Prediction-based One-shot Dynamic Parking Pricing

Yonsei University, South Korea

Seoyoung Hong
seoyoungh@yonsei.ac.kr

Heejoo Shin
hes002@uscd.edu

Jeongwhan Choi
jeongwhan.choi@yonsei.ac.kr

Noseong Park
noseong@yonsei.ac.kr

Contents

- Introduction
- Preliminary
- Method
- Experiments
- Case Studies
- Conclusion

Introduction (1/7)

Prediction-driven Optimal Decision-Making

"A pearl is <u>worthless</u> as long as it is in its shell."

- **Prediction-driven optimal decision-making** is to train a prediction model first and adjust the actionable input (e.g. price) to optimize prediction results.
- Although the prediction itself is valuable, **action** must be taken to utilize the prediction result and realize its value.

Introduction (2/7)

Prediction-driven Optimal Decision-Making

- Prediction-driven optimal decision-making has been implemented by many existing papers.
    - e.g., dynamic toll [1] and airline profit maximization [2,3]

- Based on the prediction results, an optimization problem is solved, either analytically or via optimization algorithms.
    - e.g., greedy and gradient-based optimization

[1] An et al. DyETC: Dynamic Electronic Toll Collection for Traffic Congestion Alleviation. AAAI, 2018.
[2] An et al. MAP: Frequency-based maximization of airline profits based on an ensemble forecasting approach. KDD, 2016.
[3] Li et al. Large-Scale Data-Driven Airline Market Influence Maximization. KDD, 2021.

Introduction (3/7)

Prediction-driven Optimization Dilemma

- Existing approaches failed to solve the **prediction-driven optimization dilemma**, which states that if the speed is high, the quality of the solution is low, and if the quality is high, the speed is low.
- However, we propose a "one-shot" prediction-based optimization model that can be quickly solved in **O(1)** and has a **high-quality** solution.

Introduction (4/7)

Dynamic Parking Pricing

- Dynamic pricing is to **dynamically adjust the price** to meet demand.
- Many metropolitan cities are notorious for severe shortages of parking spots.
- The benefits of dynamic parking pricing
  - reduced congestion, reduced time wasted, reduced pollution and increased revenue

Introduction (5/7)

Dynamic Parking Pricing

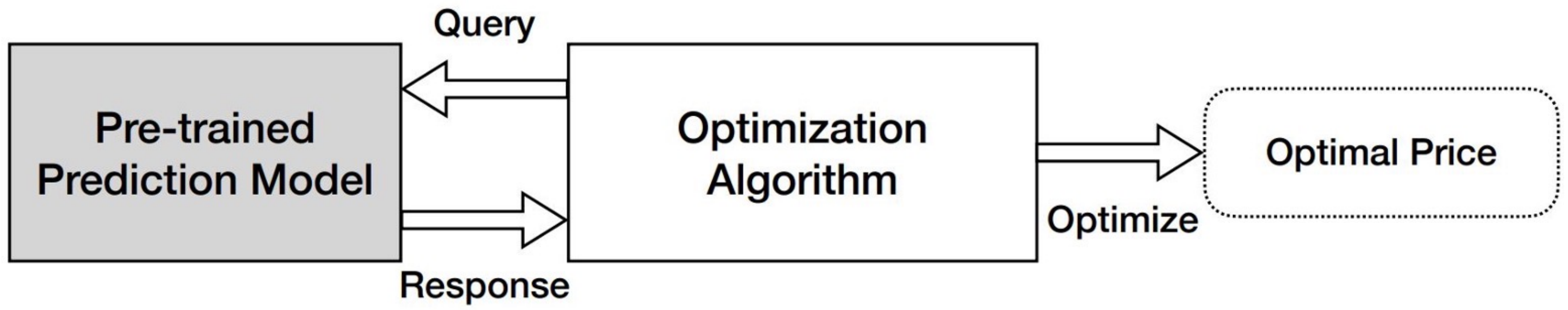
- The parking price optimization problem can be formulated as **adjusting prices to minimize the error** between predicted and target occupancy rates.
- Some papers [5,6] have adopted prediction-based optimization, formulating pricing schemes based on predicted occupancy rates by implementing various machine learning prediction models.

[5] Fabusuyi et al. Rethinking performance based parking pricing: A case study of SFpark. *Transportation Research Part A: Policy and Practice* 115, 90-101, 2018.
[6] Saharan et al. An efficient smart parking pricing system for smart city environment: A machine-learning based approach. *Future Generation Computer Systems* 106, 622-640, 2020.

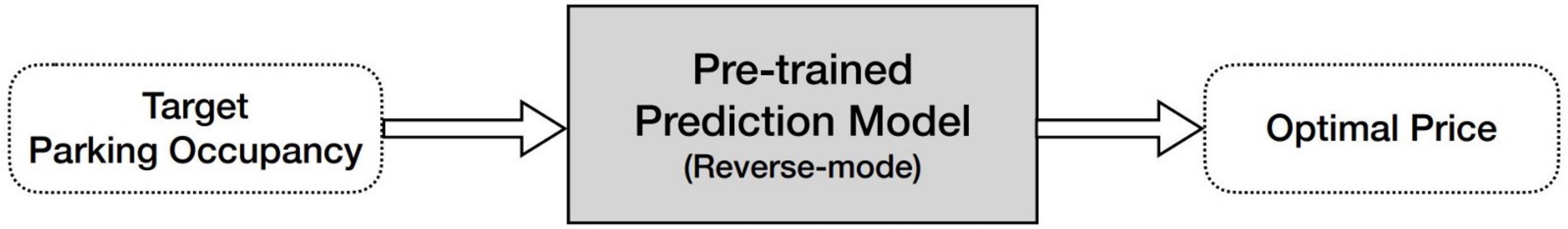
Introduction (6/7)

Main Contributions

1. We design a sophisticated future parking occupancy **prediction** model based on **NODEs**.
2. We propose a novel one-shot price **optimization** that can find the optimal price with **a single query**, owing to the continuous and bijective characteristics of Neural ODEs.



(a) Existing black-box query-based prediction-driven optimization



(b) Our “one-shot” white-box prediction-driven optimization

Introduction (7/7)

Main Contributions

3. In our experiments with two real-world datasets, **our prediction model outperforms** many existing temporal and spatiotemporal models.

4. Our one-shot **optimization finds better solutions faster** in comparison with other prediction-driven optimization paradigms.

Preliminary (1/2)

Neural Ordinary Differential Equations (NODEs)

- We use neural ordinary differential equations [4] for parking occupancy rate prediction.
- NODEs **learn differential equations as a neural network**. They solve the following integral problem to calculate the last hidden vector $\boldsymbol{z}(T)$ from the initial vector $\boldsymbol{z}(0)$:

$$\boldsymbol{z}(T) = \boldsymbol{z}(0) + \int_0^T f\big(\boldsymbol{z}(t);\ \boldsymbol{\theta}_f\big)dt$$

- where $f\left(\mathbf{z}(t); \theta_f\right)$, which we call the ODE function, is a neural network to learn $\frac{\partial \mathbf{z}(t)}{\partial t}$.

[4] Chen et al. Neural Ordinary Differential Equations. NeurIPS, 2018.

