Decision-Focused Learning

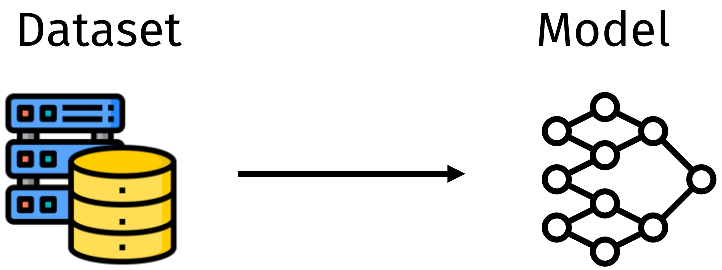
(and how to do it quickly)

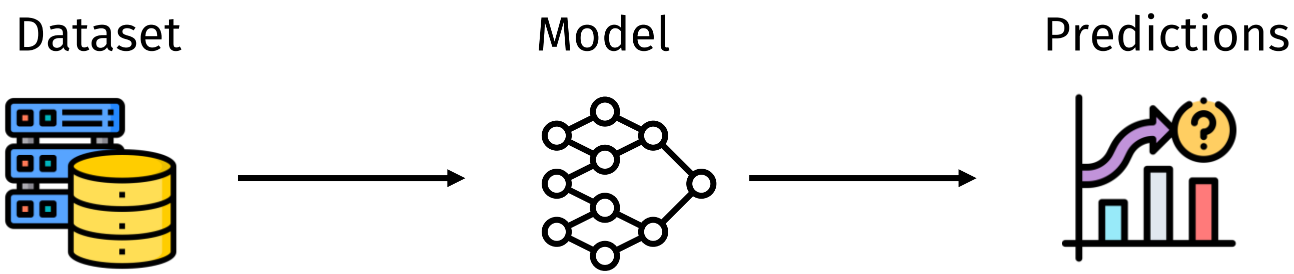
KU LEUVEN

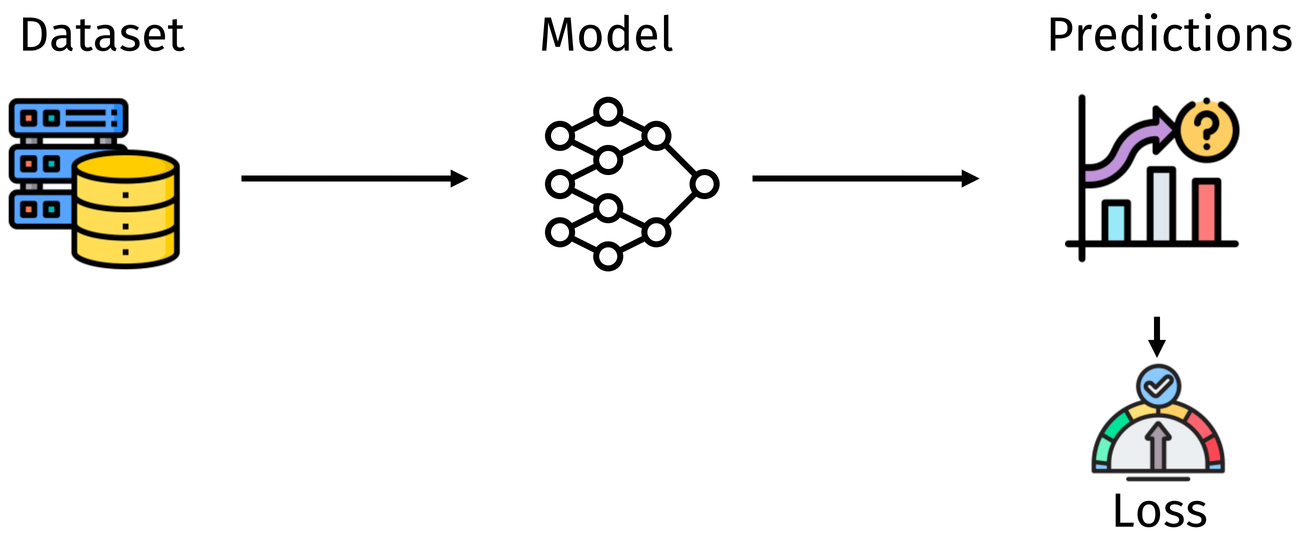
DTAI

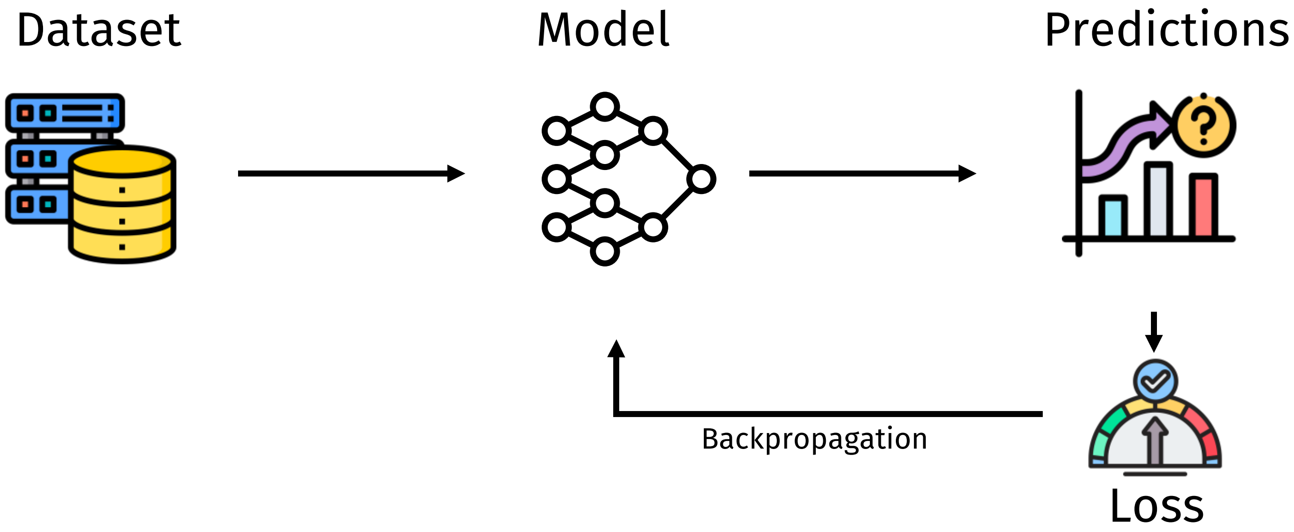
Part 1: Decision-Focused Learning

Dataset


Dataset
Model

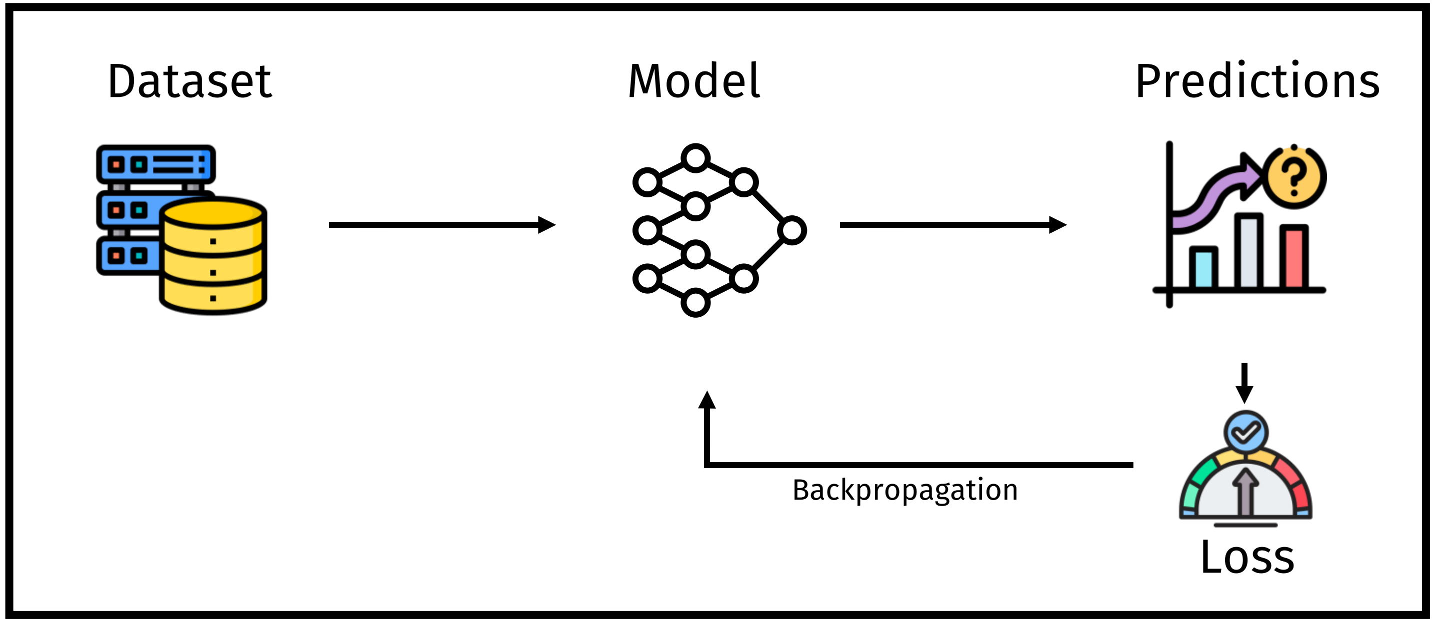

Dataset
Model
Predictions

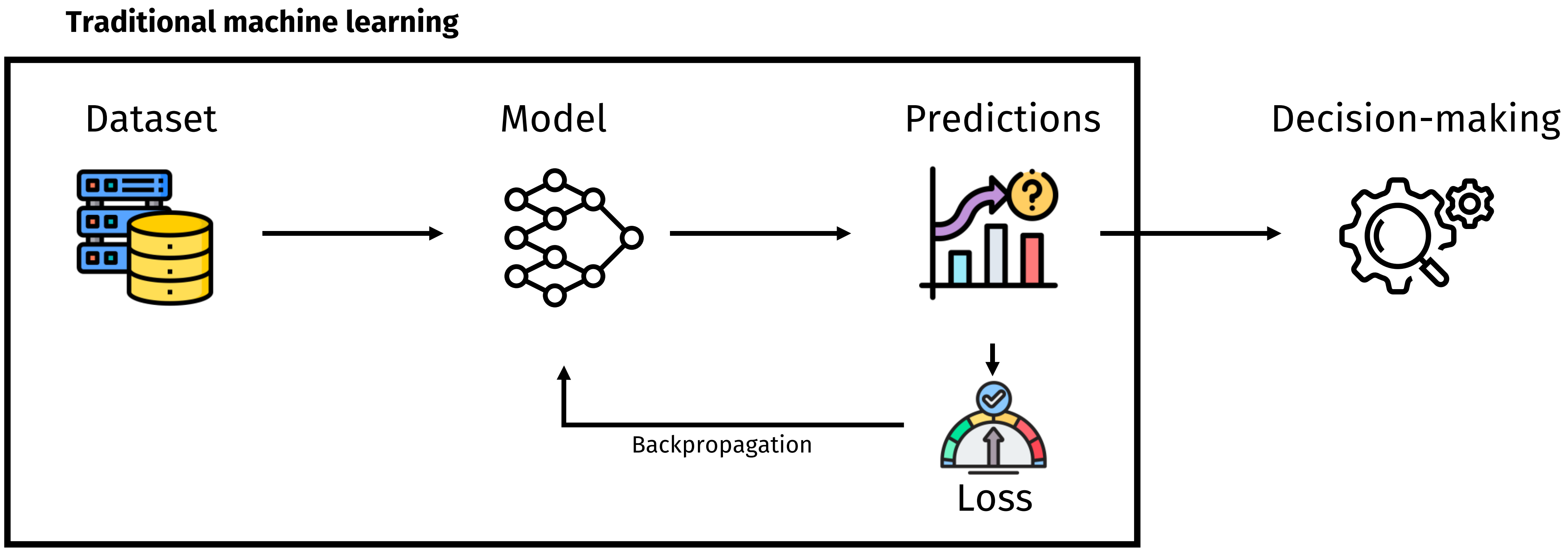

Dataset
Model
Predictions
Loss

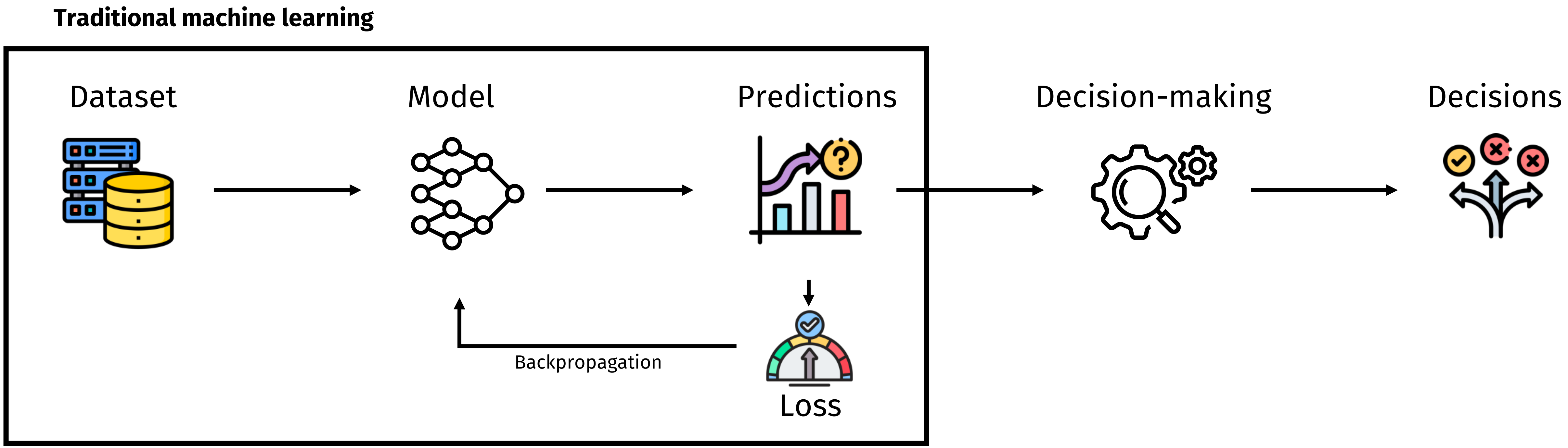


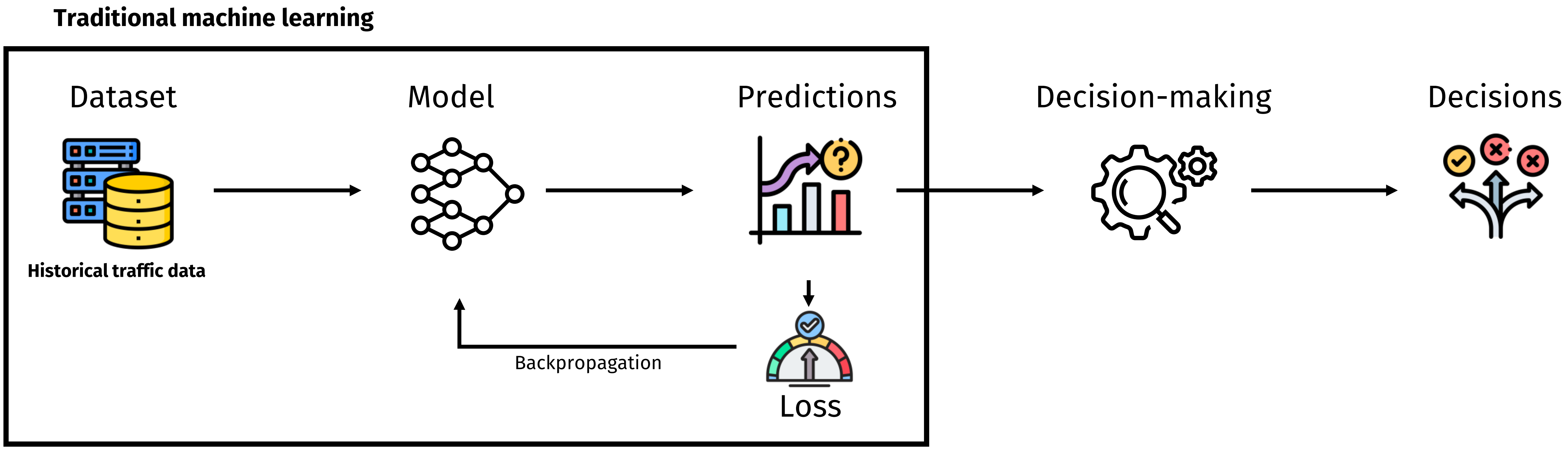
Dataset
Model
Predictions
Backpropagation
Loss

