Data-Driven Optimal Control:

An Inverse Optimization Model and Algorithm

Long, Youyuan

Supervisor: Peyman Mohajerin Esfahani

Daily Supervisors: Tolga Ok

Pedro Zattoni Scroccaro

TU Delft

Contents

TUDelft

Introduction of Inverse Optimization (IO)

TU Delft

Inverse Optimization

Given a *state (input)*, the expert determines its optimal *action (output)* by optimizing an unknown cost

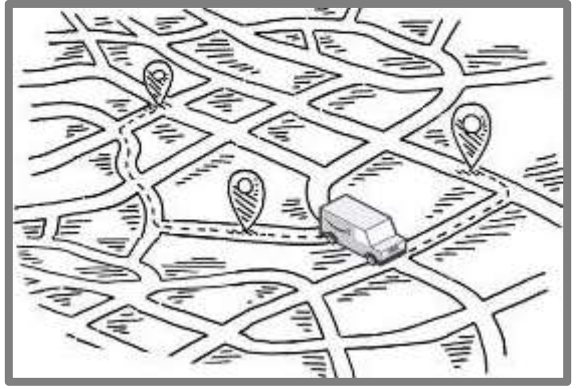
Expert

$$\hat{s} \longrightarrow \boxed{\arg\min_{u \in \mathbb{U}(\hat{s})} F(\hat{s}, u)} \longrightarrow \hat{u}$$

