

An Explainable KAN–Shapley Framework for Many-to-Many Electric Vehicle Charging Dispatch in Intelligent Transportation Systems

Wen-Dong Jiang^{ID}, *Graduate Student Member, IEEE*, Chih-Yung Chang^{ID}, *Member, IEEE*, Tzu-Chia Huang, Yue-Shi Lee^{ID}, and Diptendu Sinha Roy^{ID}, *Senior Member, IEEE*

Abstract—Mobile Charging Services (MCS) can alleviate the shortage of fixed charging infrastructure, but real-world dispatching in dense cities must jointly handle time-varying traffic and grid-price signals, hard feasibility constraints such as SOC safety, service time windows, and service regions, and the need for auditable decisions. Existing optimization-based schedulers rely on hand-crafted costs and provide limited decision transparency, whereas end-to-end learning-based dispatchers are often opaque and may violate constraints at inference time. This paper proposes an Explainable KAN–Shapley Dispatch Network (XKSDN) for many-to-many dispatch between EV charging request clusters and multiple Mobile Charging Vehicles (MCVs). XKSDN first abstracts massive requests via spatiotemporal–energy clustering, and then learns a structured utility function with a Kolmogorov–Arnold Network (KAN), whose edge-wise one-dimensional spline functions expose nonlinear effects such as distance decay and SOC-urgency transitions. To turn learned utilities into executable schedules, a constraint-aware matching layer integrates feasibility masking (Big- M), virtual nodes for partial matching, and node replication for limited MCV capacity, and solves the resulting assignment with an improved Hungarian algorithm. For accountability, XKSDN computes local Shapley-value attributions within clusters, quantifying the marginal contributions of distance, SOC, congestion, and price to each dispatch command. Experiments on New York City trip data with SUMO simulations show that XKSDN improves service success rate and response delay while reducing operational cost against representative optimization and deep learning baselines, and provides consistent, auditable explanations for safety-critical deployments.

Index Terms—Electric vehicles, explainable artificial intelligence, intelligent transportation systems, Kolmogorov–Arnold networks, mobile charging service, resource allocation.

Received 6 February 2026; revised 28 March 2026; accepted 24 April 2026. Date of publication 6 May 2026; date of current version 4 September 2026. This work was supported in part by the National Science and Technology Council of Taiwan, Republic of China and in part by NSTC under Grant 113-2221-E-032 -033 -MY2. The Associate Editor for this article was X. Hu. (Corresponding author: Chih-Yung Chang.)

Wen-Dong Jiang, Chih-Yung Chang, and Tzu-Chia Huang are with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei 25137, Taiwan (e-mail: wendongjiang@ieee.org; cychang@mail.tku.edu.tw; 142870@mail.tku.edu.tw).

Yue-Shi Lee is with the Department of Computer Science and Information Engineering, Ming Chuan University, Taoyuan 333, Taiwan (e-mail: leeysh@mail.mcu.edu.tw).

Diptendu Sinha Roy is with the Department of Computer Science and Engineering, National Institute of Technology, Shillong 793003, India (e-mail: diptendu.sr@nitm.ac.in).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TITS.2026.3688187>, provided by the authors.

Digital Object Identifier 10.1109/TITS.2026.3688187

IMPORTANT NOMENCLATURE

Key Acronyms

MCS	Mobile Charging Service.
EV	Electric Vehicle.
MCV	Mobile Charging Vehicle.
SOC	State of charge.
KAN	Kolmogorov–Arnold Network.
XKSDN	Explainable KAN–Shapley Dispatch Network.
TOU	Time-of-use electricity pricing.
ST-ECAC	Spatiotemporal–Energy Coupled Affinity Clustering.
GEM	Graph Expansion and Masking.
SCM	Structural causal model.
ATE	Average treatment effect.
SSR	Service Success Rate.
ARD	Average Response Delay.
OCR	Operational Cost Ratio.
FLB	Fleet Load Balance.

Key Symbols

$G = (\mathcal{V}, \mathcal{E})$	Road network as a directed graph.
$\tau_e(t)$	Travel time of edge e at time t .
$T(u, v, t)$	Shortest travel time from u to v starting at time t .
$\mathcal{R}$	Set of charging requests; $ \mathcal{R} = N$.
$\mathcal{K}$	Set of decision batches/time steps (indexed by k).
$\mathcal{M}$	Set of MCVs; $ \mathcal{M} = M_{\max}$.
$\mathbf{r}_i$	Request tuple $\mathbf{r}_i = (s_i, d_i, \rho_i(\cdot), SOC_i, q_i, [a_i, b_i])$.
q_i	Required replenishment energy of request i .
$[a_i, b_i]$	Acceptable service time window of request i .
ξ_m^k	State of MCV m at batch k .
$x_{m,i}^k$	Binary assignment of request i to MCV m in batch k .
y_i	Indicator of whether request i is served.
z_m	Indicator of whether MCV m is activated.

$u_{m,i}^k$	Rendezvous/charging location for pair (m, i) at batch k .
$\mathbf{h}_c$	Abstract feature vector of demand cluster c ; $c \in \{1, \dots, K\}$.
$g(t)$	Time-of-use electricity price function.
$\mathbf{U}, \mathbf{C}$	Utility matrix and constraint-aware cost matrix.
$\mathcal{C} = \{C_1, \dots, C_K\}$	Set of demand clusters, where $C_c \subseteq \mathcal{R}$ collects the requests assigned to cluster c .
$\mathbf{x}_{m,c}$	Joint MCV-cluster state vector $[\xi_m \oplus \mathbf{h}_c] \in \mathbb{R}^{D_{in}}$.
$\hat{U}_{m,c}$	Predicted dispatch utility for assigning MCV m to cluster c .
$\mathbb{F}(m, c)$	Binary feasibility indicator for assigning MCV m to cluster c .
$M_{\text{inf}}, \lambda_{\text{penalty}}$	Big- M infeasibility cost and dummy-node penalty in GEM.
$\phi_j(\mathbf{x})$	Shapley value of feature j for state $\mathbf{x}$.
$R_{\text{real}}, \mathbf{z}_{\text{exec}}$	Realized reward and observed post-dispatch performance vector.
$\theta, \eta, \lambda_{\text{reg}}$	KAN parameters, learning rate, and regularization coefficient.

I. INTRODUCTION

A GAINST the backdrop of deep integration between smart city energy and transportation systems, Mobile Charging Service (MCS) is regarded as a pivotal solution to alleviate the deficiency of fixed charging infrastructure and enhance the resilience and elasticity of urban charging systems [1], [2], [3]. Compared with traditional fixed charging stations, MCS can flexibly adjust service locations according to dynamic spatiotemporal charging demand [5], [6]. In particular, Li et al. [4] reviewed modeling and optimization approaches for EV-transportation-grid integration, while Pamidimukkala et al. [7] summarized global barriers and motivators for EV adoption. Together, these two recent reviews highlight the practical need for flexible, accessible, and well-coordinated charging services, which further strengthens the motivation of the MCS scenario studied in this work.

As shown in Fig. 1, within a smart-city MCS architecture, the system receives EV charging requests through connected transportation and energy-management layers, either periodically or in an event-triggered manner [8]. Leveraging low-latency communication technologies like 5G, EVs can instantaneously transmit charging requests to the system when a need arises. These requests generally include vehicle position, route intention, battery state, and time-sensitive service information, so practical systems often adopt batch or regional coordination mechanisms to stabilize decisions under rapidly changing demand [9], [10], [11], [12], [13], [14].

Most existing studies on EV charging scheduling and infrastructure planning are conducted under the paradigm of fixed charging station deployment [8], [9], [10], [11], [13], [22], [23]. In these works, charging station locations are assumed to be predetermined, and scheduling decisions mainly focus on vehicle assignment, charging order, or power allocation, with EVs required to deviate from their original routes to access

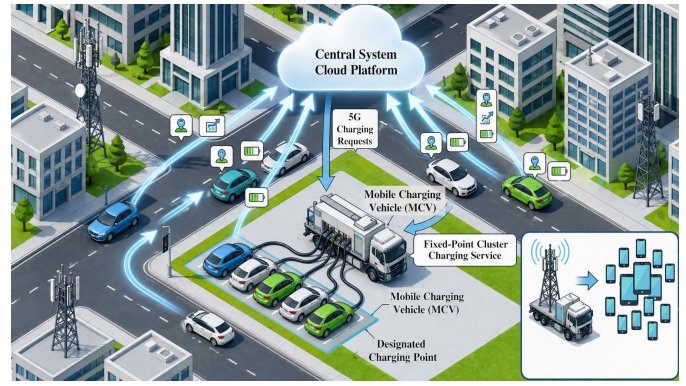


Fig. 1. Smart City Mobile Charging Service Architecture.

designated stations. In contrast, under the MCS paradigm, charging resources are mobile, and service locations become part of the scheduling decision rather than fixed inputs. This fundamental difference renders conventional methods designed for fixed charging stations difficult to directly apply to MCS scenarios, motivating a rethinking of demand aggregation and service location selection mechanisms.

In the MCS architecture, Mobile Charging Vehicles (MCVs) function as movable charging piles [24]. Charging requests are grouped according to EVs' travel routes, spatial proximity, and energy states. EVs within the same group are served by a single MCV and receive charging service at a designated location determined through the scheduling process.

This grouped service mechanism helps reduce the additional travel cost incurred by individual EVs, as the MCV relocates toward a suitable service location to accommodate the aggregated demand and performs charging in a stationary manner. From an operational perspective, this concept is analogous to dynamic resource deployment in cellular networks, where mobile base stations are repositioned toward regions with concentrated demand to improve overall service efficiency.

It should be emphasized that MCS is not intended to replace mainstream fixed charging infrastructure in all cases. Rather, its practical value lies in complementary scenarios in which fixed stations are spatially insufficient, temporarily overloaded, or operationally unavailable, such as emergency low-SOC rescue, temporary demand peaks around airports, exhibitions, and large events, overflow service for taxi or ride-hailing fleets, and resilience support during outages or other extreme conditions [2], [14], [18]. In this sense, the significance of MCS is not its share in the overall charging market, but its ability to provide elastic and location-adaptive service precisely where static infrastructure is least effective.

However, the actual performance of MCS still relies on coordinated dispatch, system-level energy-transport integration, and reliable vehicle-energy modeling [12], [15], [17]. Without a systematic and generalizable scheduling and clustering strategy, MCVs may still face frequent empty driving, redundant dispatching, and service delays, as also reflected in recent studies on adaptive charging and mobile battery swapping [14], [18]. Existing studies [19], [20], [21], [22], [23], [24] largely determine dispatching and scheduling decisions through large-scale combinatorial optimization methods

or deep learning models [25], [26], [27], [28], [29] with limited interpretability. Although these approaches have demonstrated certain effectiveness in specific scenarios, they generally suffer from two key limitations.

First, there is an excessive reliance on prior knowledge and manual rule settings. Current EV charging dispatching methods typically require pre-setting numerous parameters and rules highly correlated with system operations, such as demand weights, distance thresholds, service radii, or priority functions. These prior settings are mostly based on experience or specific scenario assumptions. Consequently, model performance often deteriorates significantly when actual demand distribution, traffic conditions, or grid loads change. The lack of effective adaptive adjustment capabilities makes it difficult for dispatching strategies to generalize stably across different city scales or time periods [8], [9], [10], [11], [12], [13], [14], [19], [20], [21], [22], [23], [24].

Second, dispatch decisions lack interpretability and decision attribution capabilities. While most optimization or deep learning-based models can output feasible schedules, their internal decision-making processes are often not explicitly explainable. They cannot clearly explain why certain charging requests are prioritized or why an MCV is dispatched to a specific location, nor can they quantify the impact of a single EV request on the overall decision. This lack of transparency not only lowers operator trust in system decisions but also limits the possibility of manual intervention, strategy adjustment, and policy evaluation during actual deployment [30], [31], [32].

To address these issues, this paper proposes an Explainable KAN-Shapley Dispatch Network (XKSDN) tailored for real-world urban operational scenarios. This framework supports the cooperative dispatch of multiple MCVs and explicitly incorporates multiple realistic constraints, including traffic congestion, grid load, vehicle SOC, service area limitations, and service time windows. Unlike existing methods [19], [20], [21] that focus only on single vehicles or idealized assumptions, the proposed XKSDN extends the scheduling object to the fleet level and adopts a layered decision process of “Learning, Matching, and Explaining” to balance feasibility, efficiency, and transparency.

