Decision Analytics Journal 18 (2026) 100691

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

An adaptive graph-based method for structured learning and decision analysis

Sovan Samanta [a,b,e,f,*], Tofigh Allahviranloo [a], Leo Mrsic [b,c], Antonios Kalampakas [d]

[a] Research Center of Performance and Productivity Analysis, Istinye University, 34320, Istanbul, Turkey
[b] Department of Technical Sciences, Algebra Bernays University, Gradiscanska 24, 10000, Zagreb, Croatia
[c] Rudolfovo Science and Technology Centre, Podbreznik 15, 8000, Novo Mesto, Slovenia
[d] College of Engineering and Technology, American University of the Middle East, 54200, Egaila, Kuwait
[e] Department of Mathematics, Tamralipta Mahavidyalaya, 721636, Tamluk, West Bengal, India
[f] Department of Technical Sciences, Western Caspian University, 1001, Baku, Azerbaijan

ARTICLE INFO

Keywords:
Machine learning
Knowledge engineering
Quantum graph structure
Parameter space
Federated learning
Analytical modeling

ABSTRACT

Many decision-support systems operate as networks of interacting units (e.g., hospital departments, transportation hubs, or organizational teams). In such systems, different units often use different data representations and model parameters, so direct comparison of local model states across units can be unreliable. We propose the *Parameter Learning Quantum Graph* (PLQG), a graph-based framework in which each node i has its own local parameter space $\mathcal{P}_i$ and each directed edge (i, j) carries a transport map $T_{ij} : \mathcal{P}_i \to \mathcal{P}_j$ that translates parameters (or decision states) into the representation used at the receiving node. PLQG defines a connection-energy regularizer that quantifies cross-unit inconsistency *after* translation and can be combined with standard local loss functions for network-wide learning and decision analysis. The framework supports time-varying graphs and explicitly distinguishes event-driven (discrete) and continuous interactions.

1. Introduction

In many real-world settings, decisions are not made in isolation. Hospitals coordinate care across departments, transportation systems manage delays across hubs, and large organizations combine recommendations from different analytical teams. In such environments, decisions interact and change over time, while data, assumptions, and analytic models often differ across units [1–3]. The main challenge is to support reliable system-level decisions when local decisions rely on heterogeneous information.

This challenge lies at the core of *decision analytics*. Decision analytics aims to support informed decision-making under uncertainty, with increasing emphasis on interpretability, transparency, and system-level coherence [1,2,4–6]. Classical decision-analytic graphical models include influence diagrams and Bayesian networks [7,8].

In networked settings, it is not enough to produce accurate local predictions; the local outputs must also remain consistent and meaningful when combined. Many existing approaches implicitly assume that model parameters or node features are directly comparable, which is often unrealistic in heterogeneous systems [9].

A local decision can be meaningful inside a unit and still be *non-comparable* across units. For example, two hospitals may train linear predictors for the same clinical outcome, but encode covariates differently (different scaling, coding, or missingness). Forcing the two parameter vectors to be equal would distort at least one model, while letting each hospital act independently can produce conflicting system-level decisions. What we need is *alignment after translation*: a downstream unit should compare its own state to an interpreted version of an upstream state. In PLQG, this translation is represented by an edge-specific transport map T_{ij}, and coherence is measured by the mismatch $\mathsf{d}_j(T_{ij}\theta_i, \theta_j)$ aggregated over edges (see Fig. 1).

To address heterogeneity directly, we adopt a network-based view of decision-making. Each decision unit is a node with its own local model, reflecting differences in data, expertise, constraints, or operating context. Connections between units represent information flow and interpretation across the system. This perspective builds on graph-based decision-support and learning approaches, while explicitly accounting for heterogeneity and uncertainty [10,11].

Within this setting, we propose the *Parameter Learning Quantum Graph* (PLQG). In a PLQG, each node has its own parameter/state space, and each edge carries a transport operator that formalizes how information should be translated from one node to another. This lets

* Corresponding author at: Department of Technical Sciences, Algebra Bernays University, Gradiscanska 24, 10000, Zagreb, Croatia.
E-mail addresses: ssamantavu@gmail.com (S. Samanta), tofigh.allahviranloo@istinye.edu.tr (T. Allahviranloo), leo.mrsic@algebra.hr (L. Mrsic), antonios.kalampakas@aum.edu.kw (A. Kalampakas).

https://doi.org/10.1016/j.dajour.2026.100691
Received 18 October 2025; Received in revised form 1 February 2026; Accepted 9 February 2026
Available online 12 February 2026

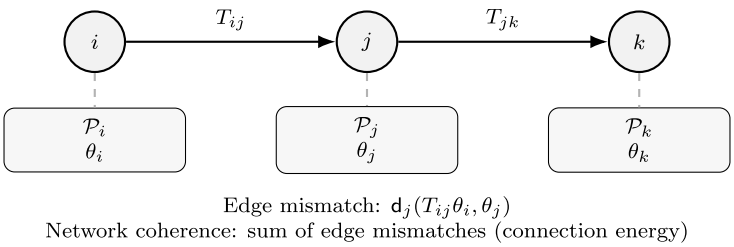

Fig. 1. PLQG: each node has its own local parameter/state space $\mathcal{P}_i$, and each edge translates states via a transport T_{ij}.

local decisions remain meaningful while enabling a global check of alignment across the network. A transport-based coherence measure guides learning and inference, building on ideas related to fuzzy and quantum graph models and transport-aligned graph tools [1,2,12].

The contributions of this study are: (i) a decision-oriented framework for learning over heterogeneous and interconnected decision systems, (ii) an interpretable way to measure and manage inconsistency across networked decisions, and (iii) a clear distinction between event-driven (discrete) and continuous interactions within the same model.

2. Literature review

Recent work in decision analytics emphasizes interpretability and the need to account for organizational heterogeneity rather than optimizing predictive accuracy alone [1,2]. In networked settings (healthcare, transportation, multi-agency systems), decisions are interdependent and errors can propagate; hence analytics must represent interactions explicitly and support coherent aggregation of local decisions [3].

Graph-based representations have long been used to study interconnected systems [13–16]. Temporal and dynamic graphs capture how relationships and decision pathways evolve over time [17–19], and dynamic graph embedding methods aim to learn useful representations under changing connectivity [20]. These results motivate decision frameworks that do not rely on static assumptions.

Graph neural networks (GNNs) provide flexible architectures for learning on relational data and have achieved strong results on node/edge prediction tasks [21]. However, standard GNN pipelines typically assume a shared feature space and comparable node representations, which limits their direct applicability to heterogeneous decision units [9,21,22]. Heterogeneous GNNs relax some assumptions, but they usually do not specify an explicit translation layer that maps parameters or beliefs between units in an interpretable way [23,24].