Preliminary (2/2)

Neural Ordinary Differential Equations (NODEs)

Reverse-mode Integral

- Our price optimization corresponds to finding the unique $\mathbf{z}^*(0)$ that leads $\mathbf{z}^*(T)$.
- For finding $\mathbf{z}^*(0)$, we can solve the below **reverse-mode integral** problem of NODEs:

$$\boldsymbol{z}^*(0) = \boldsymbol{z}^*(T) - \int_0^T f\big(\boldsymbol{z}(t);\ \boldsymbol{\theta}_f\big)dt$$

Method (1/5)

Parking Occupancy Prediction

Our prediction model consists of three modules:

1. The **initial prediction module** with spatiotemporal processing

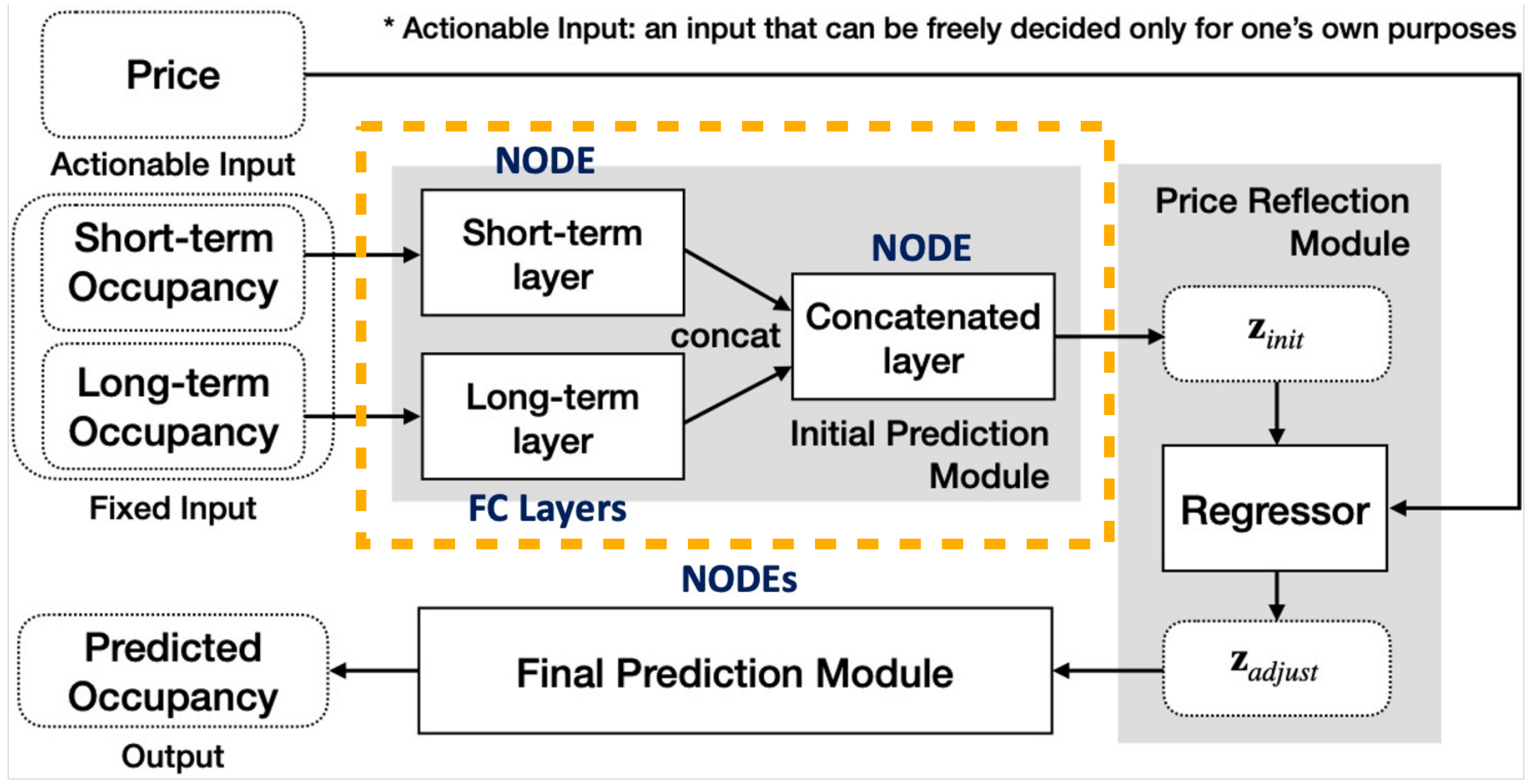
- This processes the short-term and long-term occupancy rate information and produces the initial occupancy rate prediction.

$$\mathbf{z}_{init} = \mathbf{c}(1) = \mathbf{c}(0) + \int_0^1 m(\mathbf{c}(t), \mathbf{H}_{long}; \boldsymbol{\theta}_m) dt$$

$$\mathbf{c}(0) = FC_{N\times \dim(\mathbf{H}_{short})\rightarrow N\times 1}(\mathbf{H}_{short})$$

- $\mathbf{H}_{short}$: the output of short-term layer
- $\mathbf{H}_{long}$: the output of long-term layer

$$\frac{d}{dt}\begin{bmatrix}\mathbf{c}(t)\\ \mathbf{H}_{long}\end{bmatrix} = \begin{bmatrix} m(\mathbf{c}(t), \mathbf{H}_{long}; \boldsymbol{\theta}_m) \\ 0\end{bmatrix}$$



$$\begin{aligned} m(\mathbf{c}(t), \mathbf{H}_{long}; \boldsymbol{\theta}_m) &= \psi(FC_{N\times 1\rightarrow N\times 1}(\mathbf{u}_1)),\\ \mathbf{u}_1 &= \sigma(FC_{N\times 1\rightarrow N\times 1}(\mathbf{u}_0)),\\ \mathbf{u}_0 &= \sigma(FC_{N\times (L+1)\rightarrow N\times 1}(\mathbf{c}(t) \oplus \mathbf{H}_{long})) \end{aligned}$$