Specifically, the proposed XKSDN adopts the Kolmogorov–Arnold Network (KAN) as the core function approximator to learn dispatching utility in high-dimensional state spaces. By modeling spatiotemporal and energy couplings through decomposable one-dimensional function compositions, KAN improves interpretability while reducing structural computational complexity. To transform the learned utility into executable dispatch actions, a constraint-aware cost matrix is constructed within each demand cluster, and an improved Hungarian algorithm is applied to solve the vehicle–demand assignment problem. The matching cost jointly incorporates KAN-derived utility and operational constraints, with infeasible pairings explicitly masked. The virtual nodes and vehicle-node replication are further introduced to support partial matching and limited capacity acceptance while preserving bipartite matching solvability.

Finally, to mitigate the cost and instability of global attribution in large-scale multi-vehicle scenarios, the proposed

XKSDN performs multivariate clustering on demands and vehicles to reduce the decision scope, followed by local Shapley value analysis within each cluster to quantify the marginal contributions of factors such as distance, SOC, traffic conditions, and electricity prices to individual dispatch decisions. This design enhances explanation stability and consistency with manageable computational overhead. Furthermore, by jointly modeling service success, response delay, travel energy consumption, and electricity cost within the reward function and constraint formulation, the proposed XKSDN is able to produce efficient, feasible, and deployment-ready multi-vehicle charging schedules under complex constraints and high-load conditions. Experimental results on real-world datasets and SUMO-based simulation environments validate the effectiveness of the proposed approach.

Novelty positioning: Different from (i) pure learning-based dispatchers that lack hard-constraint guarantees and (ii) pure optimization-based schedulers that require hand-crafted costs and are opaque, XKSDN couples an intrinsically interpretable utility learner (KAN), an executable constraint-enforcing matcher (GEM + Hungarian), and a decision-attribution module (local Shapley) into a single end-to-end dispatch pipeline.

Overall, the main contributions of this paper are as follows:

- **An explainable fleet-level dispatch framework for mobile EV charging services.** We introduce KAN-based structured utility learning to model spatiotemporal–energy coupling with decomposable representations, improving transparency while maintaining high predictive capacity.
- **A constraint-aware matching layer for deployable decisions.** We design an improved Hungarian-based dispatch layer with feasibility masking (Big- M), virtual nodes for partial matching, and vehicle-node replication for limited capacity, ensuring solvability and strict satisfaction of operational constraints.
- **A low-cost and stable explanation mechanism for dispatch attribution.** We perform multivariate clustering and compute local Shapley values within clusters to provide consistent, instance-level explanations of why specific assignments are selected under traffic, SOC, and price variations.

The remainder of the paper is organized as follows. Section II discusses and compares previous related work. Section III presents the problem formulation and assumptions. Section IV describes the proposed XKSDN system. Section V reports the experiments and performance evaluation, including a dedicated discussion on generalization and scalability. Section VI concludes the paper.

II. RELATED WORK

This section reviews previous work in the field of MCS scheduling. Existing scheduling strategies can generally be categorized into two streams. The first stream includes traditional methods based on rules and heuristic algorithms. The second stream includes data-driven methods based on deep learning. To make the comparison easier to follow, Table I gives a checklist-style review of representative studies and highlights the key research gaps more directly.

TABLE I
CHECKLIST-STYLE LITERATURE REVIEW FOR SECTION II

Category	Representative studies	RT-M2M	Dyn.	Interp.	Exec.
Infrastructure planning	Stoch-Plan [22], Bi-Level [23]	×	×	×	×
Fleet optimization	MILP-Coop [20], Robust-Opt [21], Mech-Design [24]	×	×	×	×
Pricing / game theory	Game-Pricing [19]	×	×	×	×
Direct DL dispatch	Fusion-Est [27], ANN-Disp [28]	×	×	×	×
Advanced learning	Deep-IL [26], GAN-QL [29], DL-Coupling [25]	×	✓	×	×
This work	XKSDN	✓	✓	✓	✓

RT-M2M: real-time many-to-many mobile charging dispatch; *Dyn.*: explicit dynamic traffic and energy coupling; *Interp.*: instance-level local interpretability; *Exec.*: hard-feasible executable dispatch.

A. Rules and Heuristic Algorithms

Traditional research in charging infrastructure and service scheduling predominantly relied on combinatorial optimization, robust planning, and game-theoretic coordination. Representative examples include long-horizon infrastructure expansion and pricing models [22], [23], cooperative fleet scheduling and robust routing frameworks [20], [21], [24], and strategic charging-lane pricing schemes [19].

As highlighted in Table I, these studies are theoretically rigorous but still rely heavily on hand-crafted objectives, pre-determined forecasts, rigid service assumptions, or idealized user-response models. Consequently, they are difficult to adapt to stochastic many-to-many MCS dispatch under abrupt traffic changes, heterogeneous request urgency, and operator-side explainability requirements.

B. Data-Driven Methods

Recent research has increasingly turned to data-driven approaches to overcome the computational bottlenecks and rigid assumptions of rule-based methods. Representative studies include state-estimation models [27], direct dispatch policies [28], imitation learning [26], GAN-enhanced reinforcement learning [29], and large-scale spatio-temporal coupling frameworks [25].

However, better nonlinear fitting alone does not close the deployment gap. Most existing data-driven methods remain weak in instance-level interpretability, explicit decision attribution, and hard-constraint enforcement during inference. Therefore, the key unresolved gap is not only higher predictive accuracy or faster inference, but a deployable framework that simultaneously provides expressive nonlinear utility learning, transparent local explanations, and executable feasibility guarantees for many-to-many urban MCS dispatch.

III. PROBLEM FORMULATION AND ASSUMPTIONS

This section presents the problem formulation and system assumptions. As illustrated in Fig. 2, consider a smart city MCS system operating over a road network represented by a directed graph $G = (\mathcal{V}, \mathcal{E})$. For any road edge $e \in \mathcal{E}$, the travel time at time t is denoted by $\tau_e(t)$. Consequently, the minimum travel time between any two nodes starting at time t is formulated as

$$T(u, v, t) = \min_{\pi: u \rightarrow v} \sum_{e \in \pi} \tau_e(t). \quad (1)$$

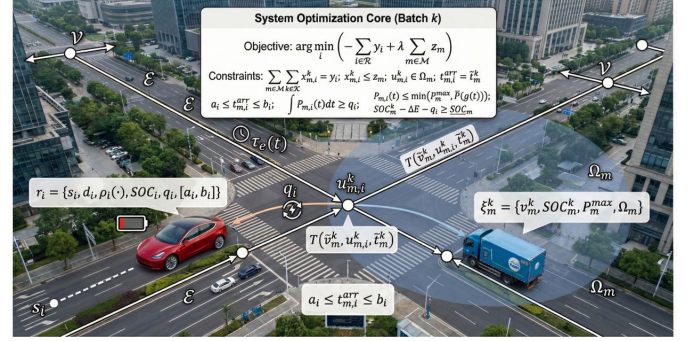


Fig. 2. Definition and environmental assumptions of a smart city MCS system.

There exists a set of EV charging requests awaiting service within the city, denoted by $\mathcal{R} = \{1, 2, \dots, N\}$. The system receives these requests in batch mode at a set of discrete decision time steps $\mathcal{K} = \{1, 2, \dots, K\}$, corresponding to physical time t_k . Each individual request $i \in \mathcal{R}$ is characterized as

$$\mathbf{r}_i = (s_i, d_i, \rho_i(\cdot), SOC_i, q_i, [a_i, b_i]), \quad (2)$$

where $s_i \in \mathcal{V}$ represents the request origin, $d_i \in \mathcal{V}$ is the expected destination, $\rho_i(\cdot)$ denotes the projected travel trajectory, $SOC_i \in (0, 1]$ indicates the current state of charge, $q_i > 0$ represents the required energy replenishment, and $[a_i, b_i]$ defines the acceptable service time window.

The MCV fleet is represented by the set $\mathcal{M} = \{1, 2, \dots, M_{\max}\}$. The state of each MCV $m \in \mathcal{M}$ at decision time step t_k is defined as

$$\xi_m^k = (v_m^k, SOC_m^k, P_m^{\max}, \Omega_m), \quad (3)$$

where $v_m^k \in \mathcal{V}$ denotes the current location, $SOC_m^k \in (0, 1]$ represents the onboard battery status, $P_m^{\max}$ is the maximum output power, and $\Omega_m \subseteq \mathcal{V}$ indicates the designated service operational area. Additionally, a time-of-use (TOU) electricity price function $g(t)$ is introduced to characterize charging costs and potential grid load constraints.

At each decision epoch t_k , the system must determine a triplet of decisions: (1) which charging requests to serve, (2) which MCV performs the service, and (3) the specific rendezvous/charging location. Let binary variable $y_i \in \{0, 1\}$ indicate whether request i is successfully served; $z_m \in \{0, 1\}$ indicate whether MCV m is activated; and $x_{m,i}^k \in \{0, 1\}$ denote the assignment of request i to MCV m in the batch k . Let $u_{m,i}^k \in \mathcal{V}$ be the rendezvous point between the MCV and the EV.

The objective of this paper is to maximize demand coverage with minimal MCV resource utilization, subject to operational feasibility constraints. This can be formulated as a two-stage lexicographical objective as

$$\begin{aligned} \mathbf{x}^* &\in \arg \max_{\{x_{m,i}^k, y_i, z_m, u_{m,i}^k\}} \sum_{i \in \mathcal{R}} y_i, \\ \mathbf{x}^{**} &\in \arg \min_{\{x_{m,i}^k, y_i, z_m, u_{m,i}^k\}} \sum_{m \in \mathcal{M}} z_m, \\ \text{s.t. } &\sum_{i \in \mathcal{R}} y_i = \sum_{i \in \mathcal{R}} y_i^*. \end{aligned} \quad (4)$$

Equivalently, this can be expressed as a single weighted objective function:

$$\arg \min_{\{x_{m,i}^k, y_i, z_m, u_{m,i}^k\}} \left(-\sum_{i \in \mathcal{R}} y_i + \lambda \sum_{m \in \mathcal{M}} z_m \right), \quad \lambda > 0. \quad (5)$$

Here, λ is the fleet-activation penalty that balances request coverage against the use of additional MCV resources.

Regarding constraints, first, each request is assigned at most once, and any assignment necessitates the activation of the corresponding vehicle:

$$\begin{aligned} \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}} x_{m,i}^k &= y_i, \quad \forall i \in \mathcal{R}, \\ x_{m,i}^k &\leq z_m, \quad \forall m \in \mathcal{M}, i \in \mathcal{R}, k \in \mathcal{K}. \end{aligned} \quad (6)$$

Second, the service area constraint requires the rendezvous point to be located within the designated operational region:

$$u_{m,i}^k \in \Omega_m, \quad \forall m, i, k \text{ with } x_{m,i}^k = 1. \quad (7)$$

The time window constraint is defined based on arrival times. Let $\tilde{v}_m^k$ and $\tilde{t}_m^k$ denote the location and time of MCV m 's previous task completion prior to batch k . The arrival time at the rendezvous point is calculated as

$$t_{m,i}^{\text{arr}} = \tilde{t}_m^k + T(\tilde{v}_m^k, u_{m,i}^k, \tilde{t}_m^k). \quad (8)$$

This must satisfy the customer's acceptable time window $a_i \leq t_{m,i}^{\text{arr}} \leq b_i$, for all m, i, k with $x_{m,i}^k = 1$.

The charging service duration is approximated by energy and power dynamics. Let $P_{m,i}(t) \leq P_m^{\text{max}}$ be the charging power, which is further constrained by the grid state $g(t)$ (imposing a total power cap $\bar{P}(g(t))$ and cost $c(g(t))$):

$$\int_{t_{m,i}^{\text{arr}}}^{t_{m,i}^{\text{dep}}} P_{m,i}(t) dt \geq q_i, \quad P_{m,i}(t) \leq \min \{P_m^{\text{max}}, \bar{P}(g(t))\}. \quad (9)$$

Here, $t_{m,i}^{\text{dep}}$ denotes the service completion time, $\bar{P}(g(t))$ is the grid-imposed charging-power ceiling under grid state $g(t)$, and $c(g(t))$ is the corresponding unit electricity price used in the operating-cost calculation.

