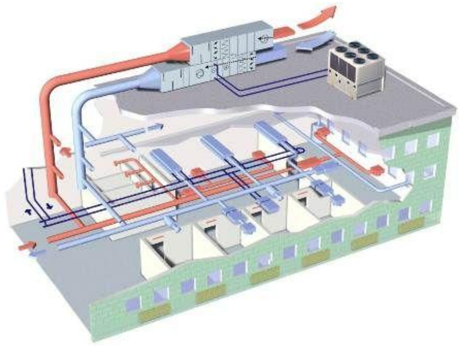
CLUE: Safe Model-Based RL HVAC ControL Using Epistemic Uncertainty Estimation

Zhiyu (Ryan) An, Xianzhong Ding, Arya Rathee, Wan Du

University of California, Merced

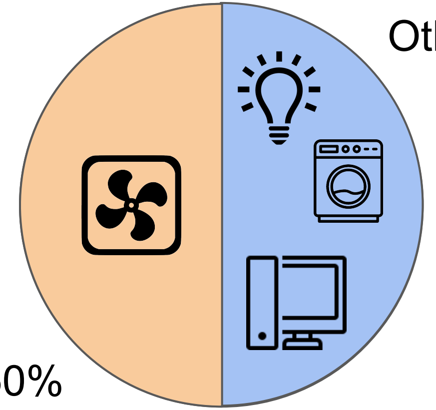
ACM BuildSys 2023, Istanbul, Turkey

HVAC control in smart buildings



Heating, Ventilation, and Air Conditioning (HVAC) system

Other appliances - 50%

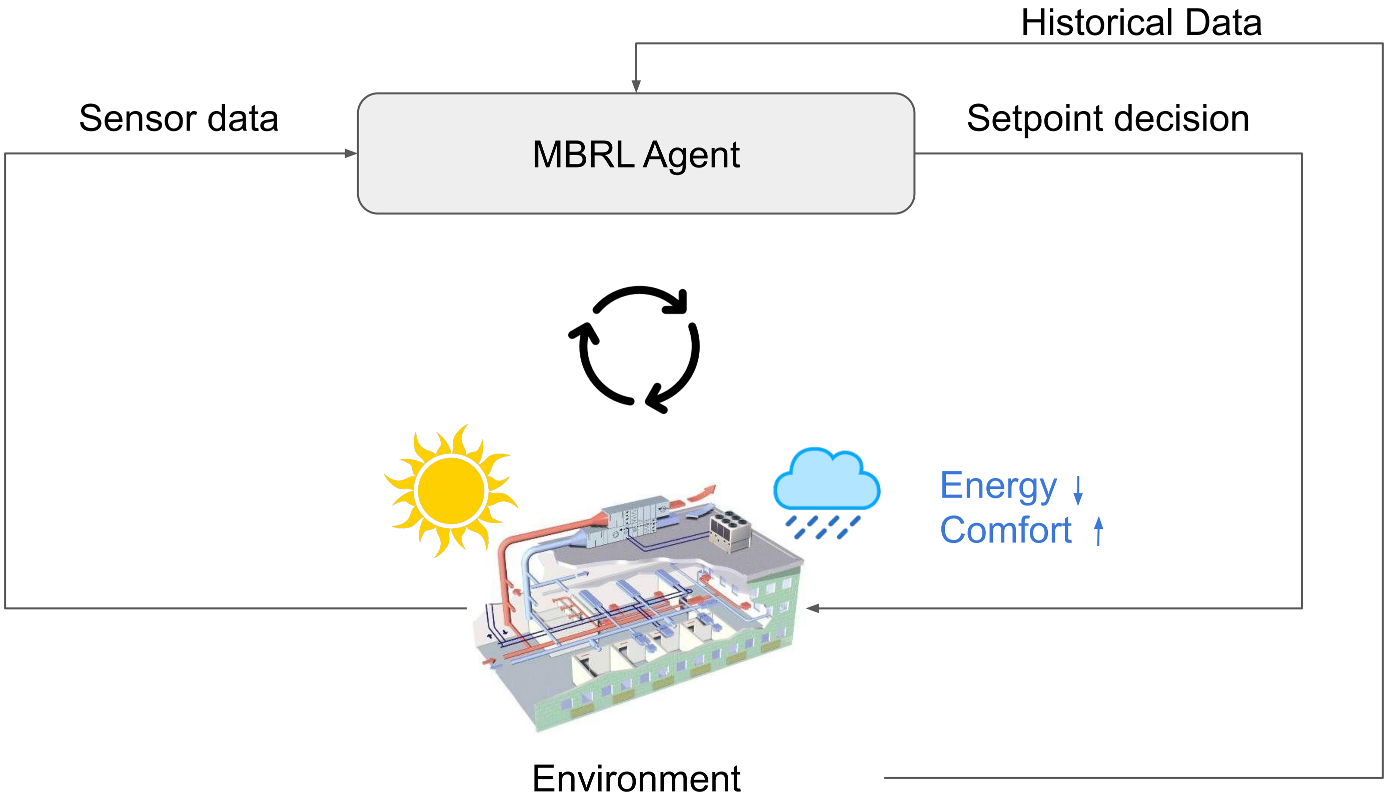


HVAC - 50%

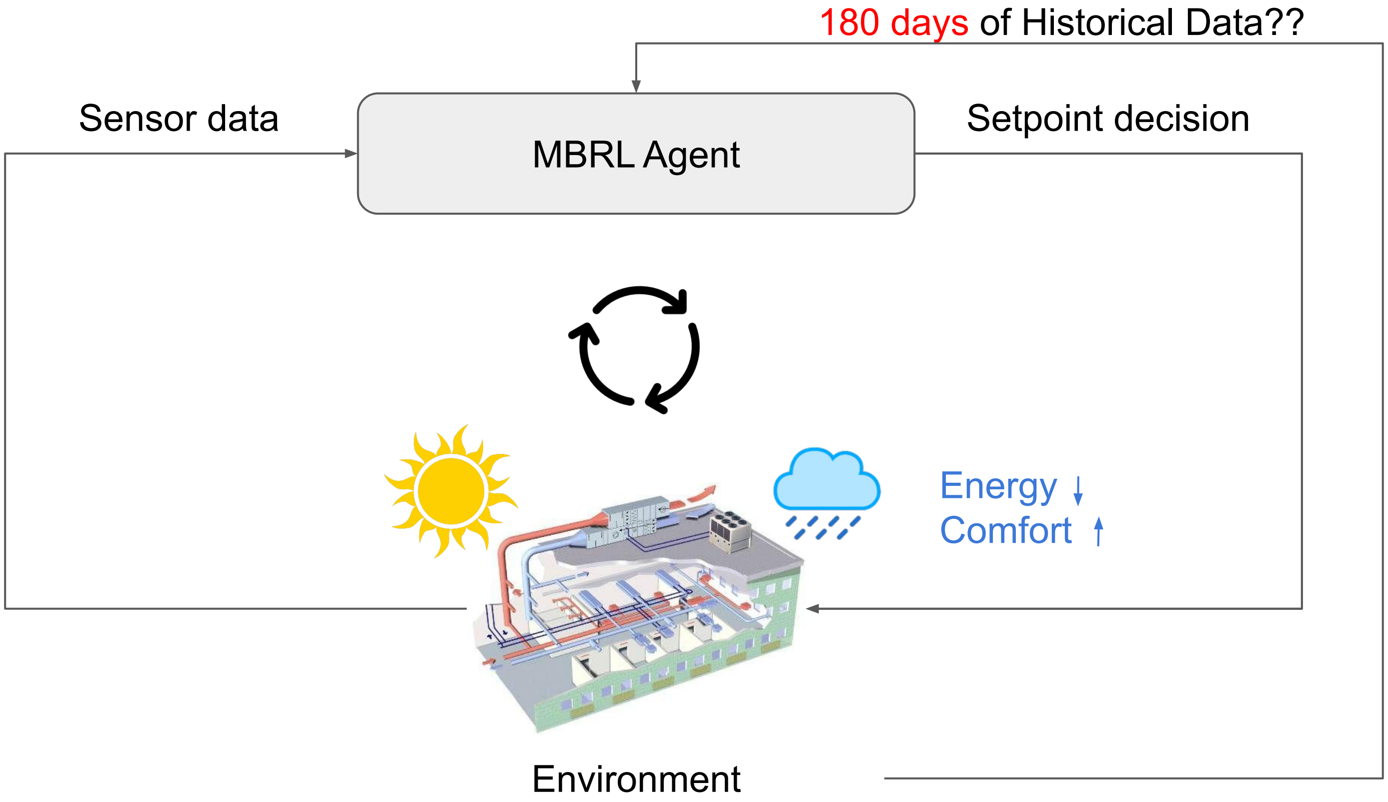
Power Consumption

Thermal Comfort

Model-based Reinforcement Learning for HVAC control



Model-based Reinforcement Learning for HVAC control



[1] Ding et al. MB2C: Model-based Deep Reinforcement Learning for Multi-zone Building Control. ACM BuildSys. 2020

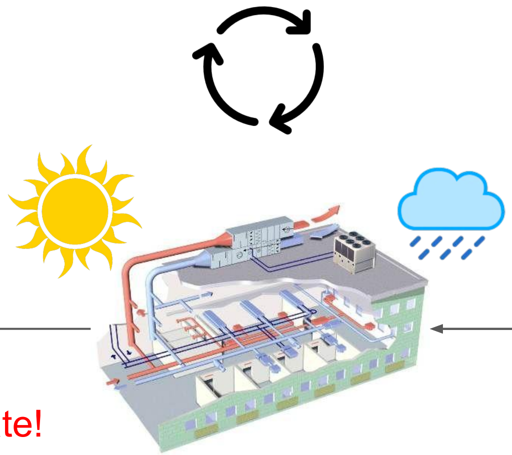
Model-based Reinforcement Learning for HVAC control

180 days of Historical Data??

Sensor data

MBRL Agent

Setpoint decision



Energy ↓
Comfort ↑

Requires too much data to safely operate!
But why?

Environment

[1] Ding et al. MB2C: Model-based Deep Reinforcement Learning for Multi-zone Building Control. ACM BuildSys. 2020