Traditional machine learning

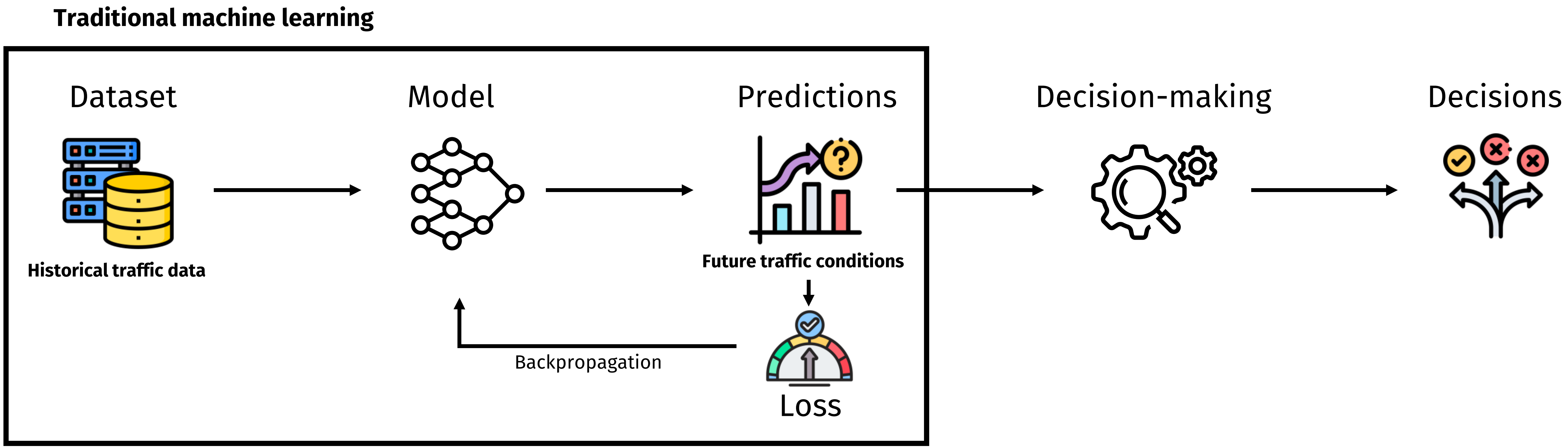


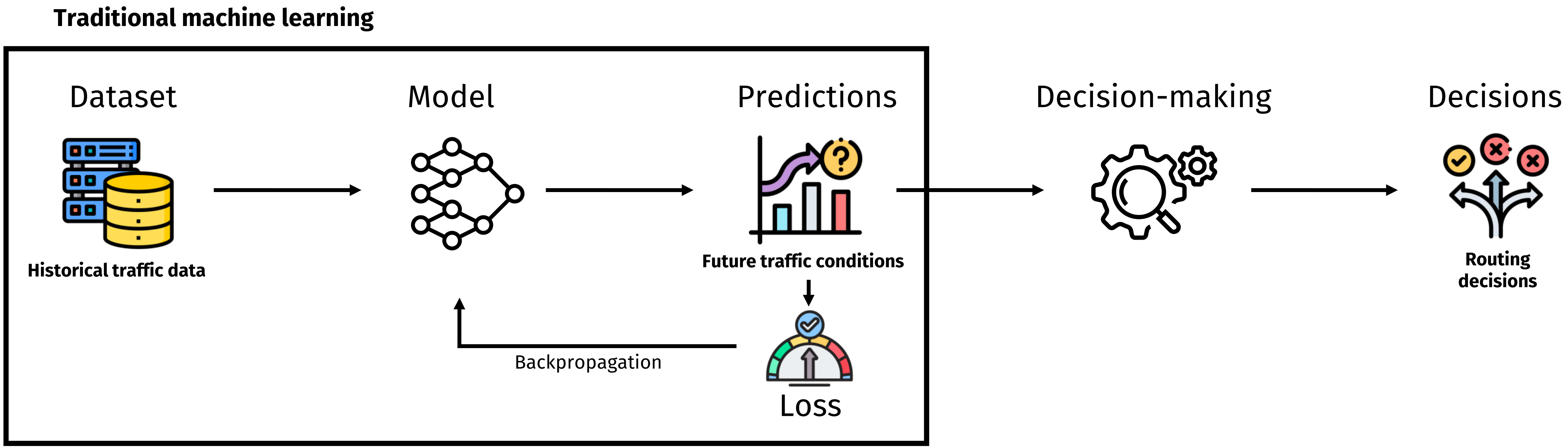

Traditional machine learning
Dataset
Model
Predictions
Decision-making
Backpropagation
Loss


Traditional machine learning
Dataset
Model
Predictions
Decision-making
Decisions
Backpropagation
Loss

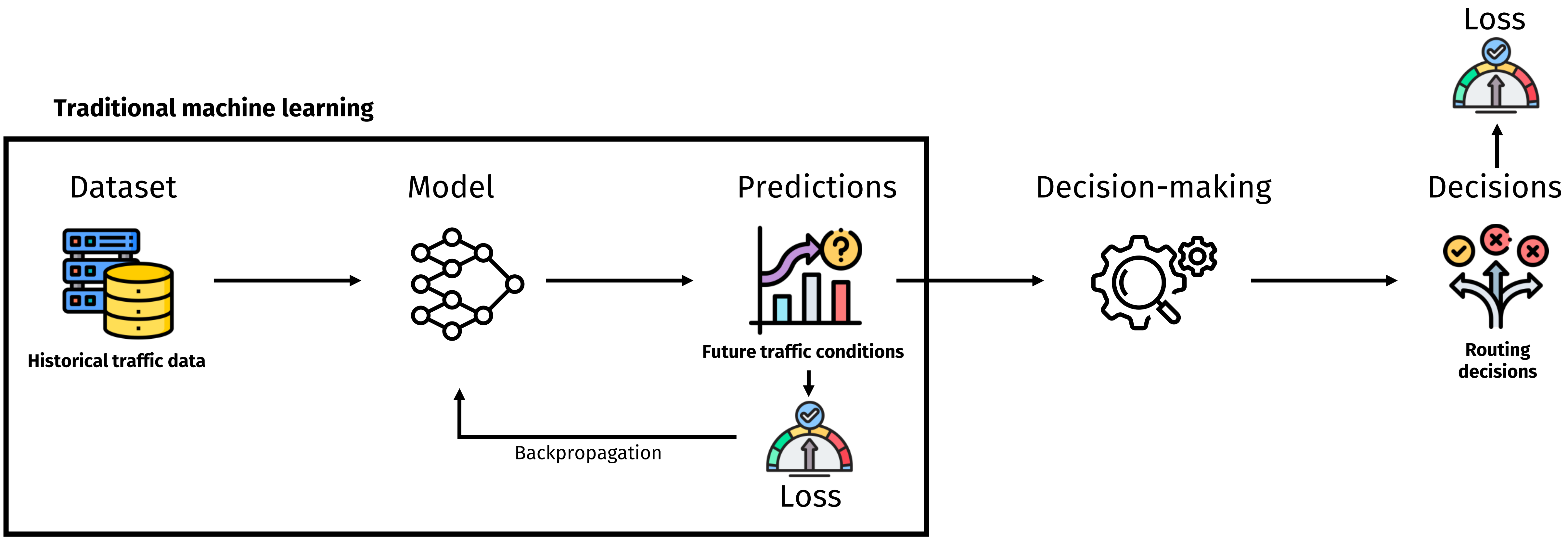


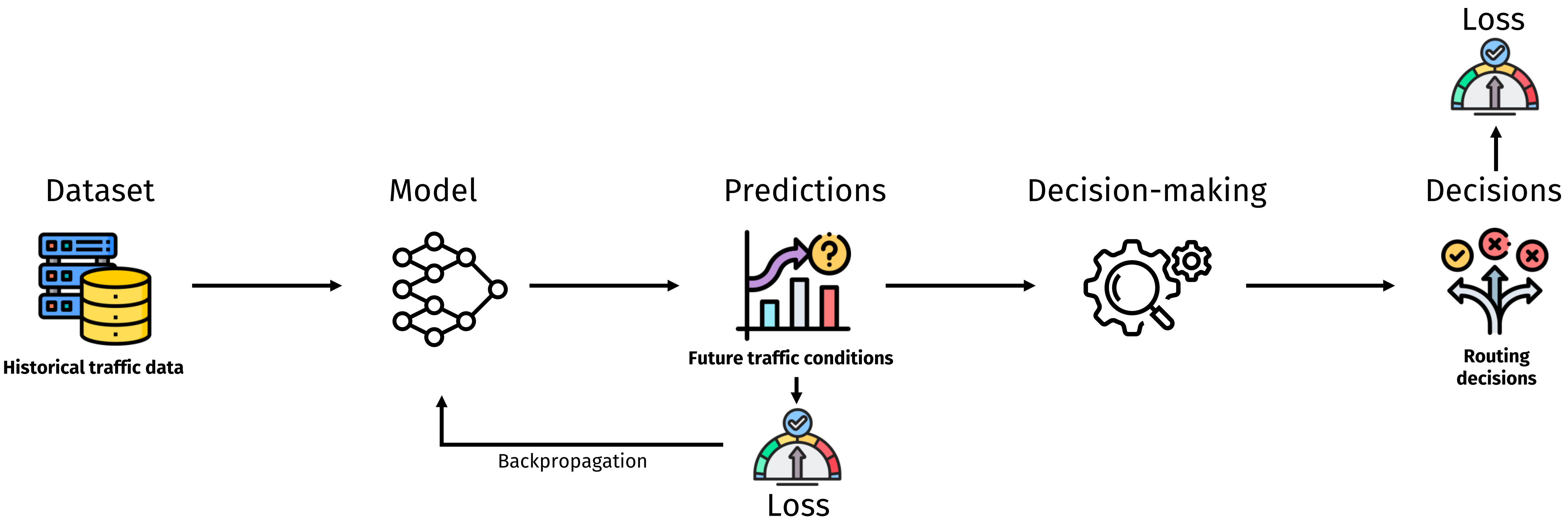
Traditional machine learning
Dataset
Historical traffic data
Model
Predictions
Loss
Backpropagation
Decision-making
Decisions

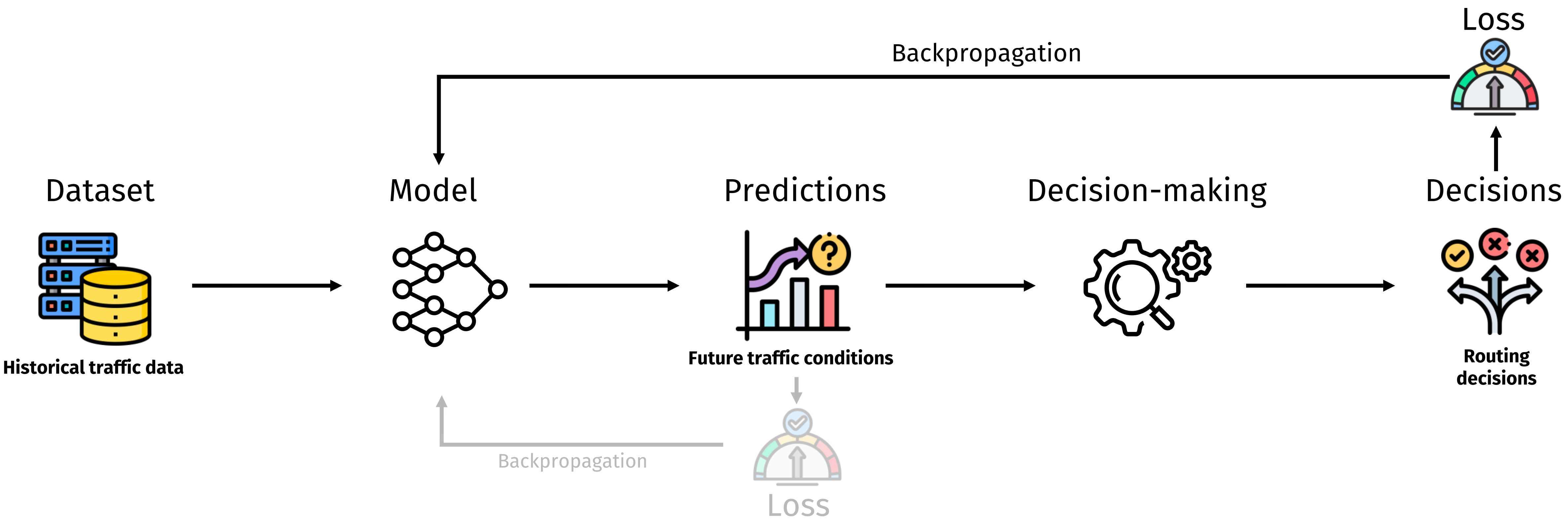

Traditional machine learning
Dataset
Model
Predictions
Decision-making
Decisions
Historical traffic data
Future traffic conditions
Backpropagation
Loss

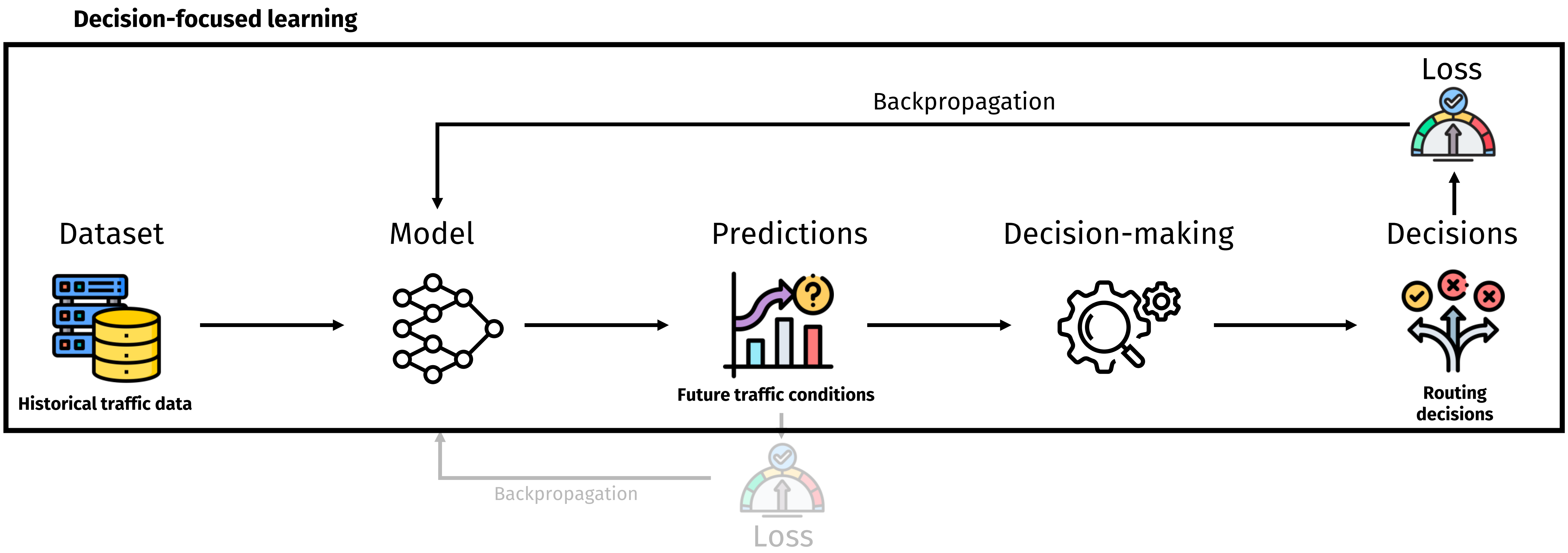


Traditional machine learning
Dataset
Historical traffic data
Model
Predictions
Future traffic conditions
Loss
Backpropagation
Decision-making
Decisions
Routing decisions