Finally, the energy feasibility constraint ensures that the MCV retains a safe battery level after traversing to the rendezvous point and delivering the requested energy:

$$SOC_m^k - \Delta E_{m:v \rightarrow u} - q_i \geq SOC_{\text{safe}}. \quad (10)$$

To ensure clarity, $\Delta E_{m:v \rightarrow u}$ represents the travel energy consumption required for the MCV to move from its current location v to the rendezvous point u . This value is dynamically

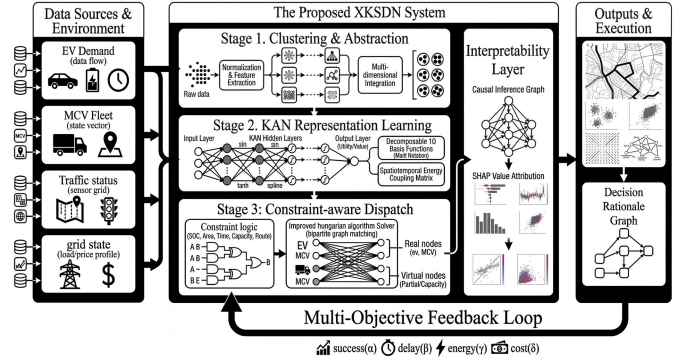


Fig. 3. The proposed XKSDN system framework. This framework integrates multi-source data and completes constraint-aware scheduling through clustering abstraction, KAN representation learning, and an improved Hungarian algorithm; it further combines a Shapley-based interpretability layer with a multi-objective feedback loop to achieve transparent and efficient fleet collaborative dispatch.

calculated based on the path distance and the traffic-dependent energy consumption rate. The SOC_{safe} is the mandatory safety battery threshold to prevent MCV breakdown.

The assumptions adopted in this paper are defined as follows:

- (A1) **Batch Arrivals.** Requests are collected in batches. At each time t_k , the current set of arrived requests $\mathcal{R}_k \subseteq \mathcal{R}$ is processed. Unserved requests may be carried over to subsequent batches or expire at time b_i . $\square$
- (A2) **Piecewise-Stationary Traffic.** Traffic conditions are assumed to be piecewise-stationary within a batch interval, i.e., estimated using $\tau_e(t_k)$ during $[t_k, t_{k+1})$. $\square$
- (A3) **Known Trajectories.** The EV trajectory $\rho_i(\cdot)$ is known or predicted via navigation; the rendezvous point $u_{m,i}^k$ is selected along this path, satisfying reachability and service area constraints. $\square$
- (A4) **Non-Preemptive Service.** The charging process is non-preemptive; once charging begins for request i , it continues until q_i is delivered or the upper time limit b_i is reached. $\square$
- (A5) **Feasible Dispatch Principle.** Batch decisions prioritize executable dispatch. The workload of each MCV within a single batch is limited by its travel and charging capabilities; more general multi-task sequencing and power allocation can be extended without altering the objective of this paper. $\square$

The above is the problem formulation and system assumptions in this paper. In Section IV, the details of the proposed XKSDN are presented.

IV. THE PROPOSED XKSDN SYSTEM

The proposed Explainable KAN-Shapley Dispatch Network (XKSDN) aims to address the limitations of conventional rule-based algorithms [19], [20], [21], [22], [23], [24] and limited-interpretability deep learning approaches [25], [26], [27], [28], [29], and to establish a mobile charging dispatch framework that is both interpretable and scalable. As illustrated in Fig. 3, the system first aggregates multi-source data, including EV

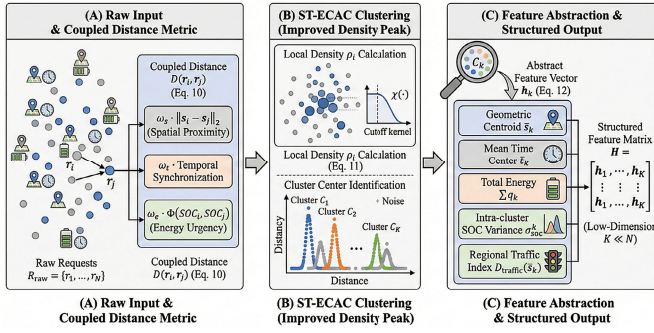


Fig. 4. The spatiotemporal clustering and feature abstraction process.

demand streams, MCV fleet states, traffic sensing networks, and grid load conditions. Through a clustering and abstraction stage, these high-dimensional raw inputs are transformed into structured representations of demand clusters.

Next, in the core KAN representation learning stage, KAN is employed as a function approximator. By leveraging decomposable one-dimensional basis functions (e.g., sine, hyperbolic tangent, and spline functions), the model efficiently captures nonlinear couplings between spatiotemporal and energy-related features, thereby enhancing transparency and computational efficiency.

At the decision execution layer, the constraint-aware dispatch stage performs hard-constraint checking and applies an improved Hungarian algorithm with dummy nodes. The resulting bipartite matching can flexibly handle partial acceptance and capacity constraints, ensuring that the produced dispatch decisions are physically feasible and practically deployable. In addition, XKSDN incorporates an interpretability layer that combines causal inference graphs with SHAP-based attribution, visualizing the marginal contributions of key factors as decision-rationale graphs, thereby improving traceability and auditability.

Finally, via a multi-objective closed-loop feedback mechanism that accounts for service success rate α , response delay β , energy consumption γ , and operational cost δ , XKSDN continuously updates and optimizes model parameters in dynamic environments, jointly improving dispatch utility and decision transparency.

For clarity, we organize the framework into four modules: spatiotemporal clustering and feature abstraction, KAN-based utility representation learning, constraint-aware bipartite matching dispatch, and interpretability mechanism with closed-loop optimization. The details are described as follows.

A. Spatiotemporal Clustering and Feature Abstraction

This stage mitigates the curse of dimensionality in large-scale mobile charging scheduling. As shown in Fig. 4, the raw input is a real-time set of EV charging requests, denoted as $\mathcal{R}_{raw} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\}$, where each request $\mathbf{r}_i$ contains heterogeneous, high-dimensional features. Directly feeding N discrete requests into the subsequent KAN module would make the computation scale poorly with N . To address this, we propose a Spatiotemporal–Energy Coupled Affinity Clustering

(ST-ECAC) algorithm that abstracts individual requests into structured demand clusters.

To quantify the mergeability between different requests, we define a coupled distance metric $D(\mathbf{r}_i, \mathbf{r}_j)$. Unlike Euclidean distance [20], [22], [23], which only reflects geographic proximity, this metric introduces a weighted coupling of energy urgency and temporal synchronization:

$$D(\mathbf{r}_i, \mathbf{r}_j) = \omega_s \|\mathbf{s}_i - \mathbf{s}_j\|_2^2 + \omega_t \left\| \frac{a_i + b_i}{2} - \frac{a_j + b_j}{2} \right\|_2^2 + \omega_e \Phi(SOC_i, SOC_j), \quad (11)$$

where $\mathbf{s}_i$ and $\mathbf{s}_j$ denote vehicle position vectors, $\|\cdot\|_2$ is the ℓ_2 norm, $[a_i, b_i]$ is the service time window, $\Phi(\cdot)$ is a penalty function based on SOC difference, and ω_s , ω_t , and ω_e are normalized weights for the spatial, temporal, and energy dimensions. The rationale is that only requests that are spatially close, temporally overlapping, and similar in energy state are suitable for grouping.

Based on this metric, we employ an improved Density Peak Algorithm for unsupervised clustering, following the density-peak clustering principle in [34]. The algorithm first computes the local density ρ_i for each node i :

$$\rho_i = \sum_{j \neq i} \chi(D(\mathbf{r}_i, \mathbf{r}_j) - d_c), \quad (12)$$

where $\chi(\cdot)$ is a cutoff kernel function and d_c is the cutoff distance. Nodes with high local density and large distance from other high-density nodes are selected as cluster centers, which adaptively determines the number of clusters K and filters isolated requests treated as noise.

After forming K clusters $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$, we perform feature abstraction by compressing all requests in each cluster into a unified state feature vector. For cluster c , the abstract feature vector $\mathbf{h}_c$ is defined as

$$\mathbf{h}_c = \left[\bar{\mathbf{s}}_c, \bar{t}_c, \sum_{i \in C_c} q_i, \sigma_{soc}^c, \mathcal{D}_{traffic}(\bar{\mathbf{s}}_c) \right]^T, \quad (13)$$

where $\bar{\mathbf{s}}_c$ is the geometric centroid, $\bar{t}_c$ is the mean center of the time windows, $\sum_{i \in C_c} q_i$ is the total energy demand, σ_{soc}^c is the intra-cluster SOC variance, and $\mathcal{D}_{traffic}(\cdot)$ denotes the current traffic congestion index. Here, $C_c \subseteq \mathcal{R}$ denotes the member set of cluster c .

Through this stage, the original N -request space is mapped to a K -cluster feature matrix $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_K]^T$. This reduces input dimensionality while preserving the key geographic and energy-demand structure, enabling efficient coarse-grained fleet-to-cluster matching. A detailed derivation of the neighborhood bound and runtime complexity is provided in Appendix A of the Supplementary Material. Appendix O further compares ST-ECAC with standard clustering baselines, including k-means, DBSCAN, and spectral clustering, under the same downstream dispatch surrogate.

B. KAN-Based Utility Representation Learning

The objective of this stage is to construct a high-accuracy, explainable utility estimator. The input is the joint state of

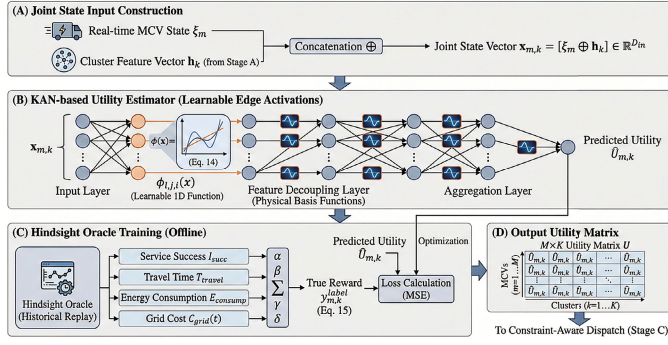


Fig. 5. The KAN-based utility representation learning process.

a mobile charging vehicle (MCV) and a demand cluster, and the output is the dispatch utility of that pairing under current conditions. As shown in Fig. 5, we define the joint state vector $\mathbf{x}_{m,c} = [\xi_m \oplus \mathbf{h}_c] \in \mathbb{R}^{D_m}$, where $\oplus$ denotes vector concatenation, ξ_m is the real-time MCV state, and $\mathbf{h}_c$ is the cluster feature vector from the previous stage. Here, c indexes demand clusters and K denotes the number of clusters. For consistency, the batch index remains k as introduced in Section III, whereas the index c is used throughout Sections IV-B and IV-C to denote a demand cluster.

Unlike multi-layer perceptrons (MLPs) [25], [26], [27], [28], [29] that apply fixed activation functions at nodes, the proposed framework adopts a Kolmogorov–Arnold Network (KAN), where learnable activation functions are placed on edges. This design captures nonlinear couplings between spatiotemporal and energy variables with fewer parameters.

Following the Kolmogorov–Arnold representation theorem and the practical KAN construction in [35], we decompose the mapping $\Psi(\mathbf{x}_{m,c})$ from the input space to the output utility into a superposition of univariate functions. For the l -th layer, the computation is $x_j^{(l+1)} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_i^{(l)})$, where $x_i^{(l)}$ is the node value in layer l , and $\phi_{l,j,i}(\cdot)$ is a learnable one-dimensional function connecting two nodes across adjacent layers. Here, n_l denotes the number of nodes in layer l , $x_j^{(l+1)}$ is the output of node j in layer $l+1$, and the final scalar utility is read from the output layer as $\Psi(\mathbf{x}_{m,c})$.

In XKSDN, we parameterize ϕ as a linear combination of B-splines with a residual term, consistent with the implementation in [35]:

$$\phi(x) = w_b x + w_s \sum_p c_p B_p(x). \quad (14)$$

In this parameterization, w_b and w_s are the residual and spline weights, c_p is the coefficient of the p -th B-spline basis, and $B_p(x)$ is the corresponding spline basis function evaluated at x .

This structure provides clear physical interpretation: the first layer acts as a feature decoupling layer that maps high-dimensional states to independent physical basis functions (e.g., distance-decay and congestion-penalty curves), while the second layer aggregates them into a utility score. The spline functions can be visualized to explain effects such as why increasing distance leads to a nonlinear decline in utility.

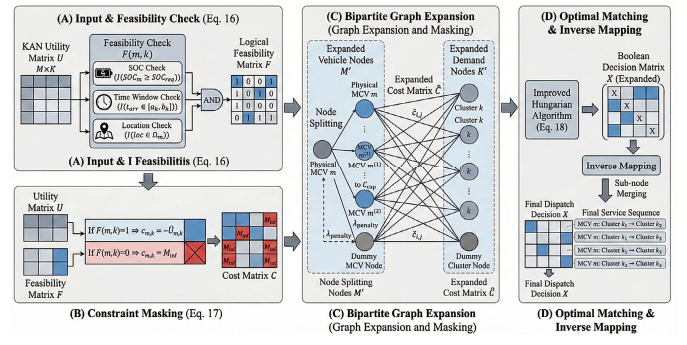


Fig. 6. The workflow of the Graph Expansion and Masking (GEM) strategy for constraint-aware bipartite matching dispatch.