Model-based Reinforcement Learning for HVAC control

MBRL Agent

Sensor data

Setpoint decision

Model-based Reinforcement Learning for HVAC control

Sensor data

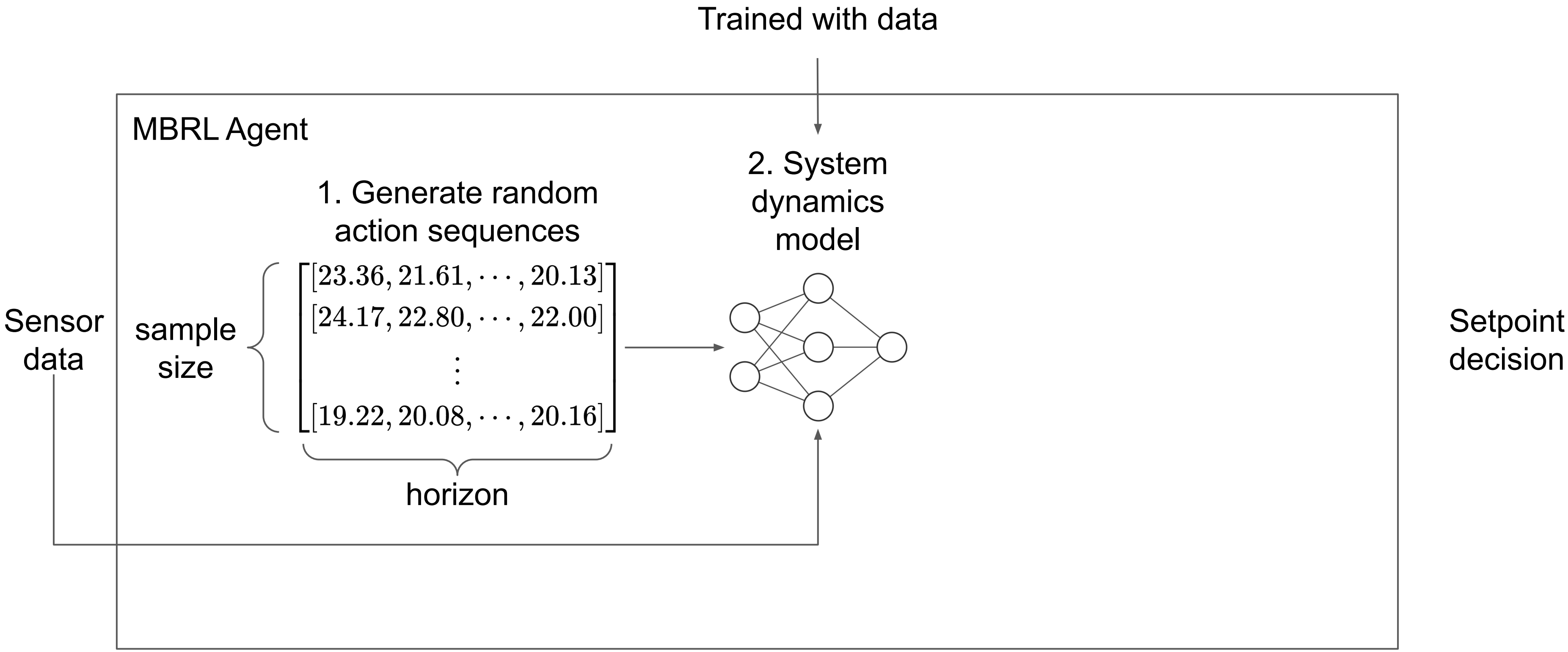
MBRL Agent

1. Generate random action sequences

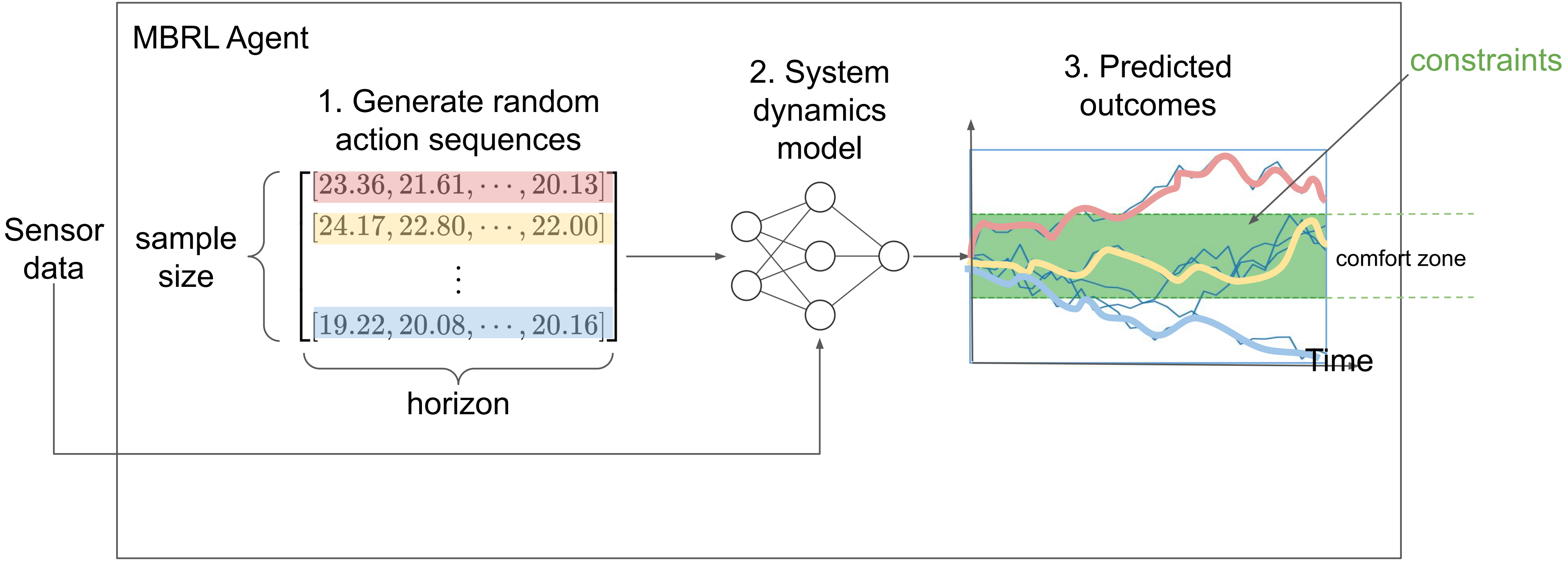
$$\text{sample size}\left\{\underbrace{\begin{bmatrix}[23.36, 