Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 34(5): 3093–3111.
DOI: 10.3934/era.2026140
Received: 13 January 2026
Revised: 24 February 2026
Accepted: 16 March 2026
Published: 09 April 2026

Theory article

Learning input-output fuzzy matrices from sensor data via Gaussian fuzzification

Xiefei He[1,*], Tao Yu[2], Zicong He[3] and Shi'an Wang[1]

[1] College of Information Engineering, Guangzhou Institute of Technology, Guangzhou 510075, China

[2] School of Manufacturing, Guangdong Polytechnic of Science and Trade, Guangzhou 511500, China

[3] Guangdong Industry Polytechnic University, Guangzhou 510300, China

* **Correspondence:** Email: 70066@gzvtc.edu.cn.

Abstract: Fuzzy relation matrices are a core representation mechanism in fuzzy inference and fuzzy system modeling, yet practical data are often raw multichannel sensor readings rather than predefined input/output fuzzy vectors. We develop an end-to-end supervised learning framework that (i) maps sensor readings to fuzzy vectors via per-channel Gaussian fuzzification and (ii) learns a max-min fuzzy relation matrix from adjacent-time sensor pairs. To overcome the nonsmooth max-min composition, we introduce a differentiable softmax/softmin surrogate with temperature annealing and derive a stochastic-gradient training procedure that jointly optimizes the relation matrix and fuzzifier parameters. Experiments on structured synthetic sensor streams show stable convergence and improved structure recovery over a fixed (data-driven) fuzzifier baseline while approaching an oracle fuzzifier.

Keywords: fuzzy matrix; max-min composition; Gaussian membership functions; differentiable smoothing; stochastic gradient descent; sensor-driven fuzzy modelling

1. Introduction

Fuzzy set theory [1, 2] provides a principled approach for representing and reasoning with graded truth values. In many applications, a central modeling object is a fuzzy relation matrix, whose entries quantify the strength of pairwise relationships among variables. When the system evolves over time, such matrices can be used as input-output operators that map an input fuzzy vector to an output fuzzy vector through fuzzy relational composition. Among common choices, max-min composition is widely used due to its close connection with fuzzy logic and its robustness in uncertain settings.

In many practical deployments, however, the available data are raw sensor readings (e.g., temperature, pressure, vibration) rather than fuzzy membership degrees. A large body of work assumes access to input-output fuzzy vectors and then infers the underlying fuzzy matrix [3]. This assumption is violated in typical sensor-driven settings, including self-learning fuzzy discrete-event models driven by external variables [4, 5] and sensor-to-fuzzy linkages in self-learning modeling for verification [6]. Hence, to learn a time-evolution fuzzy relation matrix from sensor streams, one must first connect real-valued observations to fuzzy vectors.

In this paper, we adopt a per-channel Gaussian fuzzification map with learnable centers and spreads. This design is lightweight, interpretable, and differentiable, which makes it suitable for end-to-end training. The resulting model is also parameter-efficient compared with generic black-box deep sequence models: it learns an explicit relation matrix P (amenable to inspection) and a compact set of fuzzifier parameters that encode domain knowledge about sensor scaling. This complements recent developments in explainable/interpretable fuzzy systems [7, 8].

The key technical difficulty is that max-min composition is nonsmooth. Let $\boldsymbol{s} \in [0,1]^n$ be an input fuzzy vector and $P \in [0,1]^{n\times n}$ a fuzzy relation matrix. Max–min inference produces an output fuzzy vector $\hat{\boldsymbol{s}} \in [0,1]^n$ via

$$\hat{s}_j = \max_{i\in\{1,\dots,n\}} \min(s_i, P_{ij}), \qquad j = 1,\dots,n. \tag{1.1}$$

The nested max / min operations render Eq (1.1) nondifferentiable and therefore difficult to optimize using gradient-based methods. While max-only smoothing is well studied in minimax optimization [9], learning under max-min requires handling both operations. Existing smoothing strategies for min-max type problems [10, 11] motivate the differentiable surrogate adopted in this work.

This paper makes four contributions: First, we formalize a learning setting where each fuzzy-vector component corresponds to a sensor channel and is generated by a Gaussian membership function with learnable parameters. Second, we propose a softmax/softmin approximation to the max-min composition with temperature annealing, enabling stable gradient-based training. Third, we derive a stochastic–gradient optimization procedure that jointly learns the fuzzy relation matrix and the Gaussian fuzzifier parameters from adjacent-time sensor pairs. Fourth, we evaluate the method on structured synthetic sensor streams where ground truth is available, and we report structure recovery, fuzzifier recovery, and predictive fit.

2. Related work

Classical fuzzy relational modeling includes fuzzy relational equations and inverse problems, initiated by early work such as Sanchez's studies on eigen fuzzy set equations [12]. Subsequent research developed analytical and algorithmic methods for fuzzy relational compositions and solving fuzzy relational equations [13]. In knowledge engineering, Bandler-Kohout relational products and implication-based relational calculus provide additional composition operators and semantic interpretations [14].

On the data-driven side, fuzzy systems have long been trained using gradient-based methods by embedding fuzzy rules into differentiable architectures, such as the adaptive neuro fuzzy inference system (ANFIS) [15], and by training max-min fuzzy neural networks with convergent smoothing [11]. More recently, supervised learning has been used to identify transition structures in fuzzy discrete event systems using only external variables [4, 5]. Our work aligns with these

self-learning approaches but focuses specifically on learning a general input-output fuzzy relation matrix under max-min composition while simultaneously learning the fuzzification map from sensors.

For a broader algebraic perspective on finite-state and fuzzy logical systems, we also refer to the survey on applications of algebraic state space theory (based on semitensor products) to finite state machines [16].

Softmax approximations (e.g., log-sum-exp) and their minimax variants are standard tools in non-smooth optimization [9]. For min-max-min structures, smoothing methods and temperature parameters require careful tuning to avoid gradient saturation [10]. In the fuzzy-neural literature, smoothing-based training for max-min networks establishes that properly designed smooth surrogates can yield stable convergence [11]. We build on this line by using coupled softmin and softmax operators, together with temperature annealing, to approximate the max-min composition in Eq (1.1).

3. Problem formulation

3.1. Fuzzy relations and fuzzy relation matrices

Let $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ be two finite universes. A fuzzy relation R from X to Y is a membership function $\mu_R : X \times Y \to [0, 1]$. In matrix form, a *fuzzy relation matrix* is $P \in [0, 1]^{n\times n}$ with entries

$$P_{ij} = \mu_R(x_i, y_j), \qquad i, j \in \{1, \ldots, n\}. \tag{3.1}$$

This representation is standard in fuzzy relational calculus; see also recent discussions of fuzzy relation matrices as knowledge structures [17].

3.2. Sensor pairs and fuzzy vectors

We observe an n-channel sensor vector $\boldsymbol{x}_t \in \mathbb{R}^n$ at time t and its one-step successor $\boldsymbol{x}_{t+1} \in \mathbb{R}^n$. Throughout, we assume a one-to-one correspondence between sensor channels and fuzzy-vector dimensions, that is, both $\boldsymbol{x}_t$ and the induced fuzzy vector have dimension n. Concretely, each channel is mapped to a single membership degree that reflects the "activation" of that channel around a learnable prototype value. This choice is motivated by sensor-driven fuzzy modeling scenarios where each channel (or feature) plays the role of a fuzzy atom or proposition. We discuss multi-linguistic-term extensions (multiple membership degrees per channel) in Section 7.

3.3. Per-channel Gaussian membership

For each channel $i \in \{1, \ldots, n\}$, we define a Gaussian membership function

$$\mu_i(x; c_i, \sigma_i) = \exp\left(-\frac{(x - c_i)^2}{2\sigma_i^2}\right), \tag{3.2}$$

where $c_i \in \mathbb{R}$ is the centre, and $\sigma_i > 0$ is the spread. Intuitively, $\mu_i(x; c_i, \sigma_i)$ measures the degree to which the scalar observation x is "compatible" with a prototypical value c_i under tolerance σ_i; it takes values in $(0, 1]$ and equals 1 if and only if $x = c_i$. We adopt Gaussian membership functions because they are smooth (facilitating gradient-based learning), have a clear geometric interpretation (prototype-and-tolerance), and are widely used in neuro-fuzzy systems as differentiable membership primitives [15].