To train the network when direct labels are unavailable, we use a hindsight oracle to generate ground-truth utilities offline. Specifically, historical simulation replay is used with a global optimal solver to compute the true reward $y_{m,c}^{label}$ for each pair (m, c) . The label is defined as a weighted multi-objective sum:

$$y_{m,c}^{label} = \alpha \cdot \mathbb{I}_{succ} - \beta \cdot \frac{T_{travel}}{T_{max}} - \gamma \cdot \frac{E_{consump}}{E_{bat}} - \delta \cdot C_{grid}(t), \quad (15)$$

where $\mathbb{I}_{succ}$ is the success indicator, and T_{travel} , $E_{consump}$, and C_{grid} denote travel time, energy consumption, and electricity cost, respectively. The learning objective is to minimize the mean squared error (MSE) between the predicted utility $\hat{U}_{m,c}$ and $y_{m,c}^{label}$. The coefficients α , β , γ , and δ are trade-off weights for success, delay, energy, and grid cost, while T_{max} and E_{bat} are normalization constants corresponding to the maximum tolerable travel time and the MCV battery capacity, respectively.

The output of this stage is an $M \times K$ utility matrix $\mathbf{U}$, where each element $\hat{U}_{m,c}$ represents the predicted gain for assigning MCV m to cluster c . This matrix is then used to build the constraint-aware cost matrix in the next stage, enabling the Hungarian algorithm to perform global matching while retaining the fitting power of learning-based models. Detailed derivations for the structural interpretability and approximation properties of KAN are provided in Appendices B and C of the Supplementary Material.

C. Constraint-Aware Bipartite Matching Dispatch

Traditional Hungarian algorithm [16] can only solve the standard square-matrix assignment problem. To adapt it to the MCS scenario, which involves many-to-many assignments (one vehicle serving multiple tasks), supply-demand imbalance, and hard constraint restrictions, we propose a Graph Expansion and Masking (GEM) strategy.

As shown in Fig. 6, the objective of this stage is to convert the probabilistic utility outputs from KAN into deterministic, physically feasible dispatch commands. The input is the utility matrix $\mathbf{U}$ generated in the previous stage, and the output is a binary decision matrix $\mathbf{X} = [x_{m,c}]_{M \times K}$, where $x_{m,c} = 1$ signifies assigning the m -th MCV to the c -th demand cluster. To overcome the limitations of standard bipartite matching in handling hard constraints and supply-demand imbalances, we propose a constraint masking and graph expansion mechanism.

First, we construct the constraint-aware cost matrix $\mathbf{C}$. Although KAN implicitly learns soft constraints, explicit masking is required to ensure absolute safety for critical operations. For any pair (m, c) , we define a logical feasibility function $\mathbb{F}(m, c)$ as follows:

$$\begin{aligned} \mathbb{F}(m, c) = & \mathbb{I}(SOC_m \geq SOC_{req}^c) \\ & \wedge \mathbb{I}(t_{m,c}^{arr} \in [a_c, b_c]) \\ & \wedge \mathbb{I}(\text{loc}_{m,c} \in \Omega_m), \end{aligned} \quad (16)$$

where $\wedge$ denotes the logical AND operator. If $\mathbb{F}(m, c) = 0$ (i.e., insufficient SOC, time window conflict, or out-of-service area), the corresponding cost is set to an infinity constant M_{inf} ; otherwise, the cost is set to the negative of the utility. The cost element $c_{m,c}$ is defined as

$$c_{m,c} = \begin{cases} -\hat{U}_{m,c}, & \text{if } \mathbb{F}(m, c) = 1, \\ M_{\text{inf}}, & \text{otherwise.} \end{cases} \quad (17)$$

Here, SOC_m denotes the current onboard SOC of MCV m at the decision epoch, SOC_{req}^c is the minimum energy required to reach and serve cluster c , $[a_c, b_c]$ is the aggregated feasible time window of cluster c , and $\text{loc}_{m,c}$ is the candidate rendezvous location associated with the pair (m, c) . This Big- M masking mechanism guarantees that any selected assignment strictly satisfies all physical hard constraints. In practice, we normalize utilities (and thus $-\hat{U}$) to a bounded range before matching, and then set M_{inf} to be safely larger than the maximum achievable magnitude of $-\hat{U}_{m,c}$ to forbid selecting infeasible pairs while reducing numerical issues. A more detailed discussion on practical M_{inf} selection, normalization, and numerical-stability stress tests is provided in Appendix N of the Supplementary Material.

Second, to address the multi-service capability of MCVs and the supply-demand mismatch $M \neq K$, we perform bipartite graph expansion. Considering that each MCV may possess the capacity C_{cap} to serve multiple demand clusters in one decision batch (i.e., the maximum number of cluster-level tasks that can be accepted and executed within that batch horizon), we split the original vehicle node m into C_{cap} sub-nodes $\{m^{(1)}, m^{(2)}, \dots, m^{(C_{\text{cap}})}\}$. These sub-nodes inherit the attributes and location of the parent node.

To satisfy the square matrix requirement of the Hungarian algorithm and allow for service rejection when demand exceeds supply, we introduce dummy nodes on either the vehicle or demand side. The connection cost to dummy nodes is set to a penalty threshold λ_{penalty} , representing the opportunity cost of service failure.

After constructing the expanded cost matrix $\tilde{\mathbf{C}}$, we employ an improved Hungarian algorithm to solve for the global optimal matching. The optimization objective is formulated as

$$\min_{\mathbf{X}} \sum_{i \in \mathcal{M}'} \sum_{j \in \mathcal{K}'} \tilde{c}_{i,j} x_{i,j}, \quad \text{s.t.} \quad \sum_j x_{i,j} = 1, \quad \sum_i x_{i,j} = 1. \quad (18)$$

where $\mathcal{M}'$ and $\mathcal{K}'$ denote the expanded sets of vehicles and demands, respectively. In Eq. (18), $x_{i,j} \in \{0, 1\}$ indicates whether expanded vehicle node i is matched to expanded

demand node j , and $\tilde{c}_{i,j}$ is the corresponding expanded assignment cost.

The algorithm outputs a Boolean matrix $\mathbf{X}$. The system then performs inverse mapping on the results: merging all sub-node tasks belonging to the same physical vehicle m to generate the final service sequence for that vehicle. A full feasibility and optimality derivation for the GEM dispatch module is provided in Appendix D of the Supplementary Material.

D. Interpretability Mechanism and Closed-Loop Optimization

To bridge the trust gap between opaque neural approximations and safety-critical urban operations, the proposed XKSDN integrates an explanation module rooted in cooperative game theory and causal inference around an intrinsically structured KAN utility learner. The executable dispatch decision is still produced by the KAN utility scores together with the subsequent GEM-enhanced Hungarian matcher; the explanation module is mainly applied to the learned utilities and realized assignments for instance-level interpretation and auditing, rather than for introducing additional hard matching constraints. Let $\mathcal{F} = \{1, \dots, D\}$ denote the set of input features. The contribution of feature j , denoted as the Shapley value ϕ_j , is computed by averaging its marginal impact across all possible feature coalitions $S \subseteq \mathcal{F} \setminus \{j\}$ following the standard SHAP/Shapley formulation in [30] as

$$\phi_j(\mathbf{x}) = \sum_{S \subseteq \mathcal{F} \setminus \{j\}} \frac{|S|! (|\mathcal{F}| - |S| - 1)!}{|\mathcal{F}|!} [\Psi(S \cup \{j\}) - \Psi(S)]. \quad (19)$$

where $\Psi(S)$ represents the expected output of the KAN network conditioned on the subset S . This guarantees that the sum of local attributions equals the total deviation from the baseline expectation:

$$\sum_{j=1}^D \phi_j(\mathbf{x}) = \Psi(\mathbf{x}) - \mathbb{E}[\Psi(\mathbf{x})]. \quad (20)$$

Here, $|S|$ and $|\mathcal{F}|$ denote coalition cardinalities, and $\mathbb{E}[\Psi(\mathbf{x})]$ is the baseline expected utility over the reference state distribution. Further mathematical discussion of the attribution module is provided in Appendix B of the Supplementary Material.

Beyond correlation-based attribution, we construct a structural causal model (SCM) to distinguish direct causal effects from confounding factors in the decision process, following Pearl's structural-causal framework [36]. We define the SCM as a tuple $\mathcal{M} = \langle \mathcal{V}, \mathcal{E}, \mathcal{G} \rangle$, where endogenous variables $\mathcal{V}$ and edges $\mathcal{E}$ are governed by structural equations $\mathcal{G}$. For a specific variable X_i , its value is determined by its parents PA_i in the causal graph and exogenous noise N_i :

$$X_i := f_i(\text{PA}_i, N_i), \quad \forall i \in \{1, \dots, D\}. \quad (21)$$

To evaluate the causal impact of an intervention on the dispatch utility Y , we employ the do-calculus operator [36]. The average treatment effect (ATE) of feature X_j changing from x to x' is

$$\text{ATE}_{\text{do}(X_j)} = \mathbb{E}[Y \mid \text{do}(X_j = x')] - \mathbb{E}[Y \mid \text{do}(X_j = x)]. \quad (22)$$

An operational illustration of the SCM-based causal validation corresponding to Eq. (22) is provided in Appendix G of the Supplementary Material. To clarify the operational integration of the SCM, we specify that it is used as an auxiliary analysis tool rather than an expensive real-time decision gate. In our setting, the causal graph and key ATE relationships are constructed offline (or updated at a low frequency) using historical/simulated data, and then used online only for lightweight monitoring of a few safety-critical variables (e.g., SOC urgency and congestion shocks). Therefore, Shapley/SCM explanations do not define the feasible set or alter the online combinatorial solver directly; instead, they provide post-decision accountability and low-frequency diagnostic feedback on whether the learned pReferences remain consistent with expected domain knowledge.

The system is not static but evolves via a multi-objective feedback loop. Upon the execution of a dispatch batch, the environment returns a real-world performance vector $\mathbf{z}_{exec} \in \mathbb{R}^4$, comprising binary success status $\mathbb{I}_{succ}$, delay τ_{delay} , energy consumption e_{cons} , and cost c_{cost} . The observed composite reward R_{real} is calculated as a weighted linear combination normalized by maximum tolerance thresholds:

$$R_{real}(\mathbf{z}_{exec}) = w_1 \mathbb{I}_{succ} - w_2 \frac{\tau_{delay}}{\tau_{max}} - w_3 \frac{e_{cons}}{E_{bat}} - w_4 \frac{c_{cost}}{C_{budget}}. \quad (23)$$

Here, $w_1, \dots, w_4$ are metric weights, τ_{max} is the maximum tolerable delay, E_{bat} is the MCV battery-capacity normalization constant, and C_{budget} is the reference cost budget used for normalization.

To enable online adaptation to non-stationary urban dynamics, we formulate the feedback mechanism as a supervised regression problem. The objective is to minimize the temporal-difference error between the KAN-predicted utility $\hat{U}_{m,c}$ and the realized reward R_{real} . Accordingly, the direct regularization on model behavior in the current implementation comes from the sparse KAN parameterization and the ℓ_1 penalty below, rather than from a separate Shapley-based constraint term. The loss function $\mathcal{L}(\theta)$ incorporates an ℓ_1 regularization term on the KAN spline coefficients $\mathbf{w}_{spline}$ to enforce sparsity and enhance structural interpretability:

$$\mathcal{L}(\theta) = \frac{1}{2} (R_{real}(\mathbf{z}_{exec}) - \hat{U}_{m,c}(\mathbf{x}; \theta))^2 + \lambda_{reg} \sum_{l=1}^L \|\mathbf{w}_{spline}^{(l)}\|_1. \quad (24)$$

In this loss, θ collects all trainable KAN parameters, λ_{reg} is the sparsity regularization coefficient, L is the number of KAN layers, and $\mathbf{w}_{spline}^{(l)}$ is the vector of spline coefficients in layer l .

Finally, the KAN parameters θ are updated via stochastic gradient descent to align the dispatch logic with environmental feedback. Simultaneously, we perform sensitivity analysis to determine the robustness boundary of the current policy. The parameter update and the local sensitivity are

$$\theta_{t+1} \leftarrow \theta_t - \eta \nabla_{\theta} \mathcal{L}(\theta_t), \quad (25)$$

$$S_{sens}(x_j) = \left. \frac{\partial \Psi(\mathbf{x})}{\partial x_j} \right|_{\mathbf{x}} \approx \sum_q \frac{\partial \Psi}{\partial \varphi_q} \frac{\partial \varphi_q}{\partial x_j}. \quad (26)$$

Here, η is the learning rate, $S_{sens}(x_j)$ is the local sensitivity score of feature x_j , and φ_q denotes the q -th intermediate univariate KAN response used in the chain-rule approximation, avoiding confusion with the Shapley value ϕ_j .