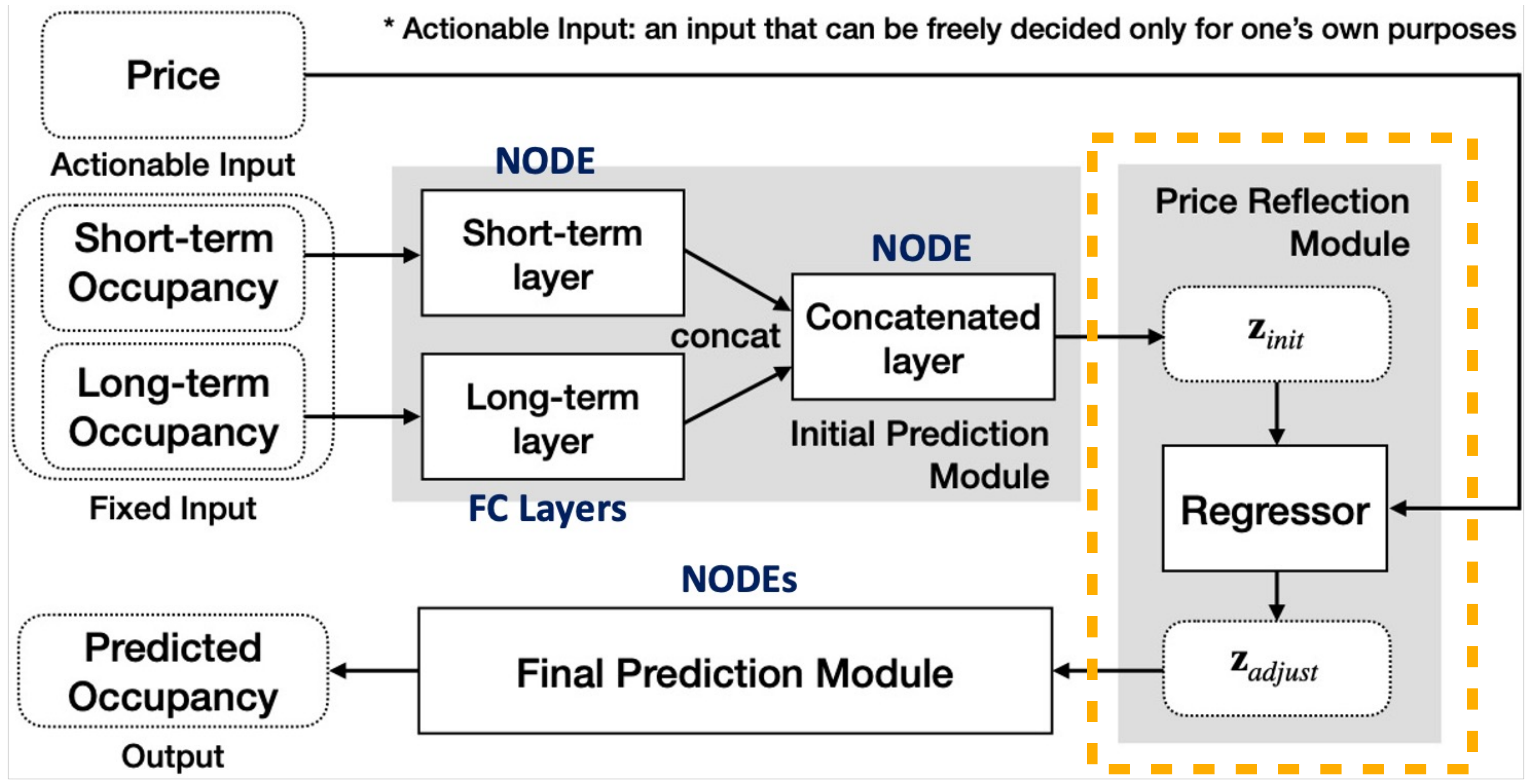
Method (2/5)

Parking Occupancy Prediction

Our prediction model consists of three modules:

2. The **price reflection** module with the regressor

- This module adjusts $\mathbf{z}_{init}$ created by the initial prediction module to $\mathbf{z}_{adjust} \in \mathbb{R}^{N \times 1}$ by considering the price information $\boldsymbol{p}$.



$$\mathbf{z}_{adjust} = \mathbf{z}_{init} - \left((\boldsymbol{c} \odot \boldsymbol{p})^T + \boldsymbol{b} \right)$$

➢ $\boldsymbol{c} \in [0, \infty]^N$: a coefficient vector (the demand elasticity on price changes)

➢ $\boldsymbol{b} \in \mathbb{R}^{N \times 1}$: the bias vector

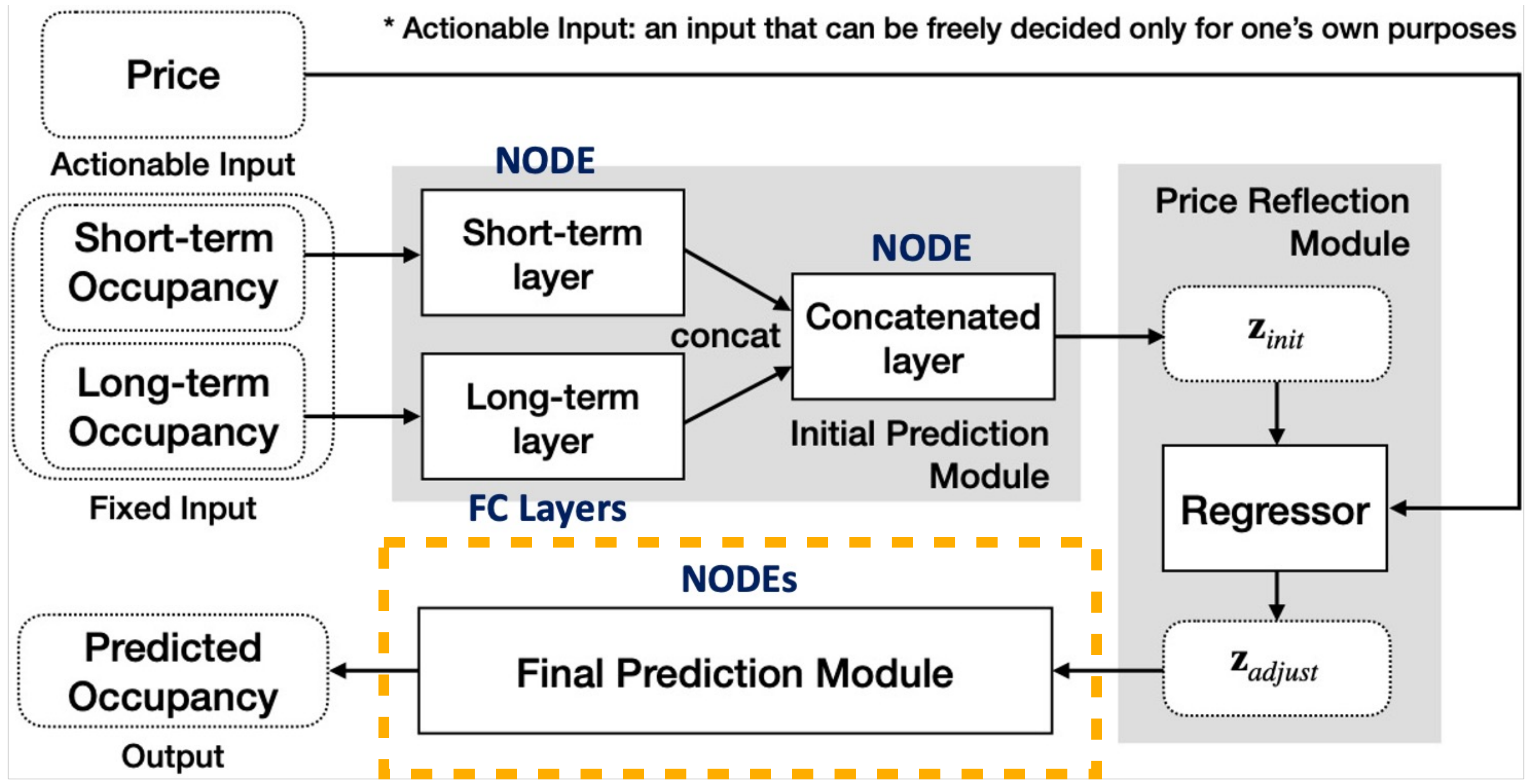
Method (3/5)

Parking Occupancy Prediction

Our prediction model consists of three modules:

3. The **final prediction** module with a series of NODEs

- This module evolves $\mathbf{z}_{adjust}$ to the final prediction $\hat{\mathbf{y}} \in [0,1]^{(N \times 1)}$.
- We adopt the following M NODE layers:



$$\boldsymbol{y}_1(1) = z_{adjust} + \int_0^1 j_1(\boldsymbol{y}_1(t); \boldsymbol{\theta}_{j_1})dt,$$

$$\boldsymbol{y}_i(1) = \boldsymbol{y}_{i-1}(1) + \int_0^1 j_i(\boldsymbol{y}_i(t); \boldsymbol{\theta}_{j_i})dt, 1 < i < M,$$

$$\boldsymbol{y}_M(1) = \boldsymbol{y}_{M-1}(1) + \int_0^1 j_M(\boldsymbol{y}_M(t); \boldsymbol{\theta}_{j_M})dt,$$

$$\begin{aligned} j_i(\boldsymbol{y}_i(t); \boldsymbol{\theta}_{j_i}) &= \psi(FC_{N\times1 \to N\times1}(\boldsymbol{o}_1)), \\ \boldsymbol{o}_1 &= \sigma(FC_{N\times1 \to N\times1}(\boldsymbol{o}_0)), \\ \boldsymbol{o}_0 &= \sigma(FC_{N\times1 \to N\times1}(\boldsymbol{y}_i(t))). \end{aligned}$$

where the future occupancy prediction $\hat{\mathbf{y}} = \mathbf{y}_M(1)$.

Method (4/5)

Parking Price Optimization

Problem definition

- Given the short-term history information $\{\mathbf{s}_i\}_{i=1}^{K}$, the long-term history information $\{\mathbf{l}_i\}_{i=1}^{L}$, and the target occupancy rates $\mathbf{y}^*$, we want to find the optimal prices $\mathbf{p}^*$ that lead $\mathbf{y}^*$ as follows:

$$
\begin{aligned}
\underset{\boldsymbol{p}^*}{\operatorname{argmin}} \quad & \frac{\| \hat{\boldsymbol{y}} - \boldsymbol{y}^* \|_1}{N}, \\
\text{subject to} \quad & \boldsymbol{p}_{min} \leq \boldsymbol{p}^* \leq \boldsymbol{p}_{max}, \\
& \hat{\boldsymbol{y}} = \xi\left(\boldsymbol{p}^*, \{\boldsymbol{s}_i\}_{i=1}^{K}, \{\boldsymbol{l}_i\}_{i=1}^{L}; \boldsymbol{\theta}_\xi\right)
\end{aligned}
$$

where ξ is our pre-trained occupancy prediction model.

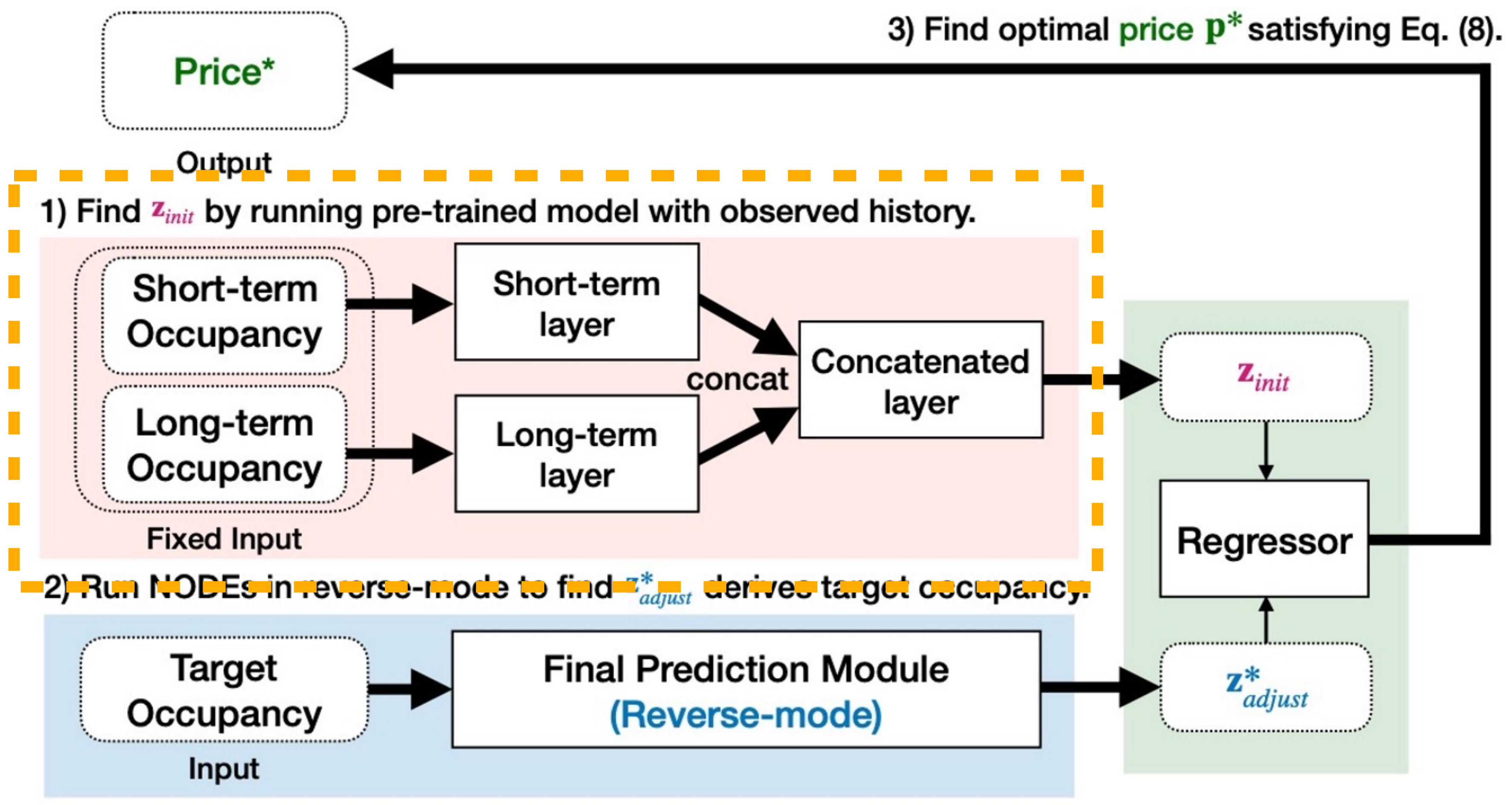
Method (5/5)

Parking Price Optimization

The following three steps comprise the optimization process:

1. We feed the given short-term and long-term history information and **derive** $\mathbf{z}_{init}$ from the initial prediction module.
2. Set the target occupancy rates $\mathbf{y}^*$ that we want to achieve and then **solve the series of reverse-mode integral problems** to derive $\mathbf{z}^*_{adjust}$.
3. The **optimal price** vector p^* can be calculated as follows:

$$\boldsymbol{p}^* = \frac{\boldsymbol{z}_{init} - \boldsymbol{z}^*_{adjust} - \boldsymbol{b}}{\boldsymbol{c}}$$