3.4. Vector-valued fuzzification

Applying (3.2) channelwise yields the fuzzy representation

$$\boldsymbol{s}_t = \boldsymbol{\mu}(\boldsymbol{x}_t; \boldsymbol{c}, \boldsymbol{\sigma}) \triangleq [\mu_1(x_{t,1}; c_1, \sigma_1), \ldots, \mu_n(x_{t,n}; c_n, \sigma_n)]^\top \in (0, 1]^n, \tag{3.3}$$

and similarly, $\boldsymbol{s}_{t+1} = \boldsymbol{\mu}(\boldsymbol{x}_{t+1}; \boldsymbol{c}, \boldsymbol{\sigma})$. The parameters $(\boldsymbol{c}, \boldsymbol{\sigma})$ are shared across all time indices and all training pairs and must be learned from data.

3.5. Training pairs

Given a dataset of adjacent sensor pairs

$$\mathcal{D} = \{(\boldsymbol{x}_t^{(r)}, \boldsymbol{x}_{t+1}^{(r)})\}_{r=1}^{R}, \tag{3.4}$$

we denote $\boldsymbol{s}_t^{(r)} = \boldsymbol{\mu}(\boldsymbol{x}_t^{(r)}; \boldsymbol{c}, \boldsymbol{\sigma})$ and $\boldsymbol{s}_{t+1}^{(r)} = \boldsymbol{\mu}(\boldsymbol{x}_{t+1}^{(r)}; \boldsymbol{c}, \boldsymbol{\sigma})$.

3.6. Input–output fuzzy relation matrix

We model one-step fuzzy dynamics via a fuzzy relation matrix $P \in [0, 1]^{n \times n}$. The entry P_{ij} represents the degree to which the fuzzy activation of channel i at time t supports the activation of channel j at time t+1.

3.7. Max-min composition

Given $\boldsymbol{s}_t$, the predicted successor $\hat{\boldsymbol{s}}_{t+1}$ is computed by max-min composition:

$$\hat{s}_{t+1,j} = (\boldsymbol{s}_t \circ P)_j \triangleq \max_i \min\Big(s_{t,i}, P_{ij}\Big), \qquad j = 1, \ldots, n. \tag{3.5}$$

This operator ensures $\hat{\boldsymbol{s}}_{t+1} \in (0, 1]^n$ and is monotone in both $\boldsymbol{s}_t$ and P.

Our goal is to jointly estimate $(P, \boldsymbol{c}, \boldsymbol{\sigma})$ so that the induced fuzzy dynamics match the observed dynamics in fuzzy space.

3.8. Illustrative example

We illustrate the construction with a simple two-channel example ($n = 2$).

Let the Gaussian membership parameters be

$$\boldsymbol{c} = (0,\ 1), \qquad \boldsymbol{\sigma} = (1,\ 0.5).$$

Consider one observed sensor pair

$$\boldsymbol{x}_t = (0.2,\ 0.6), \qquad \boldsymbol{x}_{t+1} = (0.9,\ 1.1).$$

Applying Eq (3.2) channelwise yields

$$\boldsymbol{s}_t = \boldsymbol{\mu}(\boldsymbol{x}_t; \boldsymbol{c}, \boldsymbol{\sigma}) \approx \begin{bmatrix} \exp(-0.02) \\ \exp(-0.32) \end{bmatrix} \approx \begin{bmatrix} 0.98 \\ 0.73 \end{bmatrix}$$

and

$$\boldsymbol{s}_{t+1} = \boldsymbol{\mu}(\boldsymbol{x}_{t+1}; \boldsymbol{c}, \boldsymbol{\sigma}) \approx \begin{bmatrix} \exp(-0.405) \\ \exp(-0.02) \end{bmatrix} \approx \begin{bmatrix} 0.67 \\ 0.98 \end{bmatrix}.$$

Suppose the fuzzy relation matrix is

$$P = \begin{bmatrix} 0.8 & 0.3 \\ 0.4 & 0.9 \end{bmatrix}.$$

Using the max-min composition Eq (3.5), the predicted successor is

$$\hat{\boldsymbol{s}}_{t+1} = \boldsymbol{s}_t \circ P,$$

with components

$$\hat{s}_{t+1,1} = \max\{\min(0.98, 0.8),\ \min(0.73, 0.4)\} = 0.8,$$
$$\hat{s}_{t+1,2} = \max\{\min(0.98, 0.3),\ \min(0.73, 0.9)\} = 0.73.$$

Hence,

$$\hat{\boldsymbol{s}}_{t+1} \approx (0.8,\ 0.73),$$

which can be directly compared with the observed fuzzy vector $\boldsymbol{s}_{t+1} \approx (0.67, 0.98)$. This example demonstrates how the Gaussian fuzzification and max-min composition jointly induce one-step fuzzy dynamics.

When the relation matrix P is not constrained (i.e., learned) from training data, the max-min composition in Eq (3.5) merely represents a feasible fuzzy propagation mechanism rather than an approximately optimal predictor of $\boldsymbol{s}_{t+1}$. Consequently, a noticeable discrepancy between the prediction $\hat{\boldsymbol{s}}_{t+1}$ and the observed successor $\boldsymbol{s}_{t+1}$ is expected. Reducing such discrepancies by jointly estimating $(P, \boldsymbol{c}, \boldsymbol{\sigma})$ from data is precisely the goal of the subsequent learning procedure.

4. Proposed method: Differentiable learning with Gaussian fuzzification

We present a differentiable surrogate for Eq (3.5) and a learning objective that supports stochastic gradient optimization.

4.1. Softmin and softmax surrogates

For two scalars $a, b \in \mathbb{R}$, define a softmin operator with temperature $\tau > 0$:

$$\operatorname{softmin}_\tau(a, b) \triangleq -\tau \log\left(e^{-a/\tau} + e^{-b/\tau}\right). \tag{4.1}$$

As $\tau \to 0^+$, $\operatorname{softmin}_\tau(a, b) \to \min(a, b)$. Similarly, for a vector $\boldsymbol{z} \in \mathbb{R}^n$, define softmax:

$$\operatorname{softmax}_\tau(\boldsymbol{z}) \triangleq \tau \log\Big(\sum_{i=1}^{n} e^{z_i/\tau}\Big) \xrightarrow[\tau\to 0^+]{} \max_i z_i. \tag{4.2}$$

Both operators are smooth and have well-behaved gradients as follows:

$$\frac{\partial}{\partial a} \operatorname{softmin}_\tau(a, b) = \frac{e^{-a/\tau}}{e^{-a/\tau} + e^{-b/\tau}}, \qquad \frac{\partial}{\partial b} \operatorname{softmin}_\tau(a, b) = \frac{e^{-b/\tau}}{e^{-a/\tau} + e^{-b/\tau}}, \tag{4.3}$$

$$\frac{\partial}{\partial z_i} \operatorname{softmax}_\tau(\boldsymbol{z}) = \frac{e^{z_i/\tau}}{\sum_k e^{z_k/\tau}}. \tag{4.4}$$

4.2. Differentiable max-min composition

For each output component j, define

$$\tilde{s}_{t+1,j} \triangleq \operatorname{softmax}_\tau \Big(\big[\operatorname{softmin}_\tau(s_{t,1}, P_{1j}), \ldots, \operatorname{softmin}_\tau(s_{t,n}, P_{nj}) \big]^\top \Big). \tag{4.5}$$

The mapping $\boldsymbol{s}_t \mapsto \tilde{\boldsymbol{s}}_{t+1}$ is differentiable with respect to both $\boldsymbol{s}_t$ and P. In training we anneal τ from a smoother value τ_{start} to a sharper value τ_{end} to improve optimization stability while ultimately approximating true max-min behaviour.

4.3. Learning objective

Given Eqs (3.3) and (4.5), we define the primary loss in fuzzy space:

$$\mathcal{L}_{\text{fit}}(P, \boldsymbol{c}, \boldsymbol{\sigma}) = \frac{1}{2R} \sum_{r=1}^{R} \left\| \tilde{\boldsymbol{s}}_{t+1}^{(r)} - \boldsymbol{s}_{t+1}^{(r)} \right\|_2^2 . \tag{4.6}$$

To promote a stable structure recovery of P, we incorporate regularization terms:

$$\mathcal{L}(P, \boldsymbol{c}, \boldsymbol{\sigma}) = \mathcal{L}_{\text{fit}} + \lambda_1 \|P\|_1 + \lambda_d \left\| \operatorname{diag}(P) - \mathbf{1} \right\|_2^2 + \lambda_c \|\boldsymbol{c} - \boldsymbol{c}_0\|_2^2 + \lambda_\sigma \left\| \log \boldsymbol{\sigma} - \log \boldsymbol{\sigma}_0 \right\|_2^2 . \tag{4.7}$$

Here, $(\boldsymbol{c}_0, \boldsymbol{\sigma}_0)$ denote a data-driven prior estimate (computed from sensor statistics) that serves as a weak anchor for the fuzzifier parameters. Figure 1 shows the configuration of the fuzzy relation matrix learning model.

