End-to-End Decision-Focused Prediction in Dynamic Bike-Sharing Rebalancing

Pengbo Fu
School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen)
Shenzhen, China
pengbofu1@link.cuhk.edu.cn

Chongyang Wan
School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen)
Shenzhen, China
chongyangwan@link.cuhk.edu.cn

Yuan Luo
School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen)
Shenzhen, China
luoyuan@cuhk.edu.cn

ABSTRACT

Dynamic rebalancing is essential for maintaining service quality in bike-sharing systems (BSSs), where demand fluctuates significantly across space and time. Most existing approaches adopt a predict-then-optimize (PTO) framework, in which forecasting models are trained independently of downstream decision-making. However, this separation often leads to suboptimal operational outcomes: accurate forecasts may still produce ineffective rebalancing decisions. To address this disconnect, we propose PREDICT (Prediction for REbalancing with DIrect Cost Training), an end-to-end decision-focused framework that formulates the rebalancing-aware learning task directly as a bilevel mixed-integer optimization problem. Unlike PTO pipelines, PREDICT aligns the training of the predictive model with the downstream rebalancing objective, explicitly accounting for spatiotemporal uncertainty, mixed-integer action spaces, and real-time operational constraints.Building on this formulation, we further introduce Activated Constraint Embedding (ACE), a scalable algorithm that iteratively activates only the most relevant constraints during training, making the bilevel structure tractable in practice. Experiments on the real-world Hubway dataset demonstrate that PREDICT reduces system loss by up to 56% compared to state-of-the-art PTO baselines. While our study focuses on bike-sharing, the proposed framework applies broadly to forecast-driven decision-making problems in logistics, mobility, and energy systems.

KEYWORDS

Decision-focused learning; Bike-sharing rebalancing; Transportation dispatch

ACM Reference Format:

Pengbo Fu, Chongyang Wan, and Yuan Luo . 2026. End-to-End Decision-Focused Prediction in Dynamic Bike-Sharing Rebalancing. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/ITBP8245

1 INTRODUCTION

Bike-sharing systems (BSSs) have become an integral component of modern urban transportation, offering sustainable and flexible mobility for short-distance travel. However, ensuring consistent system performance requires dynamic rebalancing of bikes across stations to avoid starvation (no bikes available for pickup) and congestion (no docks available for return). These imbalances, driven by weather-sensitive and spatiotemporally varying demand, degrade service quality and lead to unmet user needs. To mitigate this, operators deploy rebalancing vehicles to redistribute bikes throughout the day—yet determining when, where, and how to rebalance remains a challenging optimization problem [12, 13, 15].

A large body of prior work adopts a predict-then-optimize (PTO) framework: a demand forecasting model is trained to minimize statistical error, and the resulting forecasts are passed to an optimization module to compute rebalancing actions [1]. While intuitive, this decoupled approach suffers from a critical limitation: forecast accuracy does not necessarily translate into effective operational decisions [7]. In BSSs, the cost of underestimating demand (e.g., lost ridership) often outweighs that of overestimating (e.g., minor overstocking), yet most forecasting models treat all errors symmetrically [5, 22–24], by minimizing symmetric metrics like Mean Absolute Error or Root Mean Squared Error (MAE) and Root Mean Squared Error (RMSE). Furthermore, the optimization module typically assumes forecasts are noise-free, producing rebalancing plans that are overly sensitive to prediction errors. As a result, even highly accurate forecasting models can yield unstable or operationally inefficient decisions.

This disconnect has motivated interests in decision-focused learning (DFL), a paradigm that trains predictive models to directly improve downstream decision quality rather than statistical fit [6, 20]. DFL has shown promise in domains such as power systems [4, 25, 26], inventory control [11, 17, 19], and logistics [2], where decision costs are asymmetric and constraints are complex. However, applying DFL to BSSs introduces several unique challenges:

- **Multidimensional, mixed-integer decisions:** Bike rebalancing often requires hundreds of binary routing and integer loading variables, resulting in large-scale combinatorial optimization problems.
- **Spatiotemporal and weather-driven demand:** Fine-grained temporal dynamics, local station topology, and external contextual factors (e.g., rainfall, temperature) significantly influence user behavior, increasing the complexity of both prediction and rebalancing.
- **Non-differentiable optimization layer:** The rebalancing problem is typically formulated as a Mixed-Integer Linear Program (MILP), which introduces discontinuities that make gradient-based training unstable or inapplicable.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/ITBP8245

To address these challenges, we propose PREDICT (Prediction for REbalancing with DIrect Cost Training), an end-to-end decision-focused learning framework for dynamic bike-sharing rebalancing. Unlike traditional PTO pipelines, PREDICT trains the predictive model directly with respect to the downstream rebalancing objective, aligning model learning with operational goals. This integration produces forecasts that are not only statistically informed but also robust and cost-effective in practice.

A key innovation in PREDICT is the Activated Constraint Embedding (ACE) algorithm, which makes the resulting bilevel mixed-integer optimization problem tractable. ACE leverages problem structure and prior knowledge to iteratively activate only the most relevant constraints, significantly reducing the computational burden of solving the inner optimization loop. This results in a scalable training procedure that supports large-scale deployment and near-real-time inference[1].

We advance the state of the art in dynamic bike-sharing rebalancing through the following contributions:

- **Forecasting for rebalancing by Decision-Focused Learning:** We propose PREDICT, the first end-to-end learning framework tailored to dynamic bike-sharing rebalancing. PREDICT bridges the gap between forecasting and optimization by directly training models to minimize operational cost, rather than prediction error.
- **Scalable Bilevel Optimization with Activated Constraint Embedding:** We introduce ACE, a novel and highly efficient bilevel optimization technique that dynamically activates constraints during training, thereby enabling tractable optimization over high-dimensional, mixed-integer decision spaces without the need to computing full MILP gradients.
- **Empirical Validation on Real-World Deployment Scenarios:** We evaluate PREDICT on the Hubway data set and demonstrate that our method achieves up to 40% lower rebalancing cost compared to leading PTO baselines, with consistent performance across various demand regimes and planning horizon. Although we use the Hubway dataset, the proposed framework is not tailored to this network. We have tested PREDICT on additional real-world datasets, including Ningbo [23] and NYC CitiBike [5], with consistent performance gains.

2 RELATED WORK

Bike-sharing rebalancing has emerged as a central challenge in urban mobility systems, attracting sustained interest from the optimization and operations research communities. For instance, Ghosh et al. [9] introduced the Dynamic Repositioning and Routing Problem with demand Uncertainty (DRRPU), formulating the task as a two-player game between worst-case demand scenarios and a robust rebalancing policy. This was later extended in DRRPT [10] by incorporating trailers as lightweight redistribution agents and leveraging historical demand data to improve flexibility. Luo et al. [14] proposed a workload-aware mixed-integer nonlinear programming (MINLP) formulation that jointly penalizes both bike and dock shortages, while Chen et al. [3] introduced a target-based stochastic distributionally robust optimization (TSDRO) framework that explicitly incorporates user dissatisfaction, service efficiency, and spatial fairness into the rebalancing objective.