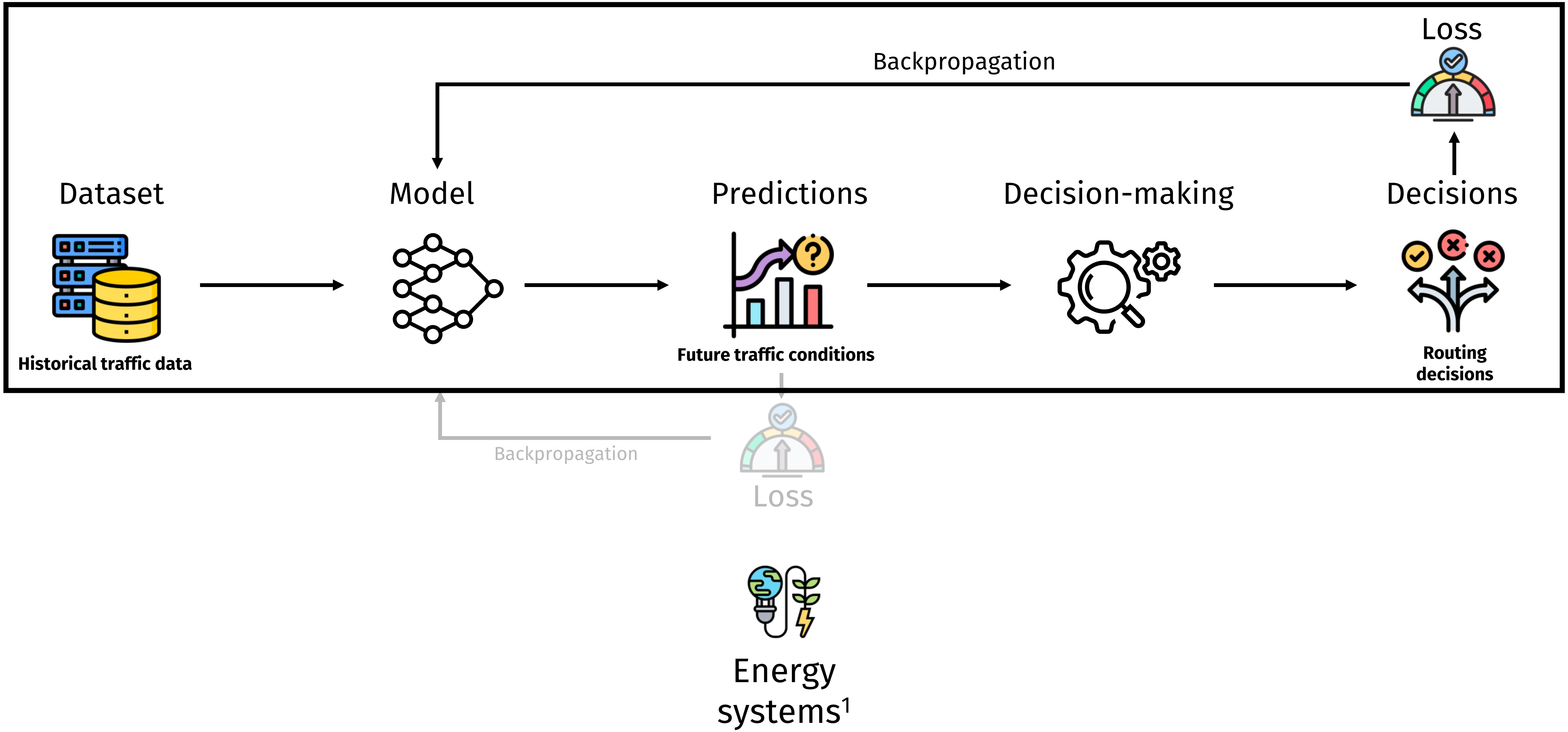

Loss
Traditional machine learning
Dataset
Model
Predictions
Decision-making
Decisions
Historical traffic data
Future traffic conditions
Routing decisions
Backpropagation
Loss


Loss
Dataset
Model
Predictions
Decision-making
Decisions
Historical traffic data
Future traffic conditions
Routing decisions
Backpropagation
Loss


Loss
Backpropagation
Dataset
Model
Predictions
Decision-making
Decisions
Historical traffic data
Future traffic conditions
Routing decisions
Backpropagation
Loss

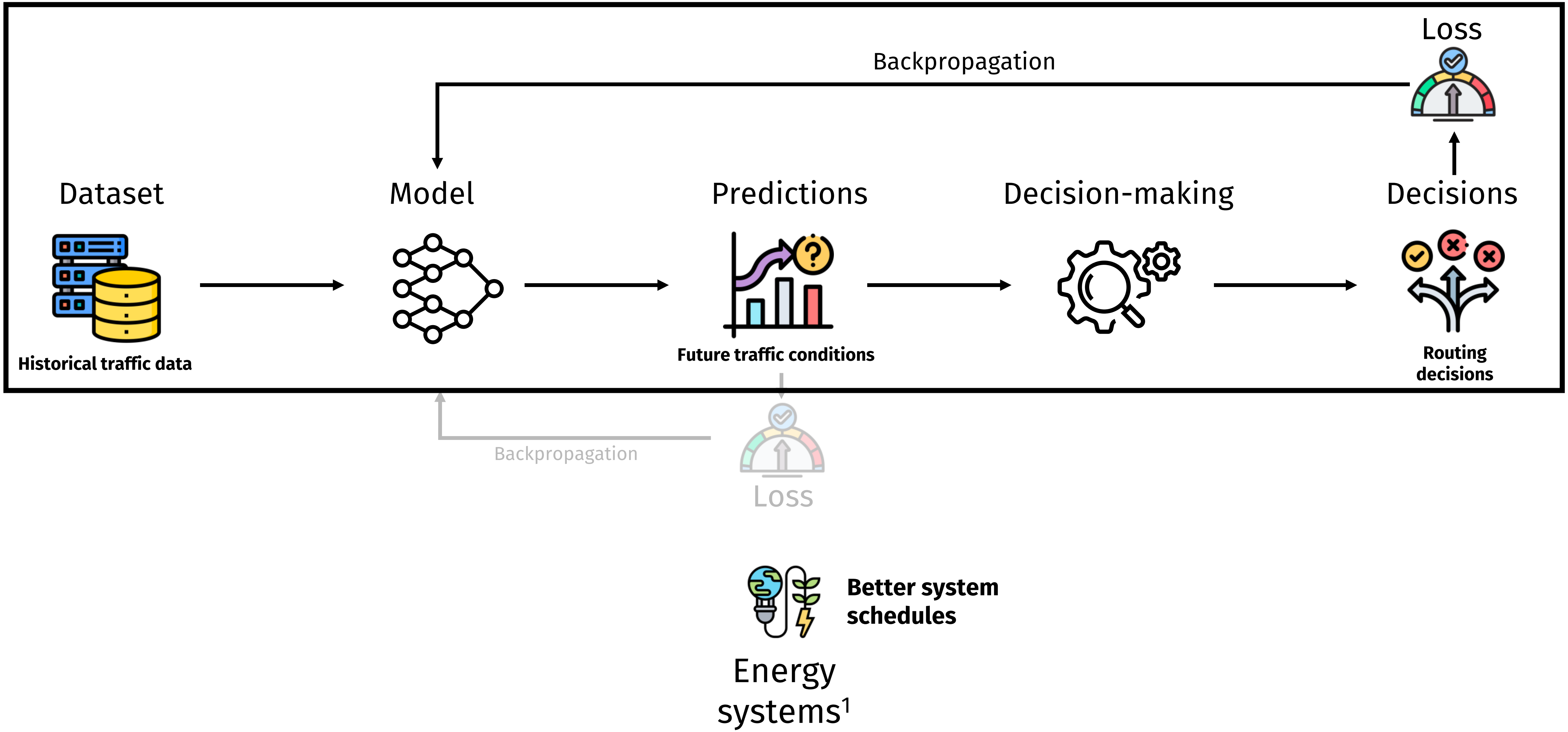

Decision-focused learning
Loss
Backpropagation
Dataset
Model
Predictions
Decision-making
Decisions
Historical traffic data
Future traffic conditions
Routing
decisions
Backpropagation
Loss

Decision-focused learning



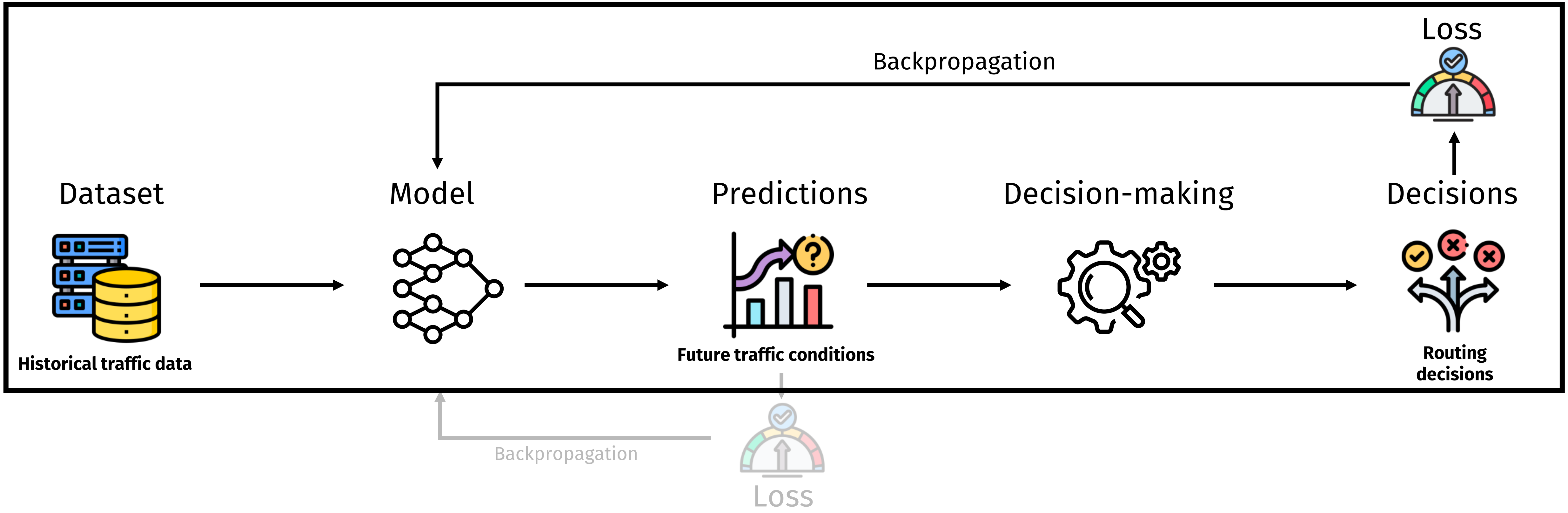
1: Wahdany, Dariush, Carlo Schmitt, and Jochen L. Cremer. "More than accuracy: end-to-end wind power forecasting that optimises the energy system."

Decision-focused learning



1: Wahdany, Dariush, Carlo Schmitt, and Jochen L. Cremer. "More than accuracy: end-to-end wind power forecasting that optimises the energy system."

Decision-focused learning



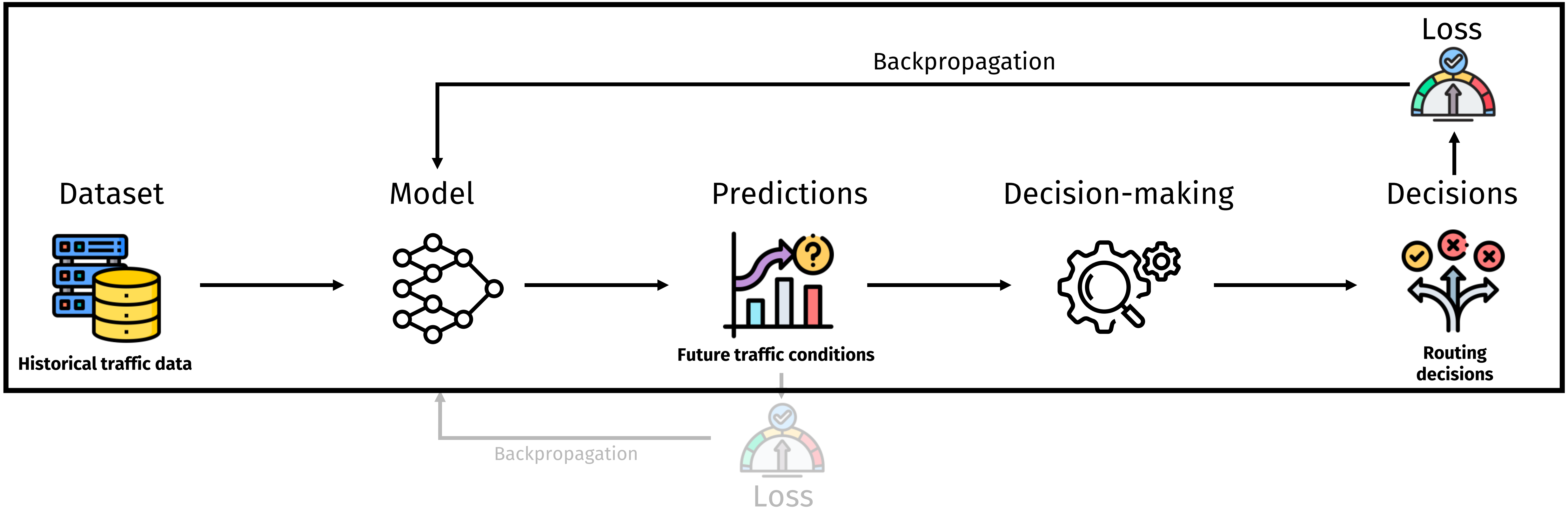
Healthcare[2]

Better system schedules

Energy systems[1]

1: Wahdany, Dariush, Carlo Schmitt, and Jochen L. Cremer. "More than accuracy: end-to-end wind power forecasting that optimises the energy system."
2: Killian, Jackson A., et al. "Learning to prescribe interventions for tuberculosis patients using digital adherence data."

Decision-focused learning



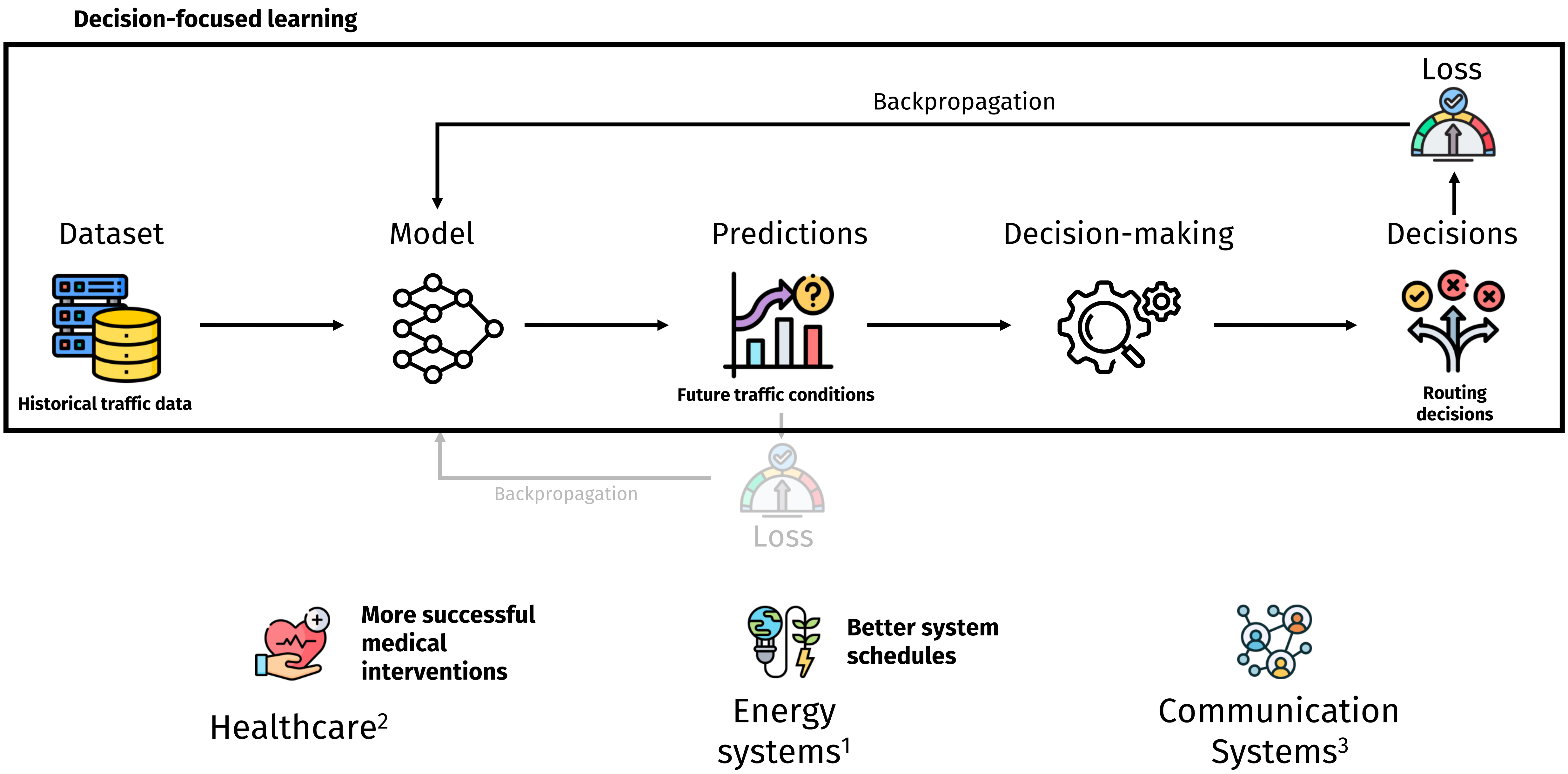
More successful medical interventions

Healthcare[2]