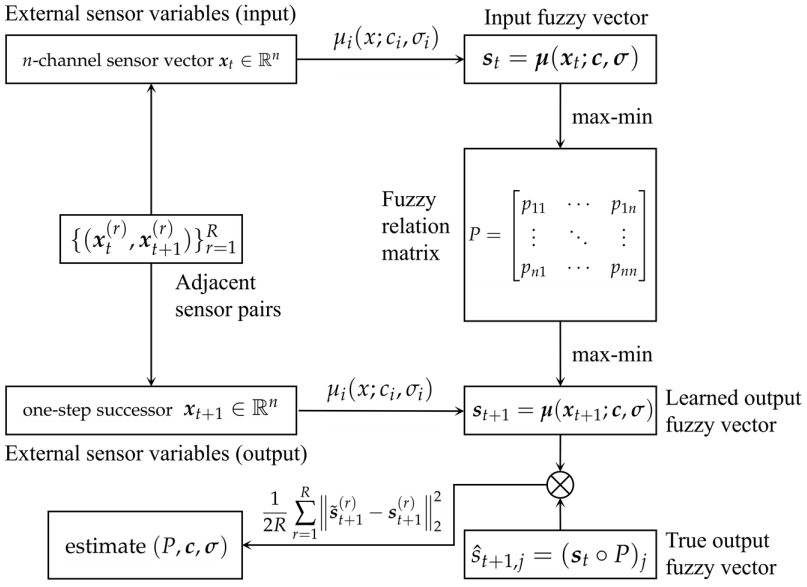


Figure 1. Configuration of the fuzzy relation matrix learning model.

5. Optimization algorithm

5.1. Parameterization and constraints

We optimize the constrained variables $(P, \boldsymbol{\sigma})$ through unconstrained parameters $(W, \boldsymbol{\rho})$. For the relation matrix, we enforce strict feasibility $P \in (0,1)^{n\times n}$ by

$$P_{ij} = \sigma(W_{ij}), \qquad W \in \mathbb{R}^{n\times n}, \tag{5.1}$$

where $\sigma(u) = 1/(1+e^{-u})$ is the logistic sigmoid. This parameterization is convenient because it avoids boundary artifacts (e.g., exact zeros) while remaining differentiable. It also eliminates the need for explicit projection steps onto $[0,1]$, which can introduce nonsmooth updates. If sparse, exactly-zero relations are desired in downstream use, one can post-process the learned P by thresholding small entries while keeping the training dynamics smooth. By the chain rule, gradients in W are obtained from gradients in P via

$$\frac{\partial \mathcal{L}}{\partial W_{ij}} = \frac{\partial \mathcal{L}}{\partial P_{ij}} P_{ij}(1 - P_{ij}). \tag{5.2}$$

For the Gaussian spreads, we enforce positivity by

$$\sigma_i = \sigma_{\min} + \mathrm{softplus}(\rho_i), \qquad \mathrm{softplus}(u) = \log(1 + e^u) \tag{5.3}$$

with unconstrained $\rho_i \in \mathbb{R}$ and a small $\sigma_{\min} > 0$ (used throughout) to prevent degenerate memberships. The resulting derivative used in backpropagation is

$$\frac{\partial \sigma_i}{\partial \rho_i} = \mathrm{sigmoid}(\rho_i) = \frac{1}{1 + e^{-\rho_i}}. \tag{5.4}$$

5.2. Gradients through Gaussian fuzzification

Let $s_{t,i} = \mu_i(x_{t,i}; c_i, \sigma_i)$ be the ith fuzzy membership degree computed from the raw sensor value $x_{t,i}$ using Eq (3.2). For numerical safety, we implement fuzzification with a small $\varepsilon > 0$ in the denominator, that is, $z_{t,i} = (x_{t,i} - c_i)/(\sigma_i + \varepsilon)$ and $s_{t,i} = \exp(-1/2 z_{t,i}^2)$. Differentiating yields closed-form partial derivatives

$$\frac{\partial s_{t,i}}{\partial c_i} = s_{t,i} \frac{x_{t,i} - c_i}{(\sigma_i + \varepsilon)^2}, \tag{5.5}$$

$$\frac{\partial s_{t,i}}{\partial \sigma_i} = s_{t,i} \frac{(x_{t,i} - c_i)^2}{(\sigma_i + \varepsilon)^3}. \tag{5.6}$$

These expressions are used at both time indices: $s_t = \boldsymbol{\mu}(\boldsymbol{x}_t; \boldsymbol{c}, \boldsymbol{\sigma})$ affects the prediction $\tilde{\boldsymbol{s}}_{t+1}$ through Eq (4.5), and $s_{t+1} = \boldsymbol{\mu}(\boldsymbol{x}_{t+1}; \boldsymbol{c}, \boldsymbol{\sigma})$ appears directly in the fit loss Eq (4.6). Finally, gradients in $\boldsymbol{\rho}$ follow from Eq (5.4) by $\partial \mathcal{L}/\partial \rho_i = (\partial \mathcal{L}/\partial \sigma_i)(\partial \sigma_i/\partial \rho_i)$.

5.3. Gradients through differentiable max-min composition

For a fixed output index j, define intermediate values

$$m_{ij} \triangleq \mathrm{softmin}_\tau(s_{t,i}, P_{ij}), \qquad \tilde{s}_{t+1,j} = \mathrm{softmax}_\tau([m_{1j}, \ldots, m_{nj}]^\top), \tag{5.7}$$

which matches Eq (4.5). Let

$$\alpha_{ij} \triangleq \frac{\exp(m_{ij}/\tau)}{\sum_{k=1}^{n} \exp(m_{kj}/\tau)} = \frac{\partial \tilde{s}_{t+1,j}}{\partial m_{ij}} \in (0, 1), \tag{5.8}$$

and

$$\beta_{ij} \triangleq \frac{\exp(-s_{t,i}/\tau)}{\exp(-s_{t,i}/\tau) + \exp(-P_{ij}/\tau)} = \frac{\partial m_{ij}}{\partial s_{t,i}}, \qquad 1 - \beta_{ij} = \frac{\partial m_{ij}}{\partial P_{ij}}. \tag{5.9}$$

For one training pair, the fit loss contribution is $\ell = 1/2 \sum_{j=1}^{n} (\tilde{s}_{t+1,j} - s_{t+1,j})^2$. Let $g_j \triangleq \partial \ell / \partial \tilde{s}_{t+1,j} = \tilde{s}_{t+1,j} - s_{t+1,j}$. Applying the chain rule gives explicit gradients after the smooth max-min layer:

$$\frac{\partial \ell}{\partial P_{ij}} = g_j \, \alpha_{ij} \, (1 - \beta_{ij}), \tag{5.10}$$

$$\frac{\partial \ell}{\partial s_{t,i}} = \sum_{j=1}^{n} g_j \, \alpha_{ij} \beta_{ij}. \tag{5.11}$$

Equation (5.10) is then mapped to W using Eq (5.2), and Eq (5.11) is propagated further through fuzzification using Eqs (5.5) and (5.6). As $\tau \to 0^+$, α_{ij} concentrates on the maximizing index (soft-argmax) and β_{ij} concentrates on the minimizing argument (soft-argmin), recovering the subgradient behavior of the exact max-min operator.

5.4. Stochastic gradient training

We use mini-batch stochastic optimization (Adam) on Eq (4.7). Each iteration samples a mini-batch $\mathcal{B} \subset \mathcal{D}$ and evaluates the following:

(i) fuzzification $\boldsymbol{s}_t, \boldsymbol{s}_{t+1}$ using Eq (3.3);

(ii) the smooth composition $\tilde{\boldsymbol{s}}_{t+1}$ using Eq (4.5);

(iii) the total objective Eq (4.7).

The procedure is summarized in Algorithm 1.

In our implementation, we use separate learning rates for the relation logits and the fuzzifier parameters, denoted by (η_P, η_θ). To reduce seed sensitivity and improve identifiability, we employ a warm start. For the first T_{warm} steps, we update only W (equivalently P), keeping $\theta = (\boldsymbol{c}, \boldsymbol{\sigma})$ fixed to the data-driven prior; afterwards, we jointly update $(W, \boldsymbol{c}, \boldsymbol{\rho})$. We also anneal the temperature exponentially,

$$\tau_k = \tau_{\text{start}} \left(\frac{\tau_{\text{end}}}{\tau_{\text{start}}} \right)^{\frac{k-1}{T-1}}, \qquad k = 1, \ldots, T, \tag{5.12}$$

which starts from a smoother surrogate and gradually sharpens toward max-min behaviour.