While these approaches have advanced the formulation of rebalancing strategies, they typically rely on exogenously trained demand forecasting models that are decoupled from the optimization process. This separation reflects the widely used predict-then-optimize (PTO) paradigm, in which a forecasting model is trained to minimize prediction error independently of the downstream decision task [8, 12]. Although conceptually simple, PTO pipelines often lead to suboptimal operational outcomes: accurate forecasts can still yield poor rebalancing decisions, especially in the presence of asymmetric penalties, spatiotemporal uncertainty, and complex routing constraints. Several works attempt to mitigate this disconnect using robustness techniques [1,3,4], yet they continue to treat forecasting and optimization as independent stages with no feedback loop between them.

To address the limitations of PTO, decision-focused learning (DFL) has recently gained attention as a more integrated alternative. In DFL, predictive models are trained end-to-end using the quality of downstream decisions as the learning signal, thereby aligning the learning process with the actual operational objective. Notable contributions include task-based learning for stochastic optimization [25, 26], hybrid modeling for public health interventions [21], and theoretical formulations of objective-aligned learning in "Smart Predict-then-Optimize" [6, 16, 26]. However, these methods have largely focused on continuous or low-dimensional decision spaces such as inventory control or resource allocation problems. In contrast, the rebalancing problem in BSSs is complex. It involves high-dimensional, mixed-integer, and temporally coupled decisions, along with non-differentiable optimization layers. These characteristics make existing DFL techniques computationally infeasible or unstable in this context.

3 SYSTEM MODEL

We study the problem of dynamic bike rebalancing in large-scale bike-sharing systems (BSSs), where the primary objective is to reposition bikes across stations over time so as to minimize service loss caused by mismatches between user demand and bike availability. In each planning period, the system operator has a limited number of rebalancing operators, each capable of performing a single pickup and drop-off task. The system must decide how to efficiently allocate these operators to reallocate bikes among stations based on predicted user flows.

Let $\mathcal{S}$ denote the set of bike stations and $\mathcal{V}$ the set of V BBS dispatchers. Each station $s \in \mathcal{S}$ has a dock capacity C_s and a current bike inventory d_s^t at the beginning of time t. Each trailer $v \in \mathcal{V}$ has a maximal load C_v^* and a upper threshold H_{max} for dispatch distance. Let $F_{s,s'}^t$ denote the true user flow from station s to station s' at time t, and $\hat{F}_{s,s'}^t$ its predicted counterpart. The loss of station s at time t is evaluated through bike lost LB_s^t and dock lost LD_s^t. The decision variables at each time t include:

[1] Although this work is grounded in the context of bike-sharing systems, the proposed framework and algorithm are directly applicable to a broader class of problems involving forecast-driven scheduling under uncertainty. Applications include last-mile logistics, emergency vehicle dispatch, and renewable-aware energy scheduling, where multidimensional, mixed-integer decisions are impacted by uncertain, spatiotemporal forecasts. We use bike-sharing as a representative and challenging case study due to its rich spatiotemporal structure, real-time constraints, and public data availability.

- $b^t_{v,s,s'} \in \{0,1\}$: whether the operator v is dispatched from s to s': 1 symbolizes that v performs dispatch task between station s and s' while 0 means no dispatch task from s to s' for v.
- $y^t_{v,s,s'} \in \mathbb{Z}_+$: number of bikes moved from station s to s' by operator v;

Then we can formally formulate the rebalancing problem for a single time step, given a predicted flow field $\hat{\boldsymbol{F}}^t$, as the following mathematical optimization problem:

$$\min_{\boldsymbol{y}^t,\boldsymbol{b}^t,} \quad \sum_{s\in\mathcal{S}} \left(\mathrm{LB}^t_s + \mathrm{LD}^t_s\right) \tag{1}$$

$$\text{s.t.} \quad \text{(Flow conservation)} \quad y^t_{v,s,s'} = b^t_{v,s,s'} \cdot y^t_{v,s,s'}, \quad \forall s,s' \in \mathcal{S}, v \in \mathcal{V} \tag{2}$$

$$\text{(Outflow constraint)} \quad \sum_{v,s'} y^t_{v,s,s'} \le d^t_s, \quad \forall s \in \mathcal{S} \tag{3}$$

$$\text{(Inflow constraint)} \quad \sum_{v,s'} y^t_{v,s',s} \le C_s - d^t_s, \quad \forall s \in \mathcal{S} \tag{4}$$

$$\text{(Trailer capacity constraint)} \quad y^t_{v,s,s'} \le \min(d^t_s, C^*_v), \quad \forall s,s' \in \mathcal{S}, v \in \mathcal{V} \tag{5}$$

$$\text{(Bike shortage loss)} \quad \mathrm{LB}^t_s \ge \sum_{s'} \hat{F}^t_{s,s'} - d^t_s - \sum_{s'} (y^t_{s',s} - y^t_{s,s'}), \quad \forall s \in \mathcal{S} \tag{6}$$

$$\text{(Dock overflow loss)} \quad \mathrm{LD}^t_s \ge d^t_s + \sum_{s'} (y^t_{s',s} - y^t_{s,s'}) + \sum_{s'} \hat{F}^t_{s',s} - C_s, \quad \forall s \in \mathcal{S} \tag{7}$$

$$\text{(Single routing constraint)} \quad b^t_{v,s,s'} \cdot H_{s,s'} \le H_{max} \quad \forall v \in \mathcal{V} \tag{8}$$

$$\text{(Single routing constraint)} \quad \sum_{s,s'} b^t_{v,s,s'} \le 1 \quad \forall v \in \mathcal{V} \tag{9}$$

$$\text{(Non-negativity)} \quad \mathrm{LB}^t_s, \mathrm{LD}^t_s \ge 0, \quad y^t_{v,s,s'} \ge 0 \quad \forall s,s' \in \mathcal{S}, v \in \mathcal{V} \tag{10}$$

$$\text{(Binary routing)} \quad b^t_{v,s,s'} \in \{0,1\} \quad \forall s,s' \in \mathcal{S}, v \in \mathcal{V} \tag{11}$$

In the above optimization problem, Eq. (1) represents the objective to minimize the sum of bike shortage and dock congestion at time t. Eq. (3) and Eq. (4) restrict the outflow and inflow dispatch quantities so that they cannot exceed the current bike inventory and dock storage respectively. Eq. (5) enforces that the dispatch quantity for each trailer cannot exceed both its capacity and the available bikes in the source station. Eq. (6) and Eq. (7) define loss for BSSs as the gap between user demand and station status after rebalancing operations. Eq. (8) further limits the dispatch distance for trailers to be lower than H_{max}, while Eq. (9) ensures that every trailer v performs at most one dispatch task. Finally, Eqs. (10)- (11) collectively define the feasible region of decision variables.