V. EXPERIMENTS AND PERFORMANCE EVALUATION

This section presents a comprehensive empirical evaluation from multiple perspectives. We first describe the experimental platform, environment configuration, dataset preprocessing, and implementation details needed for reproducibility. We then report the parameter settings, evaluation metrics, and baseline algorithm implementations. After that, we present the ablation study and the dynamic scalability analysis, followed by a discussion on generalization and scalability. Finally, we provide the case study.

A. Experimental Platform and Environment Configuration

To jointly simulate microscopic traffic dynamics and macroscopic energy scheduling, we construct a closed-loop evaluation platform. The platform integrates the SUMO traffic simulator with the PyTorch deep learning framework. The interaction between the proposed XKSDN decision module and the SUMO environment is realized through the Traffic Control Interface (TraCI) protocol. This interface allows Python scripts to perform fine-grained control over simulation entities at each simulation step Δt . The supported operations include:

- 1) **State retrieval:** Real-time acquisition of global vehicle states, including position coordinates (x, y) , instantaneous speed v , lane ID, and the average network-level travel speed v_{avg} .
- 2) **Command injection:** The scheduling decisions computed by XKSDN are translated into explicit driving routes, which are dynamically applied to MCVs via the `traci.vehicle.setRoute` interface.
- 3) **Charging emulation:** Battery energy consumption and replenishment processes are modeled using the `traci.vehicle.getParameter` and `traci.vehicle.setParameter` interfaces, enabling explicit simulation of SOC.

For road network topology construction, Midtown Manhattan, New York City, is selected as the test area. We extract map data covering this region (approximately 4 km $\times$ 4 km) from OpenStreetMap (OSM). We then convert the data into SUMO-compatible `.net.xml` files using the built-in `netconvert` tool. The converted road network preserves real-world traffic regulations. It includes one-way street constraints, turning restrictions, lane configurations, and traffic signal phase settings.

B. Dataset Preprocessing

We model EV demand using publicly available trip records released by the New York City Taxi and Limousine Commission (TLC), which is the NYC Taxi Trip Record Data (2016 Yellow Taxi Trip Data). Each row corresponds to a single trip and includes pickup/dropoff timestamps, taxi zone

locations, trip distance, fare breakdown, and passenger count, which allows us to reconstruct OD pairs and departure times. We remove trips whose origins or destinations fall outside the Manhattan experimental region. We select data from June 2016. We further divide the data into weekday and weekend subsets to evaluate the adaptability of XKSDN under different tidal traffic patterns. In addition, we filter out trips shorter than 200 meters. We also remove trips with average speeds exceeding 80 km/h, as they are likely erroneous records.

C. Implementation and Reproducibility Details

All experiments are conducted on the TaiWan Computing Cloud (TWCC) using instance type `cm.super`, with 60 GB RAM and 30 GB shared GPU memory. The software environment is `python-25.11-py3`. The traffic simulator is Eclipse SUMO v1.25.0, controlled via TraCI, with step length $\Delta t = 1$ s. The learning framework is PyTorch v2.10. Each reported result is averaged over 50 independent runs with different random seeds (demand sampling, initial SOC, and background traffic generation).

For the proposed XKSDN, the KAN is trained using Adam with learning rate 10^{-3} , batch size 256, and a maximum of 200 epochs. Early stopping is applied on a held-out validation subset with patience 10 to prevent overfitting. Unless otherwise stated, we use weight decay 10^{-5} and gradient clipping with max norm 1.0. For fair comparison, all baselines share the same demand streams, road network, and TOU pricing function.

Since the original taxi dataset reflects trips made by internal combustion engine vehicles, we employ a demand synthesis process. This process converts taxi trips into EV charging demands. It includes energy consumption modeling, initial SOC distribution, and charging request triggering. At the beginning of the simulation, each EV is assigned an initial SOC drawn from a truncated normal distribution $\mathcal{N}(0.6, 0.2)$ bounded within $[0.2, 0.9]$. A charging request is triggered once the SOC drops below a dynamically adjusted psychological threshold.

In addition to charging demands, we incorporate background traffic flow and power grid pricing data into the environment state input of the KAN network. On the transportation side, we derive origin–destination (OD) matrices from NYC taxi data. We use these matrices with SUMO's `duarouter` tool to generate background traffic flows. The volume of background vehicles is set between 2,000 and 5,000 per hour. This setting emulates peak and off-peak traffic conditions. On the energy side, we introduce a time-of-use (TOU) electricity pricing scheme. We incorporate the electricity price function $C_{grid}(t)$ into the scheduling objective as a penalty term. Although the present benchmark is built on NYC taxi traces, the core state variables, clustering metric, and feasibility constraints are defined on transferable physical quantities rather than city-specific identifiers; Appendix J of the Supplementary Material further provides additional cross-environment transfer tests on a polycentric grid-city and a radial-ring arterial network to examine generalization beyond Manhattan.

TABLE II
THE PARAMETER SETTINGS OF THE EV AND MCV

Category	Parameter Name	Configured Value
EV	Battery Capacity (C_{EV})	60 kWh
EV	Maximum Charging Power	50 kW
EV	Average Energy Consumption Rate	0.15–0.20 kWh/km
MCV	Charging Anxiety Threshold	20%–30%
MCV	Battery Capacity (C_{MCV})	300–500 kWh
MCV	Maximum Output Power	120 kW
MCV	Self Energy Consumption Rate	0.8–1.2 kWh/km
MCV	Service Radius	2 km
MCV	Recharging Threshold	15%

TABLE III
THE PARAMETER SETTINGS OF THE PROPOSED XKSDN

Module	Parameter Name	Configured Value
KAN Network	Grid Size (G)	5
KAN Network	Spline Order (k)	3
KAN Network	Number of Network Layers	3
KAN Network	Optimizer	Adam
KAN Network	Regularization Weight (λ_{reg})	0.01
Scheduling Layer	Virtual Node Penalty ($\lambda_{penalty}$)	1000
Scheduling Layer	Clustering Density Threshold (d_c)	Adaptive
Scheduling Layer	Temporal Weight (ω_t)	0.4
Scheduling Layer	Spatial Weight (ω_s)	0.4
Scheduling Layer	Energy Weight (ω_e)	0.2

D. Parameter Settings and Evaluation Metrics

The experiment includes two types of vehicle entities. EVs serve as the service recipients. MCVs serve as the service providers. The key parameters are summarized in Table II.

Because the EV/MCV settings in Table II are scenario assumptions rather than exact deterministic constants, we do not interpret the reported gains as being tied to one single parameter point. Instead, the main-text experiments already include uncertainty-oriented evaluation through stochastic demand resampling, varying request density and fleet size, capacity-scaling analysis, and trajectory-prediction perturbations in Figs. 8–12. These tests, together with the supplementary robustness analyses, indicate that the relative advantage of XKSDN remains stable under moderate uncertainty around charging demand, SOC thresholds, and vehicle-energy characteristics.

The hyperparameters of the proposed XKSDN model are optimized using a grid search approach. The detailed settings are summarized in Table III.

Implementation note: Before GEM, we min–max normalize $\hat{U}_{m,c}$ (and thus the base costs $-\hat{U}_{m,c}$) to a bounded range to keep all matching costs comparable across scenarios. The penalty $\lambda_{penalty}$ is then chosen to be larger than typical assignment costs so that matching to dummy nodes occurs only when serving a cluster is not beneficial or not possible. Similarly, the Big- M constant M_{inf} in Eq. (17) is set sufficiently large to forbid infeasible assignments while avoiding unnecessary numerical overflow.

A multi-dimensional evaluation framework is adopted to assess the model from four perspectives: service quality, operational efficiency, computational performance, and interpretability.

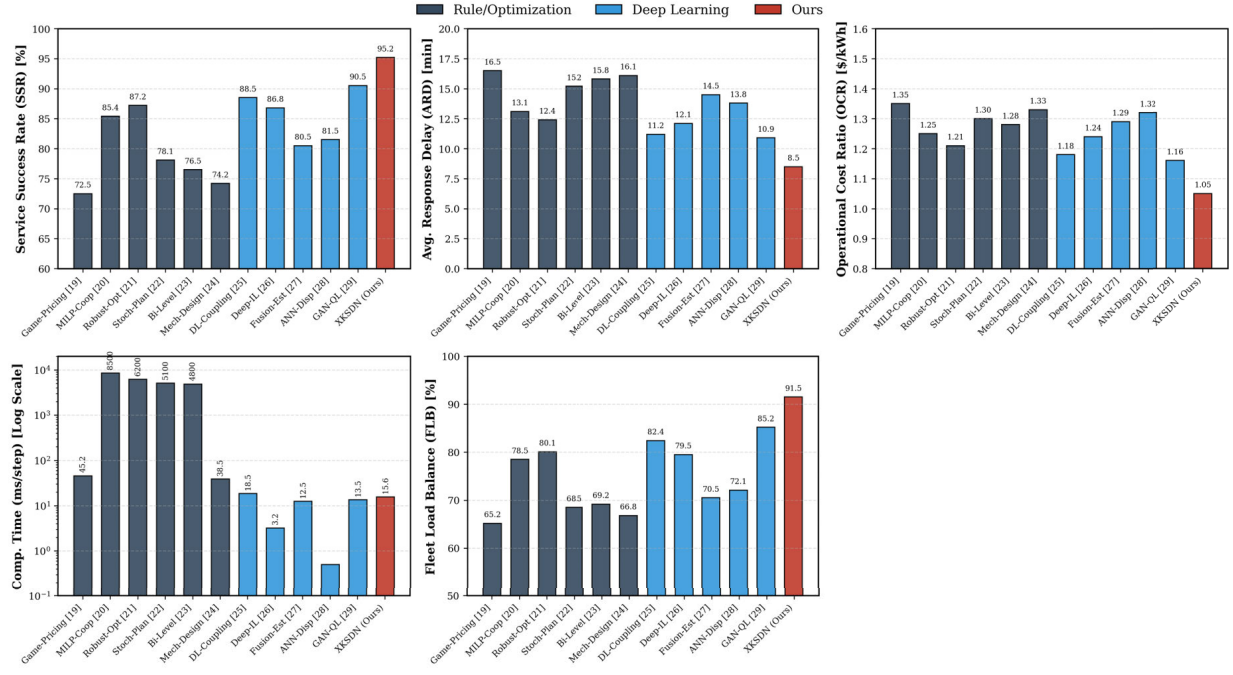


Fig. 7. Performance comparison of XKSDN and baseline methods.

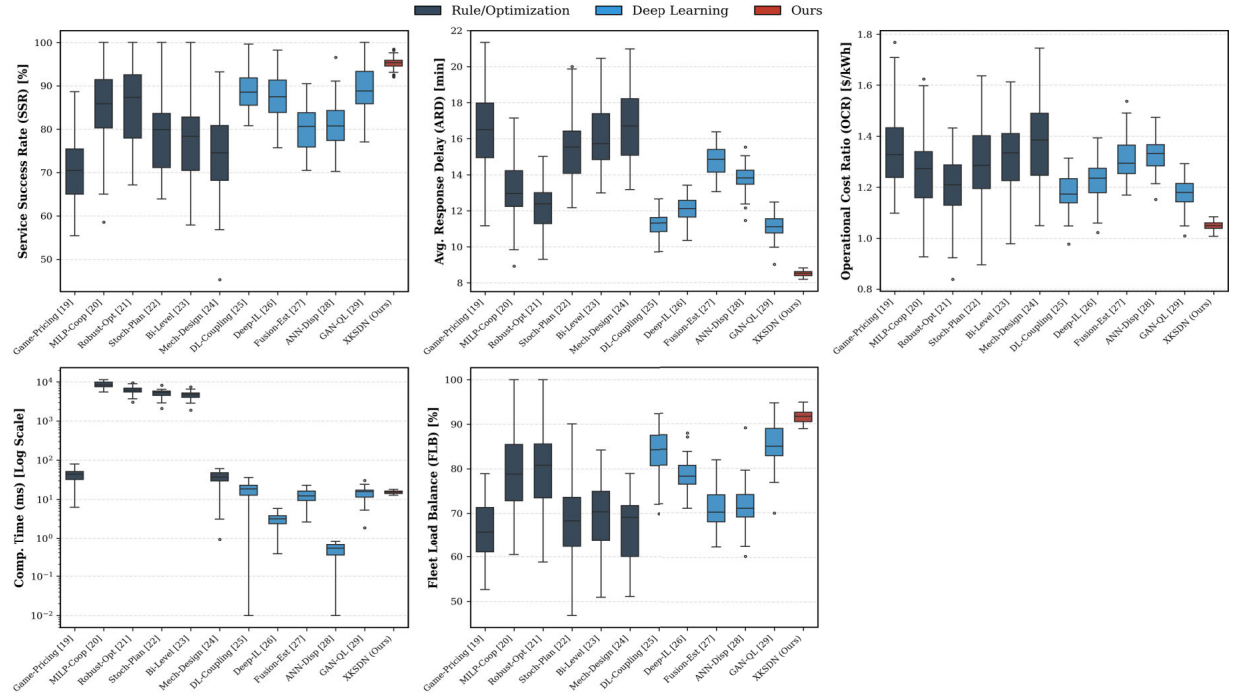


Fig. 8. Statistical distribution of key performance metrics over 50 independent simulation runs.