Algorithm 1 Proposed end-to-end learning

Require: Adjacent sensor pairs $\mathcal{D} = \{(\boldsymbol{x}_t^{(r)}, \boldsymbol{x}_{t+1}^{(r)})\}$, steps T, warm-up T_{warm}, temperatures $(\tau_{\text{start}}, \tau_{\text{end}})$, learning rates (η_P, η_θ).

Initialize $(\boldsymbol{c}, \boldsymbol{\sigma}) \leftarrow (\boldsymbol{c}_0, \boldsymbol{\sigma}_0)$ (small perturbation), $W \leftarrow 0$.
for $k = 1$ to T **do**
  Sample a mini-batch $\mathcal{B} \subset \mathcal{D}$.
  Set $\tau \leftarrow \tau_{\text{start}} \cdot (\tau_{\text{end}}/\tau_{\text{start}})^{\frac{k-1}{T-1}}$.
  Compute $\boldsymbol{s}_t, \boldsymbol{s}_{t+1}$ using (3.3).
  Compute $\tilde{\boldsymbol{s}}_{t+1}$ using (4.5) with $P = \sigma(W)$.
  Compute loss $\mathcal{L}$ using (4.7).
  if $k \le T_{\text{warm}}$ **then**
    Update W using Adam with step size η_P.
  else
    Update W with η_P and update $(\boldsymbol{c}, \boldsymbol{\rho})$ with η_θ.
  end if
end for
return learned P, $\boldsymbol{c}$, $\boldsymbol{\sigma}$.

5.5. Computational complexity

For each mini-batch of size B, evaluating Eq (4.5) forms an intermediate tensor $m \in \mathbb{R}^{B\times n\times n}$. The softmin stage computes all n^2 pairwise values m_{ij}, and the softmax stage performs a reduction over the i dimension for each j. Thus, both the forward and backward passes scale as $O(Bn^2)$ time and require $O(Bn^2)$ intermediate storage. For typical sensor channel counts (tens to low hundreds), this is feasible on commodity hardware. When n grows beyond a few hundred, the dominant bottleneck becomes the $B \times n \times n$ tensor (both memory and bandwidth). In such cases, one can (i) reduce batch size, (ii) compute the softmax reduction in blocks over j (streaming the tensor), or (iii) introduce structure or sparsity constraints on P to avoid dense n^2 interactions.

6. Experiments

6.1. Synthetic sensor stream generator

To isolate learning behavior and enable ground-truth evaluation, we generate synthetic adjacent-time sensor pairs $(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ from a known ground-truth $(P^\star, \theta^\star)$, where $\theta^\star = (\boldsymbol{c}^\star, \boldsymbol{\sigma}^\star)$. The procedure is summarized in Algorithm 2.

We split the generated pairs into 80% training and 20% test. To facilitate reproducibility, the MATLAB implementation used in this study is available at https://www.alipan.com/s/GGVF4ST41B4.

Algorithm 2 Synthetic sensor stream generator (one-step pairs)

Input: number of channels n (we use n=6), number of pairs R (we use R=6000), clipping ε, flip probability p_{flip}, noise scale η.
Sample a structured ground-truth relation matrix $P^\star \in [0,1]^{n\times n}$ with dominant diagonal entries and sparse off-diagonal couplings.
Sample ground-truth fuzzifier parameters $\boldsymbol{c}^\star \in \mathbb{R}^n$ and $\boldsymbol{\sigma}^\star \in \mathbb{R}^n_{>0}$ (heterogeneous but moderate per-channel scales).
for $r = 1$ to R **do**
    Sample $\boldsymbol{x}_t^{(r)}$ around $\boldsymbol{c}^\star$ (to avoid degenerate memberships), e.g., $x_{t,i}^{(r)} \sim \mathcal{N}(c_i^\star, (\sigma_i^\star)^2)$.
    Compute $\boldsymbol{s}_t^{(r)} \leftarrow \boldsymbol{\mu}(\boldsymbol{x}_t^{(r)}; \boldsymbol{c}^\star, \boldsymbol{\sigma}^\star)$ using Eq (3.3).
    Compute $\boldsymbol{s}_{t+1}^{(r)} \leftarrow \boldsymbol{s}_t^{(r)} \circ P^\star$ using exact max-min composition Eq (3.5).
    Clip: $\boldsymbol{s}_{t+1}^{(r)} \leftarrow \text{clip}(\boldsymbol{s}_{t+1}^{(r)}, \varepsilon, 1-\varepsilon)$.
    Map back to sensor space via the Gaussian inverse (component-wise):
    $x_{t+1,i}^{(r)} \leftarrow c_i^\star + \zeta_i \sigma_i^\star \sqrt{-2\log(s_{t+1,i}^{(r)})}$, where $\zeta_i \in \{\pm 1\}$.
    Choose branch signs to mimic one-step continuity: start with $\zeta_i \leftarrow \text{sign}(x_{t,i}^{(r)} - c_i^\star)$, then flip with probability p_{flip}.
    Add sensor noise: $x_{t+1,i}^{(r)} \leftarrow x_{t+1,i}^{(r)} + \eta\, \sigma_i^\star\, \xi$, where $\xi \sim \mathcal{N}(0,1)$.
end for
Output: dataset $\mathcal{D} = \{(\boldsymbol{x}_t^{(r)}, \boldsymbol{x}_{t+1}^{(r)})\}_{r=1}^R$.

6.2. Baselines and metrics

The experimental study considers three learning settings under the same training budget and the same differentiable max-min surrogate in Eq (4.5). The distinction among them lies solely in the treatment of the Gaussian fuzzifier parameters. In the oracle-fuzzifier setting, the fuzzifier is fixed to the ground-truth parameters $\theta^\star = (\boldsymbol{c}^\star, \boldsymbol{\sigma}^\star)$ used by the synthetic generator, and only the relation matrix P is optimised. Equivalently, optimization is performed over the unconstrained logits W with $P = \sigma(W)$. This setting serves as an idealized upper-bound reference, although such prior knowledge is unavailable in real applications.

In the fixed-fuzzifier setting, a data-driven initial estimate $\theta_0 = (\boldsymbol{c}_0, \boldsymbol{\sigma}_0)$ is first constructed from sensor marginals and then kept fixed throughout the optimization of P. Specifically, after pooling all raw observations $\{\boldsymbol{x}_t^{(r)}, \boldsymbol{x}_{t+1}^{(r)}\}_{r=1}^R$, the location and scale parameters for each channel i are defined by

$$c_{0,i} = \text{median}(\{x_{t,i}^{(r)}, x_{t+1,i}^{(r)}\}_{r=1}^R), \tag{6.1}$$

$$\sigma_{0,i} = 1.4826 \cdot \text{median}(|x_i - c_{0,i}|), \qquad \sigma_{0,i} \leftarrow \max(\sigma_{0,i}, \sigma_{\min}), \tag{6.2}$$

where the factor 1.4826 makes the median absolute deviation consistent with the standard deviation under a Gaussian assumption. The proposed setting starts from an initialisation close to $(\boldsymbol{c}_0, \boldsymbol{\sigma}_0)$ and jointly optimizes P together with the fuzzifier parameters $\theta = (\boldsymbol{c}, \boldsymbol{\sigma})$. To improve optimization stability, a warm-start strategy is adopted: During the first T_{warm} iterations, θ is frozen, and only P is updated; joint optimization of (P, θ) is enabled afterwards.