Let $M(\cdot)$ denote equality constraint (2) and $N(\cdot)$ denote inequality constraints, including Eqs. (3)–(11). Then the feasible decision space given a predicted flow field $\hat{\boldsymbol{F}}^t$ is defined as

$$\mathcal{X}_{t|\hat{F}^t} = \left\{ (\boldsymbol{y}^t, \boldsymbol{b}^t) \,\middle|\, \begin{array}{l} M(\boldsymbol{y}^t, \boldsymbol{b}^t) = 0, \\ N(\boldsymbol{y}^t, \boldsymbol{b}^t, \hat{F}^t) \le 0 \end{array} \right\} \tag{12}$$

Thus, the rebalancing problem can be compactly expressed as

$$(\boldsymbol{y}^t, \boldsymbol{b}^t) \in \operatorname*{arg\,min}_{(\boldsymbol{y}^t,\boldsymbol{b}^t)\in\mathcal{X}_{t|\hat{F}^t}} \sum_{s\in\mathcal{S}} \left(\mathrm{LB}^t_s + \mathrm{LD}^t_s\right) \tag{13}$$

Eq. (13) captures the operational constraints and service-level objectives of real-time bike rebalancing. However, traditional pipelines follow a predict-then-optimize (PTO) paradigm, where the prediction model is trained separately from the decision-making process. This can lead to poor downstream performance, as prediction errors are not penalized based on their impact on final decisions.

Decision-focused learning (DFL) offers a promising alternative by training predictive models to directly optimize decision quality. However, applying DFL to bike rebalancing presents unique challenges. The downstream optimization problem involves integer routing variables, non-convex constraints, and loss terms that depend on predicted flows in complex, non-differentiable ways. As a result, treating the rebalancing problem as a black-box loss function leads to computational intractability and unstable training.

To address these challenges, we propose **PREDICT framework**, a structured instantiation of DFL for bike rebalancing. PREDICT formulates the learning process as a bilevel optimization problem. The upper level learns a flow predictor[2] F_θ parameterized by θ, and the lower level solves the rebalancing problem Eqs. (1)–(11) based on $\hat{F}^t = F_\theta(\boldsymbol{f}^t)$.

Formally, for each scenario $k \in \mathcal{K}$ with feature input $\boldsymbol{f}^{k,t}$ and true flow $\boldsymbol{F}^{k,t}$, the PREDICT problem is:

$$\min_{\theta} \quad \sum_{k\in\mathcal{K}} A\left(\boldsymbol{y}^{k,t}, \boldsymbol{b}^{k,t}, \boldsymbol{F}^{k,t}\right) \tag{14}$$

$$\text{s.t.} \quad \hat{\boldsymbol{F}}^{k,t} = F_\theta(\boldsymbol{f}^{k,t}) \tag{15}$$

$$(\boldsymbol{y}^{k,t}, \boldsymbol{b}^{k,t}) \in \operatorname*{arg\,min}_{\substack{(\boldsymbol{y}^{k,t},\boldsymbol{b}^{k,t}) \\ \in\mathcal{X}_{k,t|\hat{F}^t}}} \sum_{s\in\mathcal{S}} \left(\mathrm{LB}^{k,t}_s + \mathrm{LD}^{k,t}_s\right) \tag{16}$$

Here, we define the realized loss function $A(\cdot)$ as:

$$A(\boldsymbol{y}^{k,t}, \boldsymbol{b}^{k,t}, \boldsymbol{F}^{k,t}) = \sum_{s\in\mathcal{S}} \left(\tilde{\mathrm{LB}}^{k,t}_s + \tilde{\mathrm{LD}}^{k,t}_s\right) \tag{17}$$

where $\tilde{\mathrm{LB}}^{k,t}_s, \tilde{\mathrm{LD}}^{k,t}_s$ are evaluated using the true flow $\boldsymbol{F}^{k,t}$ and the lower-level decision.

This closed-loop structure effectively aligns prediction and optimization by using real-world demand and supply losses to guide the training of F_θ, as illustrated in Figure 1. However, solving the bilevel problem Eqs. (14)–(16) is nontrivial due to the discrete structure of the lower-level optimization. As we can see, this bilevel setting introduces two fundamental difficulties:

[2]The ACE framework does not limit the use of expressive, nonlinear spatiotemporal forecasting models. In ACE, we introduce a lightweight linear layer F_θ on top of the forecasting model to ensure linearity of the KKT conditions (*i.e.*, Eqs. (25)-(28)). Specifically, this layer scales and shifts the raw forecast outputs to produce inputs that are compatible with the linearity requirements of the optimization layer. Only F_θ is updated during ACE training, while the nonlinear predictor remains fixed.

- **Discrete Routing**: Routing variables $b^{k,t} \in \{0,1\}$ make the lower-level problem a mixed-integer program (MIP), which is non-differentiable and non-convex.
- **Nested Optimization**: The lower-level problem is embedded as an `argmin` operator rather than as a standard constraint inside the upper-level loss, thereby preventing direct gradient computation.

In the next section, we propose a scalable solution strategy called Activated Constraint Embedding (ACE), which enables efficient bilevel training by selectively encoding the lower-level optimality conditions into the upper-level problem.

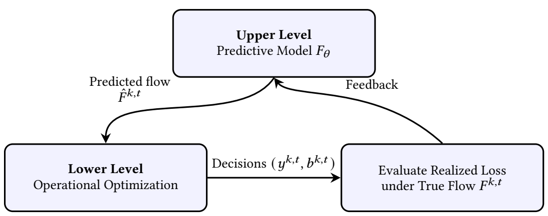

Figure 1: Closed-loop structure of the PREDICT bilevel optimization model.

4 METHODOLOGY

We present a bilevel optimization framework for decision-aware training of predictive models in bike rebalancing systems in Section 3. Our goal is to learn a flow predictor that leads to better operational decisions, by directly minimizing the realized downstream loss under a bilevel optimization structure. The lower-level problem explicitly models the rebalancing operation, which involves binary routing decisions and integer bike dispatch quantities, while the upper level updates the predictor parameters based on the realized loss under true demand. A key challenge arises from the discrete nature of the lower-level decision variables. In particular, the routing variables $b^{k,t}_{s,s'} \in \{0,1\}$ indicate whether a trailer is dispatched from station s to station s' at time t. Combined with additional constraints such as flow conservation and capacity limits, the lower-level problem becomes an integer program (IP), which is inherently non-differentiable. This makes the overall bilevel formulation intractable for standard gradient-based learning.

To solve the bilevel optimization problem, we propose the Activated Constraint Embedding (ACE) framework. ACE transforms the intractable bilevel structure into a tractable and learnable form by decomposing the problem into a master and subproblem (MP–SP) coordination framework.