Better system schedules

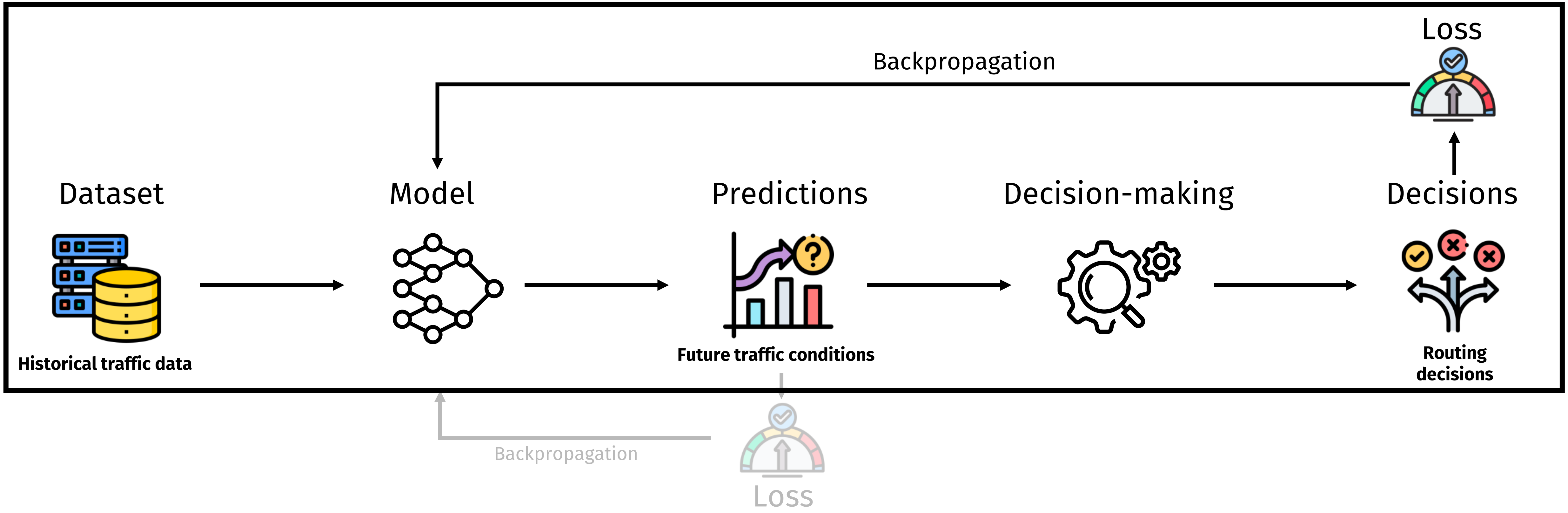
Energy systems[1]

1: Wahdany, Dariush, Carlo Schmitt, and Jochen L. Cremer. "More than accuracy: end-to-end wind power forecasting that optimises the energy system."
2: Killian, Jackson A., et al. "Learning to prescribe interventions for tuberculosis patients using digital adherence data."



1: Wahdany, Dariush, Carlo Schmitt, and Jochen L. Cremer. "More than accuracy: end-to-end wind power forecasting that optimises the energy system.“
2: Killian, Jackson A., et al. "Learning to prescribe interventions for tuberculosis patients using digital adherence data.“
3: Chai, Zhi, et al. "Port selection for fluid antenna systems."

Decision-focused learning



More successful medical interventions

Healthcare[2]

Better system schedules

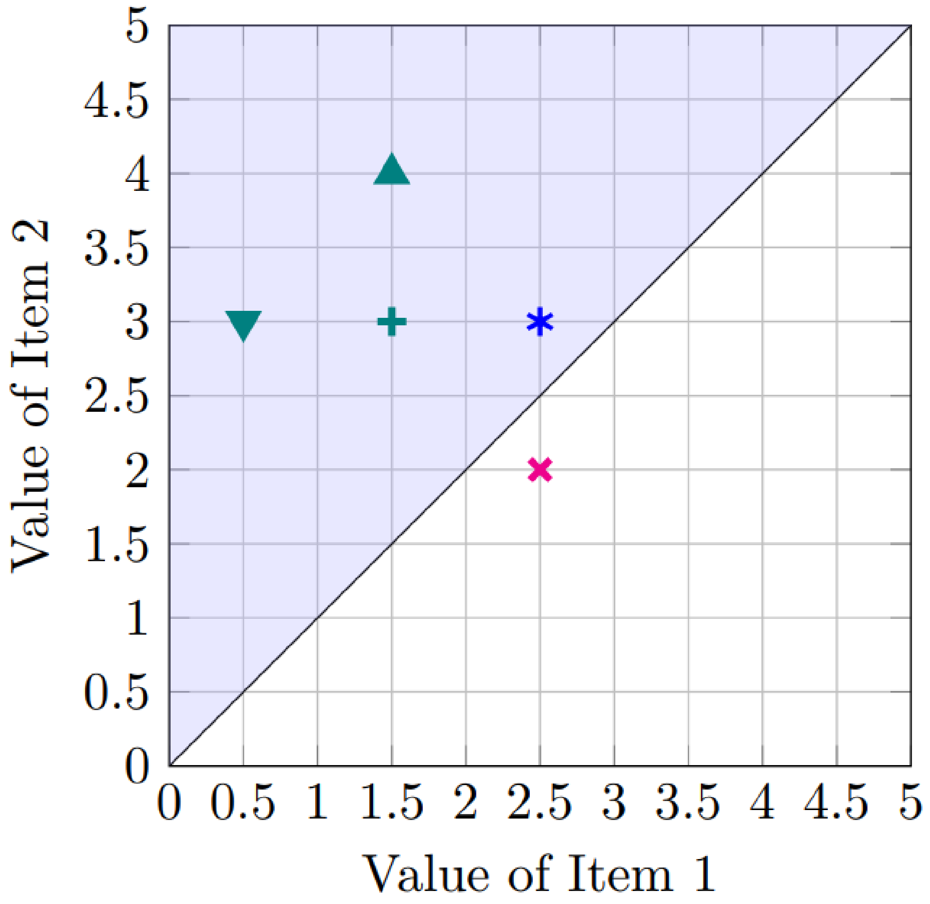
Energy systems[1]

Less system failures

Communication Systems[3]

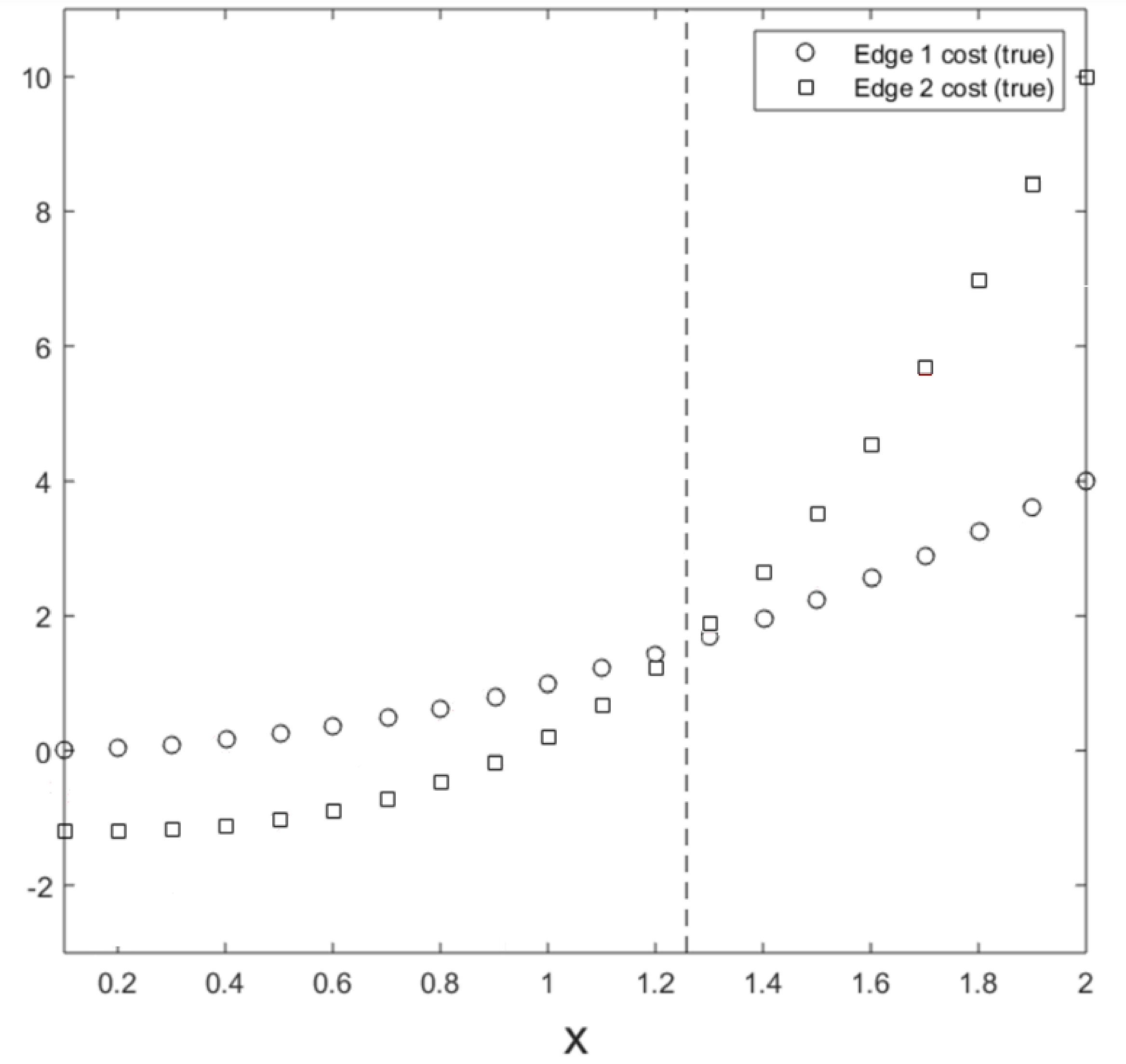
1: Wahdany, Dariush, Carlo Schmitt, and Jochen L. Cremer. "More than accuracy: end-to-end wind power forecasting that optimises the energy system.“
2: Killian, Jackson A., et al. "Learning to prescribe interventions for tuberculosis patients using digital adherence data.“
3: Chai, Zhi, et al. "Port selection for fluid antenna systems."

Why does **traditional** machine learning
not optimize decision quality?

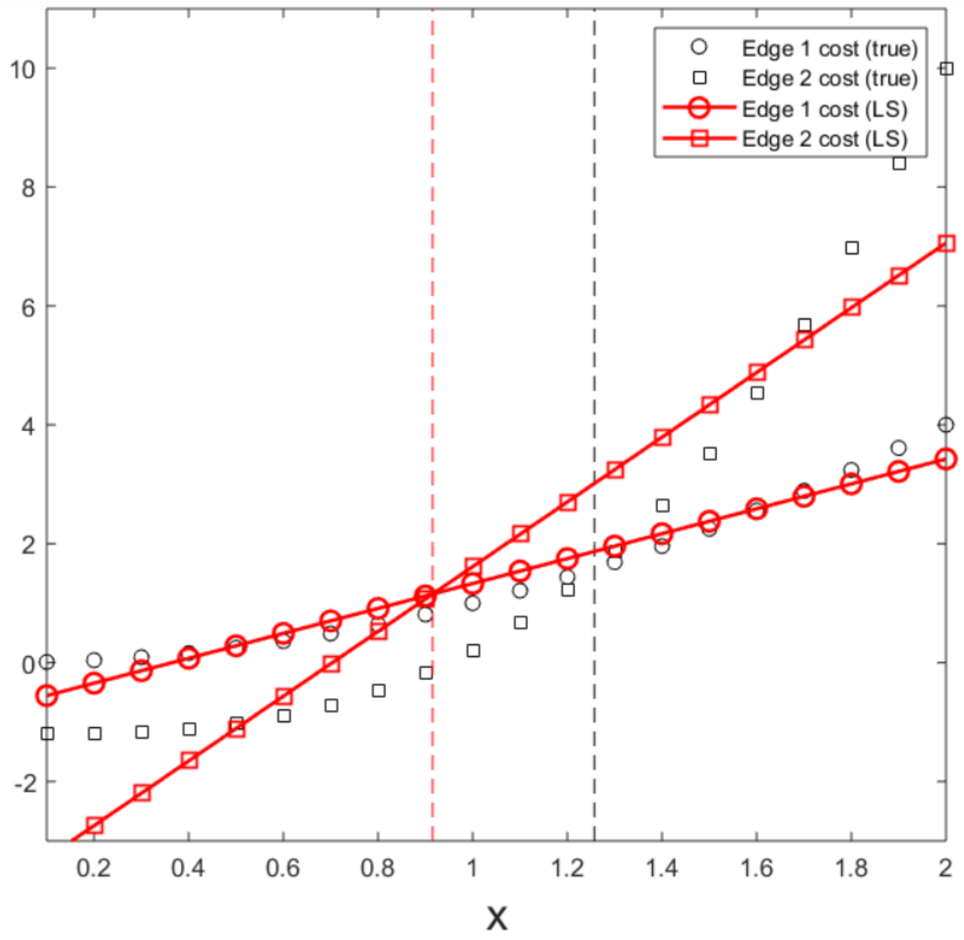


Value of Item 2
Value of Item 1
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

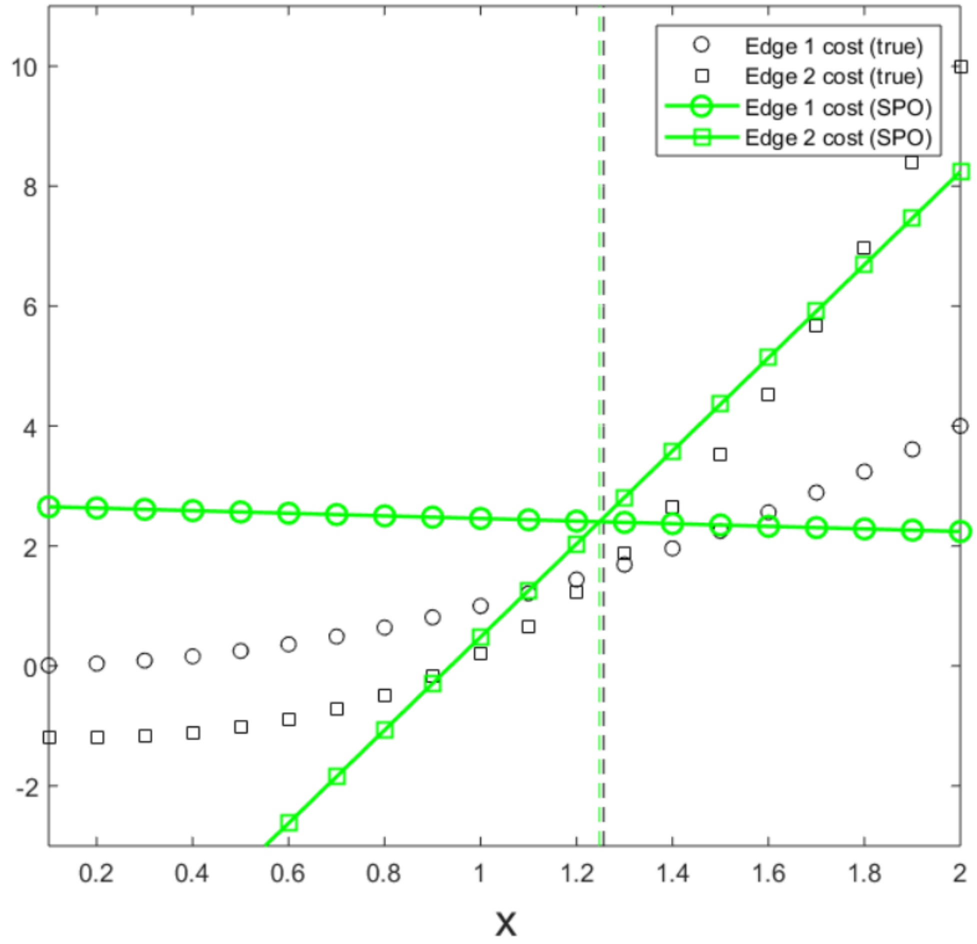
Why does **decision-focused** learning lead to better decision quality?