Uncertainty and ambiguity are central in many decision problems. Fuzzy sets and fuzzy graph models support graded membership and rule-based reasoning, which can improve interpretability in problems with linguistic assessments or imprecise data [10,11,25,26]. Work on generalized and aggregated fuzzy graphs shows how uncertainty can be encoded at node and edge levels, supporting decision rules based on partial belief rather than hard thresholds [11,27].

Tools that transport vector-valued information over graphs – such as connection Laplacians and vector diffusion maps – formalize alignment under local coordinate systems [28]. Optimal transport provides a geometric approach to mapping distributions and has been adapted to graph settings for domain alignment and transfer learning [29–32]. These tools motivate PLQG's edge-wise transports, but PLQG focuses on translating *local decision states* (parameters, beliefs, representations) rather than transporting raw distributions.

Federated learning addresses collaboration under privacy constraints and data heterogeneity by sharing updates rather than raw data [33–35]. While it offers practical deployment benefits, it often does not provide an explicit semantic alignment layer that makes cross-unit comparisons interpretable.

Networked decision analytics emphasizes coherence and interpretability, while much of graph learning assumes that node information is directly comparable. Uncertainty-based graph models handle ambiguity, but they do not by themselves define how to translate heterogeneous *local parameter spaces*. Connection-based tools formalize transport under local coordinates, but they are rarely integrated with decision-centric objectives that combine local losses with an explicit, interpretable coherence penalty. The gap addressed here is a decision-oriented framework that (i) keeps local models in their native spaces, (ii) makes cross-unit translation explicit through edge-wise transports, and (iii) measures and optimizes network-wide consistency through a clear mismatch functional. PLQG fills this gap by combining node-specific parameter spaces, transports, and a connection-energy regularizer with standard local learning objectives.

3. Preliminaries

A *directed weighted graph* is $G = (V, E, w)$ where $V = \{1, 2, \dots, n\}$ is the node set, $E \subseteq V \times V$ is the set of directed edges, and $w : E \to \mathbb{R}_{\geq 0}$ assigns a weight w_{ij} to each edge $(i, j) \in E$. Weights may represent frequency, capacity, confidence, or trust.

For a vector-valued function $f : V \to \mathbb{R}^p$, the classical Laplacian energy is

$$\mathcal{E}(f) = \frac{1}{2} \sum_{(i,j)\in E} w_{ij} \, \|f(i) - f(j)\|^2.$$

It penalizes disagreement across edges and is widely used as a smoothness regularizer.

Definition 1 (*Quantum Graph [12,36,37]*)**.** A *quantum graph* is a graph whose node and edge weights evolve in time. One representative form assigns each node v_i a time-dependent state $\nu_i(t) \in [0, 1]$ and assigns each edge (v_i, v_j) a time-dependent coupling $\gamma_{ij}(t)$ derived from the node states.

In this paper, each node i is allowed to use its own *local parameter/state space* $\mathcal{P}_i$. Depending on the application, $\mathcal{P}_i$ may be Euclidean, probabilistic, or manifold-valued. Common examples include:

- $\mathbb{R}^p$ with Euclidean distance,
- the probability simplex Δ^{p-1} with a statistical distance,
- the Stiefel manifold $\mathrm{St}(k, p)$, and
- the SPD cone $\mathrm{SPD}(p)$ with an affine-invariant distance.

A *network state* is an assignment $\theta = (\theta_i)_{i\in V}$ where each $\theta_i \in \mathcal{P}_i$. In PLQG, θ_i represents the local model parameters or the local decision state at node i (see Fig. 1).

The classical energy $\mathcal{E}(f)$ is recovered as a special case of PLQG when all node spaces coincide (e.g., $\mathcal{P}_i = \mathbb{R}^p$ for all i), all transports are identity maps ($T_{ij} = I$), and the distance is Euclidean. In that case, the PLQG edge mismatch reduces to $\|\theta_i - \theta_j\|$ and the PLQG connection energy becomes the usual graph smoothness penalty. This bridge explains why PLQG can be viewed as a transport-aligned generalization of Laplacian regularization to heterogeneous node spaces.

4. Parameter learning quantum graph (PLQG)

Definition 2 (*Parameter Learning Quantum Graph (PLQG)*)**.** Let $G = (V, E, w)$ be a weighted directed graph and let $\mathbb{T}$ be a time domain (discrete or continuous).

A *PLQG* consists of:

1. **Local spaces:** for each node $i \in V$, a metric space $(\mathcal{P}_i, \mathsf{d}_i)$. A local state is $\theta_i(t) \in \mathcal{P}_i$.
2. **Transports:** for each edge $(i, j) \in E$ and time $t \in \mathbb{T}$, a map

   $$T_{ij}(t) : \mathcal{P}_i \to \mathcal{P}_j,$$

   interpreted as how node j should translate and compare a state coming from node i.

A *network state* is $\theta(t) = (\theta_i(t))_{i \in V}$ with $\theta_i(t) \in \mathcal{P}_i$.

Remark 1. In a PLQG, the *graph* tells us who is connected, the *local space* tells us what each node stores, and the *transport* tells us how that content is translated across an edge.

Definition 3 (*Connection Energy*)**.** At time t, the *connection energy* of a network state $\theta(t)$ is

$$E_t(\theta) = \frac{1}{2} \sum_{(i,j) \in E} w_{ij}\, \mathsf{d}_j\big(T_{ij}(t)\theta_i(t),\, \theta_j(t)\big)^2.$$

Definition 4 (*Learning Objective*)**.** Assume each node i has a local loss function $\ell_i(\theta_i(t); t)$ (e.g., prediction error or constraint violation). The PLQG objective at time t is

$$J_t(\theta) = \sum_{i \in V} \ell_i(\theta_i(t); t) \;+\; \lambda\, E_t(\theta), \qquad \lambda > 0,$$

where λ balances local performance and network coherence.

Discrete-time learning. For $t = 0, 1, \ldots, T$, one can solve

$$\min_{\{\theta(t)\}} \sum_{t=0}^{T} J_t(\theta(t)).$$

Continuous-time learning. For continuous evolution, one can solve

$$\min_{\theta(\cdot)} \int_0^T J_t(\theta(t))\, dt.$$

Example 1 (*Airline Hubs: Delay and Congestion Alignment*)**.** We want a coherent network-wide state estimate for *departure delay* and *congestion* across hubs, where delays propagate along flight connections.

Nodes:

$$V = \{\text{DEL}, \text{DXB}, \text{IST}, \text{LHR}, \text{JFK}, \text{LAX}\}.$$

Edges:

$$E = \{(\text{DEL}, \text{DXB}), (\text{DXB}, \text{IST}), (\text{IST}, \text{LHR}), (\text{LHR}, \text{JFK}), (\text{JFK}, \text{LAX}), (\text{DXB}, \text{LHR})\}.$$

For simplicity set $w_{ij} = 1$ on all edges.

Use the same 2D local space at each node:

$$\mathcal{P}_i = \mathbb{R}^2, \qquad \theta_i = (d_i, c_i),$$

where d_i is delay (minutes) and c_i is congestion in $[0, 1]$.