The key idea is to preserve the discrete routing structure, while relaxing the integer bike dispatch quantities $y^{k,t}_{s,s'} \in \mathbb{Z}_+$ to continuous values $y^{k,t}_{s,s'} \in \mathbb{R}_+$. This relaxation is justified by the fact that the feasible region defined by y is a polyhedron with integer-valued data and constraints; thus, its MILP relaxation from IP still yields integral solutions in practice. More importantly, this relaxation enables us to derive and embed the KKT conditions associated with the continuous part of the lower-level problem, making the overall learning process gradient-compatible.

In more detail, ACE decomposes the learning process into two components: a subproblem (SP) that generates high-quality integer routing patterns under the current prediction model, and a master problem (MP) that embeds the optimality structure of each generated integer pattern and updates the predictor accordingly. This decomposition is structurally similar to column-and-constraint generation (CCG) [18], but differs fundamentally in three aspects: (i) we do not require full enumeration of integer patterns; (ii) we embed KKT conditions for the continuous part of the lower-level problem; and (iii) we use activation variables to bind the master solution to the optimality cut with the least KKT objective value in the present routing pattern pool from the subproblem. Table 1 lists the main symbols and variables used in our formulation to facilitate clarity in the subsequent modeling.

Table 1: Notation table for the ACE framework

Symbol	Description
SP and Routing Pool	
$(\boldsymbol{b}^{e,k,t}, \boldsymbol{y}^{e,k,t})$	Feasible routing pattern e and its dispatch solution
$\mathcal{E}$	Routing pattern pool (set of all enumerated patterns)
$e \in \mathcal{E}$	Index of routing pattern (iteration index in ACE)
λ	Regularization parameter for trailer usage in SP
Master Problem Variables	
$\text{act}_e \in \{0,1\}$	Binary variable: activation for routing pattern e
$b^{k,t}_{dv}$	Duplicated routing variable (convex combo of $b^{e,k,t}$)
$y^{k,t}_{dv}$	Duplicated dispatch variable (convex combo of $y^{e,k,t}$)
$y^{e,k,t}_{gv}$	Continuous variable optimized under fixed $b^{e,k,t}$
KKT Variables and Conditions	
$\mu^{e,k}$	Dual variables for equality constraints (Lagrange multipliers)
$\nu^{e,k}$	Dual variables for inequality constraints
$\mathcal{L}(y,\mu,\nu)$	Lagrangian of the lower-level LP under fixed b^e
$\nabla_y \mathcal{L} = 0$	Stationarity condition
$\nu^{e,k} \geq 0$	Dual feasibility
$\nu^{e,k} \cdot N(\cdot) = 0$	Complementary slackness condition

4.1 Subproblem: Instantiating the Lower-Level Model

Recall that in Section 3, the lower-level problem is a mixed-integer optimization that minimizes the total expected service loss under predicted flows $\hat{F}^{k,t} = F_\theta(\boldsymbol{f}^{k,t})$, with equality constraints $M(\cdot)$ and inequality constraints $N(\cdot)$ capturing the rebalancing system structure. These include flow conservation, routing limits, station capacity, and bike stock constraints.

To make the bilevel problem tractable, we instantiate the lower-level problem under fixed predicted flow for each training scenario $k \in \mathcal{K}$, as the following **subproblem (SP)**:

$$\min_{\boldsymbol{y}^{k,t},\boldsymbol{b}^{k,t}} \quad \sum_{s \in \mathcal{S}} \left(\mathrm{LB}_s^{k,t} + \mathrm{LD}_s^{k,t} \right) + \lambda \sum_{s,s'} b_{s,s'}^{k,t} \tag{18}$$

$$\text{s.t.} \quad M(\boldsymbol{y}^{k,t}, \boldsymbol{b}^{k,t}) = 0 \tag{19}$$

$$N(\boldsymbol{y}^{k,t}, \boldsymbol{b}^{k,t}, \hat{F}^{k,t}) \leq 0 \tag{20}$$

$$\hat{F}^{k,t} = F_\theta(\boldsymbol{f}^{k,t}) \tag{21}$$

The objective of Eq. (18) incorporates an additional regularization term $\lambda \sum_{s,s'} b_{s,s'}^{k,t}$, where $b_{s,s'}^{k,t}$ represents the activation status of routing from station s to s' in scenario k. This regularization term is introduced to address the issue that the original decision model may admit multiple optimal solutions. By imposing a small positive penalty $\lambda > 0$ on redundant trailer activations, the subproblem ensures improved consistency and practical uniqueness between the subproblem and the master problem, thereby favoring rebalancing configurations that achieve the minimum total loss while activating the fewest trailers.

The constraints $M(\cdot) = 0$ and $N(\cdot) \leq 0$ under predicted flows $\hat{F}^{k,t}$ correspond to the equality and inequality constraints of the compact rebalancing formulation in Eq. (13).

This subproblem closely mirrors the lower-level formulation of the original bilevel model and preserves its key structural components. Solving it yields a feasible decision tuple $(\boldsymbol{b}^{e,k,t}, \boldsymbol{y}^{e,k,t})$, indexed by a routing pattern e, which represents a concrete rebalancing configuration under the predicted flow. The resulting binary routing plan $\boldsymbol{b}^{e,k,t}$ is subsequently added to a candidate pattern pool $\mathcal{E}$, thereby progressively enriching the set of routing configurations available to the master problem defined in Section 4.3, which selectively embeds Karush–Kuhn–Tucker conditions only for these generated patterns. Since the dispatch variables $\boldsymbol{y}$ depend explicitly on the master-level parameters, they are not stored in the pool but are instead regenerated dynamically via the corresponding Karush–Kuhn–Tucker system.

4.2 LP Relaxation and KKT Embedding

As mentioned above, directly embedding the subproblem (SP) into the upper-level optimization using an `argmin` operator is infeasible. To address this issue, for each routing pattern e, we fix the binary routing plan $\boldsymbol{b}^{e,k,t}$ obtained from the SP and relax the dispatch variables $\boldsymbol{y}^{e,k,t}$ to be continuous. This transformation converts the original mixed-integer subproblem into a linear program in $\boldsymbol{y}^{e,k,t}$, which can be seamlessly embedded into the upper-level optimization via KKT conditions.

Although the dispatch variables are relaxed to continuous values, we empirically observe that the resulting LP solutions remain integral in practice. This empirical integrality arises from two important structural properties of the model: (i) key constraints, such as inventory balance and vehicle capacity constraints, have integer-valued right-hand sides (for example, bike counts and trailer capacities), and (ii) several constraint matrices exhibit total unimodularity. This behavior is consistently observed across all tested scenarios and is confirmed by solver diagnostics.

$$\min_{\boldsymbol{y}^{e,k,t}} \quad \sum_{s \in \mathcal{S}} \left(\mathrm{LB}_s^{k,t} + \mathrm{LD}_s^{k,t} \right) + \lambda \sum_{s,s'} b_{s,s'}^{e,k,t} \tag{22}$$

$$\text{s.t.} \quad M(\boldsymbol{y}^{e,k,t}, \boldsymbol{b}^{e,k,t}) = 0 \tag{23}$$

$$N(\boldsymbol{y}^{e,k,t}, \boldsymbol{b}^{e,k,t}, \hat{F}^{k,t}) \leq 0 \tag{24}$$

This LP preserves the structure of the original rebalancing problem while enabling tractable embedding. The solution $\boldsymbol{y}^{e,k,t}$ associated with fixed $\boldsymbol{b}^{e,k,t}$ now forms an optimality cut.