Edge 1 cost (true)
Edge 2 cost (true)
10
8
6
4
2
0
-2
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x

Traditional machine learning

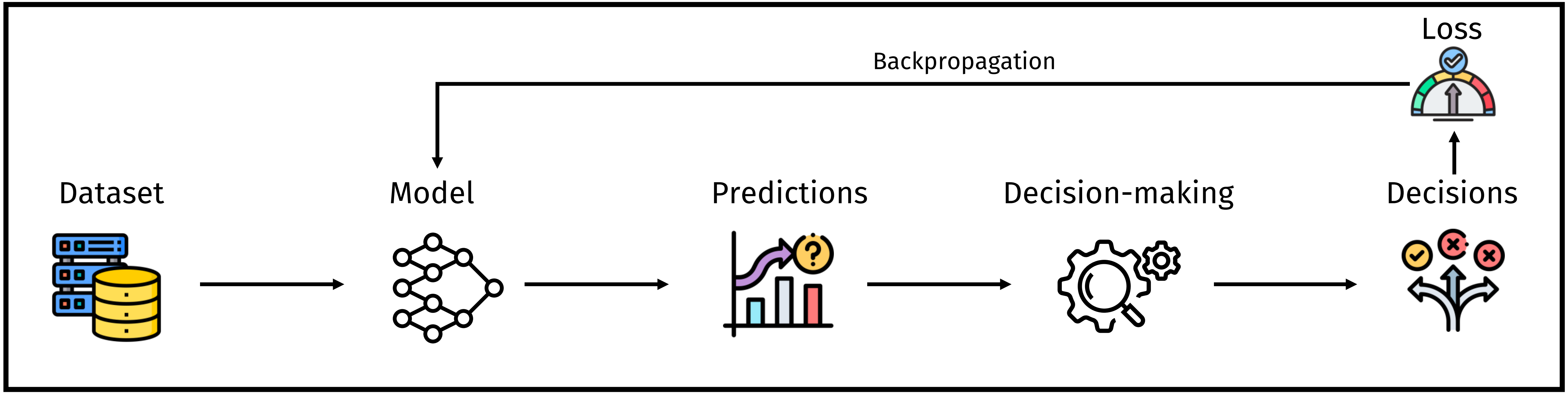


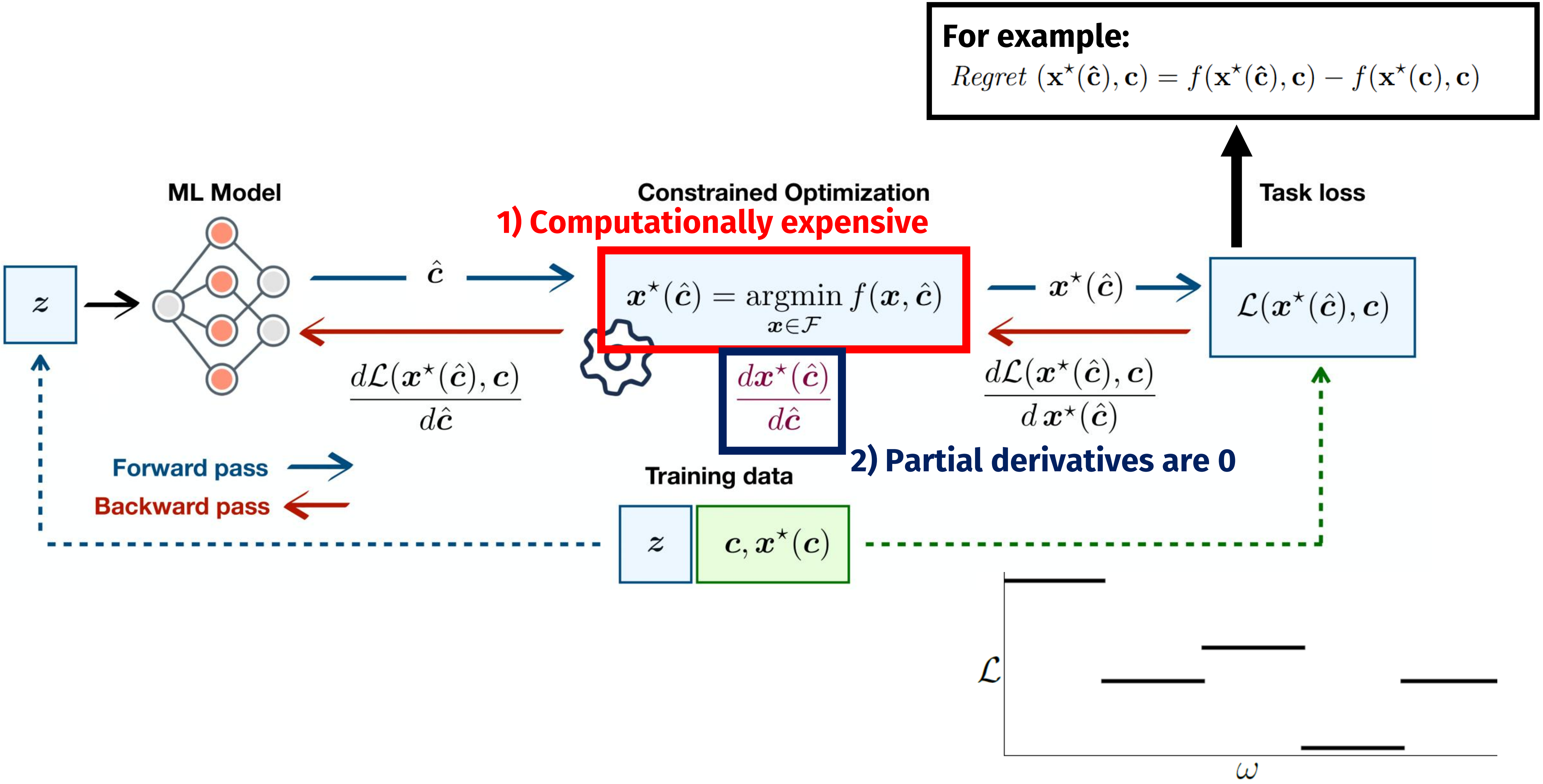
Decision-focused learning



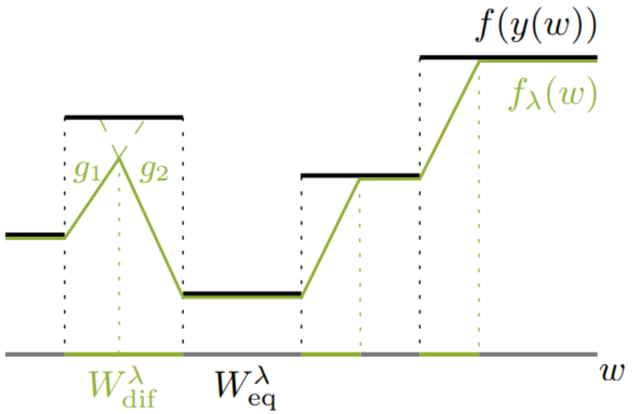
What are the challenges of decision-focused learning?

Decision-focused learning



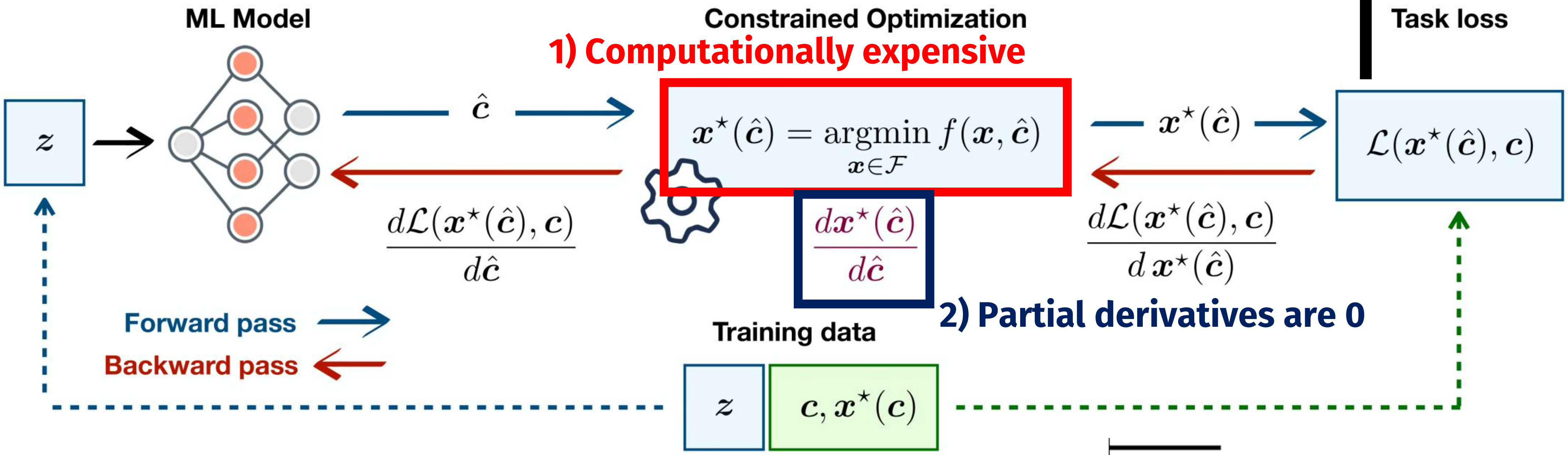

For example:
$Regret\ (\mathbf{x}^\star(\hat{\mathbf{c}}), \mathbf{c}) = f(\mathbf{x}^\star(\hat{\mathbf{c}}), \mathbf{c}) - f(\mathbf{x}^\star(\mathbf{c}), \mathbf{c})$
ML Model
Constrained Optimization
1) Computationally expensive
Task loss
z
$\hat{c}$
$x^\star(\hat{c}) = \operatorname{argmin}_{x \in \mathcal{F}} f(x, \hat{c})$
$x^\star(\hat{c})$
$\mathcal{L}(x^\star(\hat{c}), c)$
$\frac{d\mathcal{L}(x^\star(\hat{c}), c)}{d\hat{c}}$
$\frac{dx^\star(\hat{c})}{d\hat{c}}$
$\frac{d\mathcal{L}(x^\star(\hat{c}), c)}{d\,x^\star(\hat{c})}$
2) Partial derivatives are 0
Forward pass
Backward pass
Training data
z
$c, x^\star(c)$
$\mathcal{L}$
ω

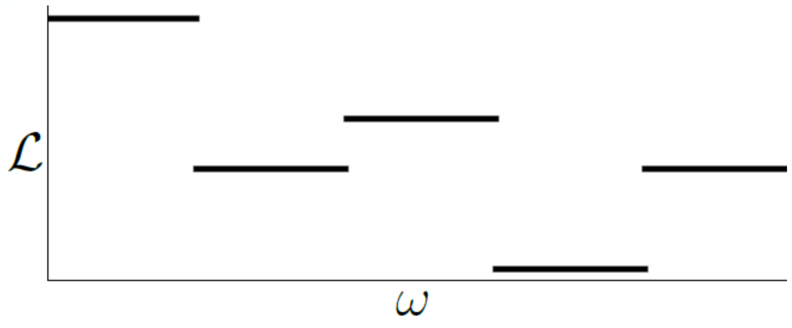
How are these challenges typically addressed?



For example:

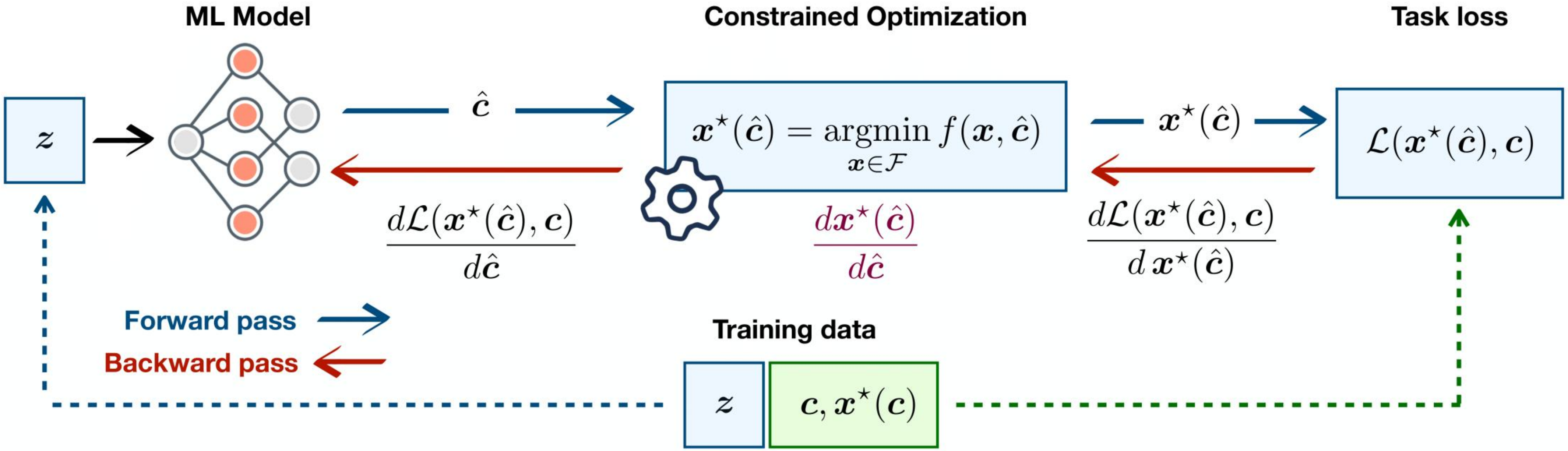
$$Regret\ (\mathbf{x}^\star(\hat{\mathbf{c}}), \mathbf{c}) = f(\mathbf{x}^\star(\hat{\mathbf{c}}), \mathbf{c}) - f(\mathbf{x}^\star(\mathbf{c}), \mathbf{c})$$





Mandi, J., Kotary, J., Berden, S., Mulamba, M., Bucarey, V., Guns, T., Fioretto, F.
Decision-Focused Learning: Foundations, State of the Art, Benchmark and Future Opportunities.
In: *Journal of Artificial Intelligence Research (JAIR)*

Part 2: how to do it quickly

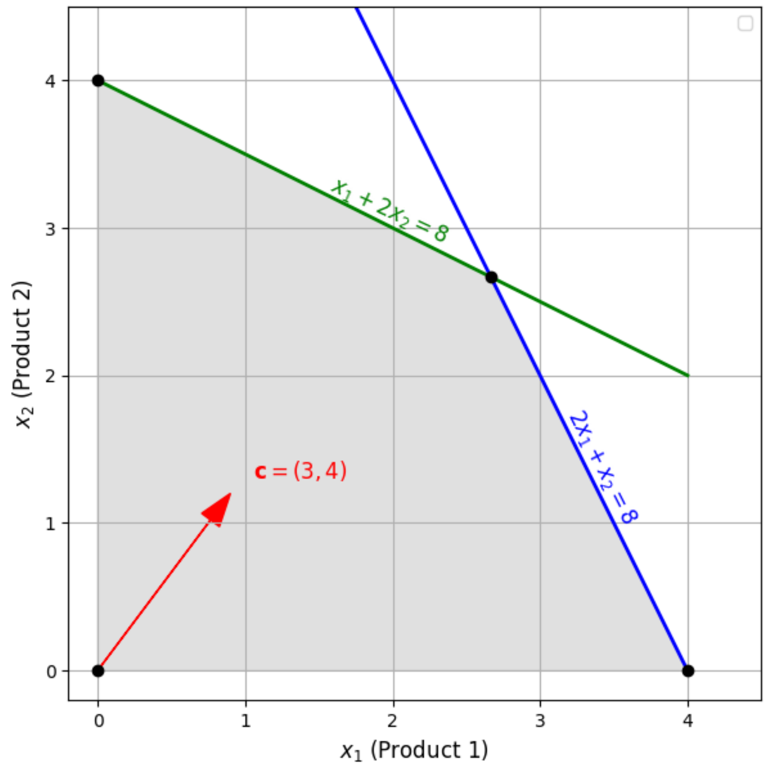

ML Model
Constrained Optimization
Task loss
z
$\hat{c}$
$x^\star(\hat{c}) = \arg\min_{x \in \mathcal{F}} f(x, \hat{c})$
$x^\star(\hat{c})$
$\mathcal{L}(x^\star(\hat{c}), c)$
$\frac{d\mathcal{L}(x^\star(\hat{c}), c)}{d\hat{c}}$
$\frac{dx^\star(\hat{c})}{d\hat{c}}$
$\frac{d\mathcal{L}(x^\star(\hat{c}), c)}{d\,x^\star(\hat{c})}$
Forward pass
Backward pass
Training data
z
$c, x^\star(c)$

Linear programming

	Product 1 (x_1)	Product 2 (x_2)	Availability
Profit	3	4	/
Labor (hours/unit)	2	1	8
Material (kg/unit)	1	2	8

Question: how much product 1 (x_1) and product 2 (x_2) should we produce?

$$\begin{aligned} \text{Maximize:} \quad & z = 3x_1 + 4x_2 \\ \text{subject to:} \quad & 2x_1 + 1x_2 \leq 8 \quad \text{(labor constraint)} \\ & 1x_1 + 2x_2 \leq 8 \quad \text{(material constraint)} \\ & x_1 \geq 0,\ x_2 \geq 0 \quad \text{(nonnegativity)} \end{aligned}$$