Suppose the current state estimates are:

City	d_i (min)	c_i
DEL	22	0.55
DXB	35	0.70
IST	18	0.50
LHR	15	0.85
JFK	28	0.80
LAX	20	0.65

For each edge (i, j), define a simple transport that predicts what j should see based on upstream delay, local congestion at j, and flight duration:

$$T_{ij}\begin{pmatrix} d_i \\ c_i \end{pmatrix} = \begin{pmatrix} \rho_{ij} d_i + \eta_{ij} c_j + \xi\, \text{Dur}_{ij} \\ c_j \end{pmatrix}.$$

We use a global $\xi = 1.5$ and the following parameters:

Edge (i, j)	ρ_{ij}	η_{ij}	Dur_{ij} (h)
DEL→DXB	0.60	6	3.5
DXB→IST	0.65	6	4.5
IST→LHR	0.70	8	4.0
DXB→LHR	0.55	8	7.5
LHR→JFK	0.60	7	7.0
JFK→LAX	0.50	5	6.0

Use Euclidean distance $\mathsf{d}_j(u, v) = \|u - v\|_2$. Because the second coordinate in $T_{ij}(\theta_i)$ is c_j, the mismatch focuses on delay consistency:

$$\Delta_{ij} = \|T_{ij}(\theta_i) - \theta_j\|_2^2 = \left(\widehat{d}_j^{(i)} - d_j\right)^2, \quad \widehat{d}_j^{(i)} = \rho_{ij} d_i + \eta_{ij} c_j + \xi\, \text{Dur}_{ij}.$$

Using the table values, the mismatches are: $\Delta_{\text{DEL,DXB}} = 152.52$, $\Delta_{\text{DXB,IST}} = 210.25$, $\Delta_{\text{IST,LHR}} = 108.16$, $\Delta_{\text{DXB,LHR}} = 497.29$, $\Delta_{\text{LHR,JFK}} = 8.41$, and $\Delta_{\text{JFK,LAX}} = 39.06$. Hence

$$E = \frac{1}{2} \sum_{(i,j) \in E} \Delta_{ij} = \tfrac{1}{2}(1015.69) = 507.845.$$

A large energy indicates global inconsistency; for example, DXB→LHR predicts much higher delay than observed at LHR, suggesting either (i) a delay shock not yet realized at LHR, (ii) parameter drift, or (iii) missing operational actions Fig. 2).

Example 2 (*Hospitals: Aligning Local Linear Predictors*)**.** Four hospitals train local linear predictors for the same clinical risk score, but features are encoded differently across sites (different scaling and coding). Forcing parameter equality is not meaningful; PLQG aligns models through transports.

Let $V = \{1, 2, 3, 4\}$ denote hospitals and use a star topology (site 1 as a reference):

$$E = \{(2, 1), (3, 1), (4, 1)\}.$$

Let each hospital use a linear model with parameters in $\mathbb{R}^3$:

$$\mathcal{P}_i = \mathbb{R}^3, \qquad \theta_i = (b_i, w_{i,1}, w_{i,2}).$$

Assume hospital i uses a local feature map $x_i = S_i x_1$ relative to hospital 1. A compatible parameter transform is

$$T_{i1}(b_i, w_{i,1}, w_{i,2}) = \big(b_i,\; w_{i,1} s_{i,1},\; w_{i,2} s_{i,2}\big),$$

where $S_i = \text{diag}(s_{i,1}, s_{i,2})$. Let

$$S_2 = \text{diag}(0.5,\, 2), \quad S_3 = \text{diag}(2,\, 1), \quad S_4 = \text{diag}(1,\, 0.25).$$

Assume current local models:

$$\theta_1 = (0.8,\, 1.2,\, -0.6), \quad \theta_2 = (0.7,\, 2.0,\, -0.3), \quad \theta_3 = (0.9,\, 0.7,\, -0.6),$$
$$\theta_4 = (0.8,\, 1.1,\, -2.0).$$

Transporting into hospital 1 coordinates gives:

$$T_{21}(\theta_2) = (0.7, 1.0, -0.6), \quad T_{31}(\theta_3) = (0.9, 1.4, -0.6), \quad T_{41}(\theta_4) = (0.8, 1.1, -0.5).$$

With Euclidean distance on $\mathcal{P}_1$, the connection energy is

$$E = \frac{1}{2} \sum_{i=2}^{4} \|T_{i1}(\theta_i) - \theta_1\|_2^2 = \frac{1}{2}(0.05 + 0.05 + 0.02) = 0.06.$$

A low energy indicates that consensus is meaningful after alignment even though raw parameters differ Fig. 3).

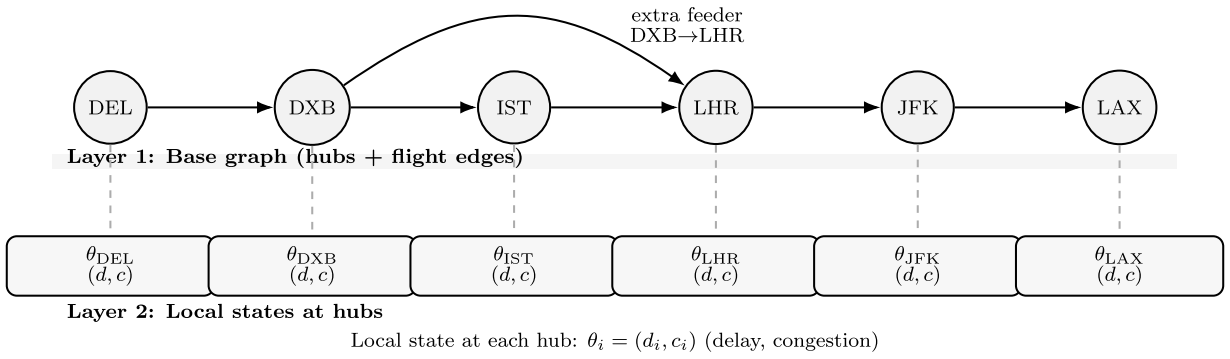

Fig. 2. Two-layer view: the top layer is the hub graph and the bottom layer shows the local state at each hub.

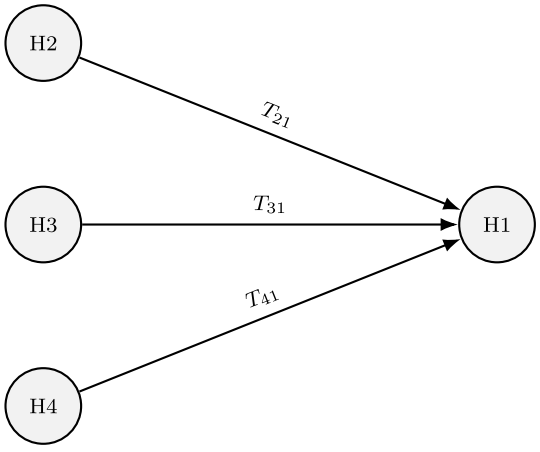

Fig. 3. A simple alignment pattern: each site transports its local model into a reference space before comparison.