Market behavior

Saez-Gallego et al. ’16

Driver preference



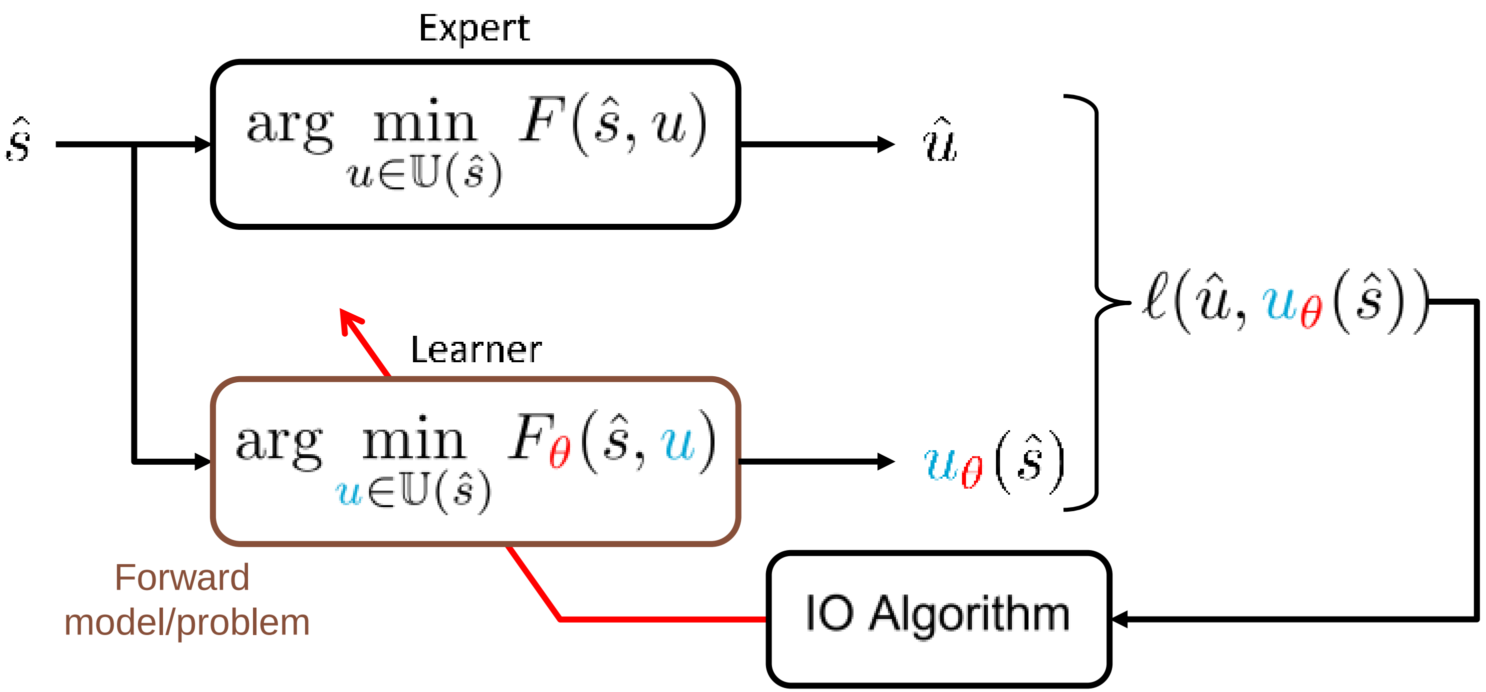
Scroccaro et al. ’23

Control policy

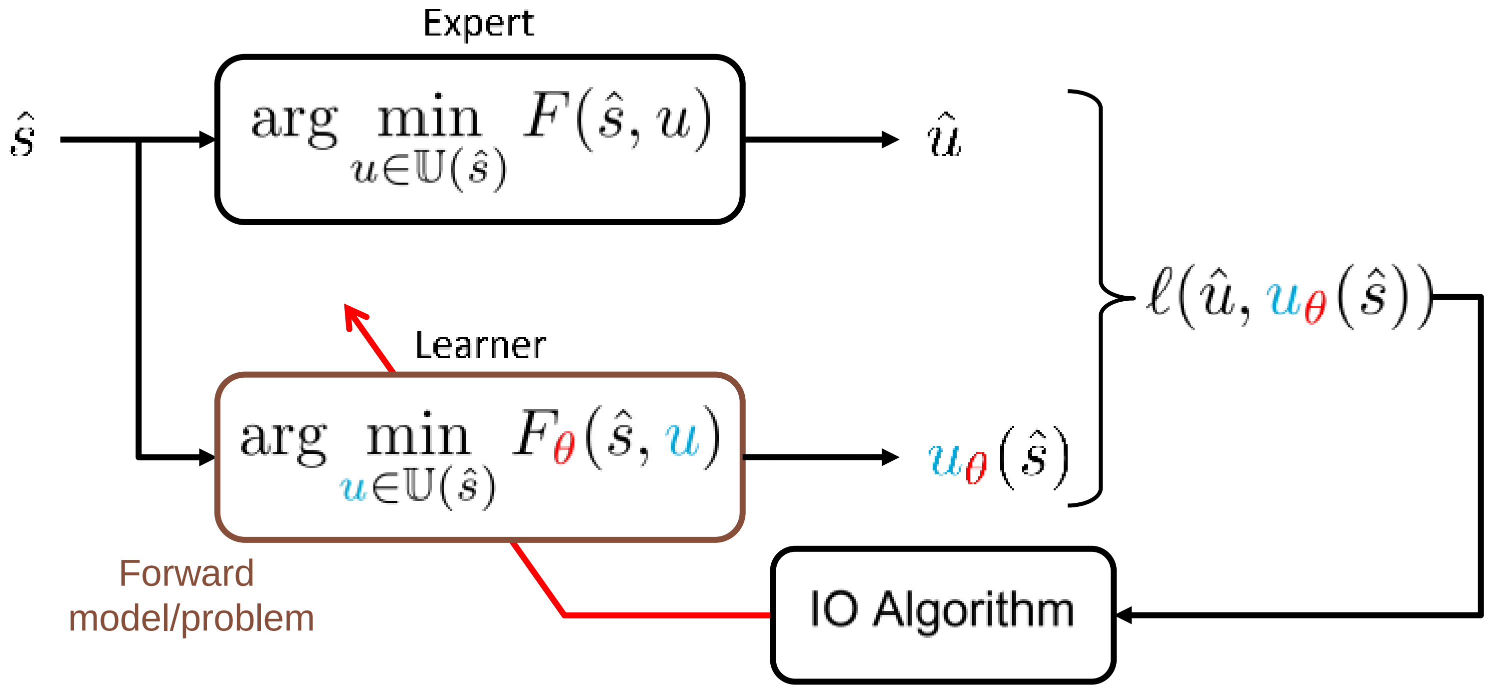
Akhtar et al. ’21

TU Delft

Supervised Learning Point of View

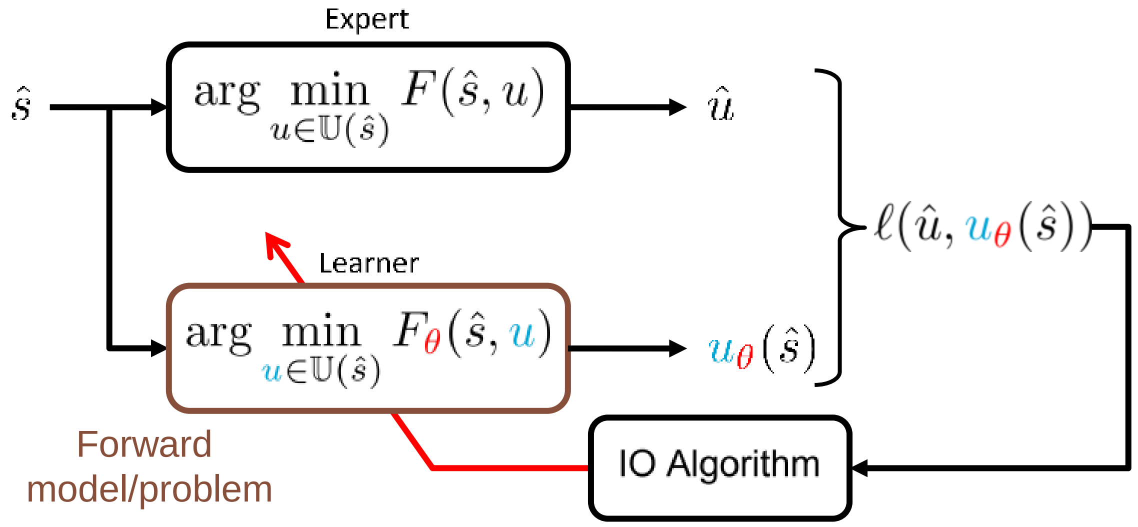


Supervised Learning Point of View



Inverse Optimization Summary

- Hypothesis space: $\{F_\theta \mid \theta \in \Theta\}$



- Loss function: $\ell(\hat{u}, u_\theta(\hat{s}))$

- Training: $\min_{\theta \in \Theta} \kappa \mathcal{R}(\theta) + \frac{1}{N} \sum_{i=1}^{N} \ell(\hat{u}_i, u_\theta(\hat{s}_i))$ $\qquad \hat{\mathcal{D}} = \{(\hat{s}_i, \hat{u}_i)\}_{i=1}^{N}$

TUDelft

Quadratic Hypothesis Space

Augmented state/feature

Quadratic cost: $F_\theta(\hat{s}, u) = u^T \theta_{uu} u + 2\phi(\hat{s})^T \theta_{su} u, \;\; \theta := (\theta_{uu}, \theta_{su})$

Linear constraints: $\mathbb{U}(\hat{s}) = \{u \in \mathbb{R}^n \mid M(\hat{s})u \leq W(\hat{s})\}$

$$\min_{u \in \mathbb{U}(\hat{s})} F_\theta(\hat{s}, u)$$

To ensure the **convexity** of the Forward Problem w.r.t u, extra constraints should be imposed on θ:

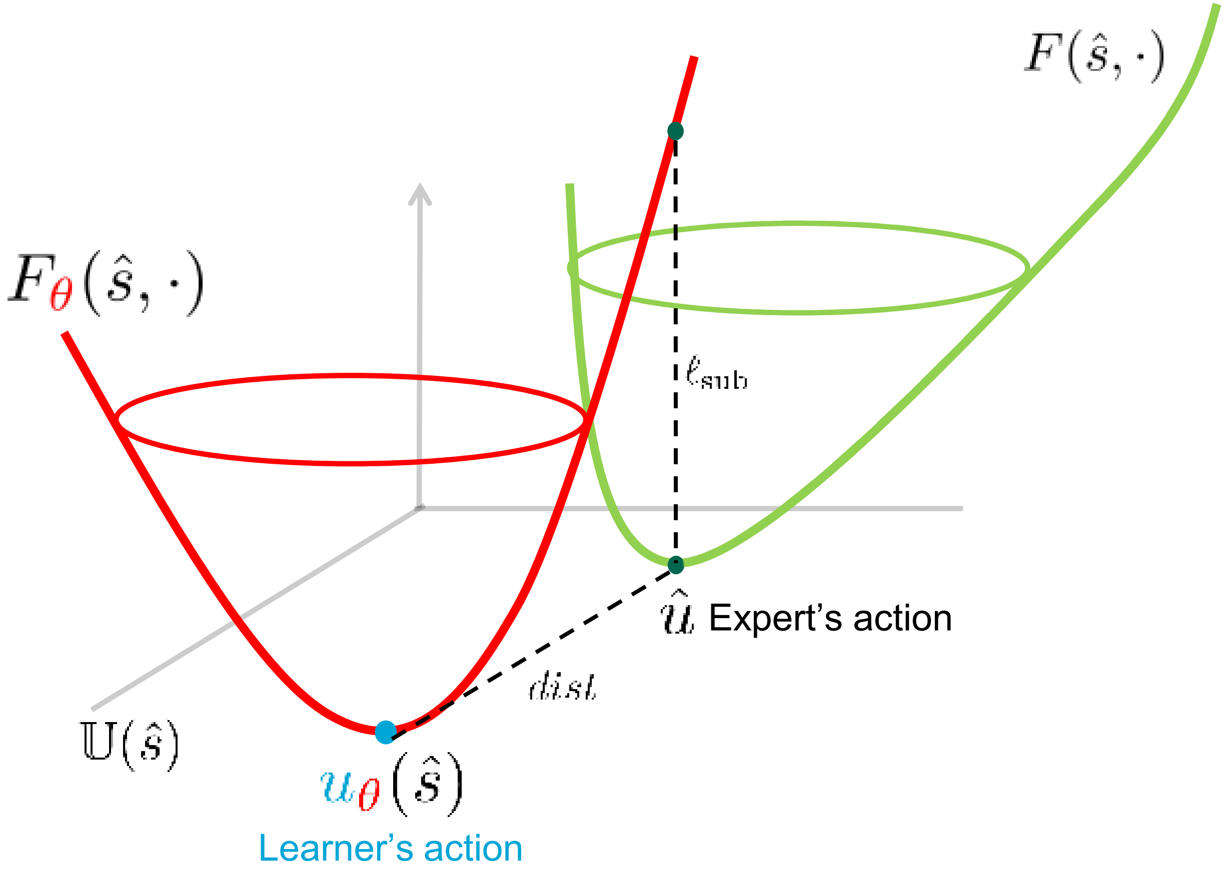
$$\theta \in \{(\theta_{uu}, \theta_{su}) \mid \theta_{uu} \succeq I\}$$

TU Delft

Suboptimality Loss Function [1]

$$\ell_{\text{sub}}(\hat{s},\hat{u}) := F_\theta(\hat{s},\hat{u}) - F_\theta(\hat{s}, u_\theta(\hat{s}))$$

$$u_\theta(\hat{s}) = \arg\min_{u\in\mathbb{U}(\hat{s})} F_\theta(\hat{s}, u)$$

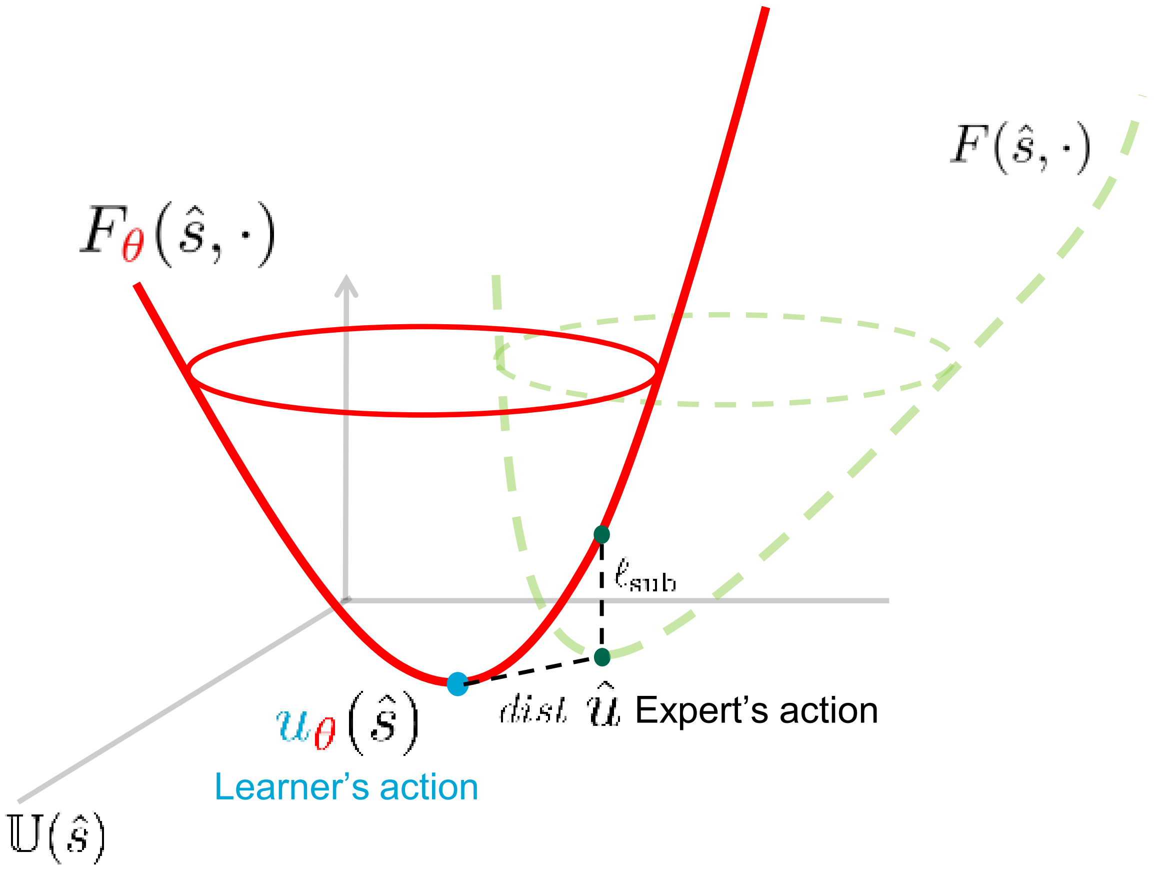


[1] Mohajerin Esfahani P, Shafieezadeh-Abadeh S, Hanasusanto G A, et al. Data-driven inverse optimization with imperfect information[J]. Mathematical Programming, 2018, 167: 191-234.