Let $\hat{P}$ and $\hat{\theta} = (\hat{\boldsymbol{c}}, \hat{\boldsymbol{\sigma}})$ denote the learned quantities, and let $P^\star$ and $\theta^\star$ denote their ground-truth counterparts in the synthetic generator. Performance is evaluated from three complementary aspects,

all reported in a lower-is-better manner. The first criterion measures structural recovery of the fuzzy relation matrix through the Frobenius error $\left\|\hat{P} - P^\star\right\|_{\mathrm{F}}$. The second criterion evaluates recovery of the fuzzifier parameters by means of scale-normalized relative mean absolute errors,

$$\text{rel. MAE}(\boldsymbol{c}) = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{c}_i - c_i^\star|}{|\sigma_i^\star| + \epsilon}, \tag{6.3}$$

$$\text{rel. MAE}(\boldsymbol{\sigma}) = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{\sigma}_i - \sigma_i^\star|}{|\sigma_i^\star| + \epsilon}, \tag{6.4}$$

where ϵ is a small constant introduced for numerical stability. The third criterion characterizes predictive accuracy in the fuzzy state space on the held-out test split, namely

$$\mathrm{MSE}_{\mathrm{test}}(s) = \frac{1}{R_{\mathrm{te}}\, n}\sum_{r\in\mathcal{D}_{\mathrm{te}}} \|\tilde{\boldsymbol{s}}_{t+1}^{(r)} - \boldsymbol{s}_{t+1}^{(r)}\|_2^2, \tag{6.5}$$

where $\tilde{\boldsymbol{s}}_{t+1}^{(r)}$ is computed using the final temperature τ_{end}. Together, these three metrics assess relation recovery, fuzzifier estimation, and out-of-sample predictive performance in a unified manner.

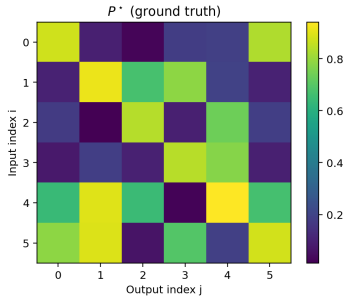
6.3. Implementation details

All methods use the same differentiable surrogate Eq (4.5) and are trained with Adam (default momentum parameters) using mini-batches of size 512. We anneal the temperature exponentially from $\tau_{\mathrm{start}} = 0.15$ to $\tau_{\mathrm{end}} = 0.05$ according to Eq (5.12). Each run uses $T = 650$ optimization steps with a warm start of $T_{\mathrm{warm}} = 200$ steps, during which only the relation parameters are updated. We use separate learning rates $(\eta_P, \eta_\theta) = (0.05, 0.015)$ for the relation logits W and the fuzzifier parameters $(\boldsymbol{c}, \boldsymbol{\rho})$. We set $\sigma_{\min} = 10^{-3}$ in Eq (5.3) and $\varepsilon = 10^{-12}$ in fuzzification to avoid division by zero. Regularization weights are $\lambda_1 = 1.5{\times}10^{-3}$ and $\lambda_d = 5{\times}10^{-2}$. For the fuzzifier anchor in Eq (4.7), we use $\lambda_c = \lambda_\sigma = 10^{-2}$; in implementation, the center deviation is normalized channelwise by $(\sigma_{0,i} + 10^{-6})$ to make the prior approximately scale-invariant. We run three random seeds $\{0, 1, 2\}$ and report mean±std.

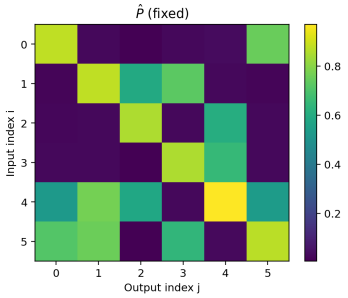
6.4. Qualitative analysis

Figure 2 visualizes the ground-truth relation matrix and the learned matrices (representative seed). The proposed method recovers the dominant diagonal and reduces spurious off-diagonal mass compared with the fixed-fuzzifier baseline.

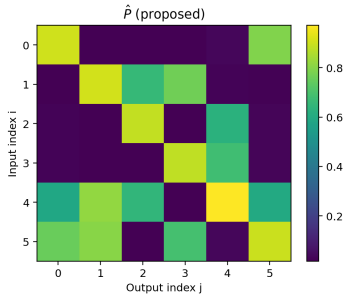
Figure 3 shows training curves for structure recovery (Frobenius error) and objective values. The warm-start and temperature annealing yield stable optimization; the proposed method consistently improves upon the fixed-fuzzifier baseline.



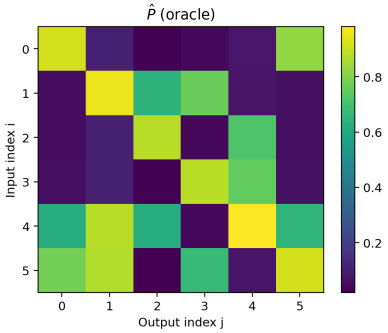
(a) $P^\star$



(b) Fixed

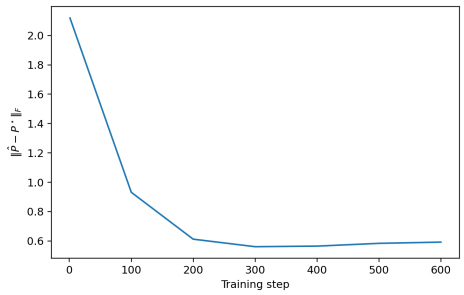


(c) Proposed

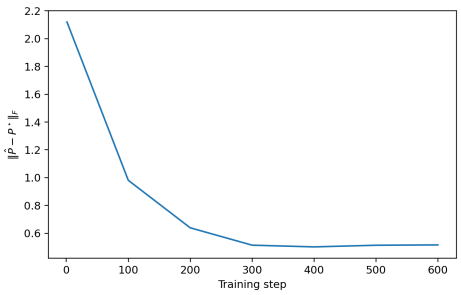


(d) Oracle

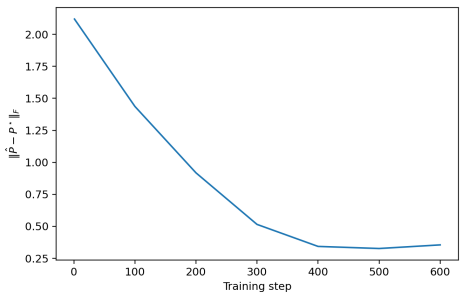
Figure 2. Heatmaps of relation matrices (representative seed). Enlarged subfigures improve axis and tick readability.



(a) $\left\|\hat{P} - P^\star\right\|_{\mathrm{F}}$ (Fixed)



(b) $\left\|\hat{P} - P^\star\right\|_{\mathrm{F}}$ (Proposed)



(c) $\left\|\hat{P} - P^\star\right\|_{\mathrm{F}}$ (Oracle)

Figure 3. Representative training curves (seed 0). Enlarged panels improve label readability.

6.5. *Elementwise learning dynamics*

To provide a fine-grained view of how parameters evolve, we record trajectories of all relation coefficients p_{ij} and fuzzifier parameters (c_i, σ_i) for a representative run (seed 0). We visualize 500 optimization steps using the same hyperparameters and warm-start schedule as the main experiments; this is close to the full 650-step budget used for the 3-seed summary in Table 1 while keeping the figures readable.