21.61, \cdots, 20.13] \\ [24.17, 22.80, \cdots, 22.00] \\ \vdots \\ [19.22, 20.08, \cdots, 20.16]\end{bmatrix}}_{\text{horizon}}\right.$$

Setpoint decision

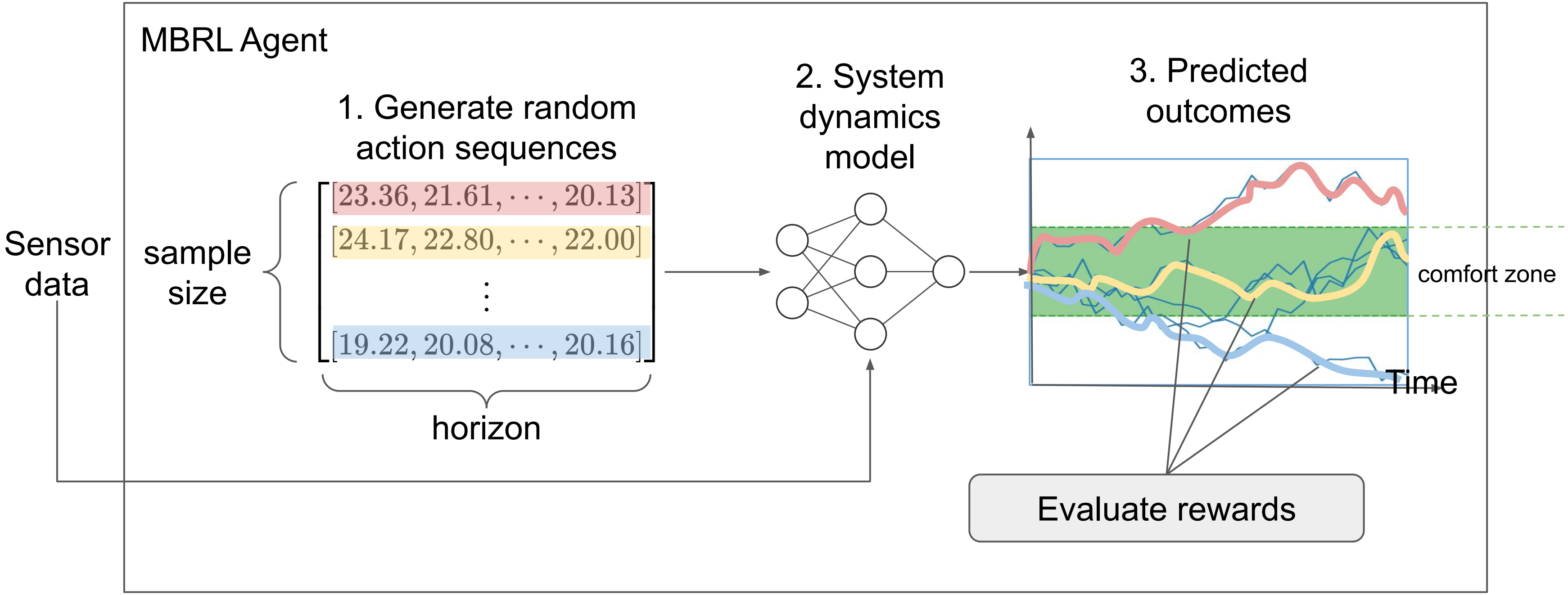
Model-based Reinforcement Learning for HVAC control



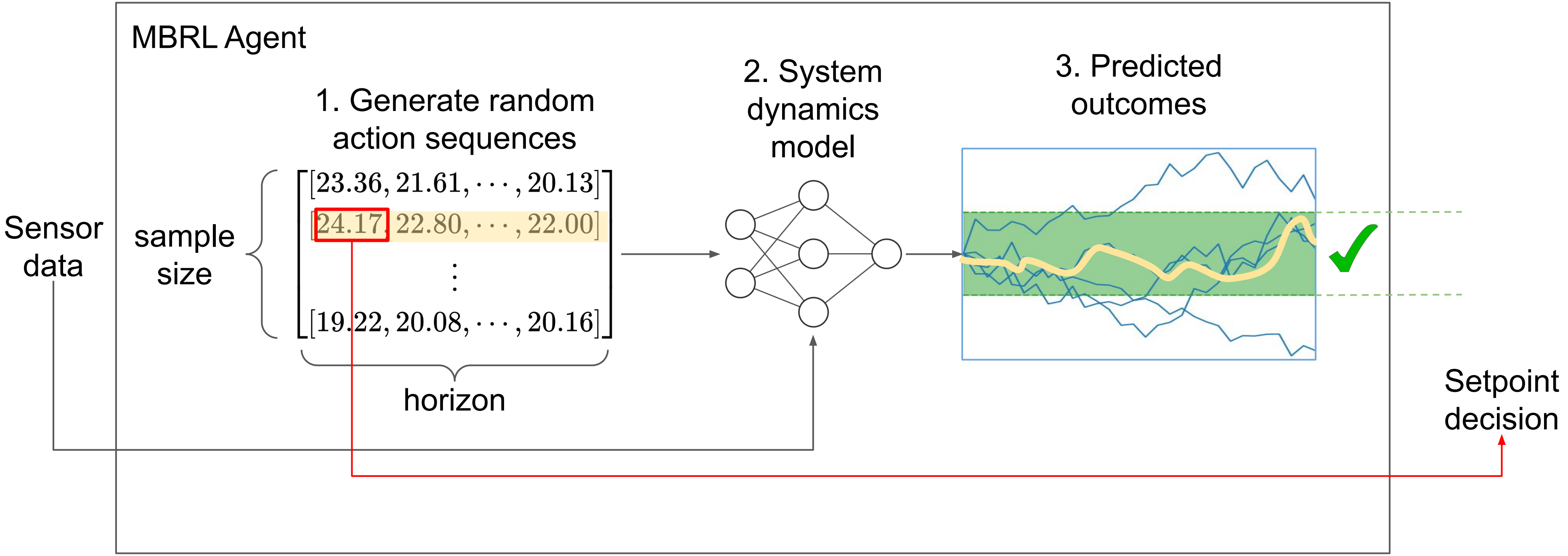
Model-based Reinforcement Learning for HVAC control