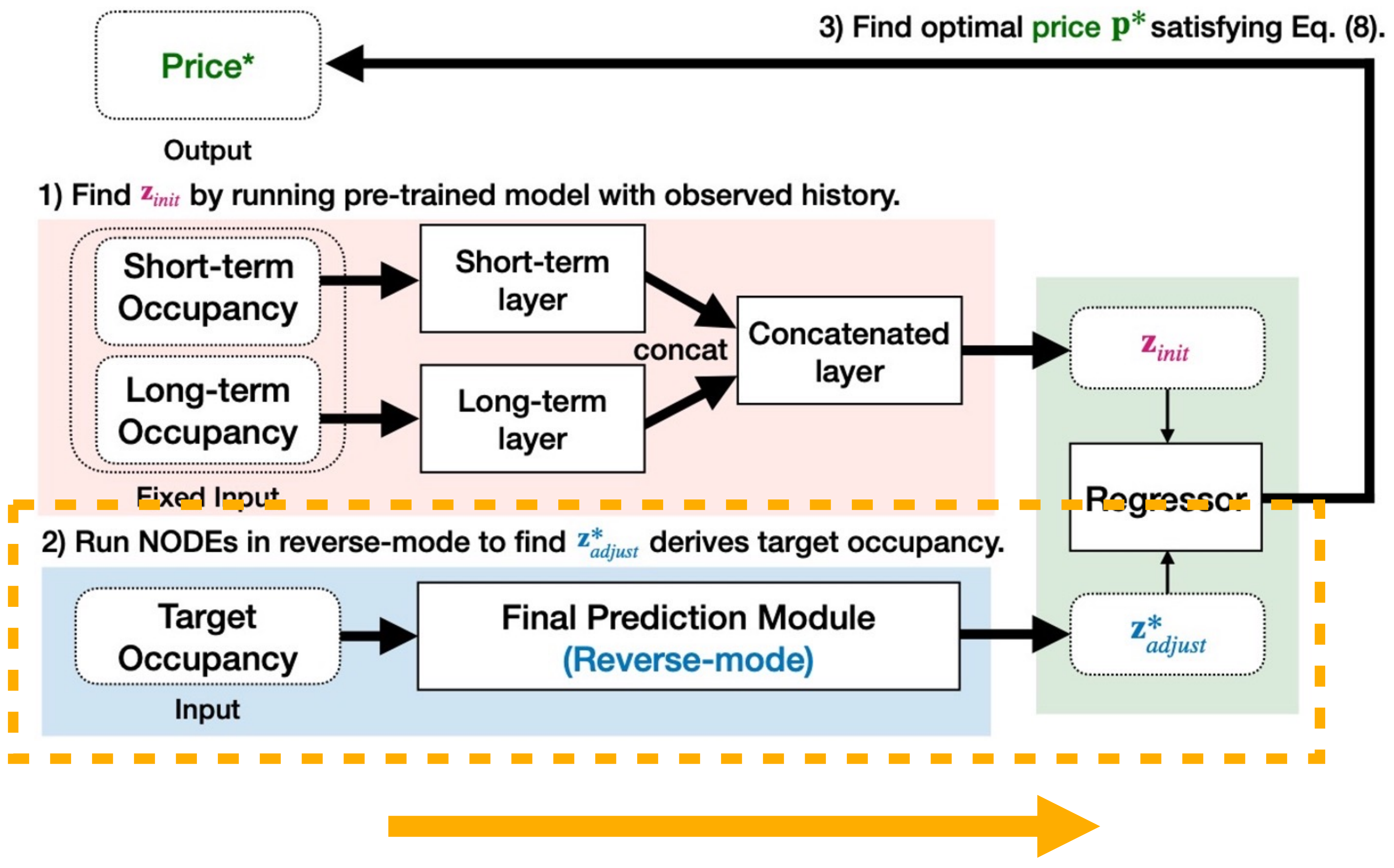
Method (5/5)

Parking Price Optimization

The following three steps comprise the optimization process:

1. We feed the given short-term and long-term history information and **derive** $\mathbf{z}_{init}$ from the initial prediction module.
2. Set the target occupancy rates $\mathbf{y}^*$ that we want to achieve and then **solve the series of reverse-mode integral problems** to derive $\mathbf{z}^*_{adjust}$.
3. The **optimal price** vector p^* can be calculated as follows:

$$\boldsymbol{p}^* = \frac{\boldsymbol{z}_{init} - \boldsymbol{z}^*_{adjust} - \boldsymbol{b}}{\boldsymbol{c}}$$



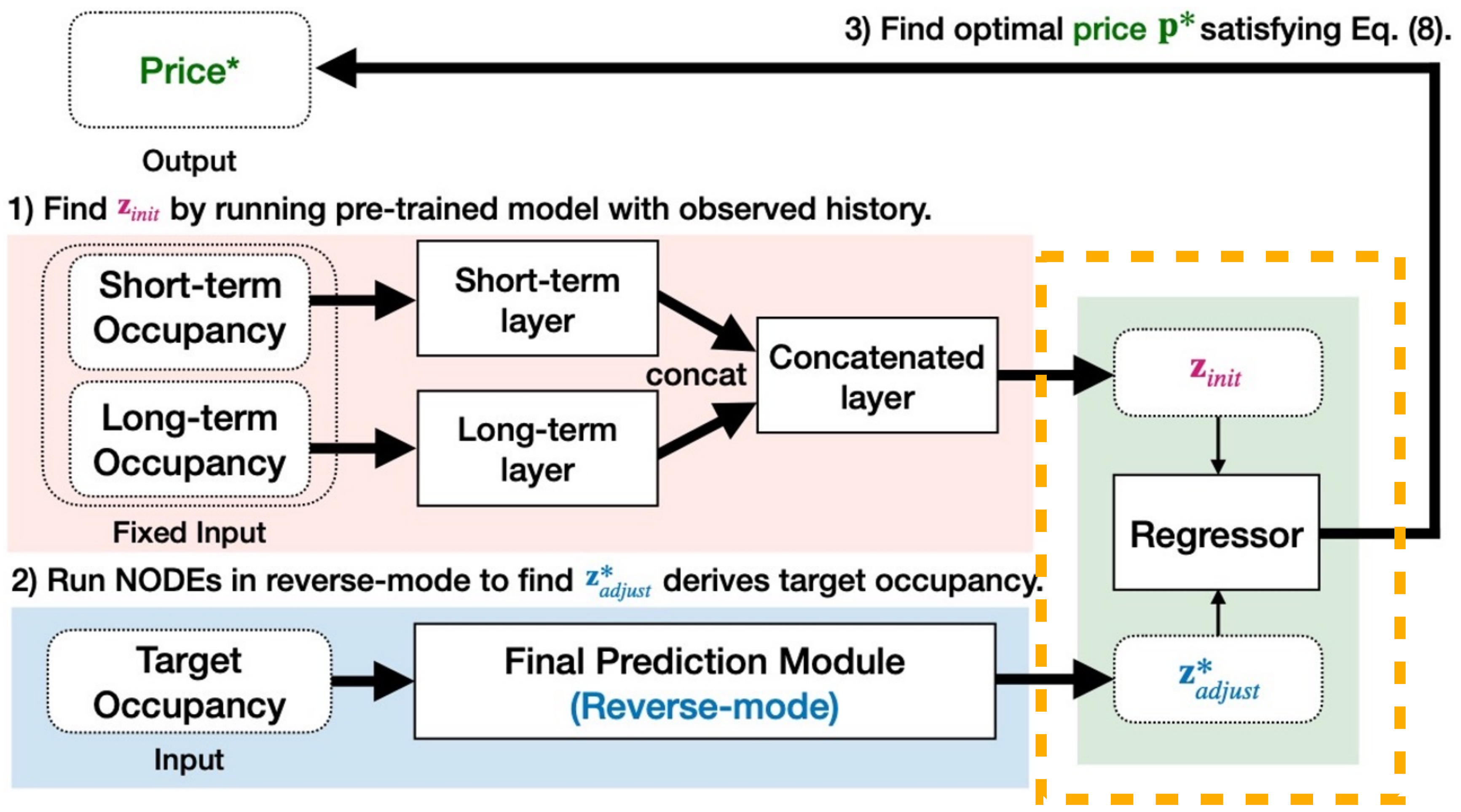
Method (5/5)