Suboptimality Loss Function[1]

$$\ell_{\text{sub}}(\hat{s}, \hat{u}) := F_\theta(\hat{s}, \hat{u}) - F_\theta(\hat{s}, u_\theta(\hat{s}))$$

$$u_\theta(\hat{s}) = \arg \min_{u \in \mathbb{U}(\hat{s})} F_\theta(\hat{s}, u)$$



[1] Mohajerin Esfahani P, Shafieezadeh-Abadeh S, Hanasusanto G A, et al. Data-driven inverse optimization with imperfect information[J]. Mathematical Programming, 2018, 167: 191-234.

Regularized training

$$\min_{\theta} k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left\{F_\theta(\hat{s}_i, \hat{u}_i) - \min_{u\in\mathbb{U}(\hat{s}_i)} F_\theta(\hat{s}_i, u)\right\}$$
$$\text{s.t. } \theta_{uu} \succeq I$$

Convex w.r.t θ !

$$\hat{\mathcal{D}} = \{(\hat{s}_i, \hat{u}_i)\}_{i=1}^N$$

Quadratic cost: $F_\theta(\hat{s}, u) = u^T\theta_{uu}u + 2\phi(\hat{s})^T\theta_{su}u, \ \theta_{uu} \succeq I$

Suboptimality loss: $\ell_{\text{sub}}(\hat{s}, \hat{u}) := F_\theta(\hat{s}, \hat{u}) - F_\theta(\hat{s}, u_\theta(\hat{s}))$

$$u_\theta(\hat{s}) = \arg\min_{u\in\mathbb{U}(\hat{s})} F_\theta(\hat{s}, u)$$

TU Delft

Theorem 1 (LMI reformulation[1])

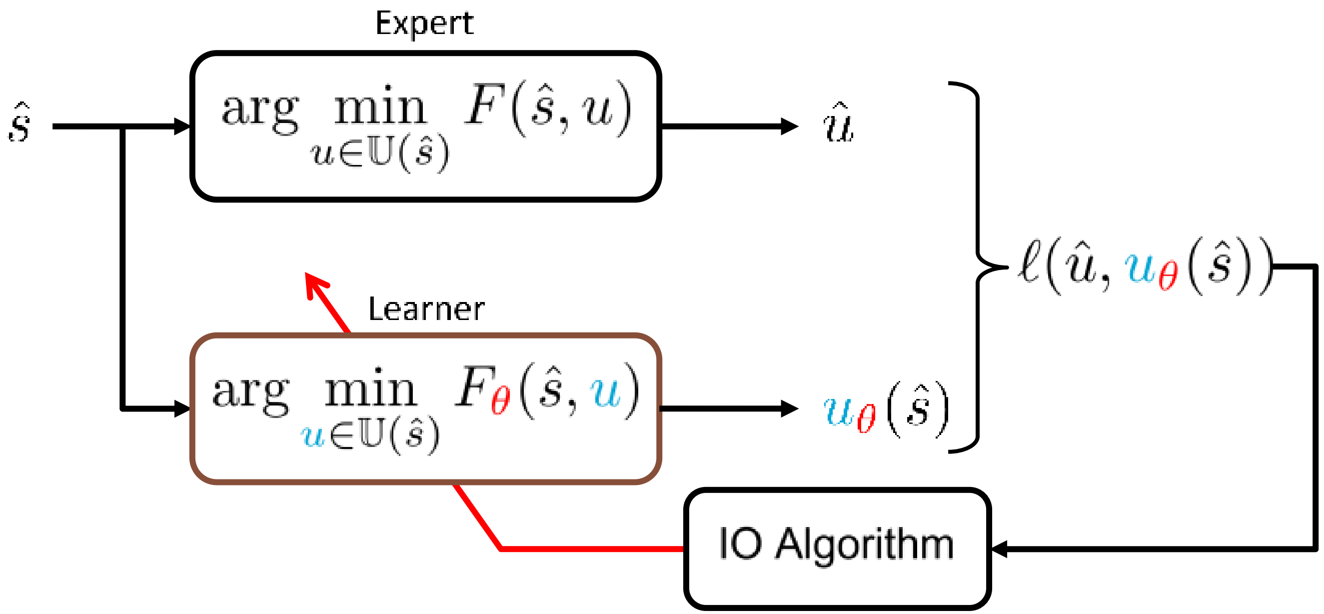
$$\min_{\theta} k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left\{F_\theta(\hat{s}_i,\hat{u}_i) - \min_{u\in\mathbb{U}(\hat{s}_i)} F_\theta(\hat{s}_i,u)\right\}$$
$$\text{s.t.}\quad \theta_{uu} \succeq I$$

$$\theta := (\theta_{uu}, \theta_{su})$$

Equivalent !

$$\begin{aligned}
\min_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i}\quad & k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 \\
& +\frac{1}{N}\sum_{i=1}^{N}\left(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\phi(\hat{s}_i)^T\theta_{su}\hat{u}_i + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right) \\
\text{s.t.}\quad & \theta_{uu}\succeq I,\ \lambda_i\in\mathbb{R}_+^d,\ \gamma_i\in\mathbb{R},\quad \forall i\leq N \\
& \begin{bmatrix} \theta_{uu} & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^T\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix}\succeq 0,\quad \forall i\leq N
\end{aligned}$$

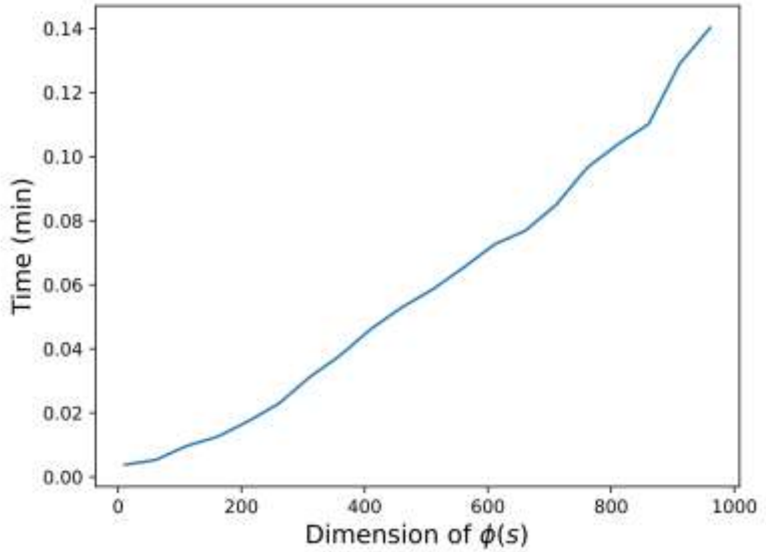
[1] Akhtar S A, Kolarijani A S, Esfahani P M. Learning for control: An inverse optimization approach[J]. IEEE Control Systems Letters, 2021, 6: 187-192.



Expert
$\hat{s}$
$\arg\min_{u \in \mathbb{U}(\hat{s})} F(\hat{s}, u)$
$\hat{u}$
$\ell(\hat{u}, u_\theta(\hat{s}))$
Learner
$\arg\min_{u \in \mathbb{U}(\hat{s})} F_\theta(\hat{s}, u)$
$u_\theta(\hat{s})$
IO Algorithm

Contributions

- Heavy computational burden by high-dimensional augmented feature



- Challenges of feature engineering

KIOM

- Scalability issues caused by large datasets

Data size	Memory (RAM)
5K	16GB
10K	64GB
15K	128GB
20K	256GB

Table: The relationship between the dataset size and the required memory on the Halfcheetah-expert task.

SSO

TU Delft

Kernel Inverse Optimization Machine

(KIOM)

---- Solution for heavy computational burden by high-dimensional augmented feature and challenges of feature engineering

TUDelft

Theorem 2 (Kernel reformulation)

$$\begin{aligned}
\min_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i} \quad & k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\phi(\hat{s}_i)^T\theta_{su}\hat{u}_i + \frac{1}{4}\gamma_i + W\left(\hat{s}_i\right)^\top\lambda_i\right) \\
\text{s.t.} \quad & \theta_{uu} \succeq I,\ \lambda_i \in \mathbb{R}_+^d,\ \gamma_i \in \mathbb{R}, \quad \forall i \leq N \\
& \begin{bmatrix} \theta_{uu} & M\left(\hat{s}_i\right)^\top\lambda_i + 2\theta_{su}^T\phi(\hat{s}_i) \\ * & \gamma_i \end{bmatrix} \succeq 0, \quad \forall i \leq N
\end{aligned}$$

[LMI reformulation]

TU Delft

Theorem 2 (Kernel reformulation)

$$\begin{aligned}\min_{\theta_{uu},\theta_{su},\lambda_i,\gamma_i} \quad & k\|\theta_{uu}\|_F^2 + k\|\theta_{su}\|_F^2 + \frac{1}{N}\sum_{i=1}^{N}\left(\hat{u}_i^T\theta_{uu}\hat{u}_i + 2\phi(\hat{s}_i)^T\theta_{su}\hat{u}_i + \frac{1}{4}\gamma_i + W(\hat{s}_i)^\top\lambda_i\right)\\ \text{s.t.} \quad & \theta_{uu} \succeq I_n,\ \lambda_i \in \mathbb{R}_+^d,\ \gamma_i \in \mathbb{R}, \quad \forall i \le N\\ & \begin{bmatrix}\theta_{uu} & M(\hat{s}_i)^\top\lambda_i + 2\theta_{su}^T\phi(\hat{s}_i)\\ * & \gamma_i\end{bmatrix} \succeq 0, \quad \forall i \le N\end{aligned}$$