Example 3 (*Function Spaces: Integral Transports*)**.** Let $G = (V, E)$ be a directed graph with $V = \{1, 2, 3\}$ and $E = \{(1, 2), (2, 3)\}$. Let the local spaces be Hilbert spaces:

$$\mathcal{P}_1 = L^2([0, 1]), \quad \mathcal{P}_2 = L^2([0, 2]), \quad \mathcal{P}_3 = L^2([0, 3]).$$

A network state is $\theta = (f_1, f_2, f_3)$ with $f_i \in \mathcal{P}_i$.

Define transports as bounded integral operators:

$$(T_{12} f)(x) = \int_0^1 e^{-|x-y|} f(y)\, dy, \quad x \in [0, 2],$$

$$(T_{23} g)(x) = \int_0^2 \cos(x - y)\, g(y)\, dy, \quad x \in [0, 3].$$

Use $\mathrm{d}_j(u, v) = \|u - v\|_{L^2}$ on each local space. Then the connection energy is

$$E(\theta) = \frac{1}{2}\Big(\|T_{12} f_1 - f_2\|^2_{L^2([0,2])} + \|T_{23} f_2 - f_3\|^2_{L^2([0,3])}\Big).$$

This example shows that PLQG naturally supports non-Euclidean and infinite-dimensional node spaces.

Remark 2 (*Event-driven vs. Continuous Edges*)**.** In a PLQG, an edge $(i, j) \in E$ represents how the state at node i influences node j through a transport $T_{ij}(t) : \mathcal{P}_i \to \mathcal{P}_j$.

Event-driven edge. An edge is *event-driven* if state transfer occurs only at isolated event times $\{t_k\}$:

$$\theta_j(t_k^+) = T_{ij}(t_k)\big(\theta_i(t_k^-)\big).$$

This models scheduled updates, transactions, or switching actions.

Continuous edge. An edge is *continuous* if influence is exerted persistently in time, for example via a coupling

$$\frac{d}{dt}\theta_j(t) = \mathcal{A}_{ij}(t)\, \theta_i(t),$$

or an equivalent integral operator. This models ongoing interactions such as mixing or flow coupling.

In practice. Many systems combine both types, and PLQG can represent such hybrid interaction patterns.

5. Properties of PLQGs

In this section we fix a time t and suppress (t) in the notation. Let $G = (V, E, w)$ be a weighted directed graph with $|V| = n$. Each node $i \in V$ carries a real Hilbert space $\mathcal{P}_i$, and each edge $(i, j) \in E$ carries a bounded linear transport $T_{ij} : \mathcal{P}_i \to \mathcal{P}_j$.

Define the direct-sum space

$$\mathbb{P} = \bigoplus_{i=1}^{n} \mathcal{P}_i,$$

and write a network state as $\theta = (\theta_1, \ldots, \theta_n) \in \mathbb{P}$.

Adjacency and degree operators. Define the (incoming) transport adjacency operator $\mathcal{A} : \mathbb{P} \to \mathbb{P}$ by

$$(\mathcal{A}\theta)_j = \sum_{i:(i,j)\in E} w_{ij}\, T_{ij}\theta_i.$$

Define the out-degree operator $\mathcal{D}_{\mathrm{out}}$ by

$$(\mathcal{D}_{\mathrm{out}}\theta)_i = \left(\sum_{j:(i,j)\in E} w_{ij}\, T_{ij}^* T_{ij}\right)\theta_i,$$

and the in-degree operator $\mathcal{D}_{\mathrm{in}}$ by

$$(\mathcal{D}_{\mathrm{in}}\theta)_j = \left(\sum_{i:(i,j)\in E} w_{ij}\right)\theta_j.$$

Definition 5 (*PLQG Laplacian*)**.** The *PLQG Laplacian* is

$$\mathcal{L} = \mathcal{D}_{\mathrm{out}} + \mathcal{D}_{\mathrm{in}} - (\mathcal{A} + \mathcal{A}^*).$$

Theorem 1 (*Quadratic Form Equals Connection Mismatch*)**.** *For any $\theta \in \mathbb{P}$,*

$$\langle \theta, \mathcal{L}\theta \rangle_{\mathbb{P}} = \sum_{(i,j)\in E} w_{ij}\, \|T_{ij}\theta_i - \theta_j\|^2_{\mathcal{P}_j}.$$

Consequently,

$$E(\theta) = \tfrac{1}{2}\langle \theta, \mathcal{L}\theta \rangle_{\mathbb{P}}.$$

Proof. Using the direct-sum inner product,

$$\langle \theta, \mathcal{D}_{\text{out}}\theta\rangle_{\mathbb{P}} = \sum_{(i,j)\in E} w_{ij}\|T_{ij}\theta_i\|^2_{\mathcal{P}_j}, \qquad \langle \theta, \mathcal{D}_{\text{in}}\theta\rangle_{\mathbb{P}} = \sum_{(i,j)\in E} w_{ij}\|\theta_j\|^2_{\mathcal{P}_j}.$$

Also,

$$\langle \theta, \mathcal{A}\theta\rangle_{\mathbb{P}} = \sum_{(i,j)\in E} w_{ij}\langle T_{ij}\theta_i, \theta_j\rangle_{\mathcal{P}_j}, \qquad \langle \theta, \mathcal{A}^*\theta\rangle_{\mathbb{P}} = \sum_{(i,j)\in E} w_{ij}\langle T_{ij}\theta_i, \theta_j\rangle_{\mathcal{P}_j}.$$

Combining these identities gives

$$\langle \theta, \mathcal{L}\theta\rangle_{\mathbb{P}} = \sum_{(i,j)\in E} w_{ij}\Big(\|T_{ij}\theta_i\|^2 - 2\langle T_{ij}\theta_i, \theta_j\rangle + \|\theta_j\|^2\Big) = \sum_{(i,j)\in E} w_{ij}\|T_{ij}\theta_i - \theta_j\|^2.$$

□

5.1. Degree concepts in PLQGs

Definition 6. For a weighted directed graph (V, E, w), define the classical scalar degrees

$$\deg^{\text{out}}(i) = \sum_{j:(i,j)\in E} w_{ij}, \qquad \deg^{\text{in}}(i) = \sum_{j:(j,i)\in E} w_{ji}.$$

Definition 7. The *operator out-degree* at node i is

$$D_i^{\text{out}} := \sum_{j:(i,j)\in E} w_{ij}\, T_{ij}^* T_{ij} \;:\; \mathcal{P}_i \to \mathcal{P}_i.$$