To retain the optimality structure of the relaxed lower-level problem, we extract the KKT conditions for the LP defined by Eqs. (22)–(24). For each pattern $e \in \mathcal{E}$, these are:

- **Stationarity**:

$$\nabla_y \mathcal{L}(\boldsymbol{y}^{e,k,t}, \mu^{e,k}, \nu^{e,k}) = 0 \tag{25}$$

- **Primal feasibility**:

$$M(\boldsymbol{y}^{e,k,t}, \boldsymbol{b}^{e,k,t}) = 0, \quad N(\boldsymbol{y}^{e,k,t}, \boldsymbol{b}^{e,k,t}, \hat{F}^{k,t}) \leq 0 \tag{26}$$

- **Dual feasibility**:

$$\nu^{e,k} \geq 0 \tag{27}$$

- **Complementary slackness**:

$$\nu^{e,k} \cdot N(\boldsymbol{y}^{e,k,t}, \boldsymbol{b}^{e,k,t}, \hat{F}^{k,t}) = 0 \tag{28}$$

To linearize the complementarity slackness condition Eq. (28) for each pattern e, we apply the standard big-M formulation:

$$\nu^{e,k} \cdot N(\cdot) = 0 \quad \Rightarrow \quad \begin{cases} -N(\cdot) \leq M(1 - z^{e,k}), \\ \nu^{e,k} \leq M \cdot z^{e,k}, \\ z^{e,k} \in \{0,1\} \end{cases} \tag{29}$$

Here, $z^{e,k}$ is a binary indicator of whether $N(\boldsymbol{y}^{e,k,t}, \boldsymbol{b}^{e,k,t}, \hat{F}^{k,t})$ is zero, and M is a sufficiently large constant.

4.3 Master Problem: Aggregating Candidate Solutions

Given the set of candidate solutions $\mathcal{E}$ and their associated KKT conditions, the master problem (MP) strategically selects only one routing plan by introducing binary activation variables $\mathrm{act}_e \in \{0,1\}$ subject to a key constraint that ensures consistent activation across the solution space:

$$\sum_{e \in \mathcal{E}} \mathrm{act}_e = 1 \tag{30}$$

Accordingly, the final routing and dispatch decisions are defined as convex combinations:

$$\boldsymbol{b}_{dv}^{k,t} = \sum_{e \in \mathcal{E}} \mathrm{act}_e \cdot \boldsymbol{b}^{e,k,t} \tag{31}$$

$$\boldsymbol{y}_{dv}^{k,t} = \sum_{e \in \mathcal{E}} \mathrm{act}_e \cdot \boldsymbol{y}^{e,k,t} \tag{32}$$

Thus, act_e is a binary indicator of whether MP determines the e-th routing pattern as the integer combination of duplicated variables. Eqs. (31) and (32) are also linearized by big-M method as (29).

Then we can write the master problem (MP) as the following problem which minimizes the realized loss under true flows:

$$\min_{\theta,\text{act}_e} \quad \sum_{k\in\mathcal{K}} A(\boldsymbol{y}^{k,t}_{dv}, \boldsymbol{b}^{k,t}_{dv}, \hat{F}^{k,t}) \tag{33}$$

$$\text{s.t.} \quad (30) - (32) \text{ hold for } \forall k \in \mathcal{K} \tag{34}$$

$$\hat{F}^{k,t} = F_\theta(\boldsymbol{f}^{k,t}), \quad \forall k \in \mathcal{K} \tag{35}$$

$$M(\boldsymbol{y}^{k,t}_{dv}, \boldsymbol{b}^{k,t}_{dv}) = 0, \quad \forall k \in \mathcal{K} \tag{36}$$

$$N(\boldsymbol{y}^{k,t}_{dv}, \boldsymbol{b}^{k,t}_{dv}, \hat{F}^{k,t}) \le 0, \quad \forall k \in \mathcal{K} \tag{37}$$

$$\text{KKT conditions defined in (25) - (29) hold for all } \forall k \in \mathcal{K}, e \in \mathcal{E} \tag{38}$$

Objective cuts:

$$\sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}}\left(\text{LB}^{k,t}_{dv,s} + \text{LD}^{k,t}_{dv,s}\right) + \lambda \sum_{k\in\mathcal{K}}\sum_{s,s'} b^{k,t}_{dv,s,s'} \le \sum_{k\in\mathcal{K}}\sum_{s\in\mathcal{S}}\left(\text{LB}^{e,k,t}_{gv,s} + \text{LD}^{e,k,t}_{gv,s}\right) + \lambda \sum_{k\in\mathcal{K}}\sum_{s,s'} b^{e,k,t}_{s,s'}, \quad \forall k \in \mathcal{K}, e \in \mathcal{E} \tag{39}$$

Here, F_θ is a parametric predictor with weights W_F bounded by $W_{\min} \le W_F \le W_{\max}$, and $A(\cdot)$ denotes the realized loss computed under actual bike flows $F_{k,t}$. Eqs. (30)-(32) and Eq. (39) ensure that the routing pattern $\boldsymbol{b}^e$ with the least KKT objective value is activated and restricted equal to duplicated routing pattern $\boldsymbol{b}^{dv}$.

This master problem can be interpreted as a tractable surrogate to the full bilevel program in Section 3, where the lower-level problem is approximated by a finite set of LP-optimal solutions, and optimality is enforced via KKT constraints.

4.4 Training Procedure

The overall ACE procedure alternates between generating new routing patterns via SP and updating the predictor via MP. In each iteration, SP produces an integer combination, and it is added to the present routing pattern pool if it is different from every item in the pool. Since we restrict the range of weights W_F, the decision solutions under the predictor will be restricted in a certain range, preventing the pool from unlimited expansion. This process is summarized in Algorithm 1.

4.5 Method Summary

Note that ACE provides a tractable approximation to the original bilevel optimization problem by decomposing it into a sequence of subproblems (SP) and a master problem (MP), coordinated through KKT-based constraint embedding. Specifically, the lower-level MIP is instantiated as a scenario-specific SP, whose relaxed LP solution is used to extract KKT conditions. These conditions are embedded into the MP, which selects the optimality with the least KKT objective value among a pool of candidate routing plans using activation variables and updates the predictor parameters θ accordingly.