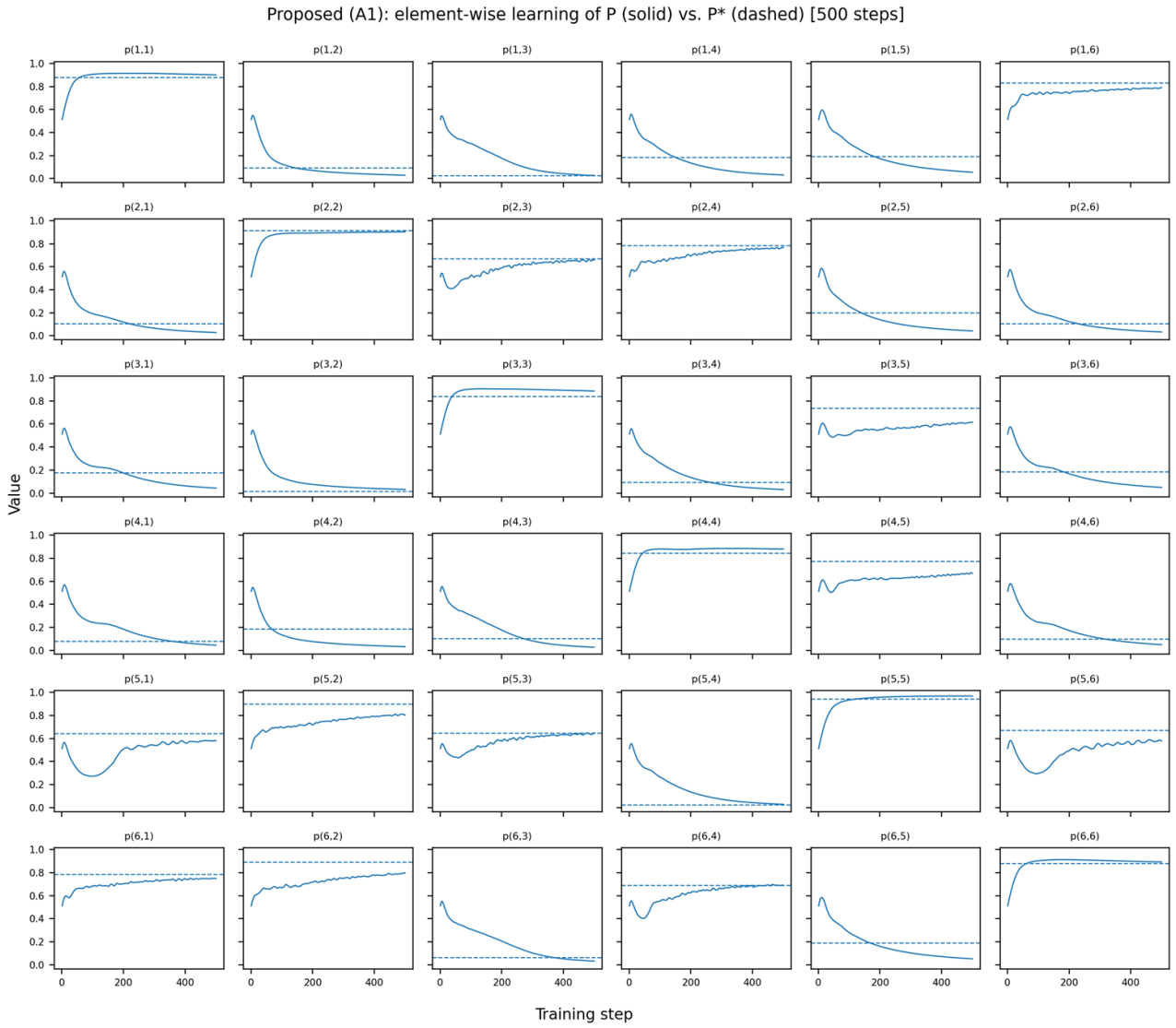


Figure 4. Elementwise trajectories of P for the proposed method (seed 0, 500 steps). Solid: learned p_{ij}; dashed: ground truth $p_{ij}^{\star}$.

Figure 4 shows the learning dynamics of each element p_{ij} (solid) together with its ground truth value $p_{ij}^{\star}$ (dashed). Dominant diagonal entries rapidly increase toward their target values, whereas most weak off-diagonal entries are suppressed by the ℓ_1 penalty and the diagonal regularizer. A small subset of moderate off-diagonal couplings converges more gradually, reflecting the need to accumulate evidence from diverse input memberships before the corresponding fuzzy relation becomes identifiable.

Table 1. Structure recovery performance (mean ± std over seeds). Lower is better.

Metric	Oracle fuzzifier	Fixed fuzzifier	Proposed (Algorithm 1)
$\left\|\hat{P} - P^{\star}\right\|_{\mathrm{F}}$	$3.75 \times 10^{-1} \pm 3.00 \times 10^{-3}$	$6.00 \times 10^{-1} \pm 6.00 \times 10^{-3}$	$5.18 \times 10^{-1} \pm 6.00 \times 10^{-3}$
rel. MAE(c)	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$1.50 \times 10^{-2} \pm 0.00 \times 10^{0}$	$3.07 \times 10^{-3} \pm 1.80 \times 10^{-3}$
rel. MAE(σ)	$0.00 \times 10^{0} \pm 0.00 \times 10^{0}$	$1.73 \times 10^{-1} \pm 0.00 \times 10^{0}$	$8.70 \times 10^{-2} \pm 4.00 \times 10^{-3}$
test MSE(s)	$5.77 \times 10^{-4} \pm 1.10 \times 10^{-5}$	$1.27 \times 10^{-3} \pm 3.10 \times 10^{-5}$	$8.14 \times 10^{-4} \pm 5.70 \times 10^{-5}$

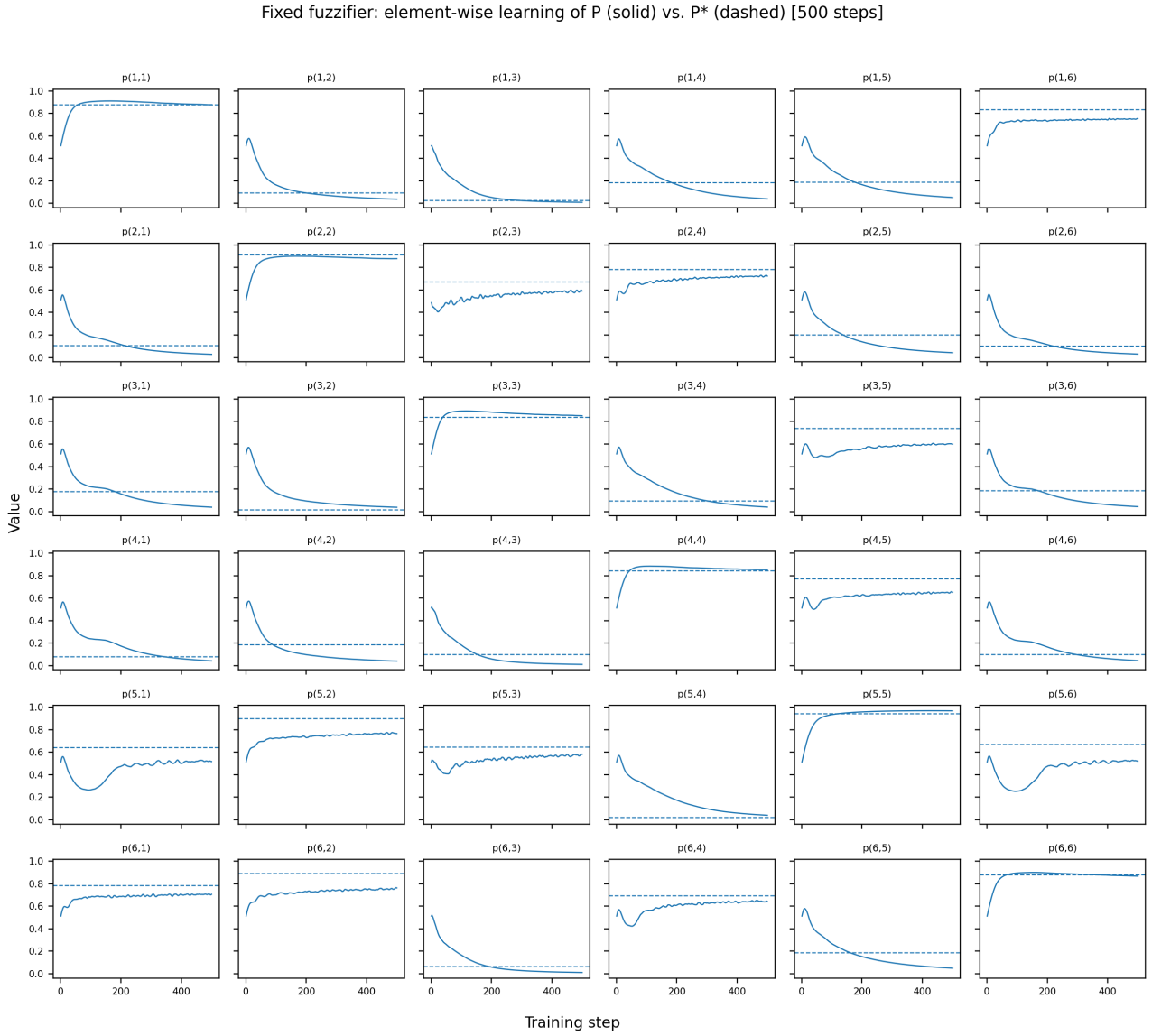


Figure 5. Elementwise trajectories of P for the Fixed baseline (seed 0, 500 steps). Solid: learned p_{ij}; dashed: ground truth $p_{ij}^{\star}$.

To highlight how the fuzzifier quality affects the identification of individual relations, Figures 5 and 6 report the same elementwise trajectories under the Fixed and Oracle baselines. With the oracle fuzzifier (true $(\boldsymbol{c}^{\star}, \boldsymbol{\sigma}^{\star})$), dominant coefficients typically approach their targets more rapidly and exhibit

less bias in their asymptotic plateaus, confirming that the remaining error primarily comes from finite-sample stochasticity and the softmax/softmin relaxation. In contrast, the fixed-fuzzifier baseline tends to settle at systematically shifted plateaus for a subset of moderate couplings, which is consistent with a persistent mismatch in membership degrees that cannot be compensated by P alone.