The *operator in-degree* (a scalar multiple of identity on $\mathcal{P}_i$) is

$$D_i^{\text{in}} := \sum_{j:(j,i)\in E} w_{ji}\, I_{\mathcal{P}_i}.$$

Definition 8. Define transport-based scalar degrees by operator norm:

$$\deg_T^{\text{out}}(i) := \|D_i^{\text{out}}\|, \qquad \deg_T^{\text{in}}(i) := \|D_i^{\text{in}}\| = \deg^{\text{in}}(i).$$

Lemma 1. *For every node $i \in V$,*

$$\deg_T^{\text{out}}(i) = \left\| \sum_{j:(i,j)\in E} w_{ij} T_{ij}^* T_{ij} \right\| \le \sum_{j:(i,j)\in E} w_{ij}\|T_{ij}\|^2.$$

If $\|T_{ij}\| \le M$ for all outgoing edges of i, then

$$\deg_T^{\text{out}}(i) \le M^2 \deg^{\text{out}}(i).$$

Proof. Use subadditivity of the operator norm and $\|T^*T\| = \|T\|^2$. □

Lemma 2. *Let $\mathcal{A} : \mathbb{P} \to \mathbb{P}$ be the transport adjacency operator. Then*

$$\|\mathcal{A}\| \le \max_{j\in V} \sqrt{\sum_{i:(i,j)\in E} w_{ij}\|T_{ij}\|^2} \cdot \max_{i\in V} \sqrt{\deg^{\text{out}}(i)}.$$

In particular, if $\|T_{ij}\| \le M$ for all edges, then

$$\|\mathcal{A}\| \le M\sqrt{\Delta_{\text{in}}\Delta_{\text{out}}}, \quad \Delta_{\text{in}} = \max_j \deg^{\text{in}}(j),\ \Delta_{\text{out}} = \max_i \deg^{\text{out}}(i).$$

5.2. Shortest paths in PLQGs based on transport

Classical shortest paths use only scalar edge lengths. In a PLQG, edges also carry transports, so we can define a distance that penalizes both low-confidence edges and highly distorting transports.

Definition 9. A directed path $p : i_0 \to i_1 \to \cdots \to i_k$ defines the composed transport

$$T_p := T_{i_{k-1}i_k} \circ \cdots \circ T_{i_0 i_1} : \mathcal{P}_{i_0} \to \mathcal{P}_{i_k},$$

and the multiplicative path weight

$$w(p) := \prod_{\ell=0}^{k-1} w_{i_\ell i_{\ell+1}}.$$

Definition 10. Fix $\varepsilon \in (0, 1)$. Define the edge cost

$$c(i,j) := -\log(w_{ij} + \varepsilon) + \log(\|T_{ij}\| + \varepsilon), \qquad (i,j) \in E,$$

and the path cost $C(p) := \sum c(i_\ell, i_{\ell+1})$. Define the transport-based distance

$$d_{\text{PLQG}}(s,t) := \inf_{p:\, s\rightsquigarrow t} C(p).$$

Remark 3. The term $-\log(w_{ij})$ prefers reliable edges, while $\log\|T_{ij}\|$ penalizes transports that amplify uncertainty. When all edge costs are nonnegative, standard shortest-path methods (e.g., Dijkstra) apply.

5.3. Decision support: selecting a reliable inference chain

Consider a hospital setting where a *Triage Committee* (TC) must decide whether to admit a patient to the ICU. Multiple experts provide assessments, but (i) trust differs by source, and (ii) the same evidence may be interpreted differently by different decision makers. We model this as a PLQG and select a reliable inference chain using transport-based shortest paths.

5.3.1. PLQG construction

Nodes. Let

$$V = \{\text{LAB}, \text{RAD}, \text{ER}, \text{ID}, \text{ICU}, \text{TC}\}.$$

Local state space (belief simplex). Use a shared belief space at each node:

$$\mathcal{P}_i = \Delta^2 = \{(p_0, p_1, p_2) \in \mathbb{R}^3_{\ge 0} : p_0 + p_1 + p_2 = 1\},$$

with hypotheses

$$H_0 : \text{no sepsis}, \quad H_1 : \text{moderate sepsis}, \quad H_2 : \text{severe sepsis}.$$

Write $\theta_i = (p_{i,0}, p_{i,1}, p_{i,2}) \in \Delta^2$.

Edges and trust. Let

$$E = \{(\text{LAB}, \text{ER}), (\text{RAD}, \text{ER}), (\text{ER}, \text{ID}), (\text{ER}, \text{ICU}), (\text{ID}, \text{TC}), (\text{ICU}, \text{TC}), (\text{ER}, \text{TC})\},$$

and set trust weights $w_{ij} \in (0, 1]$.

Transports (interpretation). Define a simple sharpening/flattening family:

$$T_{ij}(\theta) = \text{Norm}(\theta^{\alpha_{ij}}), \qquad \theta^\alpha = (p_0^\alpha, p_1^\alpha, p_2^\alpha),$$

where $\alpha_{ij} > 1$ sharpens and $\alpha_{ij} < 1$ flattens, and $\text{Norm}(x) = x/\sum_k x_k$.

5.3.2. Transport-based path cost

For a path p, define

$$w(p) = \prod_{e\in p} w_e, \qquad \delta(p) = \sum_{e\in p} |\log(\alpha_e)|, \qquad C(p) = -\log(w(p)) + \beta\,\delta(p),$$

where $\beta > 0$ trades off trust and interpretation distortion.

Use the following values (illustrative):

Edge	w_{ij}	α_{ij}
ER → TC	0.70	1.60
ER → ID	0.90	1.10
ID → TC	0.80	1.15
ER → ICU	0.85	1.05
ICU → TC	0.90	1.05

$(\beta = 2)$.

Candidate paths ER→TC:

$$p_1 : \text{ER} \to \text{TC}, \quad p_2 : \text{ER} \to \text{ID} \to \text{TC}, \quad p_3 : \text{ER} \to \text{ICU} \to \text{TC}.$$

Their costs are:

$$C(p_1) = 1.2967, \qquad C(p_2) = 0.7987, \qquad C(p_3) = 0.4631.$$

Hence the preferred inference chain is ER → ICU → TC.

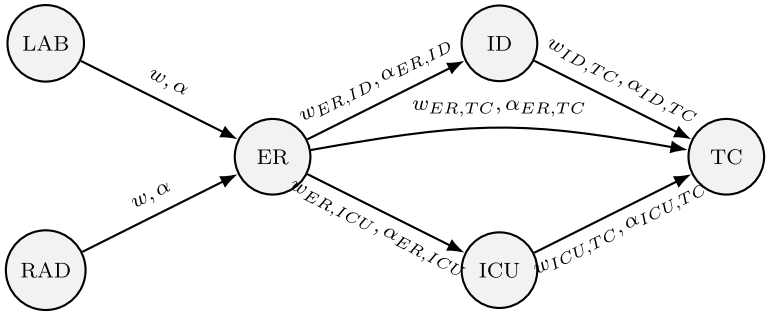

Fig. 4. Expert network with trust (w) and interpretation (α) on each edge.