Primal

Dual

$$\begin{aligned}\min_{P,\Lambda_i,\Gamma_i} \quad & \frac{1}{4k}\left\|\left(\sum_{i=1}^{N}\frac{\hat{u}_i\hat{u}_i^T}{N} - \Lambda_i\right) - P\right\|_F^2 - \mathrm{Tr}(P)\\ & + \frac{1}{k}\sum_{i=1}^{N}\sum_{j=1}^{N}\kappa(\hat{s}_i,\hat{s}_j)\left(\frac{\hat{u}_i}{N} - 2\Gamma_i\right)^T\left(\frac{\hat{u}_j}{N} - 2\Gamma_j\right)\\ \text{s.t.} \quad & P \succeq 0,\ \frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i \ge 0, \quad \forall i \le N\\ & \begin{bmatrix}\Lambda_i & \Gamma_i\\ * & \frac{1}{4N}\end{bmatrix} \succeq 0, \quad \forall i \le N\end{aligned}$$

$$\kappa(\hat{s}_i,\hat{s}_j) = \phi(\hat{s}_i)^T\phi(\hat{s}_j)$$

Recover θ

KIOM

$$\theta_{uu} = -\frac{\left(\sum_{i=1}^{N}\frac{\hat{u}_i\hat{u}_i^T}{N} - \Lambda_i\right) - P}{2k}$$

$$\theta_{su} = -\frac{\sum_{i=1}^{N}\phi(\hat{s}_i)\left(\frac{\hat{u}_i^T}{N} - 2\Gamma_i{}^T\right)}{k}$$

Theorem 2 (Kernel reformulation)

KIOM

$$\theta_{uu} = -\frac{\left(\sum_{i=1}^{N} \frac{\hat{u}_i \hat{u}_i^T}{N} - \Lambda_i\right) - P}{2k}$$

$$\theta_{su} = -\frac{\sum_{i=1}^{N} \phi(\hat{s}_i) \left(\frac{\hat{u}_i^T}{N} - 2{\Gamma_i}^T\right)}{k}$$

$$\begin{aligned} \min_{P,\Lambda_i,\Gamma_i} \quad & \frac{1}{4k}\left\| \left(\sum_{i=1}^{N} \frac{\hat{u}_i \hat{u}_i^T}{N} - \Lambda_i\right) - P \right\|_F^2 - \mathrm{Tr}(P) \\ & + \frac{1}{k}\sum_{i=1}^{N}\sum_{j=1}^{N} \kappa(\hat{s}_i, \hat{s}_j)\left(\frac{\hat{u}_i}{N} - 2\Gamma_i\right)^T \left(\frac{\hat{u}_j}{N} - 2\Gamma_j\right) \\ \text{s.t.} \quad & P \succeq 0, \quad \frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i \geq 0, \quad \forall i \leq N \\ & \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \succeq 0, \quad \forall i \leq N \end{aligned}$$

$$\kappa(\hat{s}_i, \hat{s}_j) = \phi(\hat{s}_i)^T \phi(\hat{s}_j)$$

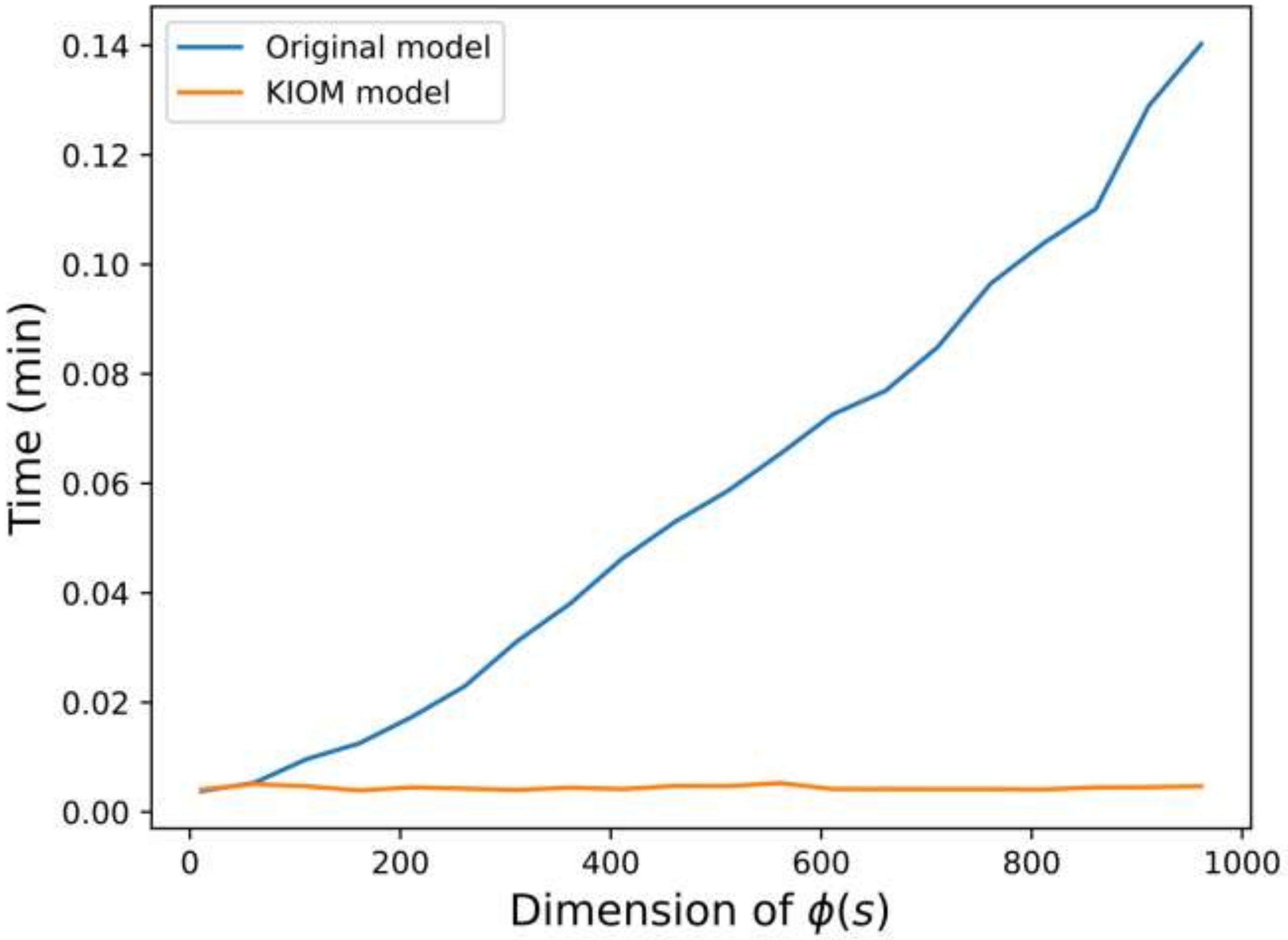
Recover θ

Forward problem:

$$\min_{u \in \mathbb{U}(s)} u^T \theta_{uu} u + \underbrace{2\phi(s)^T \theta_{su}}_{-\frac{1}{k}\sum_{i=1}^{N} \kappa(s, \hat{s}_i)\left(\frac{\hat{u}_i^T}{N} - 2{\Gamma_i}^T\right)} u$$

Does the KIOM model address the first two deficiencies mentioned before?

- Heavy computational burden by high-dimensional augmented feature
- Challenges of feature engineering



TU Delft

Variant: A Simpler Version with $\theta_{uu} = I$

KIOM

$$\theta_{uu} = -\frac{\left(\sum_{i=1}^{N} \frac{\hat{u}_i \hat{u}_i^T}{N} - \Lambda_i\right) - P}{2k}$$

$$\theta_{su} = -\frac{\sum_{i=1}^{N} \phi(\hat{s}_i)\left(\frac{\hat{u}_i^T}{N} - 2{\Gamma_i}^T\right)}{k}$$

$$\begin{aligned}
\min_{P,\Lambda_i,\Gamma_i} \quad & \frac{1}{4k}\left\|\left(\sum_{i=1}^{N} \frac{\hat{u}_i \hat{u}_i^T}{N} - \Lambda_i\right) - P\right\|_F^2 - \operatorname{Tr}(P) \\
& + \frac{1}{k}\sum_{i=1}^{N}\sum_{j=1}^{N} \kappa(\hat{s}_i, \hat{s}_j)\left(\frac{\hat{u}_i}{N} - 2\Gamma_i\right)^T\left(\frac{\hat{u}_j}{N} - 2\Gamma_j\right) \\
\text{s.t.} \quad & P \succeq 0, \quad \frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i \geq 0, \quad \forall i \leq N \\
& \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \succeq 0, \quad \forall i \leq N
\end{aligned}$$

$\theta_{uu} = I$

Simple KIOM

$$\begin{aligned}
\min_{\Lambda_i,\Gamma_i} \quad & \frac{1}{k}\sum_{i=1}^{N}\sum_{j=1}^{N} \kappa(\hat{s}_i, \hat{s}_j)\left(\frac{\hat{u}_i}{N} - 2\Gamma_i\right)^T\left(\frac{\hat{u}_j}{N} - 2\Gamma_j\right) + \sum_{i=1}^{N} \operatorname{Tr}(\Lambda_i) \\
\text{s.t.} \quad & \frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i \geq 0, \quad \forall i \leq N \\
& \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \succeq 0, \quad \forall i \leq N
\end{aligned}$$

TU Delft

Scalability issues caused by large datasets

Data size	Memory (RAM)
5K	16GB
10K	64GB
15K	128GB
20K	256GB

Table: The relationship between the dataset size and the required memory on the Halfcheetah-expert task.