This structured decomposition not only facilitates tractable computation but also lays the foundation for an iterative optimization

Algorithm 1 Activated Constraint Embedding (ACE)

Initialize predictor F_{θ_0}, routing pool $\mathcal{E} \leftarrow \emptyset$
Set bounds: $LB \leftarrow +\infty$, $UB \leftarrow 0$
for $e = 1$ to $E_{\max}$ **do**
  for each scenario $k \in \mathcal{K}$ **do**
    $\hat{F}^{k,t} \leftarrow F_\theta(\boldsymbol{f}^{k,t})$
    Solve SP (21) $\Rightarrow (\boldsymbol{b}^{e,k,t}, \boldsymbol{y}^{e,k,t})$
  end for
  $UB \leftarrow$ min(SP objective value, UB)
  if $\boldsymbol{b}^{e,k,t}$ already in $\mathcal{E}$ **then**
    Terminate: no new patterns
  else if $(UB - LB)/UB < \epsilon$ **then**
    Terminate: converged
  else
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{\boldsymbol{b}^e\}$
  end if
  Embed KKT constraints of (38) $\forall e \in \mathcal{E}$
  Solve MP (33)–(39), update θ_e
  $LB \leftarrow$ MP objective value
end for
return final predictor F_{θ^*}

process. Through this process, ACE achieves three goals simultaneously: (i) it preserves the structural logic of the original bilevel problem, (ii) it enables optimization over discrete routing plans without full enumeration, and (iii) it allows end-to-end training of the predictor via a differentiable surrogate. The result is a scalable and interpretable decision-focused learning method that maintains fidelity to the original system constraints and objectives.

5 EXPERIMENTAL EVALUATION

We conduct an empirical study to evaluate whether integrating prediction and optimization through decision-focused learning leads to improved rebalancing outcomes in bike-sharing systems. This directly addresses the central limitation of predict-then-optimize (PTO) pipelines discussed in Section 1, where accurate forecasts may still result in suboptimal decisions. We compare ACE against several baselines that decouple learning and optimization, in order to assess the benefits of aligning model training with downstream operational objectives.

5.1 Real-world Dataset

We evaluate the performance of our approach with respect to the key performance metric of total loss on a real-world Hubway dataset, which comprehensively records trip-level activity logs from the Boston bike-sharing system[3].

The dataset contains the following information: (i) customer trip records, from which we compute the detailed demand scenarios; (ii) the number of stations, their capacity and initial distribution of bikes at each of the stations; (iii) geographical locations of stations

[3]Although we use the Hubway dataset, which is used by Ghosh and Varakantham [8] for in-depth evaluation, the proposed framework is not tailored to this network. We have tested PREDICT on additional real-world datasets, including Ningbo [23] and NYC CitiBike [5], with consistent performance gains. These results are omitted for brevity but confirm the generalization capability of our method across varied station layouts, densities, and demand profiles.

(consisted of longitude and latitude), from which we calculate the relative distance between any two stations; and (iv) the number of operators and their respective capacity. The dataset consists of 95 stations and 20 operators. We let each operator have a maximal capacity of 5 bikes, a maximum travel distance to pick-up station of 50 km and a maximum travel distance from pick-up station to drop-off station of 50 km[4]. As is common in the literature (*e.g.*, [9, 10]), we consider 6 hours of planning horizon in the morning peak (6 AM-12 PM) which is divided into 12 planning periods, each having a duration of 30 minutes. We generate 60 demand scenarios for the weekdays from historical trip data. From 60 demand scenarios, 40 scenarios are used for training purposes and the other 20 scenarios are used for testing.

5.2 Experiment Setup

We compare the proposed ACE framework to several representative approaches that reflect common modeling paradigms in bike-sharing optimization[5]. These include both traditional PTO pipelines and model-free robust optimization strategies[6]:

- **Static Repositioning**: We simulate the BSS without any system rebalancing. This approach serves as a baseline and shows the amount of loss that would occur if no dynamic repositioning is applied.
- **Poisson + MINLP**: A traditional two-stage Prediction then Optimization framework that uses Poisson-based demand prediction followed by the repositioning method MINLP proposed by Luo [14]. This serves as a simple, interpretable baseline that assumes demand independence and station-level marginal rates.
- **TWIST + MINLP:** A stronger PTO approach that leverages TWIST [22], a spatiotemporal attention-based deep forecasting model. TWIST represents the current state of the art in bike-sharing demand prediction, and is included to evaluate whether improved accuracy alone can lead to better decisions. However, TWIST does not provide the repositioning policy. Hence, we combine TWIST with MINLP for finding the repositioning policy.
- **Robust Optimization:** A scenario-based method [9] that minimizes expected or worst-case loss over sampled demand realizations for robustness. This method does not rely on learned forecasts and has been widely adopted in operations literature. It serves as a model-free benchmark that directly optimizes over uncertainty.

[4]Distances are chosen based on the maximal speed of an electric vehicle and the length of a planning period. The results for other distances and other capacities of an operator are broadly similar.
[5]We do not include decision-focused learning methods such as [25] in our benchmarks. As per Section 2, existing DFL approaches typically assume that the downstream optimization problem is convex, continuous, and differentiable. However, the bike-sharing rebalancing problem involves high-dimensional, mixed-integer, and non-convex constraints, which violate these assumptions. Hence, existing DFL methods are not directly applicable, and, to the best of our knowledge, have not been successfully applied to this problem.
[6]We acknowledge that all PTO baselines use the same MINLP solver for rebalancing. This choice ensures consistency and isolates the effect of the predictive model. While more advanced solvers (e.g., commercial heuristics or neural-based optimizers) may improve results marginally, our focus is on evaluating the integration of prediction and optimization.

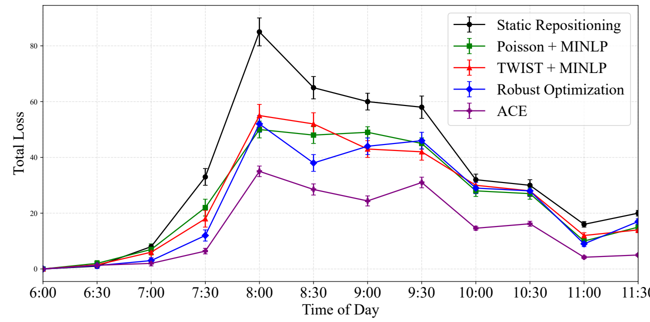

Figure 2: Total loss vs. different hours in a day

As it is common in the litertaure (*e.g.*, [4, 14]), we compare the performance of the appoaches in the following aspects[7]:

- **Loss-minimizing performance**: We compare the total loss that occurred in the experimental scenarios after applying different approaches. A lower loss means a better loss-minimizing performance, which is the ultimate goal of performing bike rebalancing.
- **Runtime performance**: We compare the running time of different rebalancing approaches to solve the same experiment problems. We prefer an algorithm that is more efficient and can solve a problem in a shorter time.