1) Service Quality: Service Success Rate (SSR):

SSR measures the proportion of charging requests that are successfully served within the user-specified time window $[a_i, b_i]$:

$$SSR = \frac{\sum_{i \in \mathcal{R}} y_i}{|\mathcal{R}|} \times 100\%. \quad (27)$$

Average Response Delay (ARD):

ARD is defined as the mean time difference between the moment when a charging request is issued ($t_{req,i}$) and the time

when the MCV arrives at the service location ($t_{arr,i}$):

$$ARD = \frac{1}{\sum_{i \in \mathcal{R}} y_i} \sum_{i \in \mathcal{R}} y_i \cdot (t_{arr,i} - t_{req,i}). \quad (28)$$

2) Operational Efficiency: Operational Cost Ratio (OCR):

The total cost is computed by jointly considering the driving energy consumption cost of MCVs and the electricity

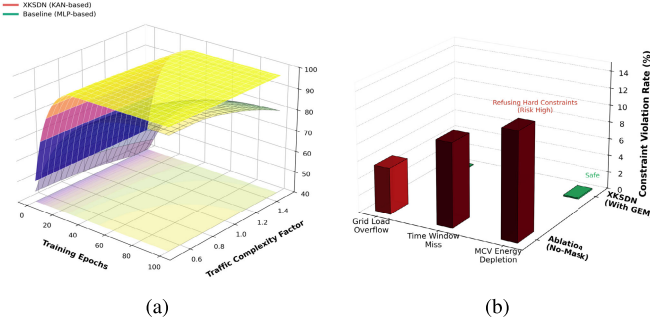


Fig. 9. The Ablation study in the proposed XKSDN. (a) 3D training landscape comparing the convergence and robustness of KAN-based vs. MLP-based networks under varying traffic complexities. (b) Safety verification showing the impact of the Constraint-Aware Gating Mechanism (GEM) on violation rates across different hard constraints.

procurement cost from the power grid:

$$\text{OCR} = \frac{\sum_m (C_{\text{travel}} \cdot E_{\text{travel},m}) + \sum_k (C_{\text{grid}}(t) \cdot E_{\text{charge},k})}{\sum E_{\text{delivered}}} \quad (29)$$

Here, C_{travel} is the unit travel-energy cost of an MCV, $E_{\text{travel},m}$ is the travel energy consumed by MCV m , $E_{\text{charge},k}$ is the charged energy procured from the grid during charging event k , and $\sum E_{\text{delivered}}$ is the total delivered charging energy used for normalization.

Fleet Load Balance (FLB): The Gini coefficient is employed to quantify the disparity in workload distribution among different MCVs.

3) *Explainability Metrics: Fidelity:* Features with the highest Shapley values are masked, and the resulting degradation in the model's output utility is examined:

$$\text{Fidelity} = \mathbb{E}_{\mathbf{x}} [\|\Psi(\mathbf{x}) - \Psi(\mathbf{x}_{\text{top-}k})\|]. \quad (30)$$

Here, $\mathbf{x}_{\text{top-}k}$ denotes the perturbed state obtained after masking the Top- k features ranked by Shapley value. **Consistency:**

Let $\mathcal{N}_{\epsilon}(\mathbf{x}_i) = \{\mathbf{x}_j : \|\mathbf{x}_i - \mathbf{x}_j\|_2 < \epsilon\}$ be the ϵ -neighborhood of state $\mathbf{x}_i$. We quantify explanation stability using the expected cosine similarity between Shapley vectors of neighboring states:

$$\text{Consistency} = \mathbb{E}_i \mathbb{E}_{j \in \mathcal{N}_{\epsilon}(\mathbf{x}_i)} \left[\frac{\phi(\mathbf{x}_i)^{\top} \phi(\mathbf{x}_j)}{\|\phi(\mathbf{x}_i)\|_2 \|\phi(\mathbf{x}_j)\|_2} \right]. \quad (31)$$

The vector $\phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \dots, \phi_D(\mathbf{x})]^{\top}$ collects all local Shapley attributions, and ϵ controls the radius of the local neighborhood. In our experiments, $\mathcal{N}_{\epsilon}(\mathbf{x}_i)$ is sampled by perturbing traffic and price features within a small range while keeping the OD pair fixed.

Action-level Faithfulness (ALF): To evaluate whether explanations are truly decision-relevant for scheduling, we measure how sensitive the final assignment is to the top- k features identified by Shapley. Let $\mathbf{X}$ be the assignment obtained from the original state, and $\mathbf{X}^{\text{top-}k}$ be the assignment after masking/removing the Shapley Top- k features and resolving GEM+Hungarian. We define the assignment change ratio as

$$\text{ALF} = \mathbb{E} \left[\frac{\|\mathbf{X} - \mathbf{X}^{\text{top-}k}\|_0}{\|\mathbf{X}\|_0} \right], \quad (32)$$

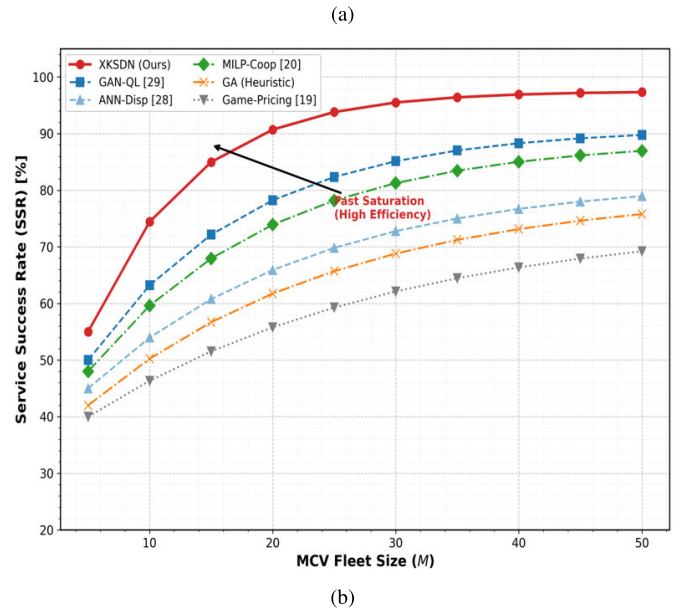
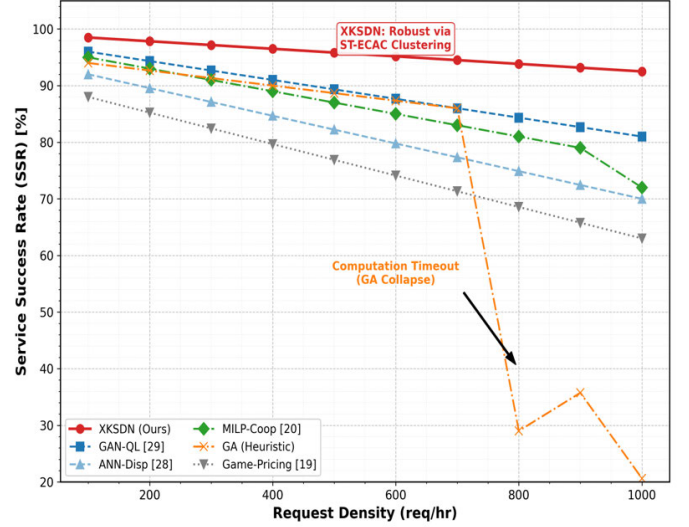


Fig. 10. Dynamic scalability analysis of the proposed XKSDN against state-of-the-art baselines. (a) Impact of Request Density (100–1000 req/hr) on Service Success Rate (SSR). XKSDN maintains robustness under extreme loads, whereas heuristic methods (e.g., GA) suffer from performance collapse due to computational timeouts. (b) Impact of MCV Fleet Size ($M = 5$ to 50) on SSR. XKSDN demonstrates superior resource efficiency, achieving high service saturation with a significantly smaller fleet compared to Deep Learning (GAN-QL [29]) and Optimization (MILP-Coop [20]) benchmarks.

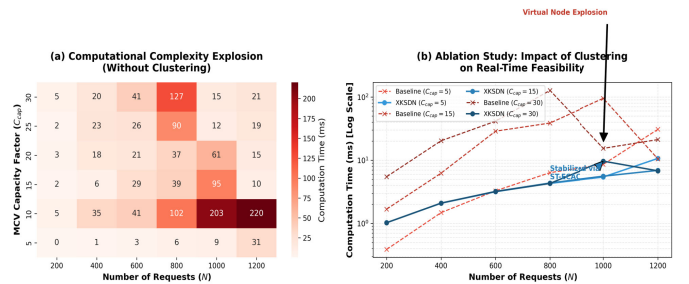


Fig. 11. Sensitivity analysis of computational complexity regarding MCV capacity and request scale.

where $\|\cdot\|_0$ counts the number of changed matched pairs. We also report the corresponding degradation in task-level

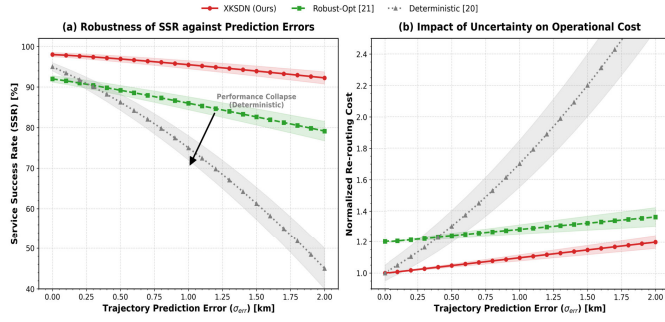


Fig. 12. Robustness evaluation under trajectory prediction errors and the effectiveness of the real-time receding horizon control (RHC) mechanism.

performance (e.g., Δ SSR) after masking Top- k features, linking explanations to operational outcomes.

E. Baseline Algorithm Implementation

The superior performance of the proposed XKSDN observed in Fig. 7 can be primarily attributed to its structured learning paradigm. XKSDN adopts the Kolmogorov–Arnold Network (KAN) as the core function approximator. It represents dispatching utility through compositions of low-dimensional functions. This design enables the model to explicitly capture spatiotemporal correlations and energy-state couplings in high-dimensional system states. It directly contributes to higher SSR and lower ARD. Compared to opaque deep neural networks [25], [26], [27], [28], [29], the decomposable structure of KAN reduces overfitting. It also improves generalization under demand fluctuations. As a result, XKSDN produces more reliable and timely dispatch decisions.

Beyond utility learning, XKSDN turns utilities into executable actions via constraint-aware bipartite matching (GEM + Hungarian), where infeasible pairs are explicitly masked and dummy/replicated nodes enable partial matching and capacity modeling. This layer contributes to lower OCR and improved FLB while keeping the solver tractable [16].

Fig. 8 presents the statistical distributions of key performance metrics over 50 independent simulation runs, providing a robustness and stability analysis of the different methods. Compared with rule-based and optimization approaches [19], [20], [21], [22], [23], [24] as well as conventional deep learning baselines [25], [26], [27], [28], [29], the proposed XKSDN exhibits markedly tighter interquartile ranges and fewer extreme outliers across all metrics. This indicates that the proposed method not only achieves strong average performance but also maintains high stability under stochastic demand realizations and randomized initial conditions. For quantitative reporting, we compute 95% confidence intervals (CIs) over the 50 runs using bootstrap resampling, and we apply paired significance testing (e.g., paired t -tests with $p < 0.05$) on key metrics to verify that the observed improvements are statistically significant.

For service-related metrics, including SSR and ARD, the proposed XKSDN shows both superior central tendencies and significantly reduced variance. In the SSR boxplot, XKSDN is centered around 95%, with a very compact interquartile range

TABLE IV
COMPARISON WITH INTERPRETABLE MACHINE LEARNING METHODS (BASELINES FROM [28], [29], [33])

Metrics	XKSDN (KAN+Shapley)	GAN-QL [29] + LIME [33]	ANN-Disp [28] + Shapley
Fidelity	0.88	0.42	0.75
Consistency	0.92	0.55	0.81

and short whiskers, whereas several rule-based or optimization baselines exhibit much wider spreads and occasional low-end failures, indicating higher sensitivity to stochastic requests and traffic perturbations. In the ARD boxplot, XKSDN also achieves the lowest median, approximately 8.5 min, with almost no outliers. This indicates that the proposed dispatch policy not only serves more requests successfully but also maintains consistently fast responses across independent runs.