Parking Price Optimization

The following three steps comprise the optimization process:

1. We feed the given short-term and long-term history information and **derive** $\mathbf{z}_{init}$ from the initial prediction module.
2. Set the target occupancy rates $\mathbf{y}^*$ that we want to achieve and then **solve the series of reverse-mode integral problems** to derive $\mathbf{z}^*_{adjust}$.
3. The **optimal price** vector p^* can be calculated as follows:

$$\boldsymbol{p}^* = \frac{\boldsymbol{z}_{init} - \boldsymbol{z}^*_{adjust} - \boldsymbol{b}}{\boldsymbol{c}}$$



→ The price optimization can be solved in **O(1)**.

Experiments (1/4)

Experimental Environments

- **Historical data from various periods** are derived via feature engineering.
- In the case of short-term history, the occupancy rate in the past K hours is given as a feature.
  - Different K settings lead to different training/testing data as shown below:

		Training Dataset	Test Dataset
San Francisco	$K = 1$	407,008	183,280
	$K = 2$	356,132	160,370
	$K = 3$	305,256	137,460
Seattle	$K = 1$	75,264	31,360
	$K = 2$	65,856	27,440
	$K = 3$	56,448	23,520

Experiments (2/4)

Prediction

- In our experiments with two real-world datasets, our prediction model **outperforms many existing temporal and spatiotemporal models**.

The results of parking occupancy prediction (mean ± std.dev.)

Dataset	Group		Model	$K = 1$ MSE	$K = 1$ R^2	$K = 2$ MSE	$K = 2$ R^2	$K = 3$ MSE	$K = 3$ R^2
San Francisco	**Ours**	Substitute Eq. (2) with	RNN	0.01374 ± 0.00027	0.60727 ± 0.00763	0.01296 ± 0.00223	0.62080 ± 0.00223	0.01373 ± 0.00016	0.59760 ± 0.00457
			LSTM	0.01701 ± 0.00027	0.51369 ± 0.01161	0.01443 ± 0.00321	0.57783 ± 0.00321	0.01517 ± 0.00009	0.55557 ± 0.00261
			GRU	0.01505 ± 0.00039	0.56983 ± 0.01113	0.01395 ± 0.00334	0.59187 ± 0.00334	0.01446 ± 0.00014	0.57618 ± 0.00406
			STGCN	0.01040 ± 0.00050	0.70287 ± 0.01419	0.01027 ± 0.00045	0.69969 ± 0.01319	0.01037 ± 0.00049	0.69619 ± 0.01433
			DCRNN	0.01022 ± 0.00030	0.70796 ± 0.00851	0.01012 ± 0.00041	0.70404 ± 0.01199	0.01020 ± 0.00053	0.70113 ± 0.01534
			AGCRN	0.01021 ± 0.00012	0.70821 ± 0.00329	0.01001 ± 0.00014	0.70727 ± 0.00417	0.01047 ± 0.00032	0.69326 ± 0.00938
		Proposed (full model)		**0.00980 ± 0.00002**	**0.71985 ± 0.00046**	**0.00975 ± 0.00001**	**0.71473 ± 0.00034**	**0.00999 ± 0.00003**	**0.70726 ± 0.00092**
		Proposed (w/o final module)		0.01005 ± 0.00003	0.71278 ± 0.00084	0.00994 ± 0.00004	0.70915 ± 0.00109	0.01009 ± 0.00003	0.70435 ± 0.00090
	Existing Method (only short-term)		RNN	0.01295 ± 0.00004	0.62994 ± 0.00120	0.01258 ± 0.00003	0.63189 ± 0.00102	0.01321 ± 0.01378	0.61225 ± 0.00145
			LSTM	0.01564 ± 0.00020	0.55307 ± 0.00562	0.01395 ± 0.00005	0.59198 ± 0.00137	0.01476 ± 0.01378	0.56762 ± 0.00296
			GRU	0.01374 ± 0.00006	0.60724 ± 0.00174	0.01280 ± 0.00006	0.62547 ± 0.00167	0.01319 ± 0.01378	0.61344 ± 0.00240
			STGCN	0.01119 ± 0.00054	0.68005 ± 0.01546	0.01100 ± 0.00047	0.67825 ± 0.01378	0.01112 ± 0.00077	0.67407 ± 0.02243
			DCRNN	0.01125 ± 0.00004	0.67850 ± 0.00127	0.01077 ± 0.00002	0.68493 ± 0.00046	0.01072 ± 0.00005	0.68578 ± 0.00136
			AGCRN	0.01092 ± 0.00010	0.68796 ± 0.00282	0.01059 ± 0.00012	0.69012 ± 0.00355	0.01095 ± 0.00066	0.67913 ± 0.01948
			NODE	**0.01058 ± 0.00005**	**0.69753 ± 0.00146**	**0.01055 ± 0.00008**	**0.69123 ± 0.00238**	**0.01063 ± 0.00006**	**0.68840 ± 0.00170**

Experiments (3/4)

Prediction

- In our experiments with two real-world datasets, our prediction model **outperforms many existing temporal and spatiotemporal models**.

The results of parking occupancy prediction (mean ± std.dev.)