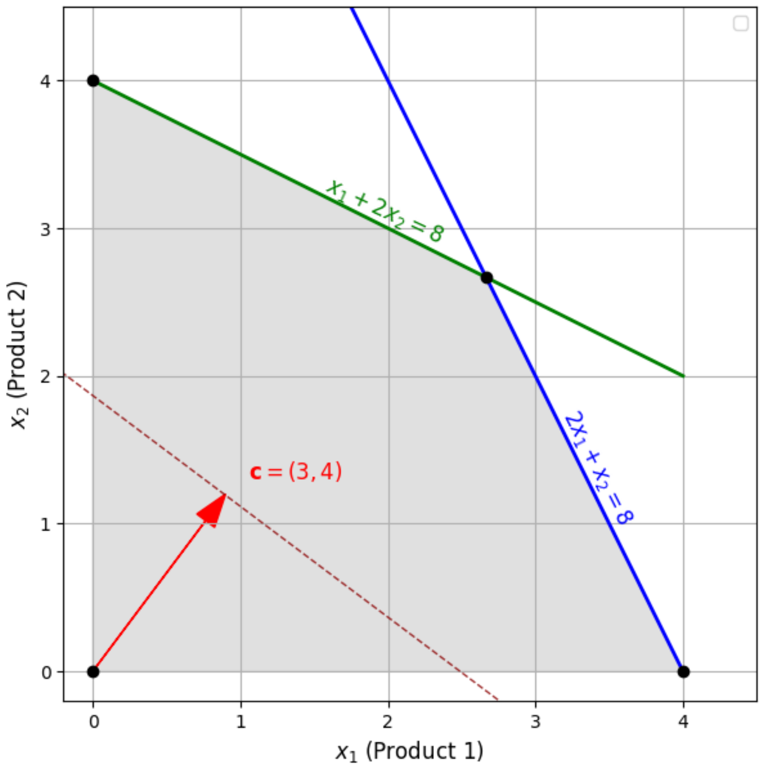


Linear programming

	Product 1 (x_1)	Product 2 (x_2)	Availability
Profit	3	4	/
Labor (hours/unit)	2	1	8
Material (kg/unit)	1	2	8

Question: how much product 1 (x_1) and product 2 (x_2) should we produce?

$$\begin{aligned} \text{Maximize:} \quad & z = 3x_1 + 4x_2 \\ \text{subject to:} \quad & 2x_1 + 1x_2 \leq 8 \quad \text{(labor constraint)} \\ & 1x_1 + 2x_2 \leq 8 \quad \text{(material constraint)} \\ & x_1 \geq 0,\ x_2 \geq 0 \quad \text{(nonnegativity)} \end{aligned}$$

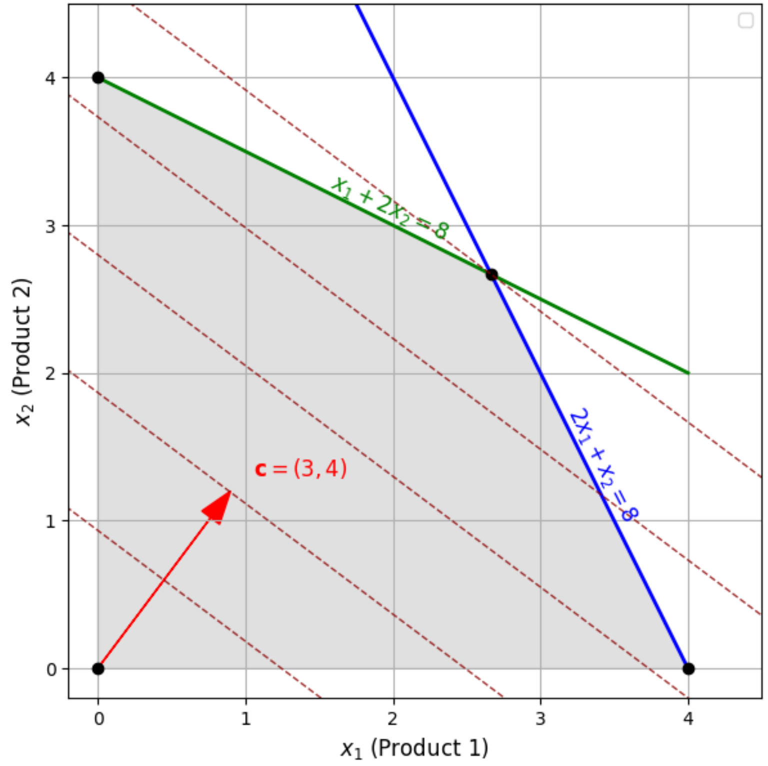


Linear programming

	Product 1 (x_1)	Product 2 (x_2)	Availability
Profit	3	4	/
Labor (hours/unit)	2	1	8
Material (kg/unit)	1	2	8

Question: how much product 1 (x_1) and product 2 (x_2) should we produce?

$$\begin{aligned}
\text{Maximize:} \quad & z = 3x_1 + 4x_2 \\
\text{subject to:} \quad & 2x_1 + 1x_2 \leq 8 \quad \text{(labor constraint)} \\
& 1x_1 + 2x_2 \leq 8 \quad \text{(material constraint)} \\
& x_1 \geq 0,\ x_2 \geq 0 \quad \text{(nonnegativity)}
\end{aligned}$$

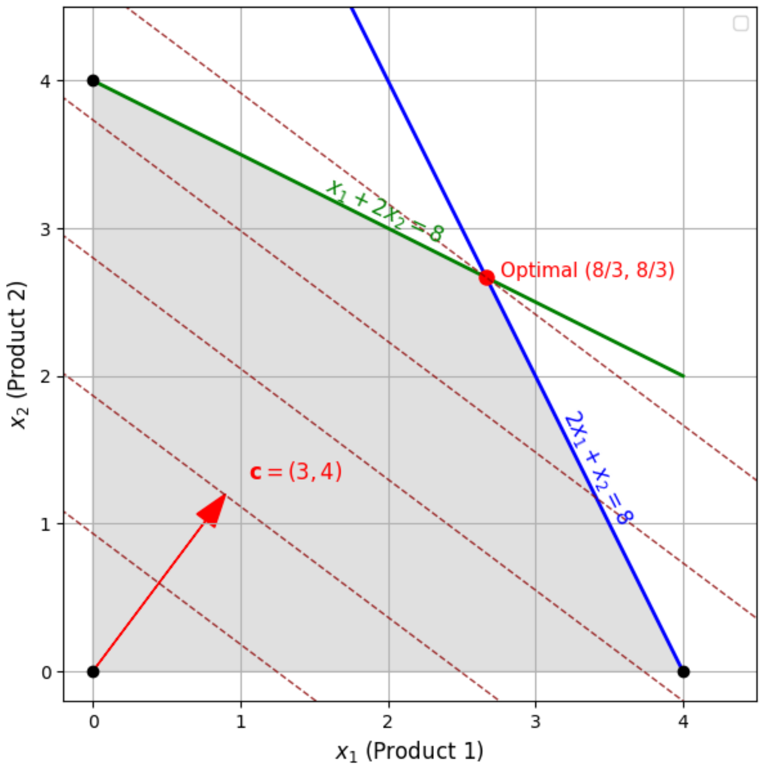


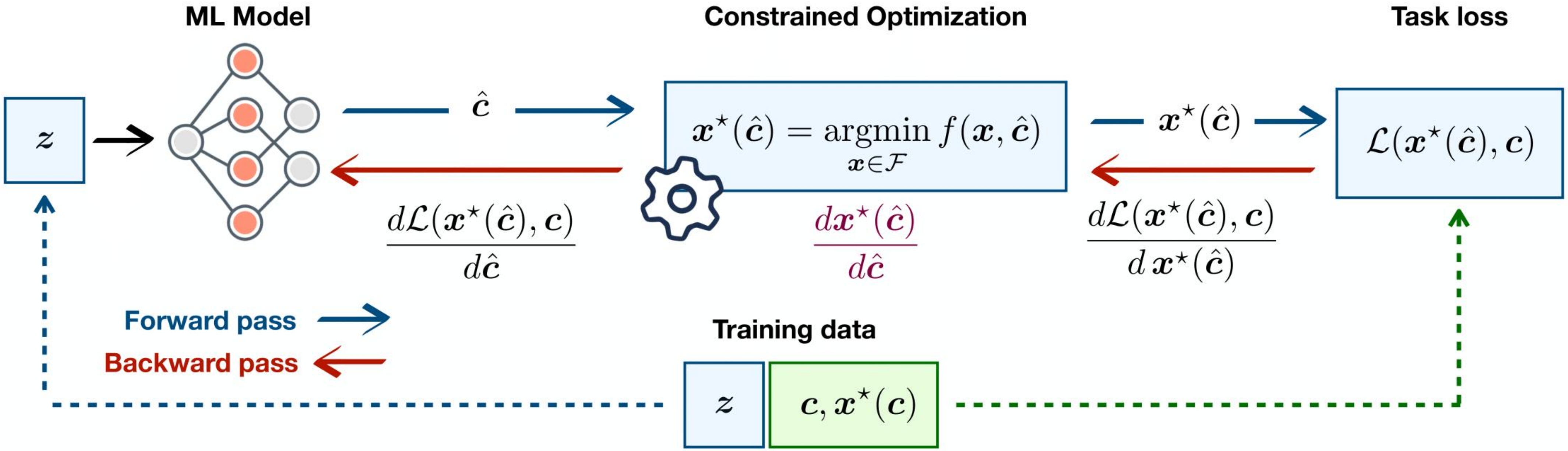
Linear programming

	Product 1 (x_1)	Product 2 (x_2)	Availability
Profit	3	4	/
Labor (hours/unit)	2	1	8
Material (kg/unit)	1	2	8

Question: how much product 1 (x_1) and product 2 (x_2) should we produce?

$$\begin{aligned} \text{Maximize:} \quad & z = 3x_1 + 4x_2 \\ \text{subject to:} \quad & 2x_1 + 1x_2 \le 8 \quad \text{(labor constraint)} \\ & 1x_1 + 2x_2 \le 8 \quad \text{(material constraint)} \\ & x_1 \ge 0,\ x_2 \ge 0 \quad \text{(nonnegativity)} \end{aligned}$$



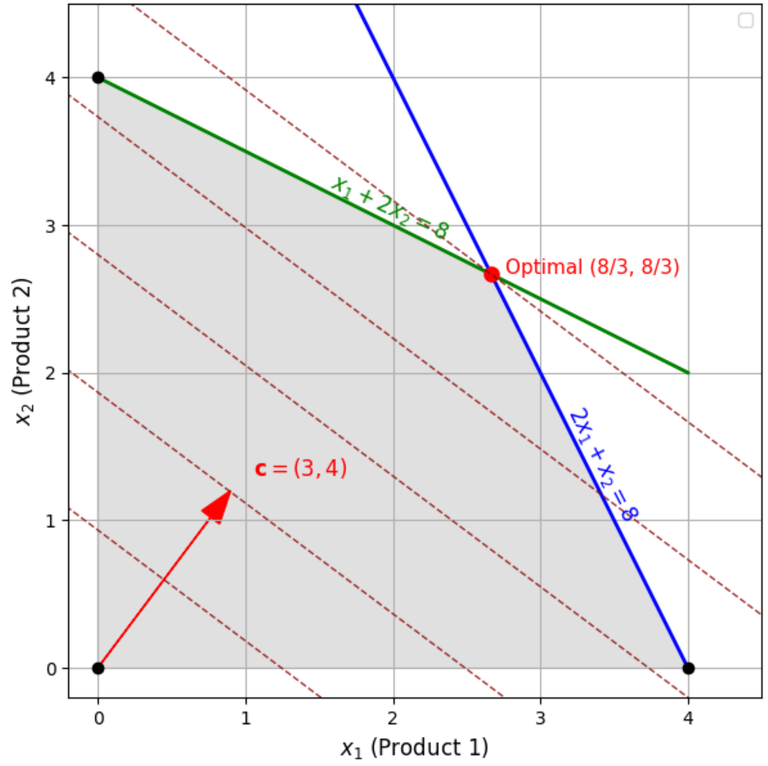


ML Model
Constrained Optimization
Task loss
z
$\hat{c}$
$x^\star(\hat{c}) = \operatorname{argmin}_{x \in \mathcal{F}} f(x, \hat{c})$
$x^\star(\hat{c})$
$\mathcal{L}(x^\star(\hat{c}), c)$
$\frac{d\mathcal{L}(x^\star(\hat{c}), c)}{d\hat{c}}$
$\frac{dx^\star(\hat{c})}{d\hat{c}}$
$\frac{d\mathcal{L}(x^\star(\hat{c}), c)}{d\,x^\star(\hat{c})}$
Forward pass
Backward pass
Training data
z
$c, x^\star(c)$

Linear programming

	Product 1 (x_1)	Product 2 (x_2)	Availability
Profit	3	4	/
Labor (hours/unit)	2	1	8
Material (kg/unit)	1	2	8

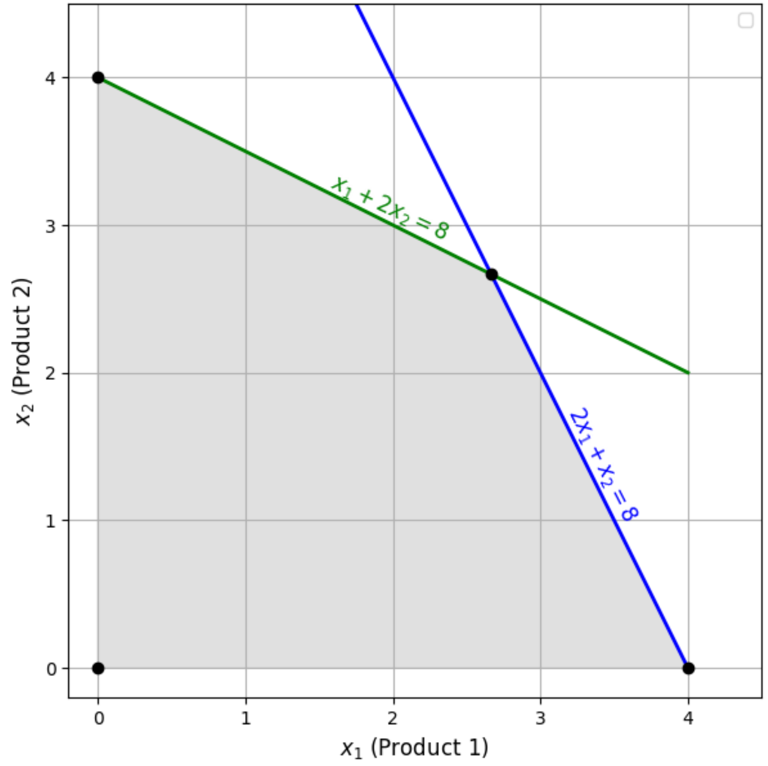
$$\begin{aligned} \text{Maximize:} \quad & z = 3x_1 + 4x_2 \\ \text{subject to:} \quad & 2x_1 + 1x_2 \leq 8 \quad \text{(labor constraint)} \\ & 1x_1 + 2x_2 \leq 8 \quad \text{(material constraint)} \\ & x_1 \geq 0,\ x_2 \geq 0 \quad \text{(nonnegativity)} \end{aligned}$$



Linear programming

	Product 1 (x_1)	Product 2 (x_2)	Availability
Profit	c_1	c_2	/
Labor (hours/unit)	2	1	8
Material (kg/unit)	1	2	8

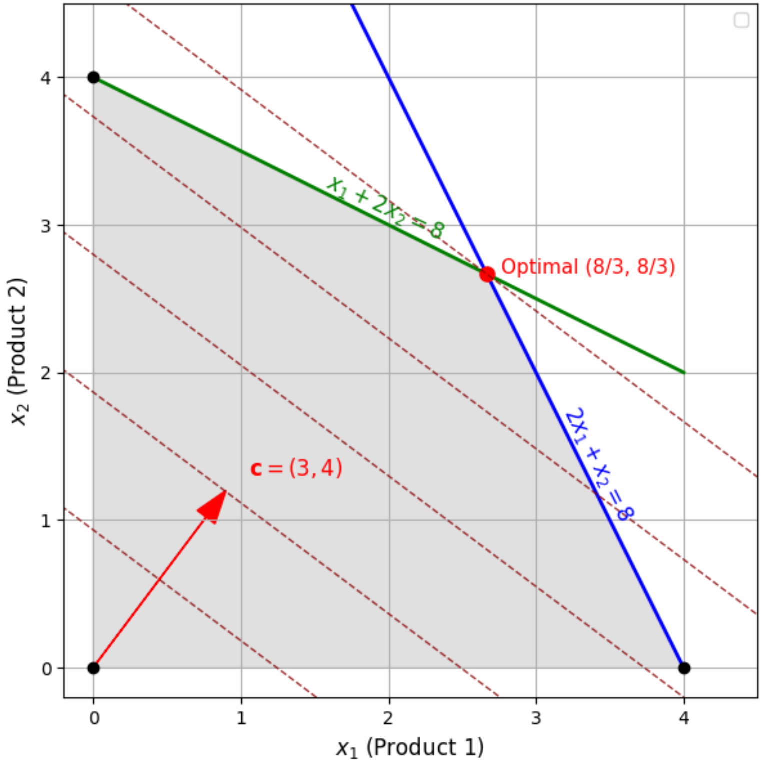
$$\begin{aligned} \text{Maximize:} \quad & z = c_1 x_1 + c_2 x_2 \\ \text{subject to:} \quad & 2x_1 + 1x_2 \leq 8 \quad \text{(labor constraint)} \\ & 1x_1 + 2x_2 \leq 8 \quad \text{(material constraint)} \\ & x_1 \geq 0,\ x_2 \geq 0 \quad \text{(nonnegativity)} \end{aligned}$$