5.3 Empirical Results

Loss-minimizing performance: Figure 2 shows the average total losses at all stations over different testing scenarios at different times in a day. Error bars in all plots represent standard error over 20 test scenarios. While not full confidence intervals, they provide a visual indication of variability across different testing scenarios.

As we can see, the total loss achieved by ACE is consistently lower than that of all other approaches throughout the planning horizon. Compared to Static Repositioning, which performs no bike redistribution, ACE reduces the total loss by up to 56% at 8:00, when system imbalance is most severe.

Compared to Poisson + MINLP and TWIST+MINLP, ACE consistently achieves a loss reduction of at least 34% across all peak periods. This demonstrates that improved forecast accuracy alone is insufficient for effective rebalancing unless it is carefully aligned with the downstream decision objective.

Compared to Robust Optimization, which optimizes over sampled demand scenarios without relying on learned forecasts, ACE achieves up to 44% lower overall loss. This is primarily because ACE effectively incorporates predictive structure while directly optimizing for operational performance objective.

Runtime performance: Having shown the effectiveness of ACE in reducing system loss, we now examine its runtime performance on real-world demand scenarios. To ensure a fair comparison, we only report decision computation times during the planning phase for methods that generate explicit rebalancing actions in each round.

[7]All optimization problems involved are solved using IBM ILOG CPLEX Optimization Studio V12.10 within python code on a NVIDIA RTX 3090 GPU.

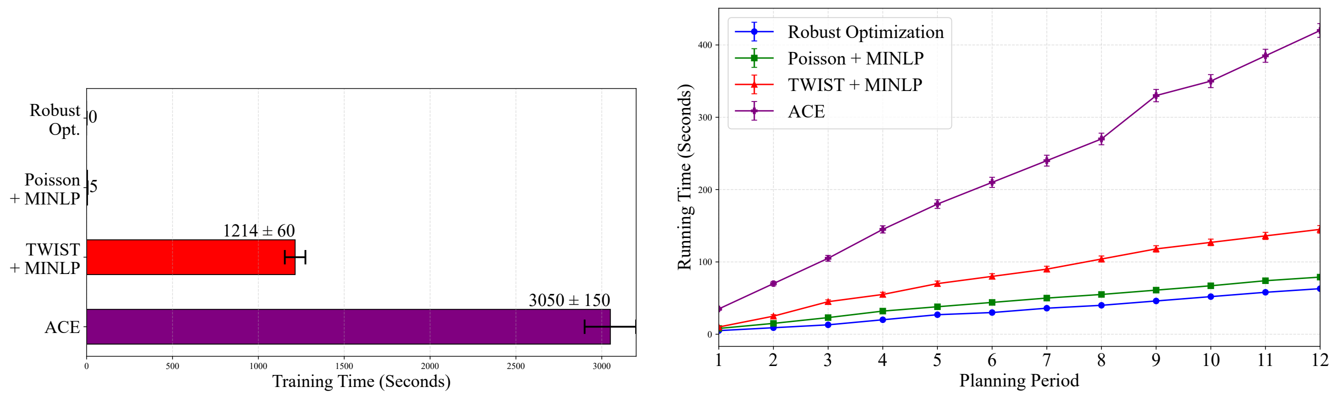

Figure 3: (a) Training time and (b) (Cumulative) Runtime time vs. different periods. The error bars represent minor variations across repeated runs.

As shown in Figure 3a, ACE generally requires more training time compared to simpler baselines. According to the design of the ACE process, the theoretical worst-case computational complexity of a single ACE iteration is $O(2^{KSV} \cdot poly(KSV))$, where K, V, and S denote the number of demand scenarios, the number of operators, and the total number of stations, respectively. However, in practical real-world systems, the value of V is relatively small (typically ranging from 10 to 20), and each operator interacts with only a limited subset of stations due to geographic and operational constraints. This effectively keeps the number of binary variables at a moderate level. In real-world experiments, each iteration takes under 30 seconds, even for the largest test cases, and no exponential behavior is observed empirically. Furthermore, the training process is performed offline only once and does not affect runtime performance during deployment. The trained model can be efficiently reused across multiple days and different planning settings, rendering this computational cost essentially negligible in operational contexts.

Figure 3b illustrates the cumulative decision runtime as the number of planning periods increases. The error bars in Figure 3b are small, indicating stable and consistent convergence across all experimental runs. As expected, all methods exhibit approximately linear growth in computation time as the planning horizon extends, without any exponential trend. Among them, ACE incurs the highest computation time, primarily due to the complexity of solving a bilevel optimization problem with an embedded normalization term. In particular, ACE requires up to 2.8× the runtime of the Robust Optimization baseline when solving for 12 planning periods. Nevertheless, this increase in runtime is accompanied by a substantial improvement in rebalancing performance and overall operational efficiency. Moreover, all ACE variants remain well within a practical runtime budget, with per-round decision time consistently under 35 seconds, which satisfies the operational requirements of most bike-sharing systems, where planning intervals typically range from 15 to 30 minutes. Additionally, our method naturally supports fully asynchronous planning and rolling-horizon deployment: at each decision step, ACE observes the current system state, solves an optimization problem over the next few planning periods, executes only the first step of the resulting solution, and then shifts the planning window forward to re-optimize. This iterative procedure enables parallelization and continuous, real-time adaptation in live, dynamic systems.

6 CONCLUSIONS

This work introduces a bilevel decision-focused learning framework for bike-sharing scheduling, combining predictive modeling with optimization-aware training. We propose the Prediction for REbalancing with DIrect Cost Training (PREDICT) framework, which integrates a flow predictor with downstream rebalancing decisions, and develop the Activated Constraint Embedding (ACE) method to enable tractable and structure-preserving training. By decomposing the bilevel problem into a subproblem and a master problem with embedded KKT conditions, ACE avoids full feasible region while maintaining fidelity to the original system structure. Empirical results show that our method significantly improves economic efficiency and reduces scheduling loss compared to PTO-based and robust optimization baselines, without incurring substantial computational overhead.

Despite these promising results, the current formulation makes structural assumptions that worth further exploration. The embedded predictor must remain linear or piecewise-linear to ensure compatibility with standard optimization solvers, limiting the expressiveness of learned models. Future work could integrate reinforcement learning to treat the PREDICT framework as a black-box environment, enabling end-to-end training of expressive models using true scheduling loss as feedback. Extensions to decentralized, multi-agent, or adaptive systems represent meaningful directions for further research, each constituting substantial standalone contributions beyond the scope of this work.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Project 62472367), Shenzhen Science and Technology Program (Project JCYJ20230807114300001, JCYJ20220818103006012), Shenzhen Stability Science Program 2023.

REFERENCES

[1] Maryam Akbari-Moghaddam, Stephen Kelly, and Douglas Down. 2024. Demand Forecasting and Rebalancing in Shared Bike Systems Using Deep Learning and Evolutionary Computation. In *Proceedings of the Twenty-Seventh IEEE International Conference on Intelligent Transportation Systems*. 3333–3338.