Dataset	Model			$K = 1$ MSE	$K = 1$ R^2	$K = 2$ MSE	$K = 2$ R^2	$K = 3$ MSE	$K = 3$ R^2
Seattle	**Ours**	Substitute Eq. (2) with	RNN	0.02392 ± 0.00015	0.62621 ± 0.00225	0.02721 ± 0.00036	0.57511 ± 0.00567	0.02799 ± 0.00053	0.56240 ± 0.00828
			LSTM	0.02564 ± 0.00034	0.59918 ± 0.00529	0.02797 ± 0.00032	0.56323 ± 0.00500	0.03128 ± 0.00085	0.51106 ± 0.01323
			GRU	0.02466 ± 0.00010	0.61449 ± 0.00151	0.02660 ± 0.00058	0.58451 ± 0.00902	0.02917 ± 0.00149	0.54399 ± 0.01746
			STGCN	0.02125 ± 0.00029	0.66792 ± 0.00451	0.02154 ± 0.00103	0.66360 ± 0.01611	0.02183 ± 0.00112	0.65870 ± 0.01746
			DCRNN	0.02128 ± 0.00023	0.66740 ± 0.00352	0.02171 ± 0.00056	0.66094 ± 0.00873	0.02203 ± 0.00088	0.65556 ± 0.01376
			AGCRN	0.02165 ± 0.00009	0.66162 ± 0.00140	0.02170 ± 0.00038	0.66137 ± 0.00643	0.02359 ± 0.00038	0.63345 ± 0.00610
		Proposed (full model)		**0.02098 ± 0.00006**	**0.67204 ± 0.00088**	**0.02126 ± 0.00008**	**0.66803 ± 0.00122**	**0.02153 ± 0.0000**	**0.66339 ± 0.00029**
		Proposed (w/o final module)		0.02266 ± 0.00009	0.64580 ± 0.00140	0.02331 ± 0.00017	0.63599 ± 0.00263	0.02386 ± 0.00021	0.62708 ± 0.00327
	Existing Method (only short-term)	RNN		0.02428 ± 0.00002	0.62045 ± 0.00035	0.02674 ± 0.00010	0.58230 ± 0.00162	0.02712 ± 0.00010	0.57598 ± 0.00155
		LSTM		0.02490 ± 0.00008	0.61076 ± 0.00118	0.02529 ± 0.00013	0.60497 ± 0.00202	0.02720 ± 0.00004	0.57485 ± 0.00063
		GRU		0.02427 ± 0.00006	0.62067 ± 0.00088	0.02466 ± 0.00004	0.61491 ± 0.00064	0.02548 ± 0.00010	0.60162 ± 0.00156
		STGCN		0.02148 ± 0.00037	0.66423 ± 0.00574	0.02167 ± 0.00007	0.66158 ± 0.00111	0.02179 ± 0.00008	0.65931 ± 0.00131
		DCRNN		0.02416 ± 0.00022	0.62232 ± 0.00338	0.02323 ± 0.00047	0.63721 ± 0.00735	0.02250 ± 0.00036	0.64833 ± 0.00558
		AGCRN		0.02158 ± 0.00010	0.66263 ± 0.00158	0.02485 ± 0.00217	0.61193 ± 0.03388	**0.02168 ± 0.00021**	**0.66108 ± 0.00321**
		NODE		**0.02138 ± 0.00006**	**0.66596 ± 0.00094**	**0.02163 ± 0.00005**	**0.66211 ± 0.00078**	0.02182 ± 0.00005	0.65894 ± 0.00080

Experiments (4/4)

Optimization

- Our one-shot optimization **finds better solutions in several orders of magnitude faster** in comparison with other prediction-driven optimization paradigms.

* The optimization performance: the ratio of failed test cases where the optimized occupancy rate exceeds the threshold τ

The optimization performance and the average runtime

	Optimization Performance $\tau = 70\%$	$\tau = 75\%$	$\tau = 80\%$	Runtime (seconds)
Observed in Data	0.5475	0.4440	0.3450	N/A
Greedy [1,2,3]	0.4685	0.2045	0.0553	446.6343
Gradient-based [1,2,3]	0.3606	0.1609	0.0488	0.5471
One-shot (Ours)	**0.1938**	**0.0065**	**0.0033**	**0.0000007**

(a) San Francisco

	Optimization Performance $\tau = 70\%$	$\tau = 75\%$	$\tau = 80\%$	Runtime (seconds)
Observed in Data	0.1307	0.1003	0.0756	N/A
Greedy	0.1533	0.0917	0.0495	0.5438
Gradient-based	0.1739	0.1171	0.0724	0.0042
One-shot (Ours)	**0.0997**	**0.0684**	**0.0440**	**0.0000007**

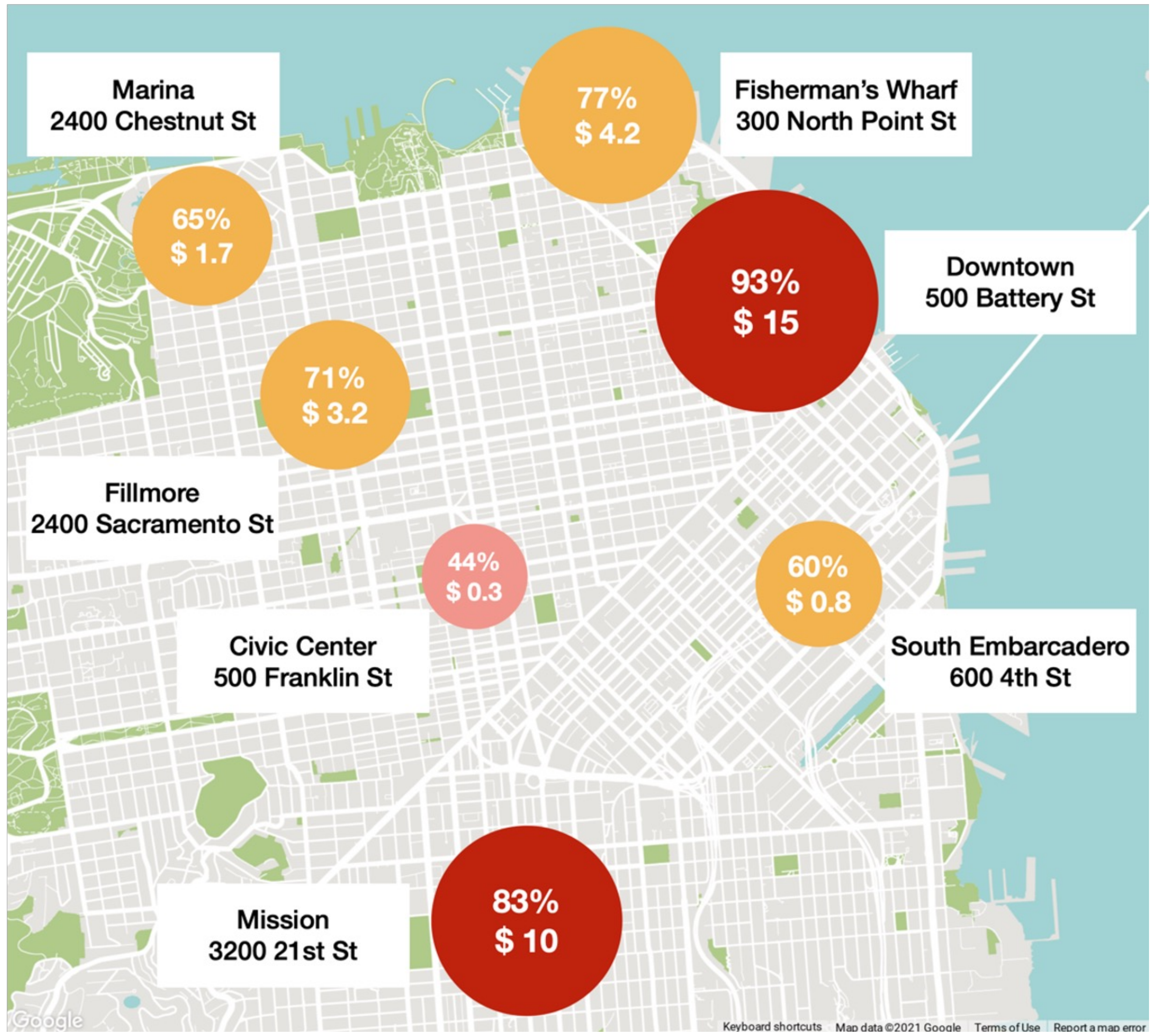
(b) Seattle

[1] An et al. DyETC: Dynamic Electronic Toll Collection for Traffic Congestion Alleviation. AAAI, 2018.
[2] An et al. MAP: Frequency-based maximization of airline profits based on an ensemble forecasting approach. KDD, 2016.
[3] Li et al. Large-Scale Data-Driven Airline Market Influence Maximization. KDD, 2021.