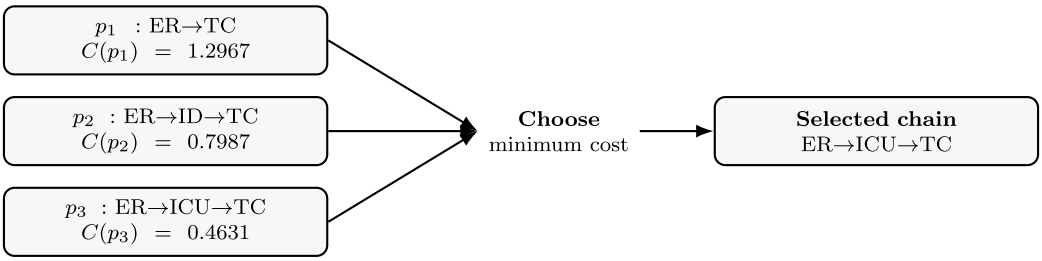

Fig. 5. Transport-based path selection balances trust and interpretation distortion.

Let

$\theta_{\mathrm{ER}} = (0.10, 0.30, 0.60)$.

Two mild sharpenings ($\alpha = 1.05$) give

$\theta_{\mathrm{ICU}} = T_{\mathrm{ER,ICU}}(\theta_{\mathrm{ER}}) \approx (0.094, 0.292, 0.614)$,
$\theta_{\mathrm{TC}} = T_{\mathrm{ICU,TC}}(\theta_{\mathrm{ICU}}) \approx (0.089, 0.284, 0.627)$.

This chain slightly increases confidence in severe sepsis without aggressive distortion, see Figs. 4 and 5).

6. Global airline operations as a PLQG

A global airline (or alliance) coordinates operations across multiple time zones, regulations, and local constraints. The system must:

- anticipate and mitigate delay propagation across hubs,
- manage crew rotations under duty/rest rules [38],
- adapt to changing congestion, weather, and disruptions, and
- keep schedules feasible in local time at each city.

City set (18 nodes)

Let the hubs be (see Figs. 6 and 7):

$V = \{$Delhi (DEL), Kolkata (CCU), Mumbai (BOM),
Dubai (DXB), Doha (DOH),
Istanbul (IST), Cairo (CAI), Nairobi (NBO), Johannesburg (JNB),
London (LHR), Paris (CDG), Frankfurt (FRA), New York (JFK),
Toronto (YYZ), Chicago (ORD), Los Angeles (LAX),
Singapore (SIN), Tokyo (HND)$\}$.

Let $G = (V, E, w)$ where $(i, j) \in E$ if a scheduled direct flight exists within the planning horizon (e.g., next 48 h). The weight w_{ij} can reflect frequency/capacity/criticality.

Local state vector

At each city i, use a 5D operational state:

$$\mathcal{P}_i = \mathbb{R}^5, \qquad \theta_i(t) = (\tau_i(t),\, d_i(t),\, r_i(t),\, c_i(t),\, f_i(t)),$$

where: τ_i is local clock time (hours), d_i is expected departure delay (minutes), r_i is crew rest buffer (hours), c_i is congestion in $[0, 1]$, and f_i is fatigue risk in $[0, 1]$.

Transport on a flight edge

For each flight edge (i, j), define a time-dependent transport:

$$T_{ij}(t)\begin{pmatrix}\tau_i\\ d_i\\ r_i\\ c_i\\ f_i\end{pmatrix} = \begin{pmatrix}\tau_i + \mathrm{Dur}_{ij}(t) + \Delta_{ij}(t)\\ \rho_{ij}\, d_i + \eta_{ij}\, c_j\\ r_i - \mathrm{Dur}_{ij}(t) - \kappa\, d_i/60\\ c_j\\ \alpha f_i + \beta\, \mathrm{Dur}_{ij}(t)/10 + \zeta\, d_i/60\end{pmatrix}.$$

Here $\mathrm{Dur}_{ij}(t)$ is flight duration (hours) and $\Delta_{ij}(t)$ converts local time from i to j (including DST and policy effects). Parameters $\rho_{ij}, \eta_{ij}, \kappa, \alpha, \beta, \zeta$ are learned from data.

The connection energy at time t is

$$E_t(\theta) = \frac{1}{2}\sum_{(i,j)\in E} w_{ij}\left\|T_{ij}(t)\theta_i(t) - \theta_j(t)\right\|_2^2.$$

Decision-oriented local penalties

One simple local penalty is:

$$\ell_i(\theta_i) = \max(0, f_i - F_{\max})^2 + \max(0, r_{\min} - r_i)^2 + \max(0, d_i - D_{\max})^2.$$

The objective is

$$J_t(\theta) = \sum_{i\in V} \ell_i(\theta_i(t)) + \lambda E_t(\theta).$$

.

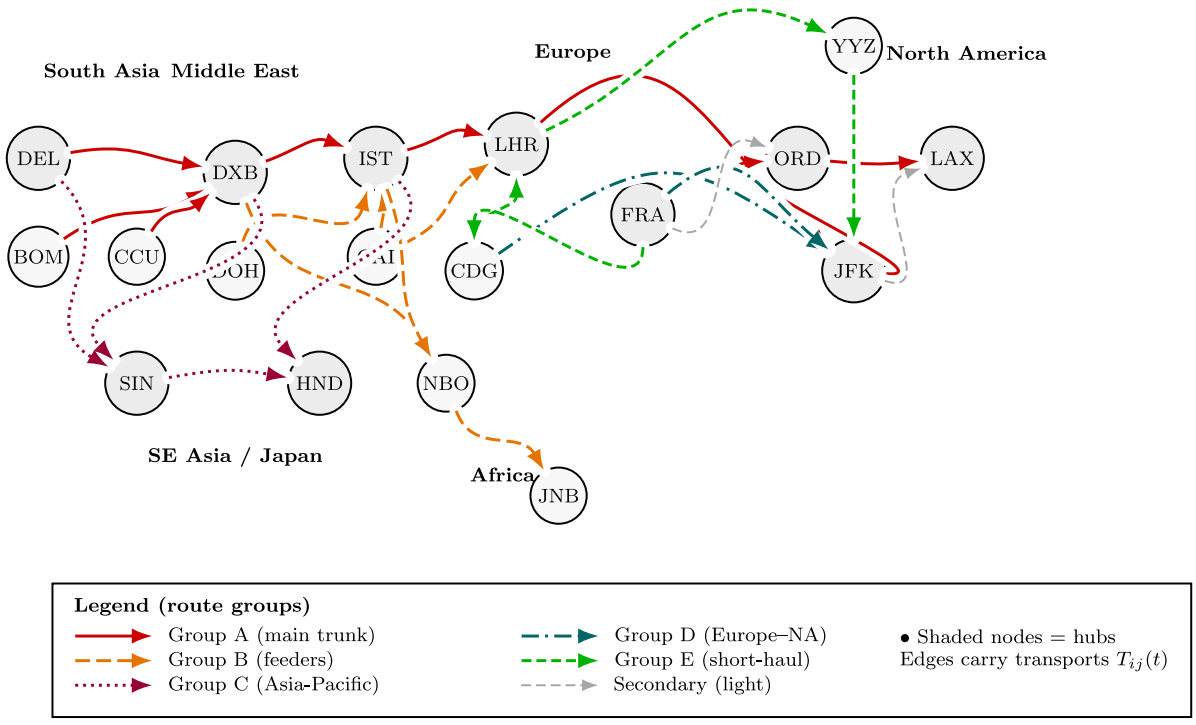

Fig. 6. Schematic 18-city network used for explanation. Route groups use different colors and line styles; dashed and dotted styles are shown explicitly in the legend.