A similar pattern can be observed for OCR, computation time, and FLB. XKSDN attains the lowest OCR median, about 1.05 \$/kWh, and the highest FLB median, around 91%, implying that the proposed method reduces operating cost while preserving a more balanced workload among MCVs. Moreover, although the computation-time subplot is presented on a logarithmic scale, XKSDN remains concentrated in the low tens-of-milliseconds range, whereas several optimization baselines fluctuate by orders of magnitude. This result is important for online deployment because it demonstrates that the proposed matching layer provides predictable latency together with stable operational quality.

Table IV compares the proposed XKSDN (KAN + Shapley) with representative interpretable machine learning baselines, including GAN-QL [29] + LIME [33] and ANN-Disp [28] + Shapley, under two quantitative interpretability metrics: fidelity and consistency. The results show that XKSDN achieves the highest scores on both metrics, indicating superior explanation quality and stability.

In terms of fidelity, the proposed XKSDN attains a value of 0.88, significantly outperforming GAN-QL [29] + LIME [33] (0.42) and also exceeding ANN-Disp [28] + Shapley (0.75). This demonstrates that the explanations generated by XKSDN more accurately reflect the true decision logic of the underlying model.

Regarding consistency, the proposed XKSDN again exhibits the strongest performance, with a score of 0.92. This indicates that for similar system states, XKSDN produces highly stable and coherent feature attributions.

F. Ablation Study

Based on the ablation study results presented in Fig. 9, we provide an in-depth analysis of the architectural advantages of XKSDN across three dimensions: convergence efficiency, robustness, and safety.

First, the training landscape in Fig. 9(a) intuitively confirms the significant advantage of the KAN architecture in feature extraction efficiency. By comparing the yellow surface (XKSDN) with the underlying blue mesh surface (MLP Baseline), it is evident that in the early training stages (Training Epochs < 20), the performance curve of XKSDN exhibits

an extremely steep ascending trajectory, rapidly converging to a high-accuracy region. This is attributed to the learnable activation functions inherent in the KAN network, which enable more efficient capture of non-linear spatio-temporal features compared to traditional MLPs, thereby significantly shortening the model training cycle.

Second, Fig. 9(a) reveals the superior generalization capability and robustness of XKSDN when facing highly dynamic environments. As the “Traffic Complexity Factor” increases from 0.6 to 1.4, the performance surface of the MLP baseline exhibits a distinct “sagging” phenomenon, indicating its difficulty in adapting to scenarios with high congestion or drastic demand fluctuations. In contrast, the XKSDN surface maintains a stable high-level plateau, remaining almost unaffected by the increased environmental complexity. This demonstrates that XKSDN can effectively cope with complex traffic flow variations in the real world, maintaining system scheduling stability.

Finally, the 3D bar chart in Fig. 9(b) profoundly illustrates the decisive role of the GEM in ensuring system safety. In the ablation variant where the masking mechanism is removed, the system relies solely on soft reward guidance, resulting in a surge of violation rates for MCV Energy Depletion and Time Window Miss to over 10%, representing an unacceptable risk for actual operations.

G. Dynamic Scalability Analysis

Fig. 10(a) illustrates the system’s robustness against surging demand. As the request density increases from 100 to 1000 req/hr, the SSR of all baselines exhibits a downward trend. However, the proposed XKSDN (red curve) displays the most graceful degradation, maintaining an SSR above 90% even at peak loads. This stability is attributed to the ST-ECAC clustering mechanism, which effectively compresses massive concurrent requests into manageable demand clusters, preventing the dimensionality explosion that cripples traditional methods. In stark contrast, the Genetic Algorithm (GA) experiences a catastrophic “performance collapse” at 700 req/hr (indicated by the sharp drop), as the exponential growth in search space leads to computation timeouts and invalid solutions.

Furthermore, Fig. 10(b) evaluates the marginal utility of fleet resources. The proposed XKSDN demonstrates the fastest convergence rate, reaching near-optimal performance with a fleet size of only $M = 25$. To achieve a comparable service level, the runner-up method requires at least 40 vehicles, while rule-based approaches (Game-Pricing [19]) fail to reach saturation even with $M = 50$. This indicates that XKSDN optimizes the spatio-temporal distribution of MCVs more effectively, allowing operators to reduce capital expenditure on vehicles without compromising service quality.

To address concerns regarding the potential “dimensionality explosion” in graph expansion methods—specifically when MCV battery capacity (C_{cap}) increases—we conducted a rigorous sensitivity analysis. Fig. 11(a) validates the limitation of the traditional graph expansion approach. As illustrated in the heatmap, the baseline method (without clustering) suffers from a significant surge in computational overhead. When

the request scale (N) exceeds 800 and C_{cap} reaches 30, the computation time enters the “danger zone” (indicated by dark red blocks), rendering real-time scheduling infeasible. This confirms that a naive expansion of virtual nodes leads to cubic complexity growth ($O(V^3)$). However, Fig. 11(b) demonstrates how the proposed XKSDN framework overcomes this bottleneck via the ST-ECAC clustering module. By aggregating individual requests into demand clusters, XKSDN decouples the matching complexity from the raw number of requests and MCV capacity. The results show that even under the most demanding conditions ($N = 1200, C_{cap} = 30$), XKSDN maintains a computation latency below 10 ms (solid blue lines). In sharp contrast, the baseline’s latency exceeds 100 ms (dashed red lines), exhibiting an exponential trend. This ablation study proves that the clustering layer is not merely an optimization but a structural necessity for ensuring scalability in large-capacity fleet operations.

To relax the idealized assumption that EV trajectories are known *a priori* or can be perfectly predicted by navigation systems, and to evaluate system performance under realistic stochastic environments, Gaussian noise is injected into the ground-truth trajectories to simulate prediction errors ranging from 0 to 2.0 km.

Fig. 12(a) reveals the fragility of deterministic approaches (e.g., MILP [20]). As σ_{err} exceeds 0.5 km, their preset rendezvous points become invalid, causing the SSR to plummet (performance collapse). Although Robust-Opt [21] mitigates this drop, it comes at the expense of higher initial operational costs due to its conservative “worst-case” planning. In contrast, XKSDN (red curve) demonstrates superior resilience. Even with a significant deviation of 2.0 km, the SSR remains above 90%. This robustness is attributed to the real-time receding horizon control (RHC) mechanism inherent in our DRL agent, which continuously re-optimizes dispatch orders based on the latest EV positions, effectively correcting prediction errors on the fly without incurring excessive re-routing costs (as shown in Fig. 12(b)). Due to space limitations, more detailed ablation experiments are provided in the appendix, including boundary condition analysis and hyperparameter studies (e.g., mesh size).

H. Discussion on Generalization and Scalability

From the generalization perspective, XKSDN is built on transferable physical variables, including travel time, travel distance, SOC urgency, congestion, energy demand, and electricity price, rather than on Manhattan-specific identifiers. This means that the KAN utility learner and the GEM dispatch layer can be migrated to other cities after re-normalizing local state ranges and re-calibrating only a small set of operating thresholds. The additional cross-environment experiments in Appendix J, which include a polycentric grid-city and a radial-ring arterial network, further show that XKSDN preserves the best overall SSR/ARD/OCR trend among the compared methods after such lightweight calibration. Therefore, the proposed framework should be interpreted as transferable across urban morphologies, although not as a completely zero-shot model.

From the scalability perspective, the main practical bottleneck lies in the size of the many-to-many assignment rather

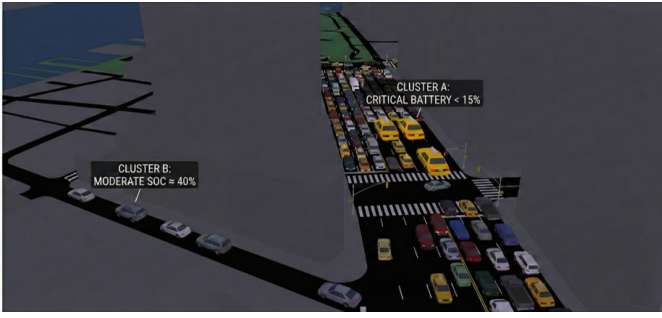


Fig. 13. A simple example of the interpretable case study.

than in the forward evaluation of the KAN itself. ST-ECAC reduces the raw request stream to a bounded set of demand clusters, after which utility scoring scales approximately with the number of candidate pairs and the GEM-enhanced matching stage remains tractable for online dispatch. The main-text results in Figs. 10–12, together with the larger stress test in Appendix I, show that all methods degrade under heavy loads, but XKSDN degrades more gracefully because clustering suppresses graph explosion and GEM avoids repeated infeasible search. This dedicated discussion clarifies the practical relevance of the framework: it offers a deployable middle ground between opaque end-to-end policies and slow full optimization, while still motivating future distributed implementations for ultra-large fleets.

I. Case Study

To intuitively demonstrate the decision-making logic of XKSDN, we present a micro-level analysis of a specific scheduling snapshot. As shown in Fig. 13, the scenario is set at 17:30 on a Friday (coinciding with both the evening rush hour and peak electricity pricing), where two competing charging demand clusters emerge along 42nd Street in Manhattan.

Cluster A: Located in the core congestion zone, consisting of three taxis with critical battery levels ($SOC < 15\%$). **Cluster B:** Located on the uncongested periphery, comprising five private vehicles with moderate battery levels ($SOC \approx 40\%$).

In this safety-critical snapshot, many baselines dispatch to Cluster B because congestion and cost dominate their objectives, while Cluster A’s breakdown risk is under-modeled. In contrast, XKSDN prioritizes Cluster A by capturing a nonlinear SOC-urgency transition in KAN utility and by enforcing hard feasibility through GEM.

In stark contrast, the proposed XKSDN is the only method that correctly dispatches the MCV to Cluster A. As shown in Fig. 14, the Shapley value analysis provides a transparent explanation for this decision. The KAN-based utility model captures a strong nonlinear transition in SOC urgency, producing a dominant positive contribution once SOC falls below 20%. Although traffic congestion imposes a negative penalty consistent with baseline judgments, this cost is outweighed by survival-critical urgency. Meanwhile, the smaller request volume in Cluster A contributes positively but modestly, insufficient to overturn the decision. These attribution results demonstrate that XKSDN’s decision is not accidental, but causally grounded in the most critical system factors.

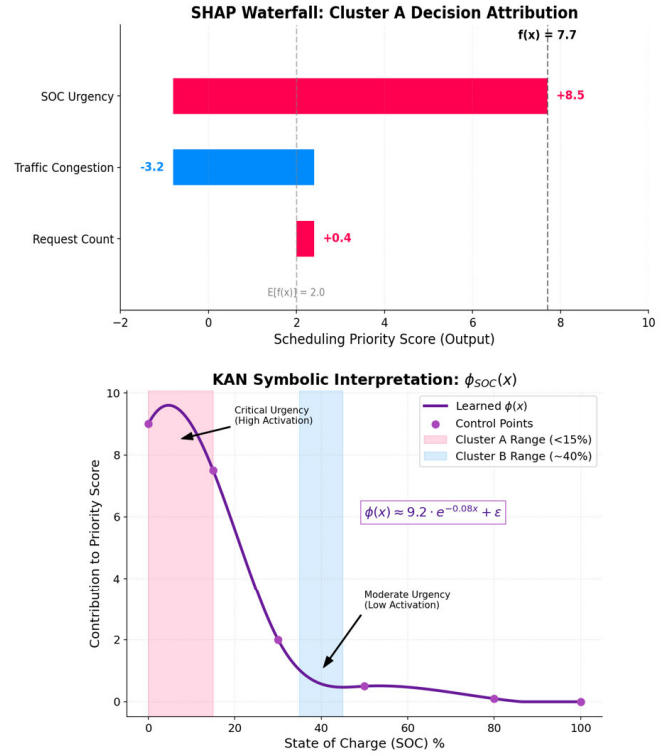


Fig. 14. The interpretable analysis by the proposed XKSDN.

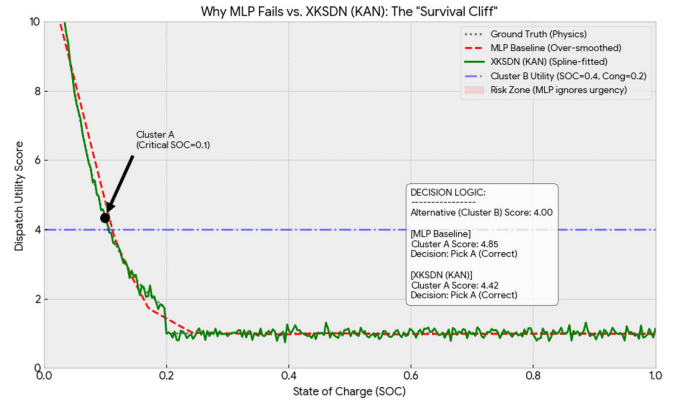


Fig. 15. Performance comparison between the MLP strong baseline (with masking) and XKSDN (KAN) in the “survival cliff” scenario.