Model-based Reinforcement Learning for HVAC control

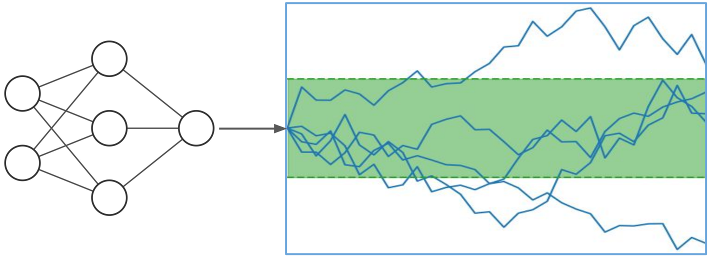


Model-based Reinforcement Learning for HVAC control



Model-based Reinforcement Learning for HVAC control

The model's prediction need to be accurate.
Can we boost accuracy with more data?



Preliminary Experiment #1

Setting | -

Building

- Single thermal zone, AHU and heating coil

Location

- Pittsburgh, PA

Time

- All year

Network architecture

- < 8 | 200 | 200 | 200 | 1 > Sigmoid Linear Units [1]

X variable

- Training dataset size

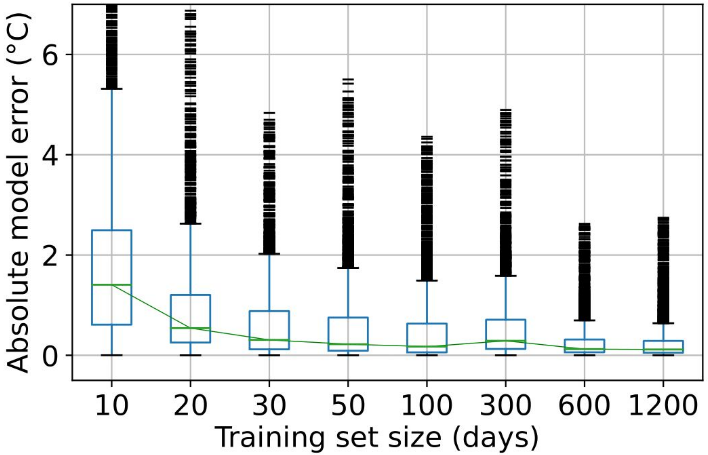
Y variable

- Absolute model errors

[1] Ding et al. MB2C: Model-based Deep Reinforcement Learning for Multi-zone Building Control. ACM BuildSys. 2020

Preliminary Experiment #1

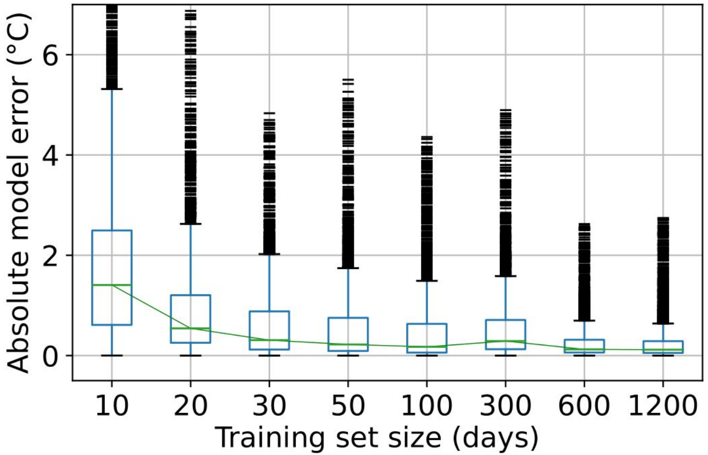
Setting | Result



2% of all predictions

Preliminary Experiment #1

Setting | Result



2% of all predictions

Model error persists with larger datasets.

Why?

Preliminary Experiment #2

Setting	-

Building

- Single thermal zone, AHU and heating coil

Location

- Pittsburgh, PA

Time

- All year

Network architecture

- < 8 | 200 | 200 | 200 | 1 > Sigmoid Linear Units [1]

Dataset size

- 1200 days

X variable

- Model inputs

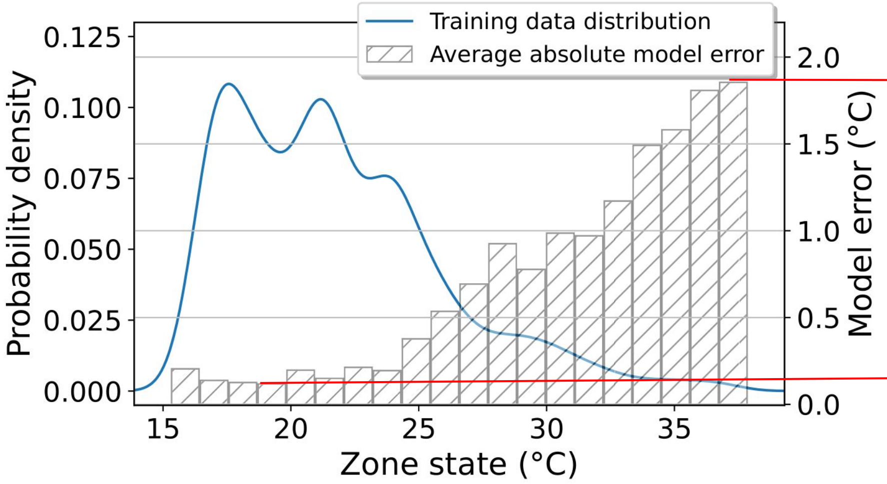
Y variable

1. Amount of training data
2. Average model error

[1] Ding et al. MB2C: Model-based Deep Reinforcement Learning for Multi-zone Building Control. ACM BuildSys. 2020

Preliminary Experiment #2

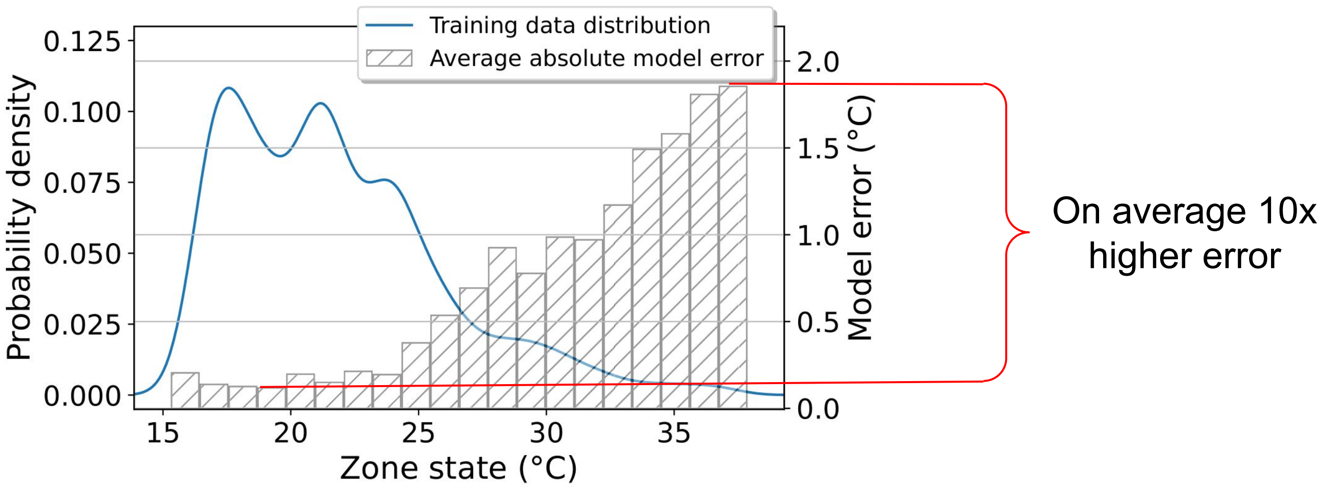
Setting | Result



On average 10x higher error

Preliminary Experiment #2

Setting | Result



Model errors are significantly higher for less frequently seen inputs.