Linear programming

	Product 1 (x_1)	Product 2 (x_2)	Availability
Profit	c_1	c_2	/
Labor (hours/unit)	2	1	8
Material (kg/unit)	1	2	8

$$
\begin{aligned}
\text{Maximize:} \quad & z = c_1 x_1 + c_2 x_2 \\
\text{subject to:} \quad & 2x_1 + 1x_2 \le 8 \quad \text{(labor constraint)} \\
& 1x_1 + 2x_2 \le 8 \quad \text{(material constraint)} \\
& x_1 \ge 0,\ x_2 \ge 0 \quad \text{(nonnegativity)}
\end{aligned}
$$

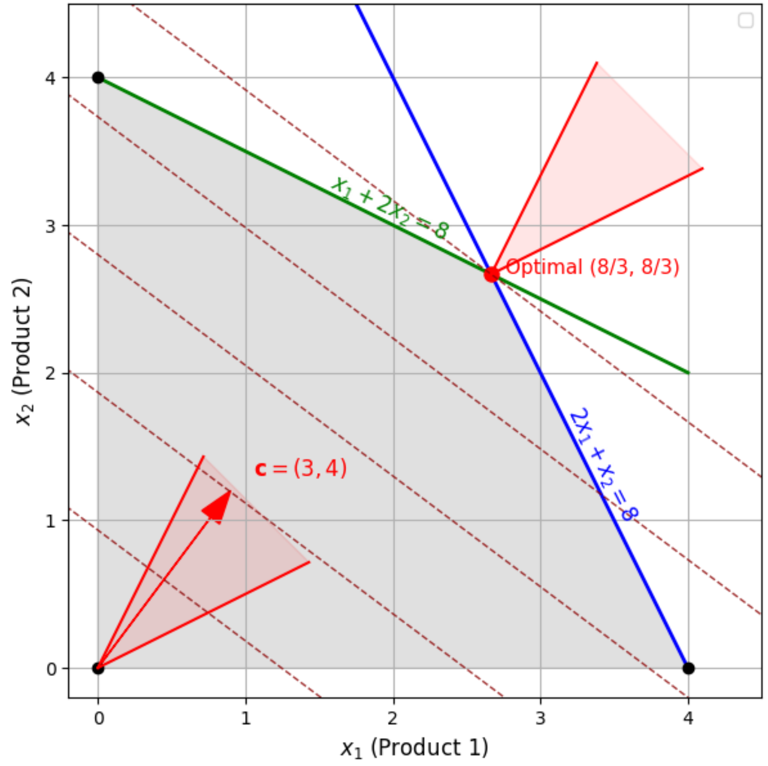


Linear programming

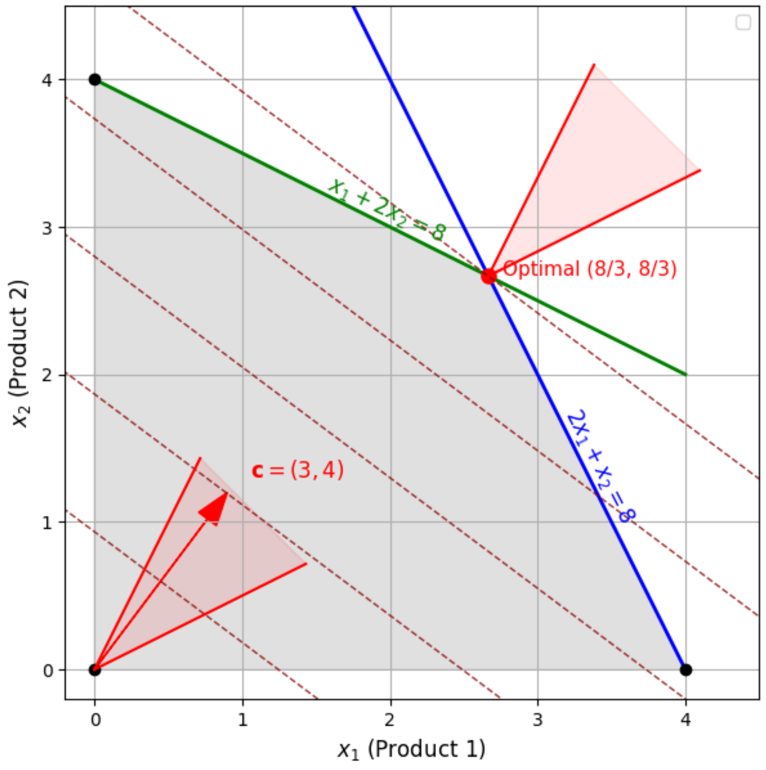
	Product 1 (x_1)	Product 2 (x_2)	Availability
Profit	c_1	c_2	/
Labor (hours/unit)	2	1	8
Material (kg/unit)	1	2	8

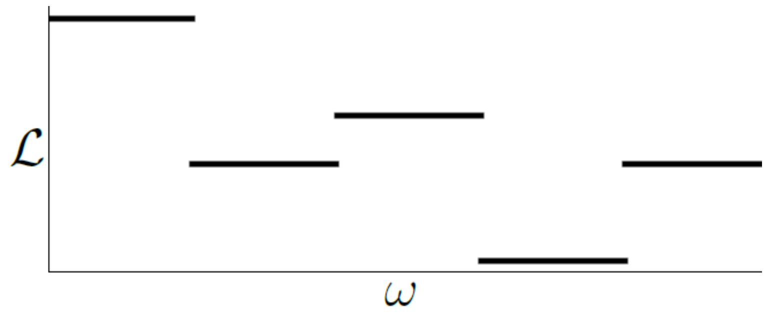
$$\text{Maximize:} \quad z = c_1 x_1 + c_2 x_2$$

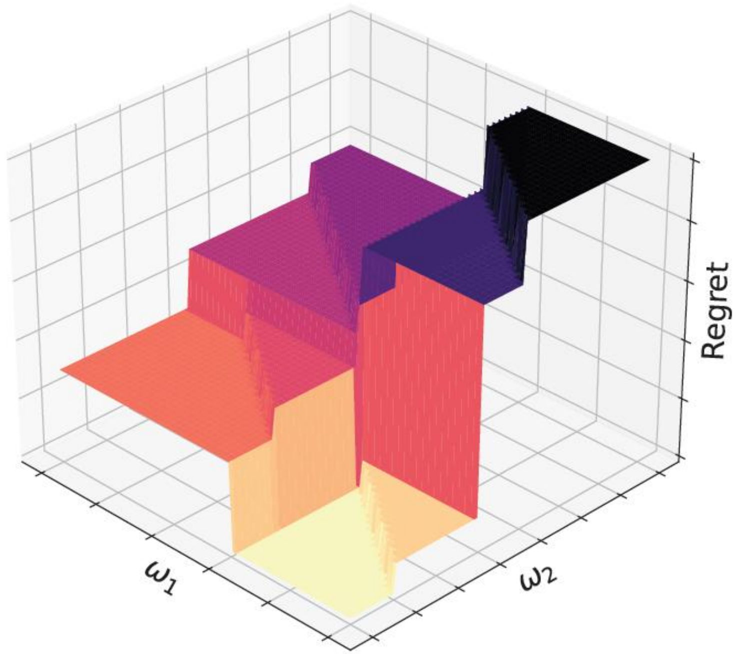
$$\begin{aligned} \text{subject to:} \quad & 2x_1 + 1x_2 \leq 8 \quad \text{(labor constraint)} \\ & 1x_1 + 2x_2 \leq 8 \quad \text{(material constraint)} \\ & x_1 \geq 0, \; x_2 \geq 0 \quad \text{(nonnegativity)} \end{aligned}$$

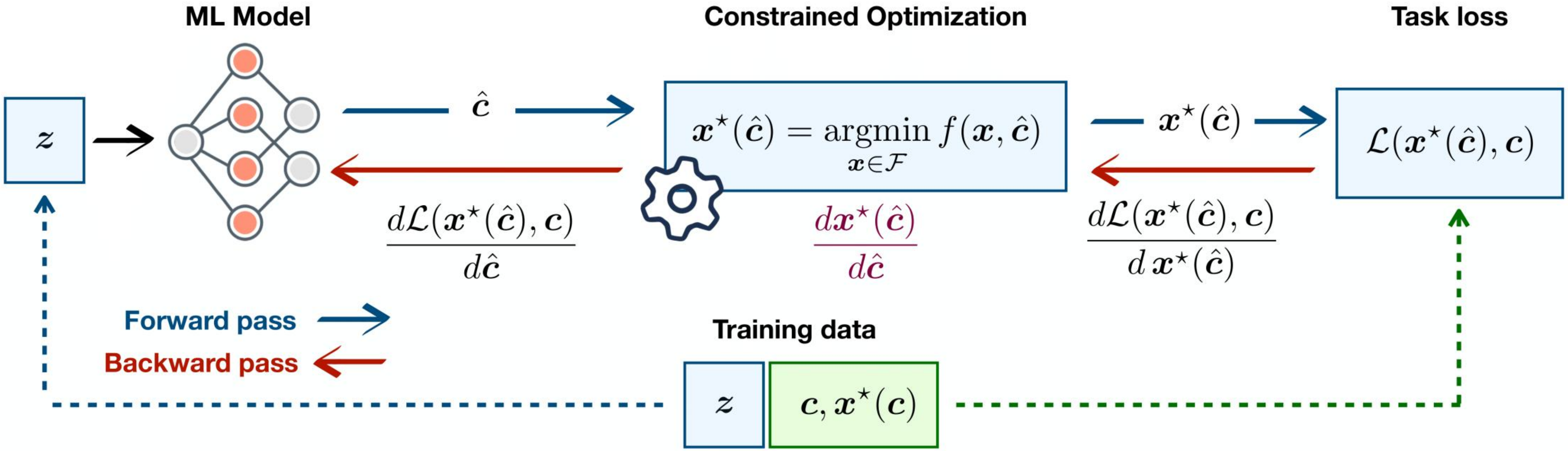


Zero-valued partial derivatives

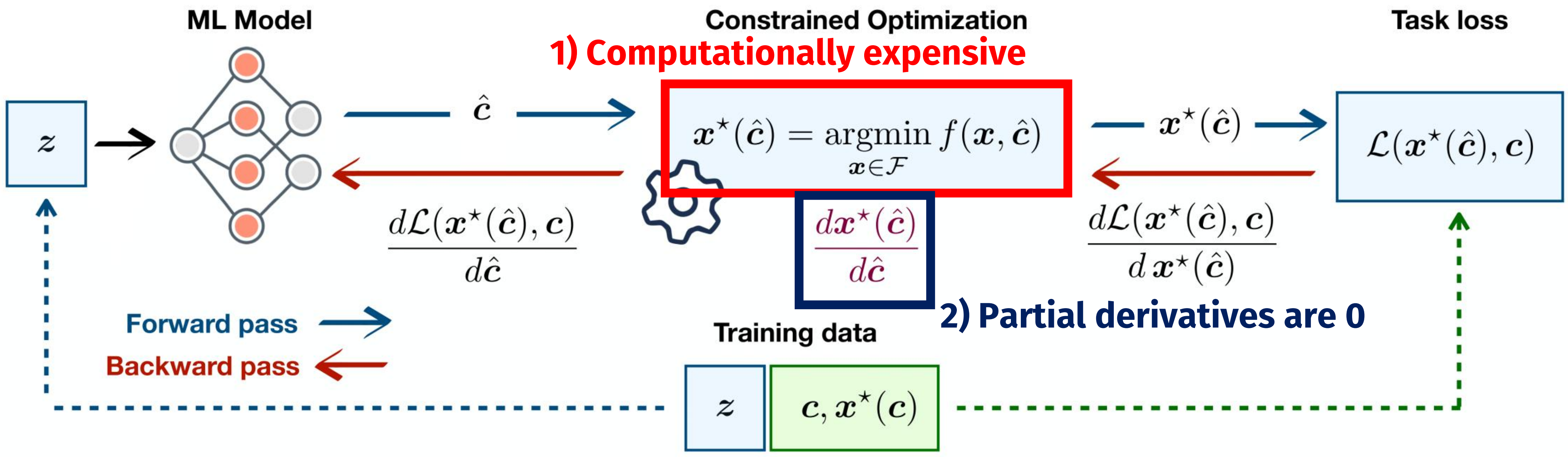




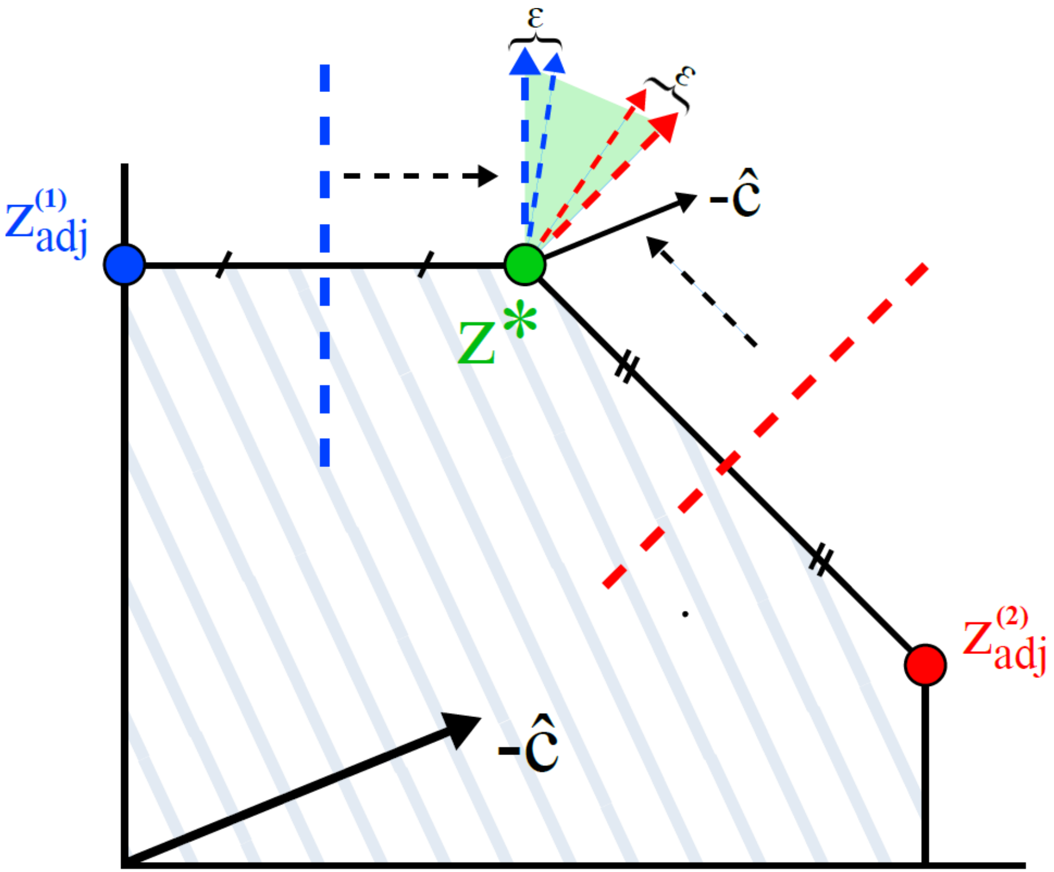




ML Model
Constrained Optimization
Task loss
$\boldsymbol{z}$
$\hat{\boldsymbol{c}}$
$\boldsymbol{x}^\star(\hat{\boldsymbol{c}}) = \underset{\boldsymbol{x}\in\mathcal{F}}{\operatorname{argmin}} f(\boldsymbol{x}, \hat{\boldsymbol{c}})$
$\boldsymbol{x}^\star(\hat{\boldsymbol{c}})$
$\mathcal{L}(\boldsymbol{x}^\star(\hat{\boldsymbol{c}}), \boldsymbol{c})$
$\frac{d\mathcal{L}(\boldsymbol{x}^\star(\hat{\boldsymbol{c}}), \boldsymbol{c})}{d\hat{\boldsymbol{c}}}$
$\frac{d\boldsymbol{x}^\star(\hat{\boldsymbol{c}})}{d\hat{\boldsymbol{c}}}$
$\frac{d\mathcal{L}(\boldsymbol{x}^\star(\hat{\boldsymbol{c}}), \boldsymbol{c})}{d\,\boldsymbol{x}^\star(\hat{\boldsymbol{c}})}$
Forward pass
Backward pass
Training data
$\boldsymbol{z}$
$\boldsymbol{c}, \boldsymbol{x}^\star(\boldsymbol{c})$