TU Delft

Sequential Selection Optimization (SSO)

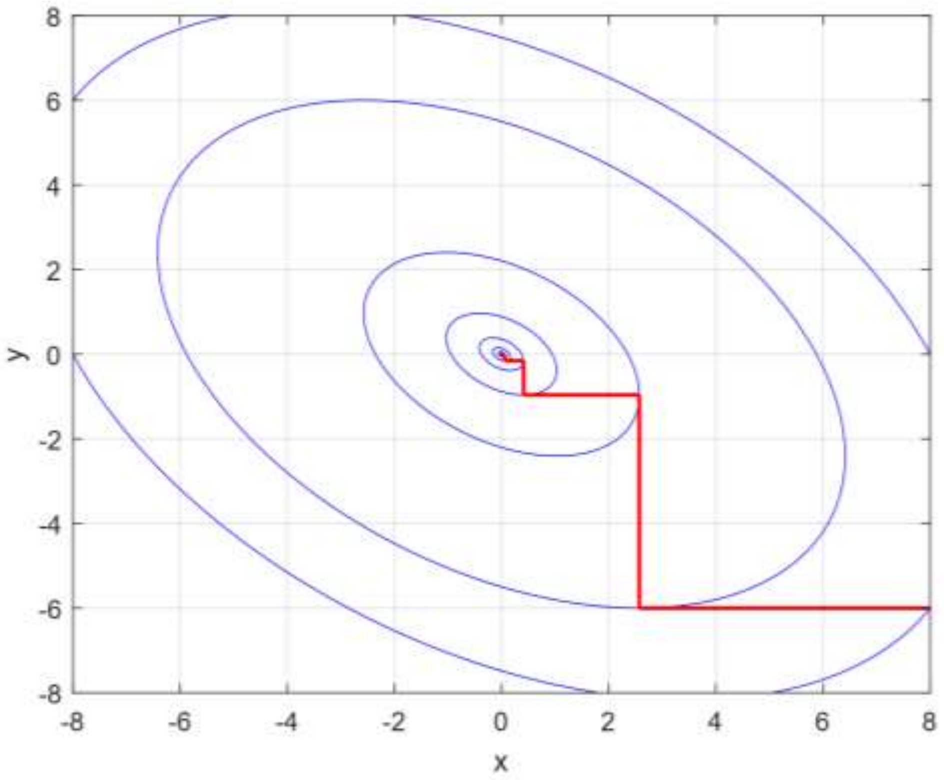
- ---- Solution for Scalability issues caused by large datasets

TUDelft

Coordinate Descent Method

- Coupled cost, decoupled constraints

$$\min_{x^1,\dots x^N} f(x^1,\dots,x^N)$$
$$\text{s.t. } x^i \in \mathcal{X}^i, \forall i \leq N$$



- Coordinate descent method

$$x^i_{k+1} = \arg\min_{\xi \in X_i} f\left(x^1_{k+1},\dots,x^{i-1}_{k+1},\xi,x^{i+1}_k,\dots,x^N_k\right)$$

TU Delft

Why Does the Coordinate Descent (CD) Method Suit?

- Simple KIOM problem

$$
\begin{aligned}
\min_{\Lambda_i,\Gamma_i} \quad & \frac{1}{k}\sum_{i=1}^{N}\sum_{j=1}^{N}\kappa(\hat{s}_i,\hat{s}_j)\left(\frac{\hat{u}_i}{N}-2\Gamma_i\right)^T\left(\frac{\hat{u}_j}{N}-2\Gamma_j\right)+\sum_{i=1}^{N}\mathrm{Tr}(\Lambda_i) \\
\text{s.t.} \quad & \frac{W(\hat{s}_i)}{N}-2M(\hat{s}_i)\Gamma_i \geq 0, \quad \forall i \leq N \\
& \begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \succeq 0, \quad \forall i \leq N
\end{aligned}
$$

$$
\rightarrow \quad
\begin{aligned}
\min_{x^1,\ldots x^N} \; & f(x^1,\ldots,x^N) \\
\text{s.t.} \; & x^i \in \mathcal{X}^i, \forall i \leq N
\end{aligned}
$$

- Definition of coordinate

$$
\text{coordinate i} := \{\Lambda_i, \Gamma_i\}
$$

Each coordinate is decoupled in constraints !

TU Delft

Two heuristics for SSO

- a. Heuristics for choosing which coordinates to optimize

Given the current values of $\{\Lambda_i, \Gamma_i\}_{i=1}^{N}$, we choose ***p*** coordinates that satisfy the condition

$$\frac{W(\hat{s}_i)}{N} - 2M(\hat{s}_i)\Gamma_i > 0$$

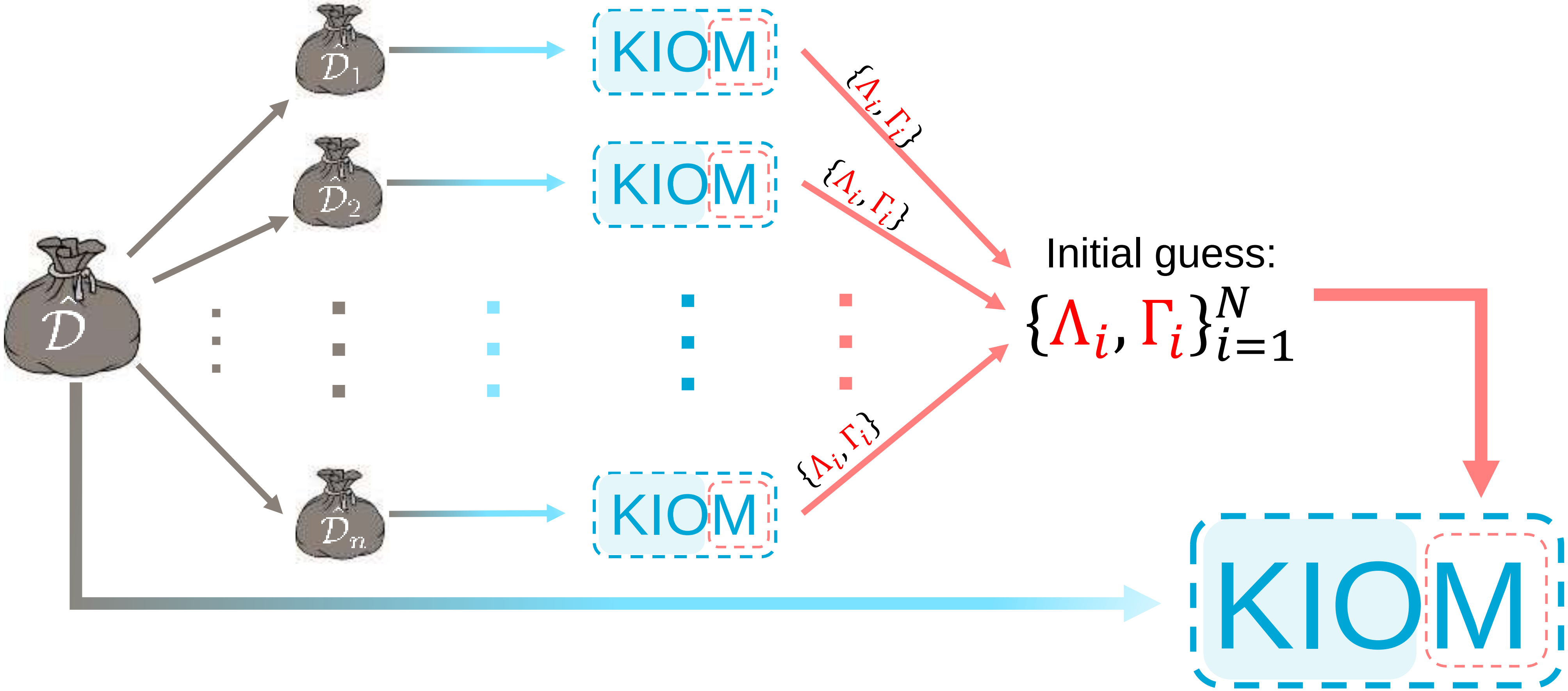
and have the largest KKT violators which is defined as

$$\text{KKT\_violator}(i) = \left| \mathrm{Tr}\left(\begin{bmatrix} \Lambda_i & \Gamma_i \\ * & \frac{1}{4N} \end{bmatrix} \begin{bmatrix} I_n & 2\theta_{su}^{\top}\phi(\hat{s}_i) \\ * & \|2\theta_{su}^{\top}\phi(\hat{s}_i)\|_2^2 \end{bmatrix} \right) \right|.$$