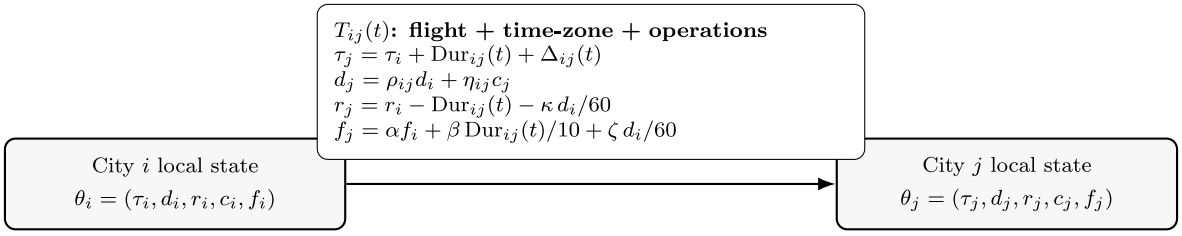

Fig. 7. A flight edge acts as a transport: it maps the local state at city i into the representation used at city j.

Operational steps

Step 1: Data preparation. Estimate congestion $c_i(t)$ and delay $d_i(t)$ at each city; compute time conversion $\Delta_{ij}(t)$.

Step 2: Build the base graph. Create edges for flights in the planning horizon and set weights w_{ij}.

Step 3: Initialize local states. Initialize $\theta_i(t_0)$ from current operational data.

Step 4: Learn transport parameters. Fit $\rho_{ij}, \eta_{ij}, \alpha, \beta, \zeta, \kappa$ from historical operations.

Step 5: Solve PLQG optimization. At each step t, solve

$$\min_{\theta(t)} \sum_{i \in V} \ell_i(\theta_i(t)) + \lambda E_t(\theta),$$

subject to operational bounds (e.g., $0 \le c_i, f_i \le 1$).

Step 6: Act on outputs. Use $\theta^\star(t)$ to trigger interventions (crew reassignment, delay buffers, rerouting, etc.).

PLQG-based prediction model for global delay forecasting

Goal. For each hub $j \in V$ and discrete time t, predict next-step departure delay:

$$\widehat{d}_j(t+1) \approx d_j(t+1).$$

Local state and projection. Use the same local state:

$$\theta_i(t) = (\tau_i(t), d_i(t), r_i(t), c_i(t), f_i(t)) \in \mathbb{R}^5,$$

and the operational projection

$$\pi(\theta_i(t)) = (d_i(t), c_i(t), r_i(t), f_i(t)) \in \mathbb{R}^4.$$

Transported message. For each edge (i, j) define:

$$m_{i \to j}(t) = \begin{pmatrix} \rho_{ij}\, d_i(t) + \eta_{ij}\, c_j(t) \\ \alpha f_i(t) + \beta\, \mathrm{Dur}_{ij}(t)/10 + \zeta\, d_i(t)/60 \end{pmatrix} \in \mathbb{R}^2. \tag{1}$$

Aggregation. Let $\mathcal{N}^-(j) = \{i : (i, j) \in E\}$. Aggregate incoming information as

$$M_j(t) = \sum_{i \in \mathcal{N}^-(j)} w_{ij}\, m_{i \to j}(t) \in \mathbb{R}^2. \tag{2}$$

Define

$$x_j(t) = \left(d_j(t), c_j(t), r_j(t), f_j(t), M_j^{(1)}(t), M_j^{(2)}(t)\right) \in \mathbb{R}^6,$$

and predict

$$\widehat{d}_j(t+1) = \sigma(a^\top x_j(t) + b), \qquad \sigma(z) = \max(z, 0).$$

Training loss. Given observed $d_j(t+1)$, learn a, b (and optionally transport parameters) by minimizing

$$\sum_t \sum_{j \in V} \left(\widehat{d}_j(t+1) - d_j(t+1) \right)^2 + \gamma \|a\|_2^2.$$

Numerical example (propagation into London)

Consider the chain

DEL → DXB → IST → LHR.

At time t:

$$\theta_{\text{IST}} = (\cdot, 25, 5, 0.50, 0.35), \quad \theta_{\text{LHR}} = (\cdot, 15, 4, 0.80, 0.45).$$

Assume IST is the dominant in-neighbor of LHR and use:

$$w_{\text{IST,LHR}} = 1, \ \rho_{\text{IST,LHR}} = 0.65, \ \eta_{\text{IST,LHR}} = 8, \ \text{Dur}_{\text{IST,LHR}}(t) = 4.0,$$

$$\alpha = 0.70, \ \beta = 0.50, \ \zeta = 0.40.$$

Then

$$m^{(1)}_{\text{IST}\to\text{LHR}}(t) = 0.65(25) + 8(0.80) = 22.65,$$

$$m^{(2)}_{\text{IST}\to\text{LHR}}(t) = 0.70(0.35) + 0.50(0.4) + 0.40(25/60) = 0.6117.$$

So $M_{\text{LHR}}(t) = (22.65, 0.6117)^\top$ and

$$x_{\text{LHR}}(t) = (15, 0.80, 4, 0.45, 22.65, 0.6117).$$

With $a = (0.40, 12, -1.0, 8, 0.55, 6.0)^\top$ and $b = -10$,

$$\widehat{d}_{\text{LHR}}(t+1) = \max(21.3277, 0) = 21.33 \text{ min}.$$

7. Conclusion

We introduced Parameter Learning Quantum Graphs (PLQG) as a framework for learning and decision-making in networks where different units use different local models and representations. By giving each node its own local parameter/state space and equipping edges with explicit translation maps (transports), PLQG avoids the unrealistic assumption of direct comparability across nodes. Connection energy provides a simple, interpretable measure of network-wide alignment and can be combined with standard local loss functions.