Cause of model errors: distribution shift!

Preliminary Experiment: Insights

Challenge

- Distribution shift causes high model errors, which misleads the controller
- Overfitting on the frequent data won't help with unusual/unseen inputs.

-

Preliminary Experiment: Insights

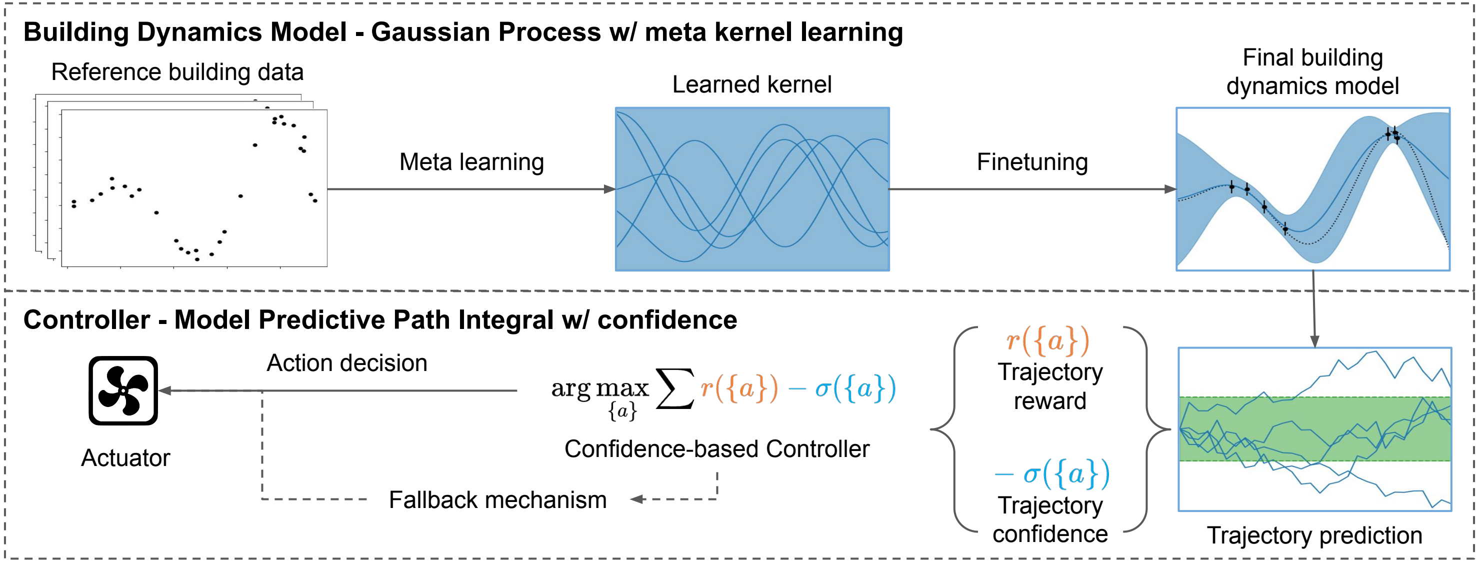
Challenge

- Distribution shift causes high model errors, which misleads the controller
- Overfitting on the frequent data won't help with unusual/unseen inputs.

Solution

- Instead of focusing on fitting an accurate model,
  Can we make the controller to be aware about the uncertainty?

CLUE: HVAC Control Framework



CLUE: a data-efficient and uncertainty-aware model-based RL method

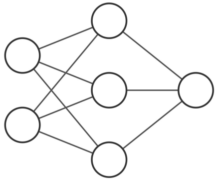
Uncertainty aware controller

Instead of using a traditional controller like this…..

Action sequences

$$\begin{bmatrix} [\cdots \vec{a_0} \cdots] \\ [\cdots \vec{a_1} \cdots] \\ \vdots \\ [\cdots \vec{a_N} \cdots] \end{bmatrix}$$

Dynamics model



Predicted outcomes

$$\begin{bmatrix} [\cdots \vec{o_0} \cdots] \\ [\cdots \vec{o_1} \cdots] \\ \vdots \\ [\cdots \vec{o_N} \cdots] \end{bmatrix}$$

Evaluate rewards

$$a^* = \arg\max_a \sum_{t=0}^{T} \gamma^t r(\arg\max_{s'} P(s'|s,a))$$

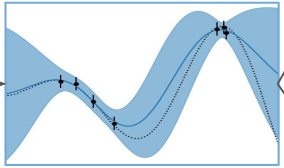
Uncertainty aware controller

Uncertainty aware controller

Action sequences

$$\begin{bmatrix} [\cdots \vec{a}_0 \cdots] \\ [\cdots \vec{a}_1 \cdots] \\ \vdots \\ [\cdots \vec{a}_N \cdots] \end{bmatrix}$$

Gaussian Process dynamics model



Predicted outcomes

$$\begin{bmatrix} [\cdots \vec{o}_0 \cdots] \\ [\cdots \vec{o}_1 \cdots] \\ \vdots \\ [\cdots \vec{o}_N \cdots] \end{bmatrix}$$

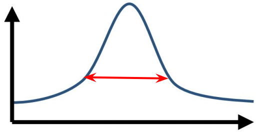
Prediction uncertainties

$$\begin{bmatrix} [\cdots \vec{e}_0 \cdots] \\ [\cdots \vec{e}_1 \cdots] \\ \vdots \\ [\cdots \vec{e}_N \cdots] \end{bmatrix}$$

considers uncertainty in the objective

$$a^* = \arg\max_a \sum_{t=0}^{T} \gamma^t \Big(r\big(\arg\max_{s'} P(s'|s,a)\big) + \lambda \mathbb{E}\big[\big|P(s'|s,a) - \widehat{P(s'|s,a)}\big|\big] \Big)$$

Uncertainty



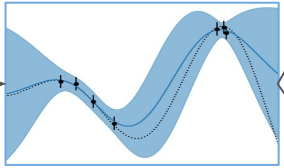
Uncertainty aware controller

Uncertainty aware controller

Action sequences

$$\begin{bmatrix} [\cdots \vec{a}_0 \cdots] \\ [\cdots \vec{a}_1 \cdots] \\ \vdots \\ [\cdots \vec{a}_N \cdots] \end{bmatrix}$$

GP model



Predicted outcomes

$$\begin{bmatrix} [\cdots \vec{o}_0 \cdots] \\ [\cdots \vec{o}_1 \cdots] \\ \vdots \\ [\cdots \vec{o}_N \cdots] \end{bmatrix}$$

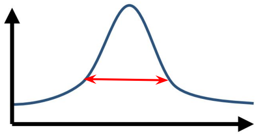
Prediction uncertainties

$$\begin{bmatrix} [\cdots \vec{e}_0 \cdots] \\ [\cdots \vec{e}_1 \cdots] \\ \vdots \\ [\cdots \vec{e}_N \cdots] \end{bmatrix}$$

considers uncertainty in the objective

$$a^* = \arg\max_a \sum_{t=0}^{T} \gamma^t \left(r(\arg\max_{s'} P(s'|s,a)) + \lambda \mathbb{E}[|P(s'|s,a) - \widehat{P(s'|s,a)}|] \right)$$

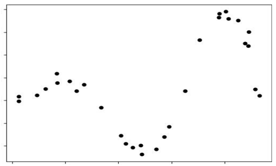
Uncertainty



This solves safety…
But what about data efficiency?