TU Delft

Two heuristics for SSO

- b. Warm-up trick for improved initial guess



TU Delft

SSO Algorithm

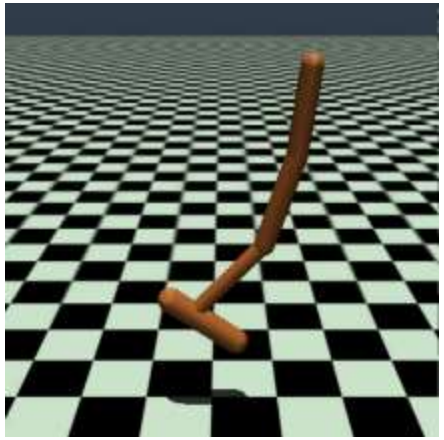
Algorithm 1 SSO

1: Initialize variable: $\{\Lambda_i, \Gamma_i\}_{i=1,\ldots,N} \leftarrow \text{WarmUp}(\{\hat{s}_i, \hat{u}_i\}_{i=1,\ldots N})$
2: **for** iteration = 1 to T **do**
3: $\quad \{\Lambda_{a_i}, \Gamma_{a_i}\}_{i=1,\ldots,p} \leftarrow \text{HeuristicSelection}(\{\hat{s}_i, \hat{u}_i\}_{i=1,\ldots N}, \{\Lambda_i, \Gamma_i\}_{i=1,\ldots,N})$
4: $\quad \text{Update}(\{\Lambda_{a_i}, \Gamma_{a_i}\}_{i=1,\ldots,p})$ (Update selected coordinates)
5: **end for**

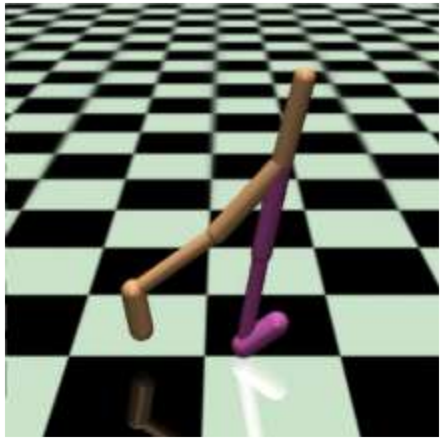
Numerical Experiments

Numerical Experiments

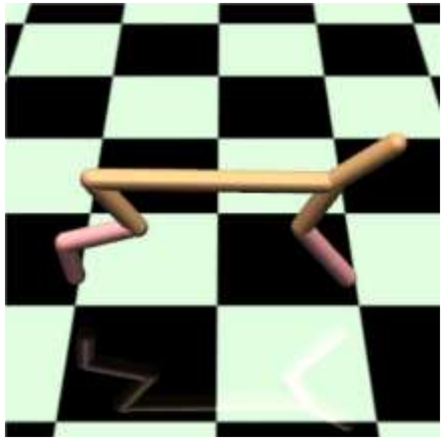
- Experimental environment



Hopper



Walker2d



HalfCheetah

Task	Action Dim	State Dim
Hopper	3	11
Walker2d	6	17
HalfCheetah	6	17

- Dataset

D4RL

- Expert dataset
- Medium dataset

- Solver

SCS
SPLITTING CONIC SOLVER

- Open-source
- Parallel processing

TU Delft

Performance Evaluation of KIOM

Task	KIOM	IO	BC(TD3+BC)[1]	BC(CQL)[2]	Teacher agent
Hopper-expert	**109.9**(5k)	31.8	**111.5**	**109.0**	**108.5**
Hopper-medium	**50.2**(5k)	20.6	30.0	29.0	44.3
Walker2d-expert	108.5(10k)	0.9	56.0	**125.7**	107.1
Walker2d-medium	**74.6**(5k)	0.0	11.4	6.6	62.1
Halfcheetah-expert	84.4(10k)	-1.7	105.2	**107.0**	88.1
Halfcheetah-medium	**39.0**(5k)	-3.1	36.6	36.1	**40.7**

TU Delft

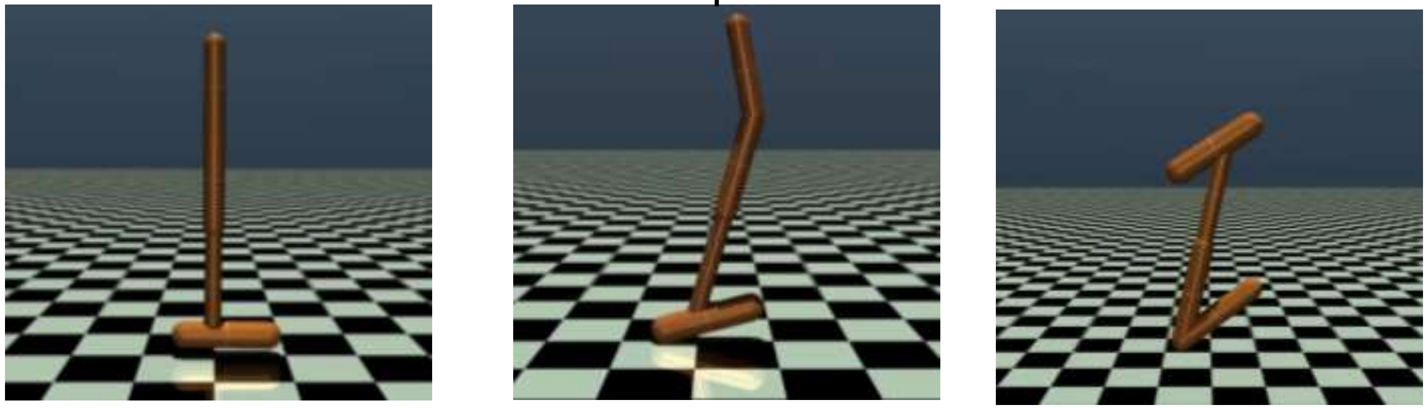
[1] Fujimoto S, Gu S S. A minimalist approach to offline reinforcement learning[J]. Advances in neural information processing systems, 2021, 34: 20132-20145.
[2] Kumar A, Zhou A, Tucker G, et al. Conservative q-learning for offline reinforcement learning[J]. Advances in Neural Information Processing Systems, 2020, 33: 1179-1191

Performance Evaluation of KIOM

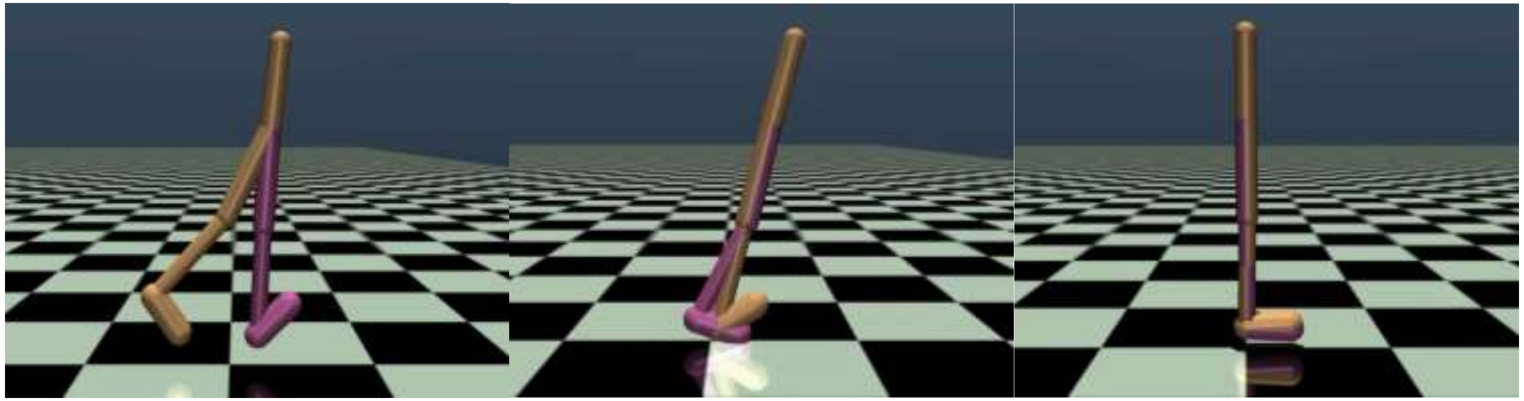
Task	KIOM	IO	BC(TD3+BC)[1]	BC(CQL)[2]	Teacher agent
Hopper-expert	**109.9**(5k)	31.8	**111.5**	**109.0**	**108.5**
Hopper-medium	**50.2**(5k)	20.6	30.0	29.0	44.3
Walker2d-expert	108.5(10k)	0.9	56.0	**125.7**	107.1
Walker2d-medium	**74.6**(5k)	0.0	11.4	6.6	62.1
Halfcheetah-expert	84.4(10k)	-1.7	105.2	**107.0**	88.1
Halfcheetah-medium	**39.0**(5k)	-3.1	36.6	36.1	**40.7**