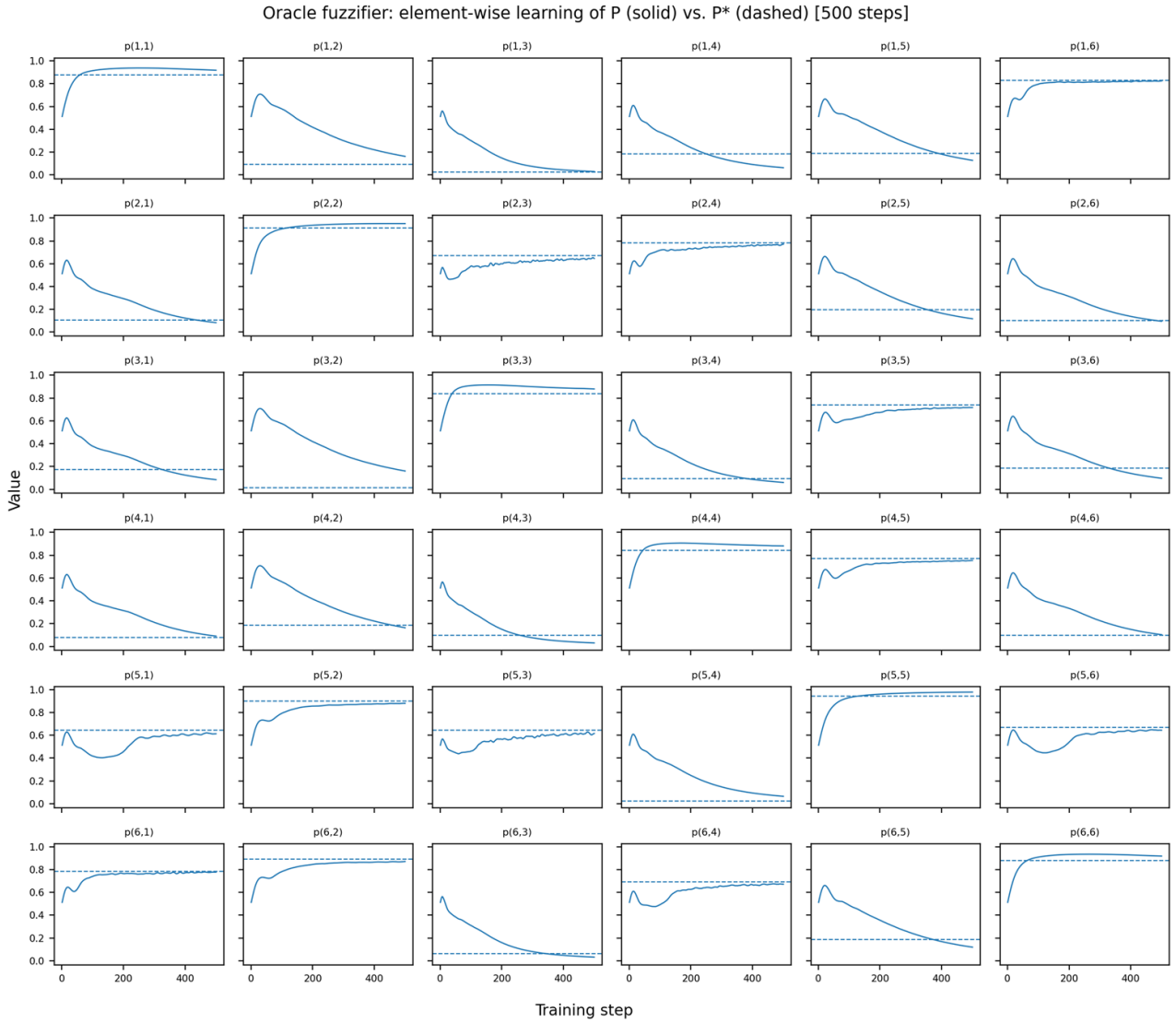


Figure 6. Elementwise trajectories of P for the Oracle baseline (seed 0, 500 steps). Solid: learned p_{ij}; dashed: ground truth $p^{\star}_{ij}$.

Figures 7 and 8 present the corresponding trajectories of c_i and σ_i. The center parameters c_i typically stabilize quickly (they are directly anchored by the sensor distribution and the prior), whereas the width parameters σ_i evolve more slowly due to their interaction with P in shaping membership degrees. After the warm-start stage (when θ learning is activated), both c_i and σ_i exhibit small corrective adjustments, while remaining stable overall a behavior, that indicates that the joint optimization is well-conditioned under the fuzzy-space supervision.

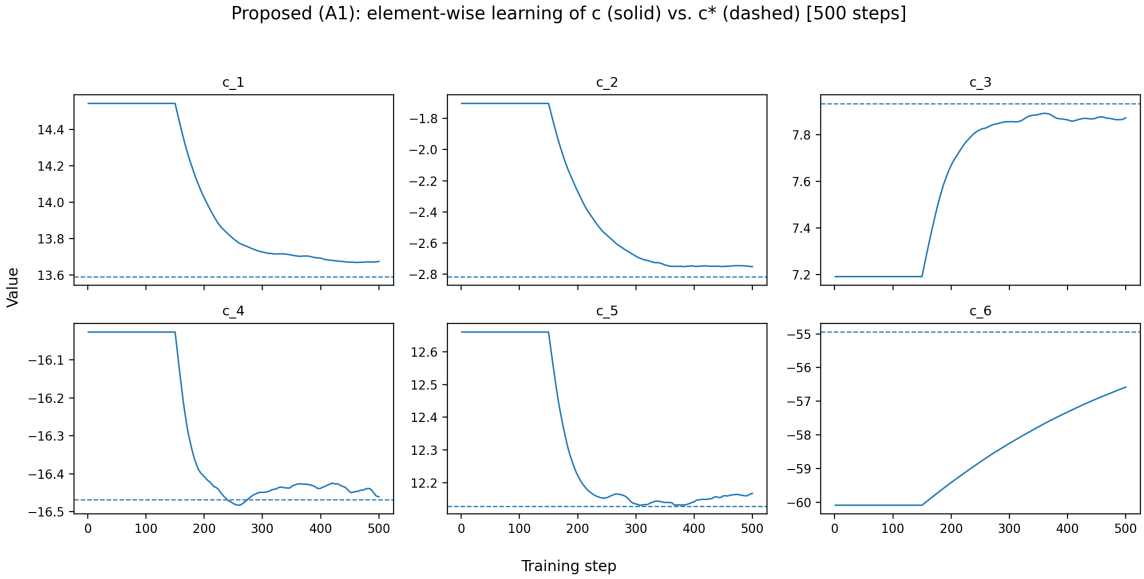


Figure 7. Elementwise learning dynamics of Gaussian fuzzifier parameters: centers c_i (seed 0, 500 steps). Solid: learned parameters; dashed: ground truth.

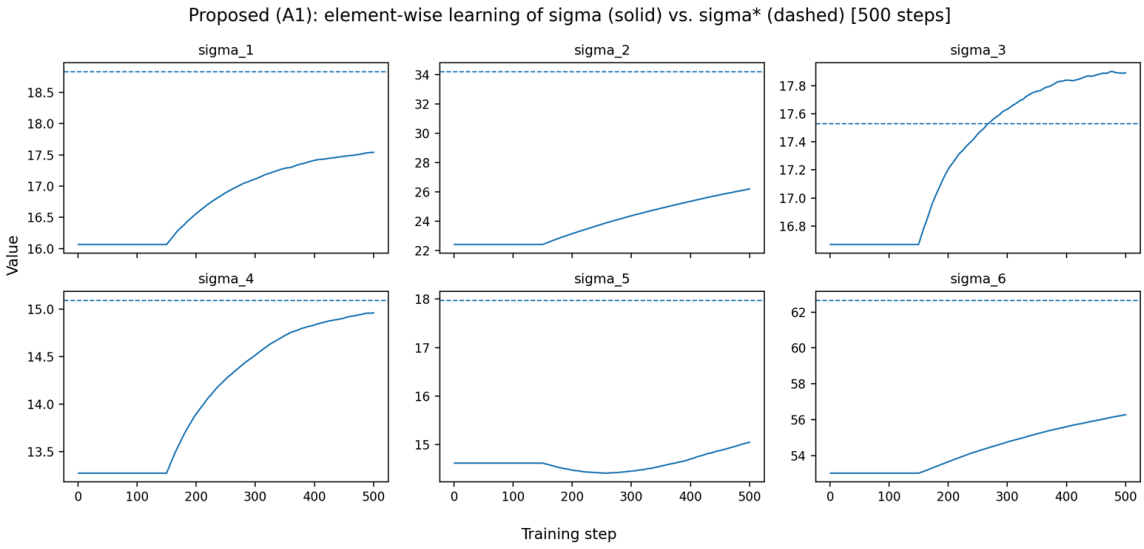


Figure 8. Elementwise learning dynamics of Gaussian fuzzifier parameters: widths σ_i (shown as "sigma" in the figure title; seed 0, 500 steps). Solid: learned parameters; dashed: ground truth.