Gaussian Process Data Efficiency

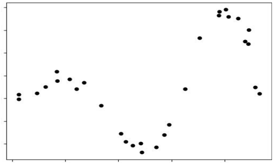
Data



+

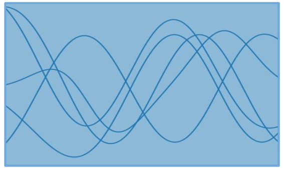
Gaussian Process Data Efficiency

Data



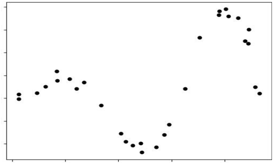
+

Kernel



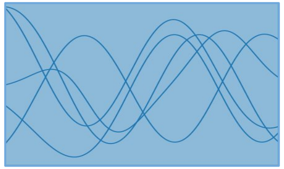
Gaussian Process Data Efficiency

Data



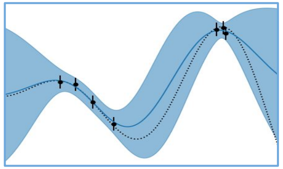
+

Kernel



=

Model



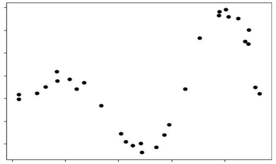
RBF kernel: $$k(x, x') = \theta_{\text{scale}} \exp\left(-\frac{1}{2}(x - x')^\top \Theta^{-2}(x - x')\right)$$

$$\theta_{\mathcal{GP}} : \{\theta_{\text{scale}}, \Theta \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}\}, \mathcal{X} : \text{input space}$$

parameter space scales quadratically with feature number.

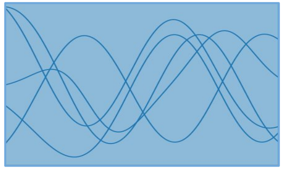
Gaussian Process Data Efficiency

Data



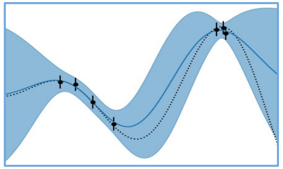
+

Kernel



=

Model



RBF kernel: $$k(x, x') = \theta_{\text{scale}} \exp\left(-\frac{1}{2}(x - x')^\top \Theta^{-2}(x - x')\right)$$

$$\theta_{\mathcal{GP}} : \{\theta_{\text{scale}}, \Theta \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}\}, \mathcal{X} : \text{input space}$$

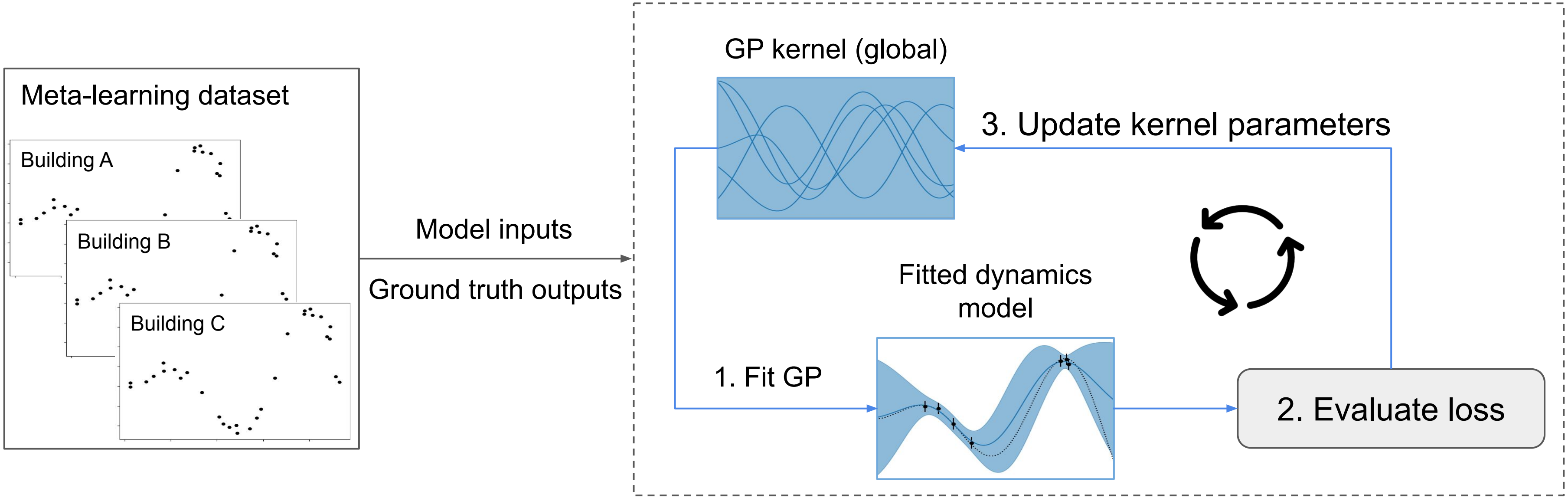
parameter space scales quadratically with feature number.

Challenge: how to efficiently tune the GP kernel hyperparameters?

Meta-Learned Gaussian Process

Solution: meta kernel learning

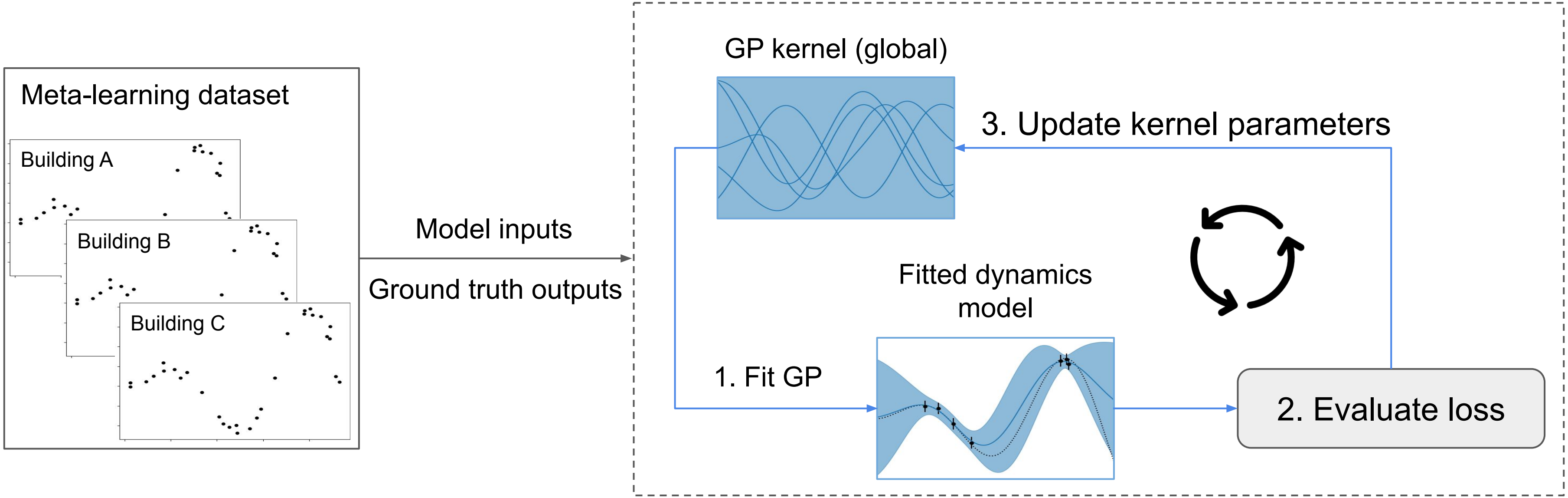
Use data from similar buildings to tune the hyperparameters!