KIOM Expert IO

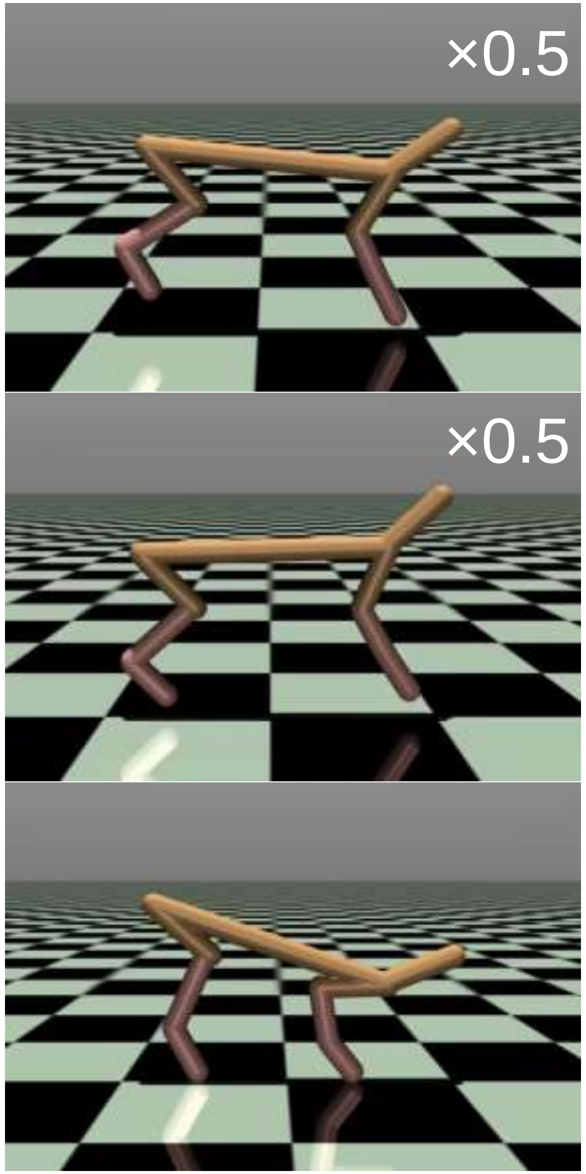
Hopper



Walker



Halfcheetah



KIOM

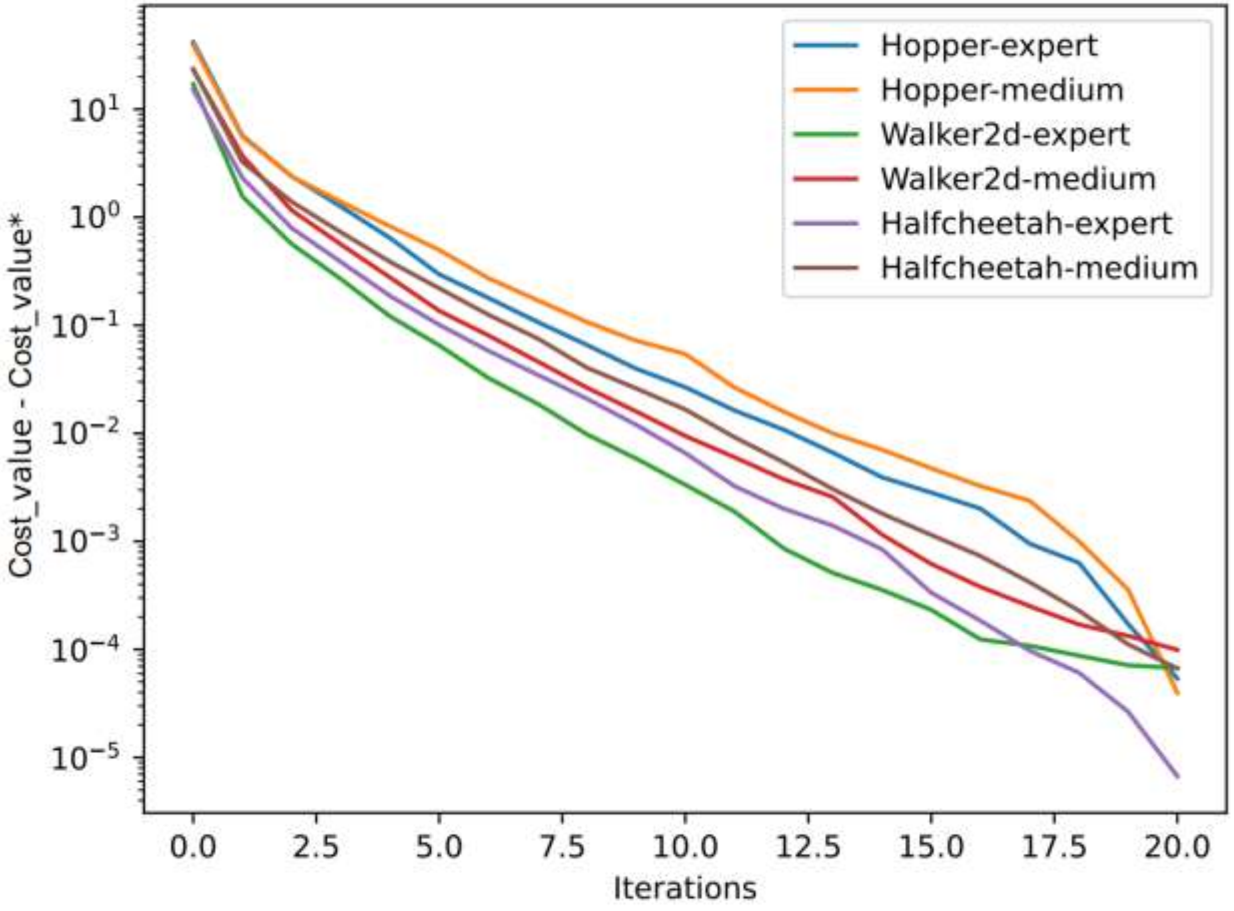
Expert

IO

Performance Evaluation of SSO

Task	SCS		SSO	
	Cost Value	Score	Cost Value	Score
Hopper-expert	185.219	109.9	185.220	110.2
Hopper-medium	218.761	50.2	218.761	51.8
Walker2d-expert	140.121	108.5	140.121	109.2
Walker2d-medium	151.117	74.6	151.117	74.9
Halfcheetah-expert	165.041	84.4	165.041	83.8
Halfcheetah-medium	188.184	39.0	188.184	39.7

Performance Evaluation of SSO

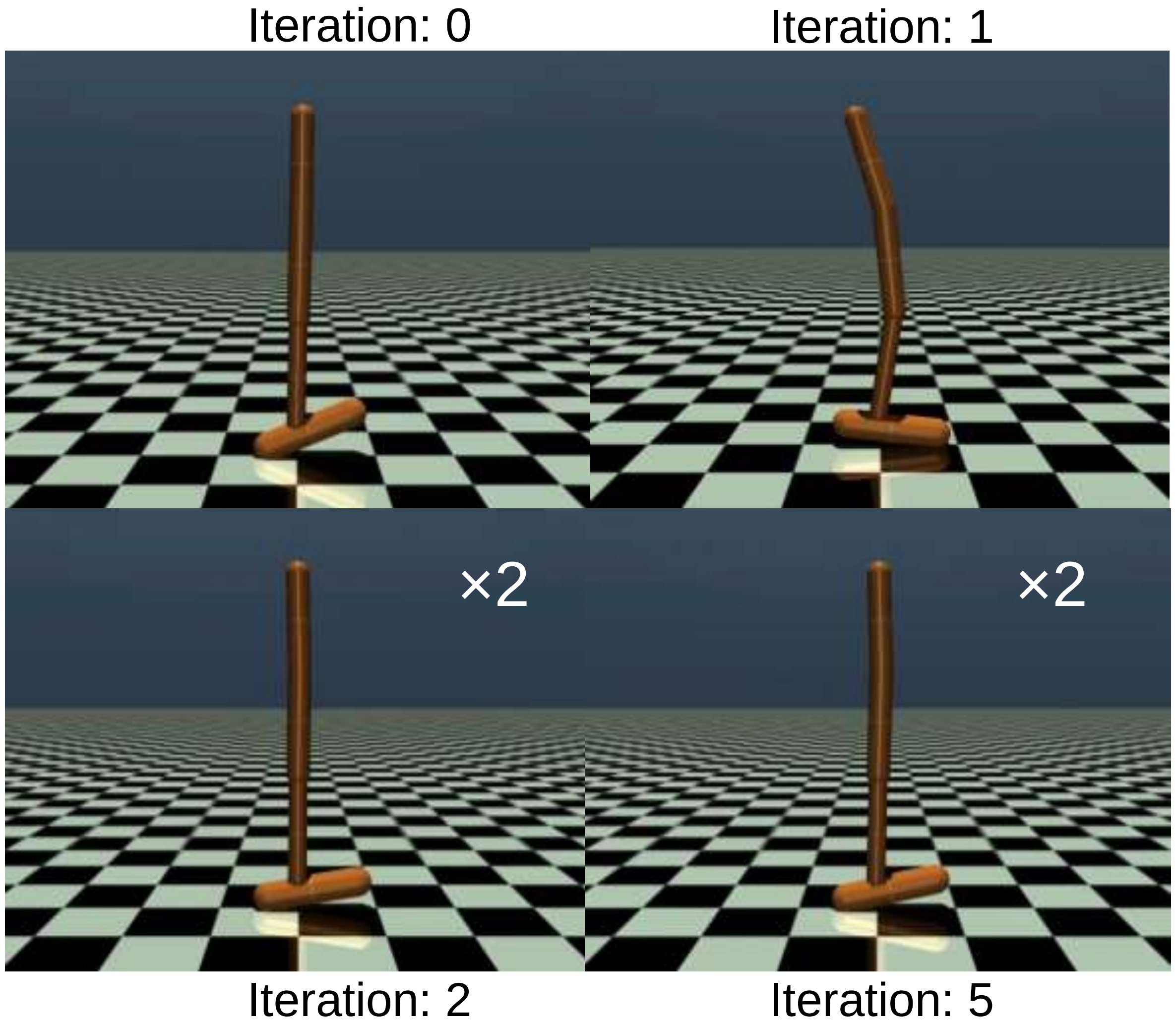


Task	SCS		SSO	
	Cost Value	**Score**	**Cost Value**	**Score**
Hopper-expert	185.219	109.9	185.220	110.2
Hopper-medium	218.761	50.2	218.761	51.8
Walker2d-expert	140.121	108.5	140.121	109.2
Walker2d-medium	151.117	74.6	151.117	74.9
Halfcheetah-expert	165.041	84.4	165.041	83.8
Halfcheetah-medium	188.184	39.0	188.184	39.7