[2] Sergey Evgenievich Barykin, Elena Aleksandrovna Smirnova, Dan Chzhao, Irina Vasilievna Kapustina, Sergey Mikhailovich Sergeev, Yuri Yurievich Mikhalchevsky, Alexander Viktorovich Gubenko, Gennady Aleksanrovich Kostin, Elena De La Poza Plaza, Lilya Saychenko, et al. 2021. Digital Echelons and Interfaces within Value Chains: End-to-End Marketing and Logistics Integration. *Sustainability* 13, 24 (2021), 13929.

[3] Qingxin Chen, Chenyi Fu, Ning Zhu, Shoufeng Ma, and Qiao-Chu He. 2023. A Target-Based Optimization Model for Bike-Sharing Systems: From the Perspective of Service Efficiency and Equity. *Transportation Research Part B: Methodological* 167 (2023), 235–260.

[4] Xianbang Chen, Yikui Liu, and Lei Wu. 2024. Towards Improving Unit Commitment Economics: An Add-On Tailor for Renewable Energy and Reserve Predictions. *IEEE Transactions on Sustainable Energy* 15, 4 (2024), 2547–2566. https://doi.org/10.1109/TSTE.2024.3426337

[5] Pan Deng, Yu Zhao, Junting Liu, Xiaofeng Jia, and Mulan Wang. 2023. Spatio-Temporal Neural Structural Causal Models for Bike Flow Prediction. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, Vol. 37. 4242–4249.

[6] Adam N. Elmachtoub and Paul Grigas. 2022. Smart "Predict, Then Optimize". *Management Science* 68, 1 (2022), 9–26.

[7] Daniele Gammelli, Yihua Wang, Dennis Prak, Filipe Rodrigues, Stefan Minner, and Francisco Camara Pereira. 2022. Predictive and Prescriptive Performance of Bike-Sharing Demand Forecasts for Inventory Management. *Transportation Research Part C: Emerging Technologies* 138 (2022), 103571.

[8] Supriyo Ghosh, Jing Yu Koh, and Patrick Jaillet. 2019. Improving customer satisfaction in bike sharing systems through dynamic repositioning. International Joint Conferences on Artificial Intelligence Organization.

[9] Supriyo Ghosh, Michael Trick, and Pradeep Varakantham. 2016. Robust Repositioning to Counter Unpredictable Demand in Bike-Sharing Systems. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 3096–3102.

[10] Supriyo Ghosh and Pradeep Varakantham. 2017. Incentivizing the Use of Bike Trailers for Dynamic Repositioning in Bike-Sharing Systems. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 27. 373–381.

[11] Md Shafikul Islam and Azmine Toushik Wasi. 2023. Optimizing Inventory Routing: A Decision-Focused Learning Approach Using Neural Networks. arXiv:2311.00983 [cs.LG]

[12] Jingjing Li, Qiang Wang, Wenqi Zhang, Donghai Shi, and Zhiwei Qin. 2021. Dynamic Rebalancing Dockless Bike-Sharing Systems Based on Station Community Discovery. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. 4136–4143.

[13] Jiaqi Liang, Defeng Liu, Sanjay Dominik Jena, Andrea Lodi, and Thibaut Vidal. 2025. Single- vs. Dual-Policy Reinforcement Learning for Dynamic Bike Rebalancing. arXiv:2402.03589 [cs.LG]

[14] Yuan Luo. 2023. Dynamic Workload-Aware Bike Rebalancing for Bike-Sharing Systems. In *Proceedings of the Twenty-Sixth European Conference on Artificial Intelligence*. IOS Press, 1641–1647.

[15] Ji Ma, Shenggen Zheng, Shangjing Lin, and Yonghong Cheng. 2024. A City Shared Bike Dispatch Approach Based on Temporal Graph Convolutional Network and Genetic Algorithm. *Biomimetics* 9, 6 (2024), 368.

[16] Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. 2024. Decision-Focused Learning: Foundations, State of the Art, Benchmark, and Future Opportunities. *Journal of Artificial Intelligence Research* 80 (2024), 1623–1701.

[17] Wuyang Mao, Chuanren Liu, Yundu Huang, Zhonglin Zu, M. Harshvardhan, Liang Wang, and Bo Zheng. 2023. End-to-End Inventory Prediction and Contract Allocation for Guaranteed Delivery Advertising. In *Proceedings of the Twenty-Ninth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1677–1686.

[18] Mouad Morabit, Guy Desaulniers, and Andrea Lodi. 2021. Machine-Learning–Based Column Selection for Column Generation. *Transportation Science* 55, 4 (2021), 815–831.

[19] Meng Qi, Yuanyuan Shi, Yongzhi Qi, Chenxin Ma, Rong Yuan, Di Wu, and Zuo-Jun Shen. 2023. A Practical End-to-End Inventory Management Model with Deep Learning. *Management Science* 69, 2 (2023), 759–773.

[20] Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. 2025. A Survey of Contextual Optimization Methods for Decision-Making Under Uncertainty. *European Journal of Operational Research* 320, 2 (2025), 271–289.

[21] Sanket Shah, Arun Suggala, Milind Tambe, and Aparna Taneja. 2024. Efficient Public Health Intervention Planning Using Decomposition-Based Decision-Focused Learning. arXiv:2403.05683 [cs.AI]

[22] Peng Wang, Longxi Feng, Wenhao Zhang, and Kanghua Hui. 2025. TWIST: An Efficient Spatial–Temporal Transformer with Temporal Window and Sparse Attention for Traffic Forecasting. *IEEE Internet of Things Journal* 12, 14 (2025), 26799–26815.

[23] Yuxi Wang and Yuan Luo. 2023. WeaGAN: Weather-Aware Graph Attention Network for Traffic Prediction. In *Proceedings of the Twenty-Sixth European Conference on Artificial Intelligence*. IOS Press, 2591–2598.

[24] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. 2021. Spatial-Temporal Transformer Networks for Traffic Flow Forecasting. arXiv:2001.02908 [eess.SP]

[25] Qun Yu, Zhiyi Li, Xutao Han, Ping Ju, and Mohammad Shahidehpour. 2024. End-to-End Learning for Stochastic Preventive Dispatch of Renewables-Rich Power Systems in Abnormal Weather Conditions. *Renewable Energy* 234 (2024), 121107.

[26] Haipeng Zhang, Ran Li, Qintao Du, Junyi Tao, Salvador Pineda, Georges Kariniotakis, Simon Camal, Claire Bizon Monroc, Mingyang Sun, Can Wan, Wangkun Xu, and Fei Teng. 2026. Decision-Focused Learning for Power System Decision-Making Under Uncertainty. *IEEE Transactions on Power Systems* 41, 1 (2026), 307–323. https://doi.org/10.1109/TPWRS.2025.3597806