7. Discussion

The fixed-fuzzifier baseline estimates a fuzzifier from marginal sensor statistics only. This prior is generally insufficient to disambiguate how sensor readings should be mapped to membership degrees to best explain temporal evolution, especially when channels have different signal-to-noise characteristics. Jointly optimizing $(\boldsymbol{c}, \boldsymbol{\sigma})$ with P allows the fuzzifier to adapt to temporal consistency

constraints induced by Eq (4.5), reducing mismatch between predicted and observed fuzzy states and, consequently, improving recovery of P.

The mapping $(P, \boldsymbol{c}, \boldsymbol{\sigma}) \mapsto \tilde{\boldsymbol{s}}_{t+1}$ can be nonidentifiable: Different parameter tuples may generate similar fuzzy transitions, particularly when τ is large (over-smoothing), when the sensor distribution lacks excitation, or when multiple channels have highly correlated memberships. Heuristically, uniqueness becomes more plausible when (i) the fuzzifier is fixed (or well anchored), (ii) the temperature is sufficiently small so that Eq (4.5) closely approximates true max-min composition, and (iii) the dataset explores diverse fuzzy states so that each column of P is "excited" by some samples. In practice, we therefore use diagonal/sparsity regularizers and a warm start (optimizing P before releasing θ) to bias the solution toward stable and interpretable structures.

Our implementation scales as $O(Bn^2)$ per mini-batch due to the dense pairwise tensor in Eq (4.5). For n in the low hundreds, this remains practical on modern GPUs/CPUs; for substantially larger n, potential mitigations include sparsity masks on P, low-rank parameterizations, and block-structured relations (domain-driven grouping of channels).

The optimization also depends on the warm-start length T_{warm} and the temperature schedule $(\tau_{\text{start}}, \tau_{\text{end}}, \gamma)$. A systematic sensitivity study (and guidelines for selecting these hyperparameters across datasets) is an important direction for future work.

First, our current fuzzifier uses one Gaussian per channel, producing a single membership degree per sensor. In applications that require multiple linguistic terms per channel (e.g., low/medium/high), the framework can be extended by using multiple Gaussians per channel and enlarging $\boldsymbol{s}_t$ accordingly. Second, the experiments in this paper are synthetic to enable ground-truth structural evaluation; validating the approach on real or semi-real sensor datasets (and comparing with additional black-box sequence baselines) is left for future work. Third, robustness and stability under distribution shift, sensor bias, or drift are not explicitly tested here. In practical deployments, one can incorporate standard sensor normalization, robust losses, and drift-aware updating of the fuzzifier prior $(\boldsymbol{c}_0, \boldsymbol{\sigma}_0)$; designing principled bias-handling mechanisms for fuzzy relational learning is a promising topic for further study.

8. Conclusions

We proposed an end-to-end method for learning a max-min input-output fuzzy relation matrix directly from adjacent-time sensor data by linking sensor readings to fuzzy vectors through Gaussian fuzzification. By introducing a differentiable softmax/softmin surrogate with temperature annealing and a warm-start optimization strategy, we obtained stable training and improved structure recovery compared with a fixed fuzzifier, while approaching oracle performance.

The current study focuses on a controlled synthetic setting to enable ground-truth evaluation. Future work will (i) validate the approach on real or semi-real sensor datasets, (ii) extend the fuzzifier to multiple linguistic terms per channel, and (iii) investigate robustness, hyperparameter sensitivity, and identifiability conditions for unique recovery.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the 2025 Key Domain Research Project of the Guangdong Provincial Department of Education, Research on Intelligent Diagnosis Methods and Applications of Middleware Performance Bottlenecks Based on Machine Learning (No. 2025ZDZX1077); the 2023 Guangdong Provincial University Engineering Technology Center Project, Guangdong VR/AR Application Integration Engineering Technology Development Center (No. 2023GCZX012); and the 2025 Guangdong Provincial Science and Technology Innovation Strategy Special Fund (University Student Science and Technology Innovation Cultivation) Project, Design and Implementation of an Intelligent Virtual Training System for Agricultural Production-Supporting the Agricultural Transformation and Upgrading of Jiaoling, Meizhou (No. pdjh2025bk367).

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. L. A. Zadeh, Fuzzy sets, *Inf. Control*, **8** (1965), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X
2. L. A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy Sets Syst.*, **1** (1978), 3–28. https://doi.org/10.1016/0165-0114(78)90029-5
3. M. Ye, N. Wang, X. Yu, X. Wang, W. Liu, Supervised learning fuzzy matrix based on input-output fuzzy vectors, *Axioms*, **14** (2025), 126. https://doi.org/10.3390/axioms14020126
4. H. Ying, F. Lin, Self-learning fuzzy automaton with input and output fuzzy sets for system modelling, *IEEE Trans. Emerging Top. Comput. Intell.*, **7** (2023), 500–512. https://doi.org/10.1109/TETCI.2022.3192890
5. H. Ying, F. Lin, Discrete-time finite fuzzy Markov chains realized through supervised learning stochastic fuzzy discrete event systems, *IEEE Trans. Fuzzy Syst.*, **32** (2024), 6088–6100. https://doi.org/10.1109/TFUZZ.2024.3440184
6. W. Liu, Q. He, Z. Li, Y. Li, Self-learning modeling in possibilistic model checking, *IEEE Trans. Emerging Top. Comput. Intell.*, **8** (2024), 264–278. https://doi.org/10.1109/TETCI.2023.3300189
7. J. M. A. Moral, C. Castiello, L. Magdalena, C. Mencar, *Explainable Fuzzy Systems: Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems*, 1st edition, Springer, 2021. https://doi.org/10.1007/978-3-030-71098-9
8. J. Cao, T. Zhou, S. Zhi, S. Lam, G. Ren, Y. Zhang, et al., Fuzzy inference system with interpretable fuzzy rules: Advancing explainable artificial intelligence for disease diagnosis–A comprehensive review, *Inf. Sci.*, **662** (2024), 120212. https://doi.org/10.1016/j.ins.2024.120212
9. S. Xu, Smoothing method for minimax problems, *Comput. Optim. Appl.*, **20** (2001), 267–279. https://doi.org/10.1023/A:1011211101714
10. A. Tsoukalas, P. Parpas, B. Rustem, A smoothing algorithm for finite min-max-min problems, *Optim. Lett.*, **3** (2009), 49–62. https://doi.org/10.1007/s11590-008-0090-9

11. L. Li, Z. Qiao, Y. Liu, Y. Chen, A convergent smoothing algorithm for training max-min fuzzy neural networks, *Neurocomputing*, **260** (2017), 404–410. https://doi.org/10.1016/j.neucom.2017.04.046
12. E. Sanchez, Resolution of Eigen fuzzy sets equations, *Fuzzy Sets Syst.*, **1** (1978), 69–74. https://doi.org/10.1016/0165-0114(78)90033-7
13. B. De Baets, E. E. Kerre, A primer on solving fuzzy relational equations on the unit interval, *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.*, **2** (1994), 205–225. https://doi.org/10.1142/S021848859400016X
14. W. Bandler, L. J. Kohout, Semantics of implication operators and fuzzy relational products, *Int. J. Man-Mach. Stud.*, **12** (1980), 89–116. https://doi.org/10.1016/S0020-7373(80)80055-1
15. J. R. Jang, ANFIS: Adaptive-network-based fuzzy inference system, *IEEE Trans. Syst. Man Cybern.*, **23** (1993), 665–685. https://doi.org/10.1109/21.256541
16. Y. Yan, D. Cheng, J. Feng, H. Li, J. Yue, Survey on applications of algebraic state space theory of logical systems to finite state machines, *Sci. China Inf. Sci.*, **66** (2023), 111201. https://doi.org/10.1007/s11432-022-3538-4
17. W. Li, Y. Wei, W. Xu, General expression of knowledge granularity based on a fuzzy relation matrix, *Fuzzy Sets Syst.*, **440** (2022), 149–163. https://doi.org/10.1016/j.fss.2022.01.007

AIMS Press