Case Studies (1/2)

San Francisco

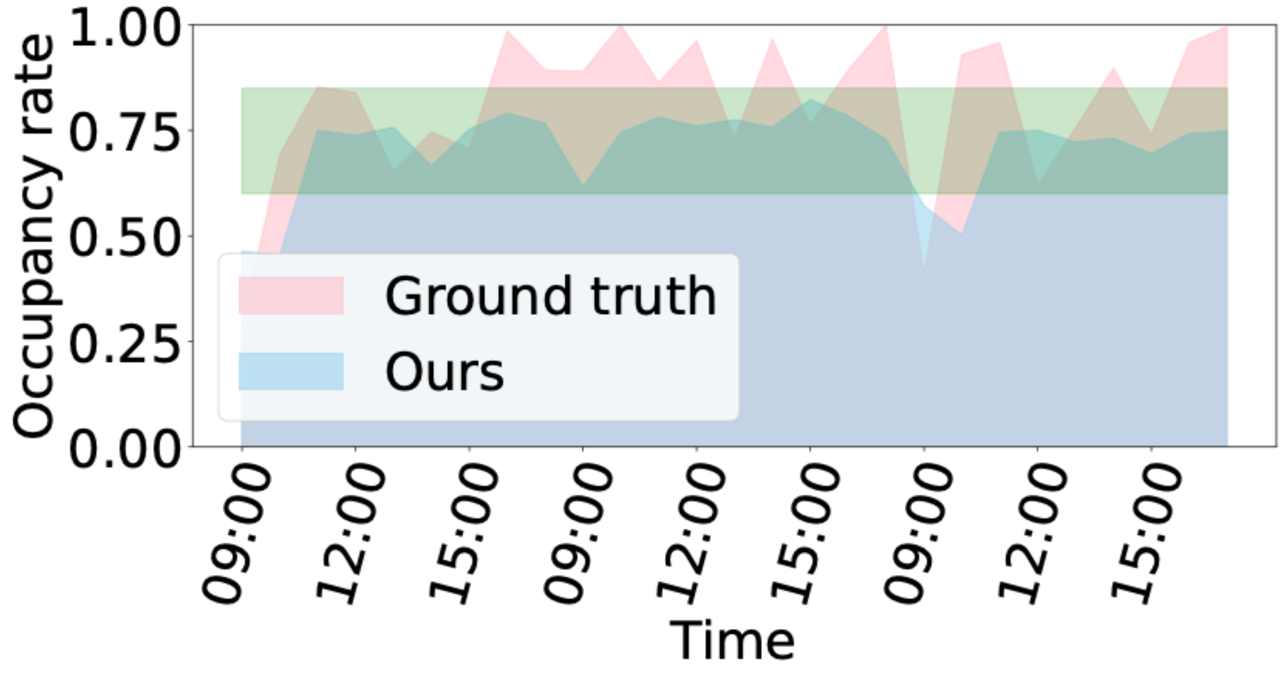
- As shown in the figure, the optimized parking price and the observed parking occupancy rate (the ground truth) are **highly correlated**.
- The correlation between the mean hourly observed occupancy rate and the mean optimized price on each block is 0.724 in our San Francisco's test set.

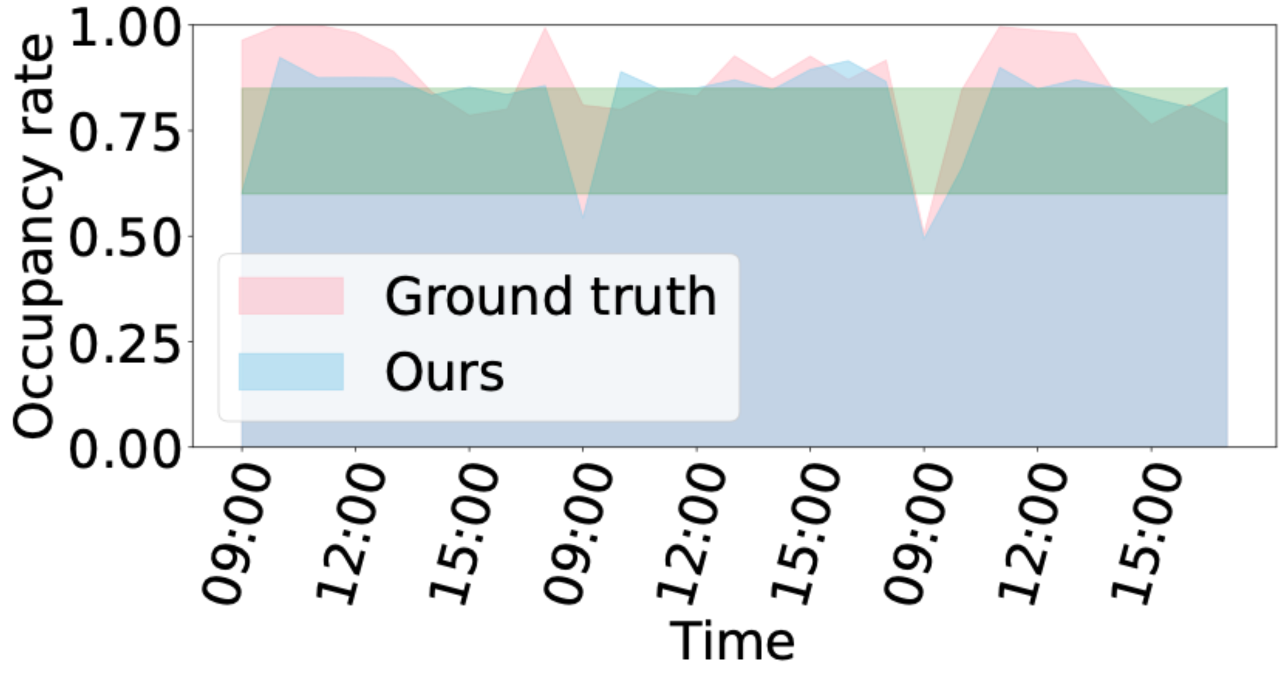


Case Studies (2/2)

Seattle

- In Seattle, the ground-truth occupancy rates are above the ideal range, 60%~ 85%, whereas **our method successfully suppresses the occupancy rates on or below the range**.





Conclusion

- We presented **a one-shot prediction-driven optimization** framework which is featured by i) an effective prediction model for the price-occupancy relation and ii) a one-shot optimization method.
- Our prediction model is **carefully tailored for the price-occupancy relation** and therefore, it outperforms other general spatiotemporal forecasting models.
- Our experiments show that the presented dynamic pricing works well as intended, **outperforming other optimization methods**.

Thank you!