Future work can focus on scalable optimization for large networks, learning transports directly from data, and applying PLQG to additional decision settings such as supply chains, healthcare coordination, and federated organizations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Project 101187046 Healthy and resilient mindset with an organized, nurturing, and digital workplace for you (HARMONY), ERASMUS-EDU-2024-PI-ALL-INNO (Partnerships for Innovation - Alliances), ERASMUS-EDU-2024-PI-ALL-INNO-EDU-ENTERP

Data availability

No data was used for the research described in the article.

References

[1] S. MajidiParast, R.N. Monemi, S. Gelareh, A graph convolutional network for optimal intelligent predictive maintenance of railway tracks, Decis. Anal. J. 14 (2025) 100542.
[2] N. Das, B. Sadhukhan, R. Chatterjee, S. Chakrabarti, Integrating sentiment analysis with graph neural networks for enhanced stock prediction: A comprehensive survey, Decis. Anal. J. 10 (2024) 100417.
[3] J. Jiang, W. Li, H. Cui, Z. Zhu, L. Zhang, Q. Hu, et al., Feasibility of applying graph theory to diagnosing generalized anxiety disorder using machine learning models, Psychiatry Res.: Neuroimaging 333 (2023) 111656.
[4] B. Roy, Decision-aiding today: What should we expect? in: In Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications, Springer US, Boston, MA, 1999, pp. 1–35.
[5] G. Montibeller, D. Von Winterfeldt, Cognitive and motivational biases in decision and risk analysis, Risk Anal. 35 (7) (2015) 1230–1251.
[6] G.S. Parnell, T.A. Bresnick, E.R. Johnson, S.N. Tani, E. Specking, Handbook of Decision Analysis, John Wiley & Sons, 2025.
[7] Shachter R. D., Evaluating influence diagrams, Oper. Res. 34 (6) (1986) 871–882.
[8] J. Poropudas, Bayesian networks, influence diagrams, and games in simulation metamodeling, 2011.
[9] B. Khemani, S. Patil, K. Kotecha, S. Tanwar, A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions, J. Big Data 11 (1) (2024) 18.
[10] S. Kalathian, S. Ramalingam, N. Deivanayagampillai, A fuzzy planar subgraph formation model for partitioning very large-scale integration networks, Decis. Anal. J. 9 (2023) 100339.
[11] S. Samanta, B. Sarkar, A study on generalized fuzzy graphs, J. Intell. Fuzzy Systems 35 (3) (2018) 3405–3412.
[12] S. Samanta, T. Allahviranloo, A. Amirteimoori, M.H. Behzadi, L. Mršić, Signed quantum graphs: a dynamical system, 2025.
[13] F. Riaz, K.M. Ali, Applications of graph theory in computer science, in: Proc. International Conference on Computational Intelligence, Communication Systems and Networks, 2011, pp. 142–145.
[14] A. Majeed, I. Rauf, Graph theory: A comprehensive survey about graph theory applications in computer science and social networks, Inventions 5 (1) (2020) 10.
[15] M. Dehmer, F. Emmert-Streib, Y. Shi, Quantitative graph theory: A new branch of graph theory and network science, Inform. Sci. 418 (2017) 575–580.
[16] R.V. Ravi, P.K. Dutta, S.B. Goyal, Graph data science: Applications and future, Appl. Graph Data Sci. 22 (2025) 7–243.
[17] V. Kostakos, Temporal graphs, Phys. A 388 (6) (2009) 1007–1023.
[18] O. Michail, An introduction to temporal graphs: An algorithmic perspective, Internet Math. 12 (4) (2016) 239–280.
[19] H. Wu, J. Cheng, S. Huang, Y. Ke, Y. Lu, Y. Xu, Path problems in temporal graphs, Proc. VLDB Endow. 7 (9) (2014) 721–732.
[20] C.D. Barros, M.R. Mendonça, A.B. Vieira, A. Ziviani, A survey on embedding dynamic graphs, ACM Comput. Surv. 55 (1) (2021) 1–37.
[21] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2020) 4–24.
[22] C. Morris, F. Frasca, N. Dym, H. Maron, S. Jegelka, Future directions in the theory of graph machine learning, in: In Proceedings of the 41st International Conference on Machine Learning, (ICML), 2024.
[23] S. Yang, R. Liu, A review of graph neural networks for rolling bearing fault diagnosis, Meas. Sci. Technol. (2025).
[24] K. Yu, Y. Li, Q. Zhan, Y. Zhang, B. Xing, Intelligent fault diagnosis for cross-domain few-shot learning of rotating equipment based on mixup data augmentation, Machines 13 (9) (2025) 807.
[25] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (3) (1965) 338–353.
[26] W. Pedrycz, Granular computing: Fundamentals and system modeling, in: Developments in Advanced Control and Intelligent Automation for Complex Systems, Springer International Publishing, Cham, 2021, pp. 167–192.
[27] F.J. Talavera, C. Bejines, S. Ardanza-Trevijano, J. Elorza, Aggregation of fuzzy graphs, Internat. J. Approx. Reason. 172 (2024) 109243.
[28] A. Singer, H.T. Wu, Vector diffusion maps and the connection Laplacian, Comm. Pure Appl. Math. 65 (8) (2012) 1067–1144.
[29] G. Peyré, M. Cuturi, Computational optimal transport: With applications to data science, Found. Trends® Mach. Learn. 11 (5–6) (2019) 355–607.
[30] S. Kolouri, G.K. Rohde, H. Hoffmann, Sliced wasserstein distance for learning gaussian mixture models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3427–3436.
[31] D. Bon, G. Pai, G. Bellaard, O. Mula, R. Duits, Optimal transport on the lie group of roto-translations, SIAM J. Imaging Sci. 18 (2) (2025) 789–821.
[32] E. Facca, M. Benzi, Fast iterative solution of the optimal transport problem on graphs, SIAM J. Sci. Comput. 43 (3) (2021) A2295–A2319.
[33] B. Liu, N. Lv, Y. Guo, Y. Li, Recent advances on federated learning: A systematic survey, Neurocomputing 597 (2024) 128019.
[34] B. Yurdem, M. Kuzlu, M.K. Gullu, F.O. Catak, M. Tabassum, Federated learning: Overview, strategies, applications, tools and future directions, Heliyon 10 (19) (2024).

[35] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, et al., Advances and open problems in federated learning, Found. Trends® Mach. Learn. 14 (1–2) (2021) 1–210.

[36] S. Samanta, A study on quantum graphs, in: Quantum Theory and Fuzzy Systems: Traversing Uncertainty in Group Decision-Making and Social Networks: Quantum and Fuzzy Approaches To Social Network Analysis and Group Decisions, Springer Nature Switzerland, Cham, 2025, pp. 29–42.

[37] S. Samanta, T. Allahviranloo, Introduction to quantum graphs and their stability analysis, Comput. Methods Differ. Equations. (2025).

[38] B. Vaidyanathan, K.C. Jha, R.K. Ahuja, Multicommodity network flow approach to the railroad crew-scheduling problem, IBM J. Res. Dev. 51 (3.4) (2007) 325–344.