 25	1.709	388.527	5.696	< 1
	0.05, 0.03	555	5, 26	2.513	616.520	11.222	< 1
Bank Loan (4)	0.05, 0.05	365	7, 27	1.927	387.599	8.975	< 1
	0.05, 0.03	608	4, 27	2.855	1299.196	17.998	< 1
Theorem Prover (6)	0.05, 0.05	338	4, 20	0.767	3.392	1.138	< 1
	0.05, 0.03	703	3, 28	2.051	18.148	3.643	< 1

Performance

Summary

Summary

- Pareto-optimal interpretation synthesis

Summary

- Pareto-optimal interpretation synthesis
- Pareto optimality is the best we can hope for when synthesizing interpretations

Summary

- Pareto-optimal interpretation synthesis
- Pareto optimality is the best we can hope for when synthesizing interpretations
- A MaxSAT-based solution for finite domain

Summary

- Pareto-optimal interpretation synthesis
- Pareto optimality is the best we can hope for when synthesizing interpretations
- A MaxSAT-based solution for finite domain
- Algorithm for exploring the Pareto-optimal space

Summary

- Pareto-optimal interpretation synthesis
- Pareto optimality is the best we can hope for when synthesizing interpretations
- A MaxSAT-based solution for finite domain
- Algorithm for exploring the Pareto-optimal space
- Statistical guarantees based on PAC learnability

Summary

- Pareto-optimal interpretation synthesis
- Pareto optimality is the best we can hope for when synthesizing interpretations
- A MaxSAT-based solution for finite domain
- Algorithm for exploring the Pareto-optimal space
- Statistical guarantees based on PAC learnability
- For technical details: https://arxiv.org/pdf/2108.07307.pdf

Summary

- Pareto-optimal interpretation synthesis
- Pareto optimality is the best we can hope for when synthesizing interpretations
- A MaxSAT-based solution for finite domain
- Algorithm for exploring the Pareto-optimal space
- Statistical guarantees based on PAC learnability
- For technical details: https://arxiv.org/pdf/2108.07307.pdf

- Future work:

Summary

- Pareto-optimal interpretation synthesis
- Pareto optimality is the best we can hope for when synthesizing interpretations
- A MaxSAT-based solution for finite domain
- Algorithm for exploring the Pareto-optimal space
- Statistical guarantees based on PAC learnability
- For technical details: https://arxiv.org/pdf/2108.07307.pdf

- Future work:
    - extended to work with interpretation classes of infinite cardinality but finite VC dimension

Summary

- Pareto-optimal interpretation synthesis
- Pareto optimality is the best we can hope for when synthesizing interpretations
- A MaxSAT-based solution for finite domain
- Algorithm for exploring the Pareto-optimal space
- Statistical guarantees based on PAC learnability
- For technical details: https://arxiv.org/pdf/2108.07307.pdf

- Future work:
  - extended to work with interpretation classes of infinite cardinality but finite VC dimension
  - investigating oracle-guided approaches for refining interpretation

Summary

- Pareto-optimal interpretation synthesis
- Pareto optimality is the best we can hope for when synthesizing interpretations
- A MaxSAT-based solution for finite domain
- Algorithm for exploring the Pareto-optimal space
- Statistical guarantees based on PAC learnability
- For technical details: https://arxiv.org/pdf/2108.07307.pdf

- Future work:
  - extended to work with interpretation classes of infinite cardinality but finite VC dimension
  - investigating oracle-guided approaches for refining interpretation

Thank you!