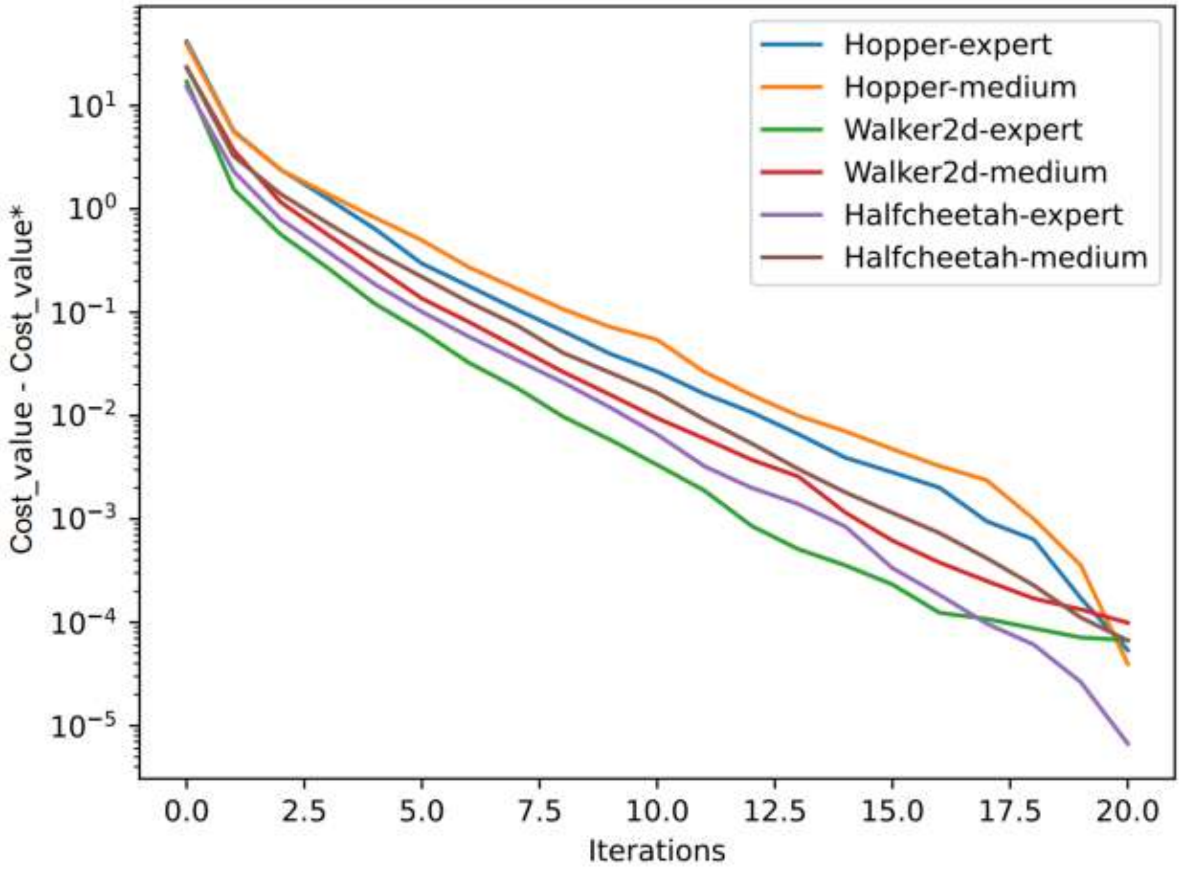
TU Delft

Performance Evaluation of SSO

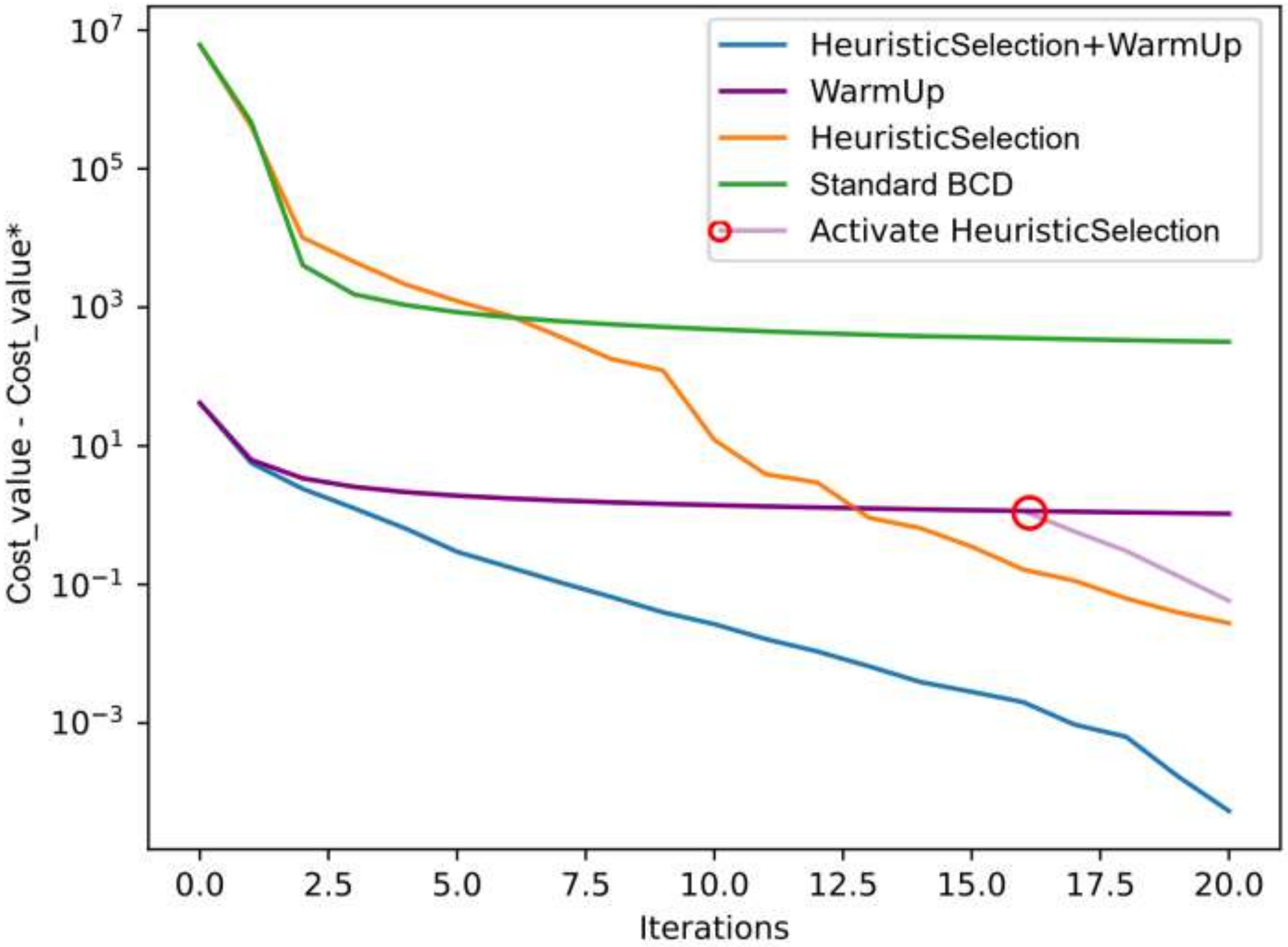
- a. Performance evaluation



Task	SCS		SSO	
	Cost Value	Score	Cost Value	Score
Hopper-expert	185.219	109.9	185.220	110.2
Hopper-medium	218.761	50.2	218.761	51.8
Walker2d-expert	140.121	108.5	140.121	109.2
Walker2d-medium	151.117	74.6	151.117	74.9
Halfcheetah-expert	165.041	84.4	165.041	83.8
Halfcheetah-medium	188.184	39.0	188.184	39.7



Ablation studies of SSO



Conclusion and Future Study

Conclusion and Future Study

- Conclusion

Introduction of Inverse Optimization

Primal

IO Model

Quadratic hypothesis space
Suboptimality loss

Dual

KIOM

Kernel Method

$\theta_{uu} = I$

Simple KIOM

SSO (CD based)

HeuristicSelection
WarmUp

- Future study

Establish the convergence analysis for the SSO

Thank you for your attention

Long, Youyuan

TU Delft