ML Model
Constrained Optimization
Task loss
1) Computationally expensive
z
$\hat{c}$
$x^\star(\hat{c}) = \operatorname{argmin}_{x \in \mathcal{F}} f(x, \hat{c})$
$x^\star(\hat{c})$
$\mathcal{L}(x^\star(\hat{c}), c)$
$\frac{d\mathcal{L}(x^\star(\hat{c}), c)}{d\hat{c}}$
$\frac{dx^\star(\hat{c})}{d\hat{c}}$
$\frac{d\mathcal{L}(x^\star(\hat{c}), c)}{d\,x^\star(\hat{c})}$
2) Partial derivatives are 0
Forward pass
Backward pass
Training data
z
$c, x^\star(c)$

Our solution: a surrogate loss



$$\mathcal{L}_{\text{AVA}}(\hat{c}, z^\star(c)) = \sum_{z_{adj} \in Z_{adj}(z^\star(c))} \max(\hat{c}^\top z^\star(c) - \hat{c}^\top z_{adj}, -\epsilon) \quad \text{where } \epsilon \geq 0$$

LAVA loss

$$\mathcal{L}_{\text{AVA}}(\hat{c}, z^\star(c)) = \sum_{z_{adj} \in Z_{adj}(z^\star(c))} \max(\hat{c}^\top z^\star(c) - \hat{c}^\top z_{adj}, -\epsilon) \quad \text{where } \epsilon \geq 0$$

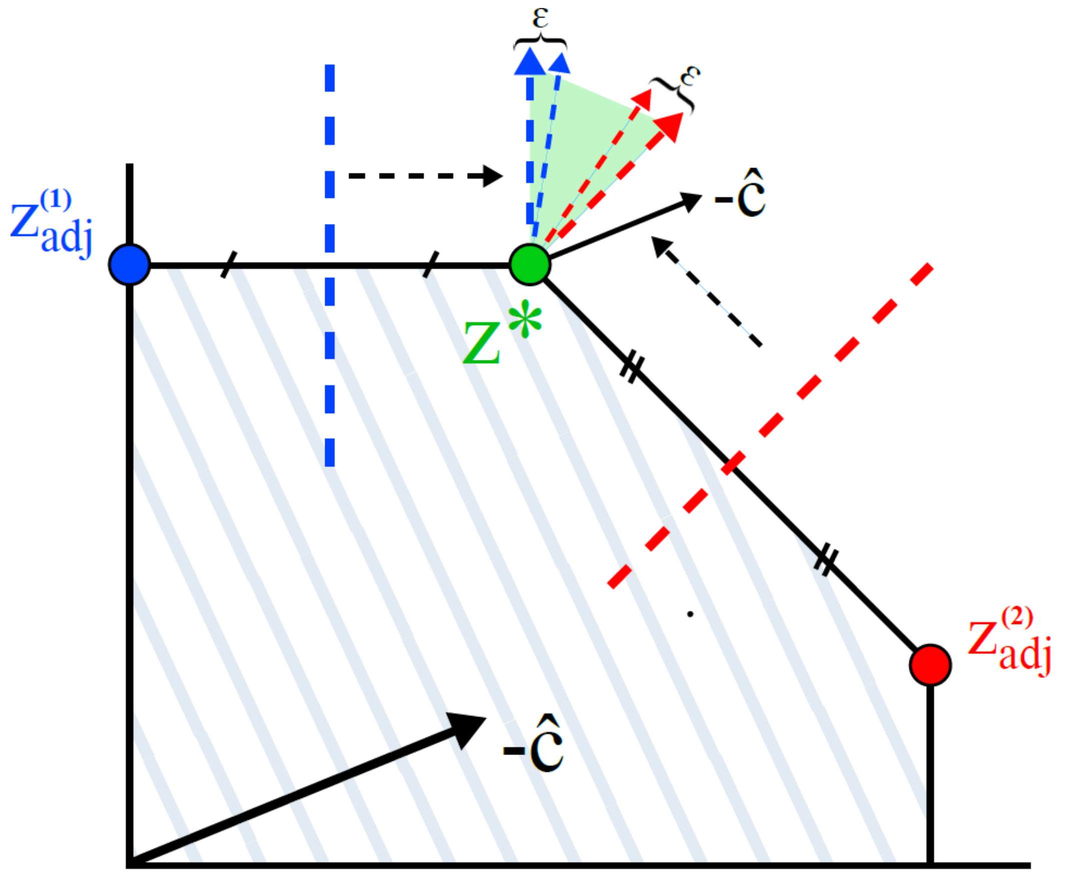
Advantages:

- Differentiable
- Convex
- Does not require access to true *c*
- Efficient to evaluate (solver-free)

Disadvantage:

- Must compute adjacent vertices

Our solution: a surrogate loss



$$\mathcal{L}_{\text{AVA}}(\hat{c}, z^\star(c)) = \sum_{z_{adj} \in Z_{adj}(z^\star(c))} \max(\hat{c}^\top z^\star(c) - \hat{c}^\top z_{adj}, -\epsilon) \quad \text{where } \epsilon \geq 0$$

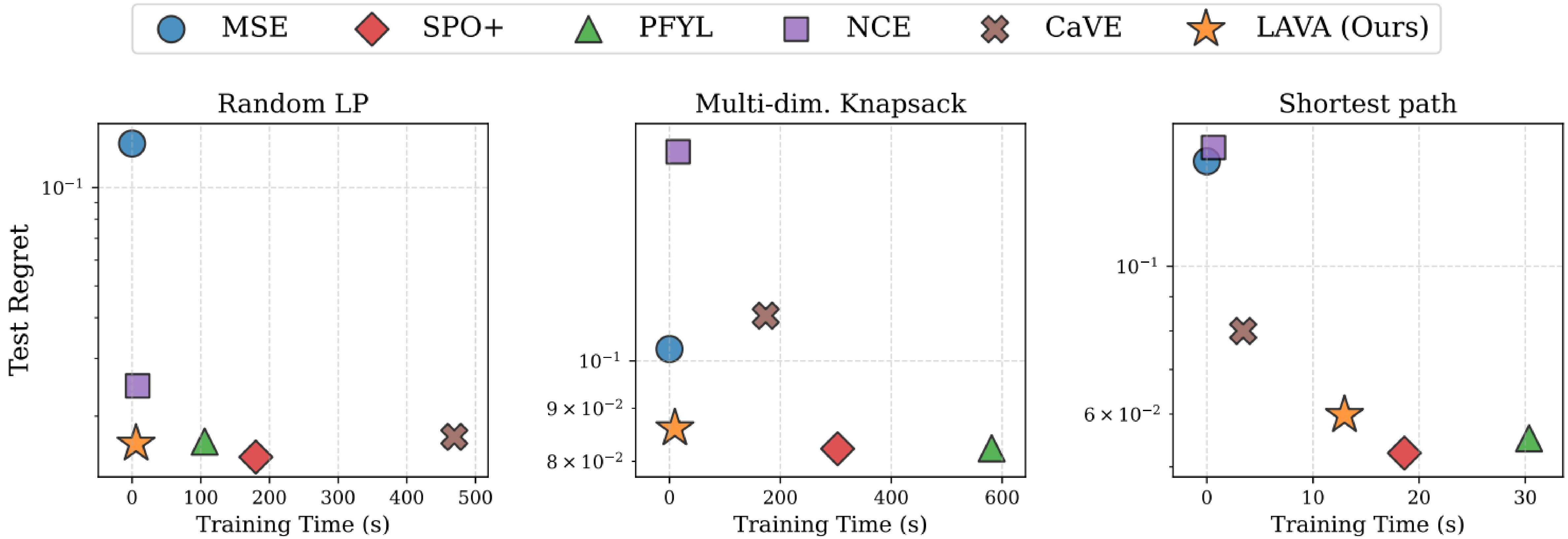
Precomputing adjacent solutions

Algorithm 1 Find adjacent vertices of a basic feasible solution

```
Input: Basic feasible solution z*, basic indices B(1), ..., B(m), non-basic indices
  N(1), ..., N(n − m), constraint matrix A
Output: Set Z_adj of vertices adjacent to z*
B ← [A_{*B(1)} ... A_{*B(m)}]
N ← [A_{*N(1)} ... A_{*N(m)}]
Z_adj ← ∅
Queue ← [(B, N)]
Visited ← {(B, N)}
while Queue is not empty do
    Dequeue a basis B_curr and corresponding N
    Compute directions of movement D = − ...
    for each non-basic variable z*_j as ent...
        d′ = D_{*j} is the direction of ... basic variables when increasing z*_j by 1
        Perform minimum ra... ..._i<0} (− z*_{B_curr(i)} / d_i)
        if θ* > 0 then                                   ▷ Nondegenerate pivot
            Let d ... ∈ ℝ^n
            for ...
                d_{B_curr(k)} = d′_k for basic variable B_curr(k)
            ...jacent vertex Z_adj = Z_adj ∪ {z* + θ*d}
                                                         ▷ Degenerate pivot
            ...tify B_curr(i) as the leaving basic variable according to the TNP rule
            Construct new basis B_new and corresponding N_new by replacing B(i) with N(j)
            if (B_new, N_new) ∉ Visited then
                Add (B_new, N_new) to Queue and to Visited
return Z_adj
```

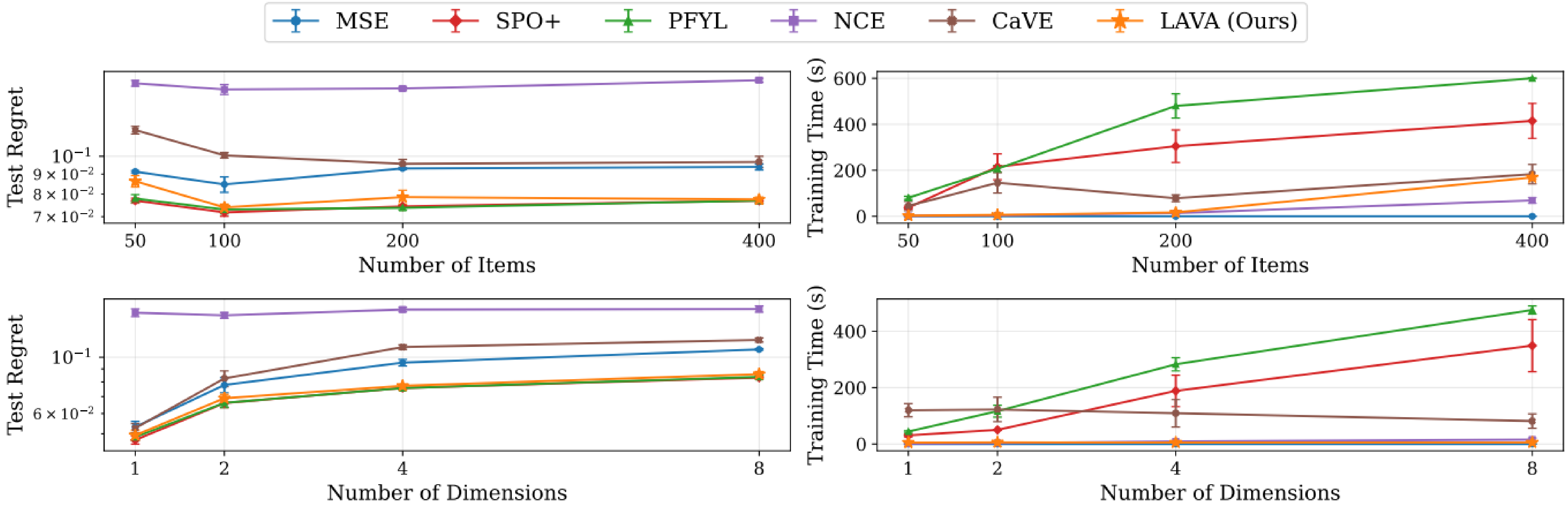
O(n³) in well-behaved cases

Results



MSE
SPO+
PFYL
NCE
CaVE
LAVA (Ours)
Random LP
Multi-dim. Knapsack
Shortest path
Test Regret
10^{-1}
9×10^{-2}
8×10^{-2}
6×10^{-2}
0
100
200
300
400
500
600
10
20
30
Training Time (s)
Training Time (s)
Training Time (s)

Results

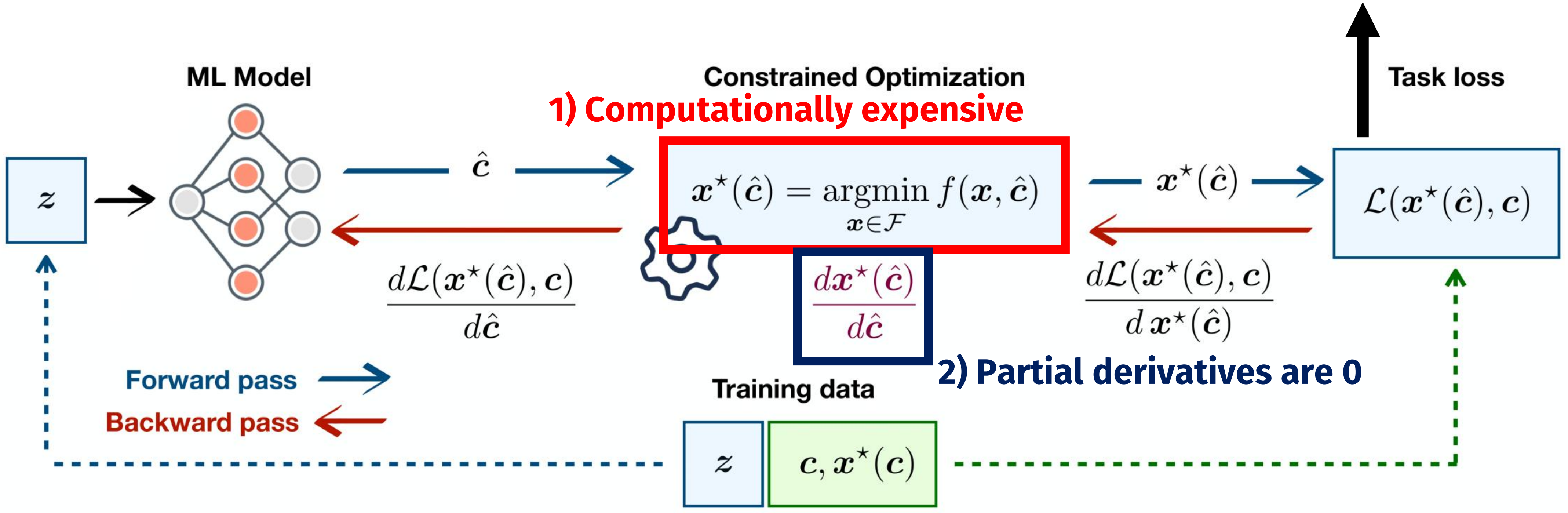


MSE
SPO+
PFYL
NCE
CaVE
LAVA (Ours)
Test Regret
10^{-1}
9×10^{-2}
8×10^{-2}
7×10^{-2}
50
100
200
400
Number of Items
Training Time (s)
600
400
200
0
50
100
200
400
Number of Items
Test Regret
10^{-1}
6×10^{-2}
1
2
4
8
Number of Dimensions
Training Time (s)
400
200
0
1
2
4
8
Number of Dimensions

Conclusion

For example:

$$\textit{Regret}\ (\mathbf{x}^\star(\hat{\mathbf{c}}), \mathbf{c}) = f(\mathbf{x}^\star(\hat{\mathbf{c}}), \mathbf{c}) - f(\mathbf{x}^\star(\mathbf{c}), \mathbf{c})$$



$$\mathcal{L}_{\text{AVA}}(\hat{c}, z^\star(c)) = \sum_{z_{adj} \in Z_{adj}(z^\star(c))} \max(\hat{c}^\top z^\star(c) - \hat{c}^\top z_{adj}, -\epsilon) \quad \text{where } \epsilon \geq 0$$

Conclusion

Applications:

- Healthcare
- Telecommunication
- Energy
- ...

Open challenges:

- Scalability on other problems than linear programs
- Parameters in the constraints
- Benchmarks