Moreover, GEM serves as a final safety barrier by masking infeasible or unsafe dispatch options. Fig. 15 further shows that, even with action masking, an MLP baseline may underestimate utilities in the low-SOC “risk zone”, while KAN better captures the sharp nonlinear transition.

VI. CONCLUSION

This paper proposed an Explainable KAN-Shapley Dispatch Network (XKSDN) for many-to-many mobile charging service dispatch under dynamic traffic, energy, and hard feasibility constraints. By combining ST-ECAC demand abstraction, KAN-based utility learning, GEM-enhanced Hungarian matching, and local Shapley plus causal analysis, the framework

jointly improves dispatch quality, operational safety, and decision transparency.

Using real New York City taxi traces and SUMO simulations, XKSDN achieves a median SSR of about 95%, a median ARD of about 8.5 min, and a median OCR of about 1.05 \$/kWh, while maintaining an FLB of about 91% and low-tens-of-milliseconds online inference time. Relative to representative strong baselines, these results correspond to roughly a 5–10 percentage-point SSR improvement, about 2 min less response delay, and approximately 10%–20% operating-cost savings, with statistically more stable behavior across 50 independent runs.

At the same time, the proposed method still has several limitations. Its centralized pipeline can degrade under very large request volumes; its performance remains sensitive to clustering quality, state-estimation accuracy, and communication latency; and severe extreme events, such as incident-driven road blockages with abrupt local demand surges, can still push the system away from its nominal operating region. In addition, cross-city transferability, long-horizon field deployment, and closed-loop interaction with imperfect sensing infrastructure remain to be further validated. Therefore, recommended future work includes distributed or hierarchical matching for ultra-large fleets, delay-compensated and noise-aware state estimation, event-aware rerouting under rare disruptions, and broader real-world trials that jointly model user behavior, grid uncertainty, sensing noise, and communication delay.

REFERENCES

- [1] Y. Su, J. Teh, and C. Chen, "Optimal dispatching for AC/DC hybrid distribution systems with electric vehicles: Application of cloud-edge-device cooperation," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 3, pp. 3128–3139, Mar. 2024.
- [2] S. Belakaria, M. Ammous, S. Sorour, and A. Abdel-Rahim, "Fog-based multi-class dispatching and charging for autonomous electric mobility on-demand," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 762–776, Feb. 2020.
- [3] Y. Lai, Z. Lv, K.-C. Li, and M. Liao, "Urban traffic Coulomb's law: A new approach for taxi route recommendation," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 3024–3037, Aug. 2019.
- [4] M. Li, Y. Wang, P. Peng, and Z. Chen, "Toward efficient smart management: A review of modeling and optimization approaches in electric vehicle-transportation network-grid integration," *Green Energy Intell. Transp.*, vol. 3, no. 6, Dec. 2024, Art. no. 100181.
- [5] Z. Zhang, H. Hui, and Y. Song, "Response capacity allocation of air conditioners for peak-valley regulation considering interaction with surrounding microclimate," *IEEE Trans. Smart Grid*, vol. 16, no. 2, pp. 1155–1167, Mar. 2025.
- [6] D. Harris, F. L. Da Silva, W. Su, and R. Glatt, "A review on simulation platforms for agent-based modeling in electrified transportation," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 2, pp. 1131–1147, Feb. 2024.
- [7] A. Pamidimukkala, S. Kermanshachi, J. M. Rosenberger, and G. Hladik, "Barriers and motivators to the adoption of electric vehicles: A global review," *Green Energy Intell. Transp.*, vol. 3, no. 2, Apr. 2024, Art. no. 100153.
- [8] X. Li et al., "Electric vehicle behavior modeling and applications in vehicle-grid integration: An overview," *Energy*, vol. 268, Apr. 2023, Art. no. 126647.
- [9] T. Yang, Q. Guo, L. Xu, and H. Sun, "Dynamic pricing for integrated energy-traffic systems from a cyber-physical-human perspective," *Renew. Sustain. Energy Rev.*, vol. 136, Feb. 2021, Art. no. 110419.
- [10] X. Sun, Z. Chen, and Y. Yin, "Integrated planning of static and dynamic charging infrastructure for electric vehicles," *Transp. Res. D, Transp. Environ.*, vol. 83, Jun. 2020, Art. no. 102331.
- [11] J. Cao, Y. Han, N. Pan, J. Zhang, and J. Yang, "A data-driven approach to urban charging facility expansion based on bi-level optimization: A case study in a Chinese city," *Energy*, vol. 300, Aug. 2024, Art. no. 131529.
- [12] Y. Wang, S. Gan, Y. Zhao, Y. Lin, and K. Li, "An integrated multi-regional cooperative strategy for real-time electric taxi dispatching and charging," *Control Eng. Pract.*, vol. 164, Nov. 2025, Art. no. 106492.
- [13] Z. Lai and S. Li, "Towards a multimodal charging network: Joint planning of charging stations and battery swapping stations for electrified ride-hailing fleets," *Transp. Res. B, Methodol.*, vol. 183, May 2024, Art. no. 102928.
- [14] S. An, J. Qiu, J. Lin, Z. Yao, Q. Liang, and X. Lu, "Planning of a multi-agent mobile robot-based adaptive charging network for enhancing power system resilience under extreme conditions," *Appl. Energy*, vol. 395, Oct. 2025, Art. no. 126252.
- [15] M. Mahjoubnia, T. Niknam, A. Taghavi, and H. Heydari-Doostabad, "System-level integration of electric transportation in smart grids: An aggregator-centric review of control architectures, optimization algorithms, and business models," *eTransportation*, vol. 27, Jan. 2026, Art. no. 100542.
- [16] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, 1955.
- [17] X. Zhang, Z. Zhang, Y. Liu, Z. Xu, and X. Qu, "A review of machine learning approaches for electric vehicle energy consumption modelling in urban transportation," *Renew. Energy*, vol. 234, Nov. 2024, Art. no. 121243.
- [18] J. Xiao, S. Wang, J. Xiao, and G. Q. Huang, "Delivery routing for electric vehicles with en-route mobile battery swapping," *Transp. Res. E, Logistics Transp. Rev.*, vol. 192, Dec. 2024, Art. no. 103838.
- [19] S. Lai, Z. Yang Dong, J. Qiu, Y. Tao, J. Zhao, and G. Wang, "Competitive pricing strategy for the wireless charging lane operator considering range anxiety of electric vehicle users," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 2, pp. 2187–2201, Feb. 2025.
- [20] P. Li, M. Jiang, Y. Zhang, and Y. Zhang, "Cooperative optimization of bus service and charging schedules for a fast-charging battery electric bus network," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5362–5375, May 2023.
- [21] X. Peng, Y. Zhang, X. Hu, and W. Lu, "Integrated design of mobile battery-swapping and charging services for electric vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 27, no. 1, pp. 1525–1537, Jan. 2026.
- [22] J. Xu, J. Ma, C. Zhao, and Y. Li, "Dynamic expansion planning of charging stations with fixed and mobile chargers," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 12, pp. 21954–21970, Dec. 2025.
- [23] S. Zheng, Y. Liu, K. Fu, H. Xiao, and X. Lin, "Optimal of siting and pricing for multi-type charging facility considering electric vehicle and grid bidirectional power transfer," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 10, pp. 17371–17385, Oct. 2025.
- [24] A. Ghafelebashi, M. Razaviyayn, and M. Dessouky, "Incentive systems for fleets of new mobility services," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 2, pp. 1826–1839, Feb. 2025.
- [25] C. Pan et al., "Spatio-temporal EV task offloading, energy, and traffic management for 6G communication-power-transportation coupling network," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 10, pp. 18044–18057, Oct. 2025.
- [26] Z. Huang, J. Wang, X. Fan, R. Yue, C. Xiang, and S. Gao, "Economical electric vehicle charging scheduling via deep imitation learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 11, pp. 18196–18210, Nov. 2024.
- [27] M. Lin, S. Chen, J. Meng, W. Wang, and J. Wu, "Instantaneous energy consumption estimation for electric buses with a multi-model fusion method," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 1, pp. 371–381, Jan. 2025.
- [28] L. Hu and J. Dong, "An artificial-neural-network-based model for real-time dispatching of electric autonomous taxis," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1519–1528, Feb. 2022.
- [29] S. K. Das, S. P. Maity, and C. Chakraborty, "SG-GAN for electric vehicle road networks and Q-learning in energy efficient routing," *IEEE Trans. Intell. Transp. Syst.*, early access, Jul. 14, 2025, doi: [10.1109/TITS.2025.3585310](https://doi.org/10.1109/TITS.2025.3585310).
- [30] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 4765–4774.
- [31] G. Sariyer, S. K. Mangla, M. E. Sozen, G. Li, and Y. Kazancoglu, "Leveraging explainable artificial intelligence in understanding public transportation usage rates for sustainable development," *Omega*, vol. 127, Sep. 2024, Art. no. 103105.

- [32] J. Xie et al., “Is human-like decision making explainable? Towards an explainable artificial intelligence for autonomous vehicles,” *Transp. Res. Interdiscipl. Perspect.*, vol. 29, Jan. 2025, Art. no. 101278.
- [33] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why should I trust you?’: Explaining the predictions of any classifier,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144.
- [34] A. Rodriguez and A. Laio, “Clustering by fast search and find of density peaks,” *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
- [35] Z. Liu et al., “KAN: Kolmogorov–Arnold networks,” 2024, *arXiv:2404.19756*.
- [36] J. Pearl, *Causality: Models, Reasoning, and Inference*. Cambridge, U.K.: Cambridge Univ. Press, 2009.



Wen-Dong Jiang (Graduate Student Member, IEEE) received the B.S. degree in multimedia and game science development from the Lunghwa University of Science and Technology, Taoyuan, Taiwan, in 2021, and the M.S. degree in computer science and information engineering from Ming Chuan University, Taoyuan, in 2023, under the supervision of Prof. Yue-Shi Lee and Prof. Show-Jane Yen. He is currently pursuing the Ph.D. degree in computer science and information engineering with Tamkang University, New Taipei, Taiwan, under the supervision of Distinguished Prof. Chih-Yung Chang.

He has published several research articles in leading journals, including IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, IEEE TRANSACTIONS ON BIG DATA, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, IPM, and ESWA. His research interests include interpretable machine learning and its applications in smart cities.

Dr. Jiang was a recipient of the Best Paper Award at WASN 2024 and DLT 2024, as well as the Best Presentation Award at the ICCE-TW 2025 Special Session on Artificial Intelligence Applications and Technologies in the Internet of Things. He also serves as a reviewer for multiple international journals.



Chih-Yung Chang (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Taoyuan, Taiwan, in 1995.

He is currently a Distinguished Professor with the Department of Computer Science and Information Engineering and the Department of Artificial Intelligence, Tamkang University, New Taipei, Taiwan. His current research interests include artificial intelligence, deep learning and machine learning, the Internet of Things, and wireless sensor networks.

Dr. Chang is the Co-Chair of ACM SIGMOBILE, Taiwan. He has been serving as an Associate Guest Editor for several SCI-indexed journals, including *International Journal of Ad Hoc and Ubiquitous Computing* since 2011, *Journal of Applied Science and Engineering* since 2018, *International Journal of Distributed Sensor Networks* from 2012 to 2014, *IET Communications* in 2011, *Telecommunication Systems* in 2010, *Journal of Information Science and Engineering* in 2008, and *Journal of Internet Technology* from 2004 to 2008.



Tzu-Chia Huang is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. Her research interests include machine learning.



Yue-Shi Lee received the Ph.D. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1997.

He is currently a Professor with the Department of Computer Science and Information Engineering, Ming Chuan University, Taoyuan, Taiwan. His initial research interests were computational linguistics and Chinese language processing, and over time he evolved toward data warehousing, data mining, information retrieval and extraction, and the Internet technology.



Diptendu Sinha Roy (Senior Member, IEEE) received the Ph.D. degree in engineering from the Birla Institute of Technology, Mesra, India, in 2010. In 2016, he joined the Department of Computer Science and Engineering, National Institute of Technology (NIT) Meghalaya, Shillong, India, as an Associate Professor, where he has been working as the Chair of the Department of Computer Science and Engineering since January 2017. Prior to his stint at NIT Meghalaya, he worked with the Department of Computer Science and Engineering,

National Institute of Science and Technology, Berhampur, India. His current research interests include software reliability, distributed and cloud computing, and the Internet of Things (IoT), specifically applications of artificial intelligence/machine learning for smart integrated systems. He is a member of the IEEE Computer Society.