Meta-Learned Gaussian Process

Solution: meta kernel learning

Use data from similar buildings to tune the hyperparameters!



Solved data efficiency!

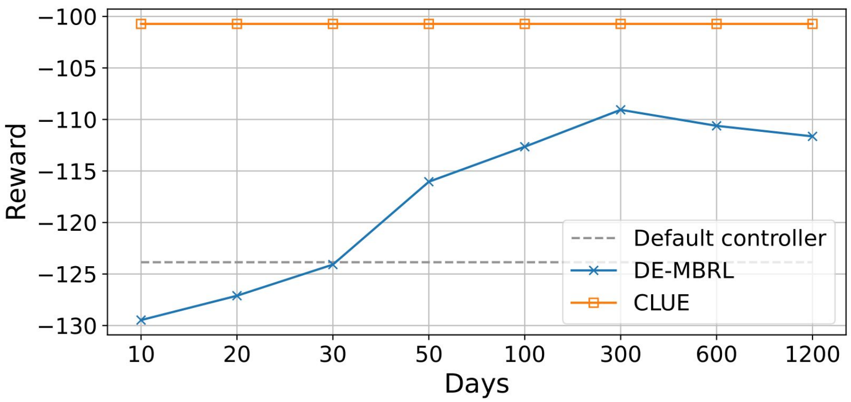
Evaluation of *CLUE*

Setting	-

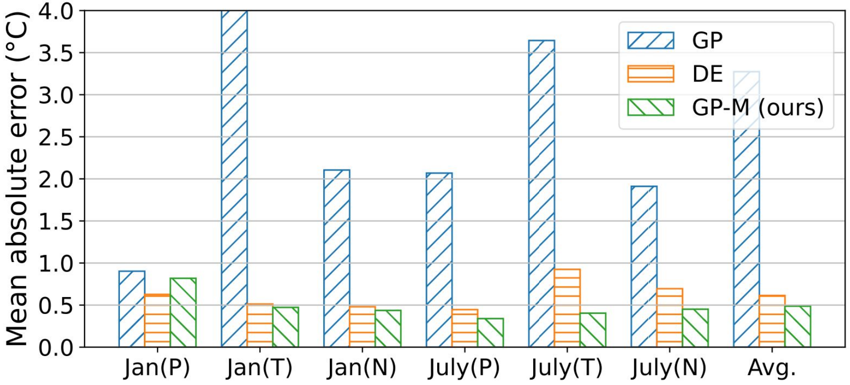
- Three locations: Pittsburgh, Tucson, New York
- Data efficiency
- Building control performance
  - Energy usage
  - Comfort violation rate

Evaluation - Data Efficiency

Setting | Result



CLUE converges >30x faster than previous SOTA

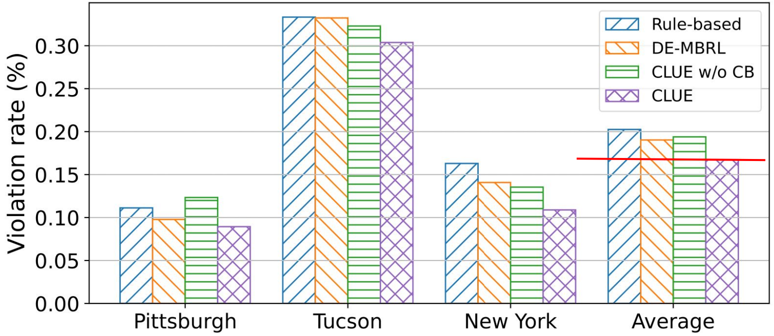


Produces more accurate model given the same data

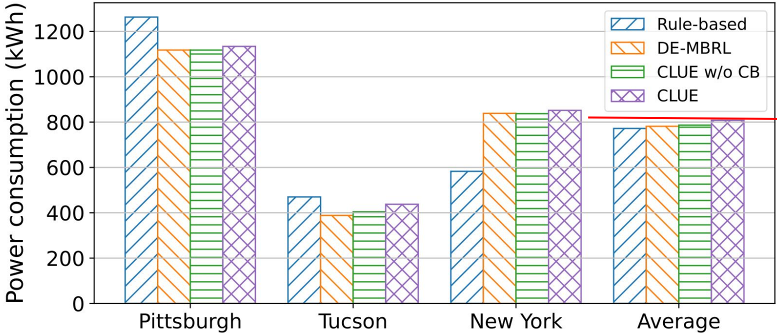
[1] Ding et al. MB2C: Model-based Deep Reinforcement Learning for Multi-zone Building Control. ACM BuildSys. 2020

Evaluation - Building Control

Setting | Result



CLUE produces 12.07% lower violation rates compared w/ previous SOTA



Similar energy saving w/ previous SOTA

Conclusion

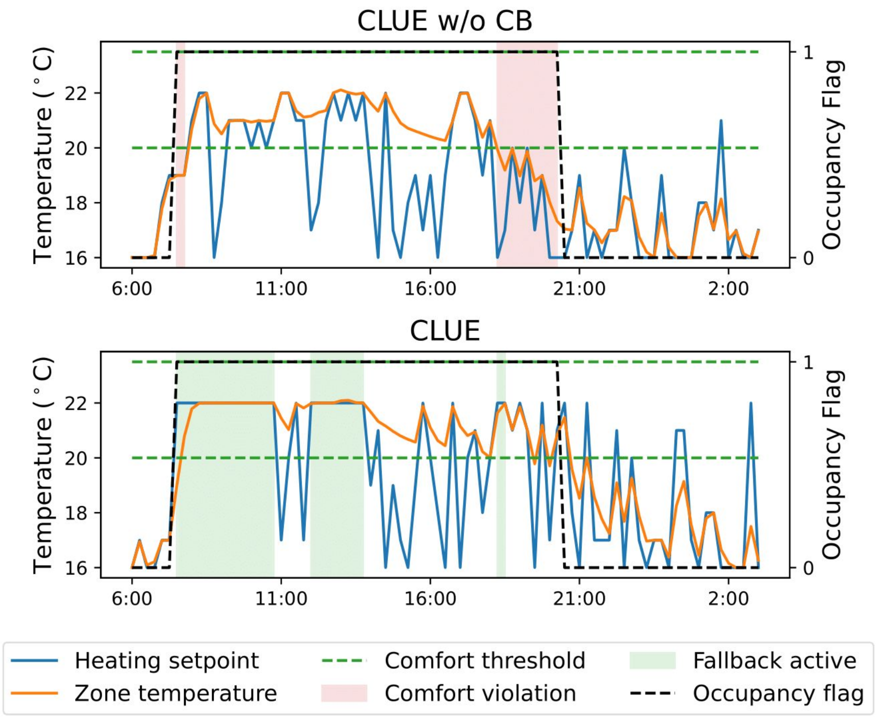
- We are the first to include epistemic uncertainty estimation in shooting-based control for HVAC.
- We proposed *CLUE*, a data-efficient and safe MBRL control method for HVAC, consists of meta kernel learning and confidence-based control.
- We evaluated *CLUE* with extensive simulation experiments in three different locations.

Code+data available at https://github.com/ryeii/CLUE/

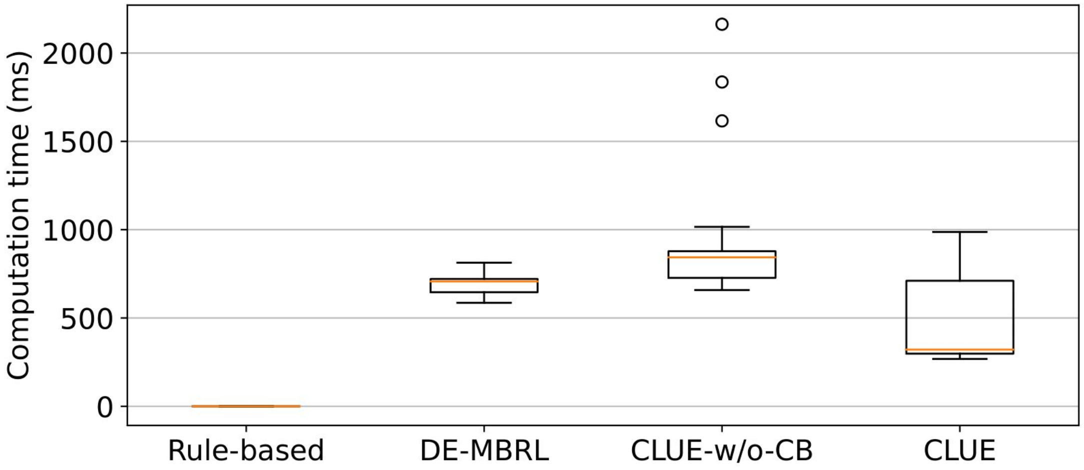
Uncertainty estimation

Location	Time	Deep Ensemble [11]			GP			GP-M (ours)		
		Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Pittsburgh, PA	January	.796±.00	.521±.01	.740±.01	.877±.00	**.803±.00**	**.958±.00**	**.884±.00**	.768±.00	.677±.00
	July	.831±.01	**.851±.09**	.160±.10	.840±.00	.809±.00	.763±.00	**.961±.00**	.056±.00	**.999±.00**
Tucson, AZ	January	.736±.01	.439±.08	.693±.12	.847±.00	**.697±.00**	**.844±.00**	**.932±.00**	.341±.00	.694±.00
	July	.650±.00	.489±.00	.827±.00	.844±.00	**.854±.00**	.860±.00	**.947±.00**	.036±.00	**.999±.00**
New York, NY	January	.830±.00	.403±.02	.816±.00	.855±.00	**.883±.00**	.728±.00	**.965±.00**	.299±.00	**.900±.00**
	July	.679±.00	.373±.01	.812±.01	.797±.00	**.934±.00**	.718±.00	**.953±.00**	.205±.00	**.947±.00**

Performance analysis



Computation overhead

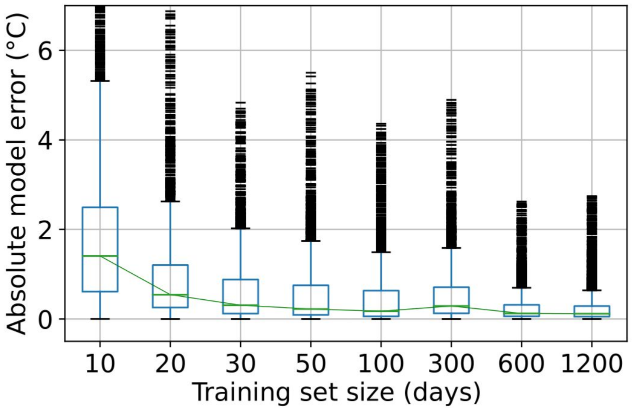


Motivation Experiments

1. Can we mitigate high model errors by training on more data?

Motivation Experiments

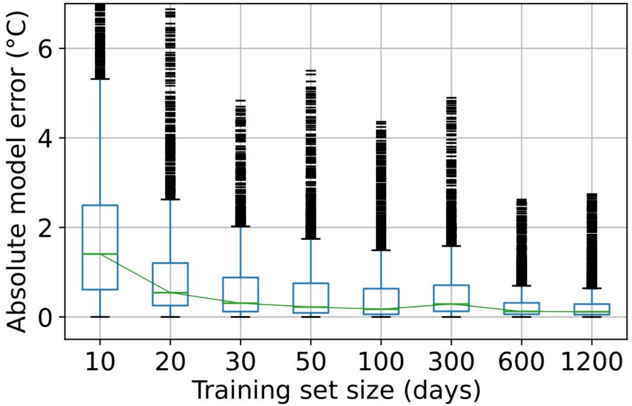
1. Can we mitigate high model errors by training on more data?



No, high model errors persists even after very large dataset is used

Motivation Experiments

1. Can we mitigate high model errors by training on more data?

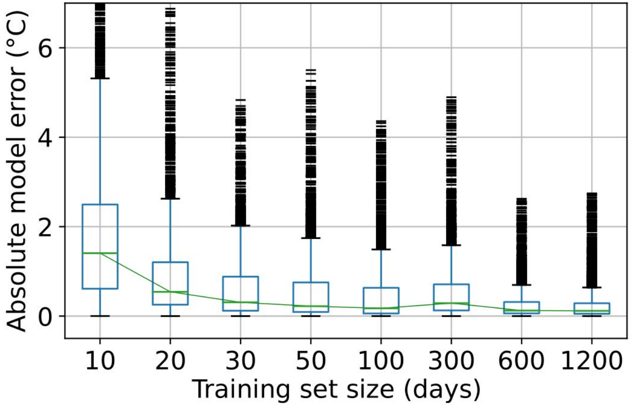


No, high model errors persists even after very large dataset is used

2. Can we let the building system tolerate short periods of controller glitches?

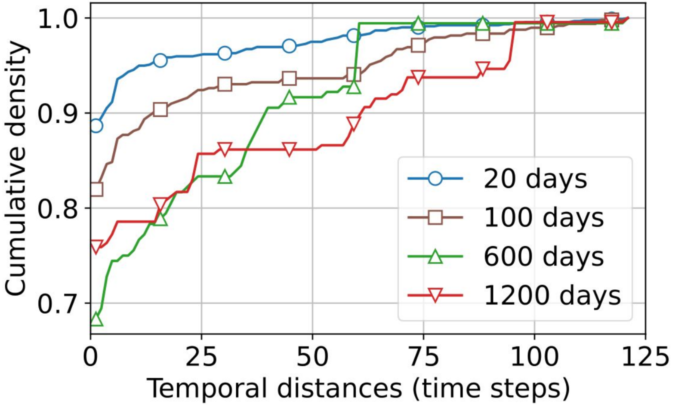
Motivation Experiments

1. Can we mitigate high model errors by training on more data?



No, high model errors persists even after very large dataset is used

2. Can we let the building system tolerate short periods of controller glitches?



No, high model errors often appear in clusters