GAUSSIAN GRAPH(ICAL) NEURAL NETWORKS FOR REPEATED OBSERVATIONS OVER A FIXED GRAPH WITH PERSISTENT EDGE RELATIONSHIPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many networked systems are observed repeatedly over a fixed physical or relational structure. In applications such as environmental monitoring, individual observations may be sparse, noisy, or incomplete, making the current state difficult to estimate from measurements alone. However, repeated observations provide an opportunity to learn dependencies between nodes that persist across measurements and can constrain future inference. Existing graph neural networks do not naturally encode such dependencies, while classical Gaussian graphical models encode them explicitly but offer limited representational flexibility. We introduce Gaussian Graphical Neural Networks, which combine expressive neural representations with Gaussian graphical inference by learning a sparse precision matrix over the fixed graph. We develop three precision-matrix parameterizations that guarantee convergent inference, characterize the spectral behavior of the resulting operator and show that it imposes no inherent smoothing bias, and derive an implicit differentiation procedure that enables end-to-end training without unrolling inference. Experiments on spatial reconstruction and hydrological forecasting demonstrate that learning persistent graphical dependencies and performing structured inference over latent representations can improve estimation in repeated-observation settings.

1 INTRODUCTION

Many critical environmental systems depend on distributed sensor measurements for monitoring and prediction. River gauges support flood forecasting, weather stations inform responses to severe conditions, groundwater wells help characterize drought, and seismic arrays provide evidence about evolving geological events. Yet measurements from these networks are often incomplete or uncertain because sensors may fail or drop out, leaving portions of the system unobserved, while available measurements may be noisy or difficult to assess in isolation. Crucially, these networks are monitored repeatedly over long periods of time, producing a plethora of observations for the same underlying relational system. Across these observations, the state of the system changes, but the identities of the sensors and the relationships between them persist. When individual observations are insufficient to characterize the system reliably, these persistent relationships provide a critical source of information across observations. This repeated-observation setting therefore necessitates learning persistent dependencies that characterize how the system evolves, so that each new estimate is constrained by the system's expected behavior even when current measurements are incomplete or unreliable.

Existing graph learning approaches are not built around the problem of reconciling persistent dependencies across observations with incomplete or uncertain measurements of the current state. Neighborhood aggregation (Kipf & Welling, 2017; Hamilton et al., 2017; Velickovic et al., 2018), graph diffusion (Gasteiger et al., 2019; Chamberlain et al., 2021), graph equilibrium models (Gu et al., 2020; Liu et al., 2021), and graph transformers (Rampášek et al., 2022; Ying et al., 2021) transform and propagate node features using rules that are largely generic across edges, rather than specific to the particular pair of nodes they connect. Some spatiotemporal forecasting methods do learn persistent, node-specific adjacency structures (Wu et al., 2019; Bai et al., 2020), but the learned structure shapes feature aggregation rather than reconciling evidence against the graph.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Gaussian Graphical Models (GGMs), on the other hand, have long been used in spatial statistics and geostatistics for exactly this repeated-observation, fixed-topology problem (Rue & Held, 2005; Lindgren et al., 2011). GGMs model node variables as jointly Gaussian and use the graph to encode their conditional dependencies directly; inference then solves precisely the configuration of variables most consistent with both the observed evidence and the graph's dependency structure. This makes GGMs a natural fit for persistent relational systems, but classical GGMs pair this structure with simple, typically linear observation models, and are not designed to exploit expressive, nonlinear representations of the underlying signal.

This gap suggests a remedy: pair the GGM's capacity for structural reconciliation with expressive neural representations. Methods have combined neural networks with graphical or belief-propagation machinery before. Deep Gaussian Markov Random Fields and their extension to general graphs build a GMRF precision structure from neural network layers, but infer a single latent field from a single set of observations using coefficients shared across the graph (Sidén & Lindsten, 2020; Oskarsson et al., 2022). Deep Factor Graphs applies Gaussian belief propagation to the computation graph of a neural network, rather than to a persistent relational system (Nabarro et al., 2024). Neural Enhanced Belief Propagation assumes the graphical model is known and learns only a correction to its messages, so no dependency structure is estimated (Garcia Satorras & Welling, 2021). Work combining graph neural networks with conditional random fields instead targets discrete labels in a single graph snapshot, rather than continuous states across repeated observations (Qu et al., 2019; Wang et al., 2021). Yet, none of these methods learns persistent, edge-specific dependencies from a repeatedly observed graph while retaining the representational flexibility of neural models.

In light of this, we introduce Gaussian Graphical Neural Networks (GGNNs). GGNNs embed a Gaussian graphical model inside a graph neural network so that each estimate is reconciled against persistent, learned edge relationships. Unlike existing graph learning approaches, this provides an expressive model of persistent relational structure that can characterize the system across observations. To do this, we represent the graph's persistent dependencies as a learned precision matrix over its edges and develop three parameterizations that guarantee convergent graphical inference. We characterize the spectral behavior of the resulting operator, showing that GGNNs impose no inherent low-pass or high-pass bias, and analyze the learned precision matrices to characterize the structures they learn. We further develop a memory-efficient training strategy that reuses the same belief-propagation solver for the backward pass. Finally, across spatial reconstruction and hydrological forecasting tasks, we demonstrate the effectiveness of incorporating this graphical inductive bias into learned representations.

2 Gaussian Graphical Models

GGMs represent a system of continuous random variables as a multivariate Gaussian distribution $X \sim \mathcal{N}(\mu, \Sigma)$. The graph is represented through the precision matrix $J = \Sigma^{-1}$, which encodes the conditional dependency structure of the distribution. Nonzero off-diagonal entries J_{ij} indicate conditional dependencies between variables x_i and x_j, thereby defining the edges of the graph, while diagonal entries J_{ii} encode the local precision (self strength) associated with each variable (Malioutov et al., 2006; Bickson, 2008; Ortiz et al., 2021).

For a particular realization of the system, the observed evidence is encoded by an information vector $h \in \mathbb{R}^N$, where each entry h_i specifies the evidence associated with variable x_i. Inference on the graph therefore amounts to finding the configuration of variables that best explains the observed evidence while remaining consistent with the dependency structure encoded by the graph. This is formulated as the quadratic optimization problem

$$\mu = \arg\min_x \frac{1}{2} x^\top J x - h^\top x = J^{-1} h.$$

2.1 Constraints on Inference

Although the GGM's objective is defined over the full graph, its formulation allows for distributed inference through Gaussian Belief Propagation (GaBP). On tree-structured GGMs, messages propagate along a unique path between variables, allowing inference to be completed exactly after a finite sequence of upward and downward message-passing passes. The resulting marginals recover the exact solution to the global inference problem.

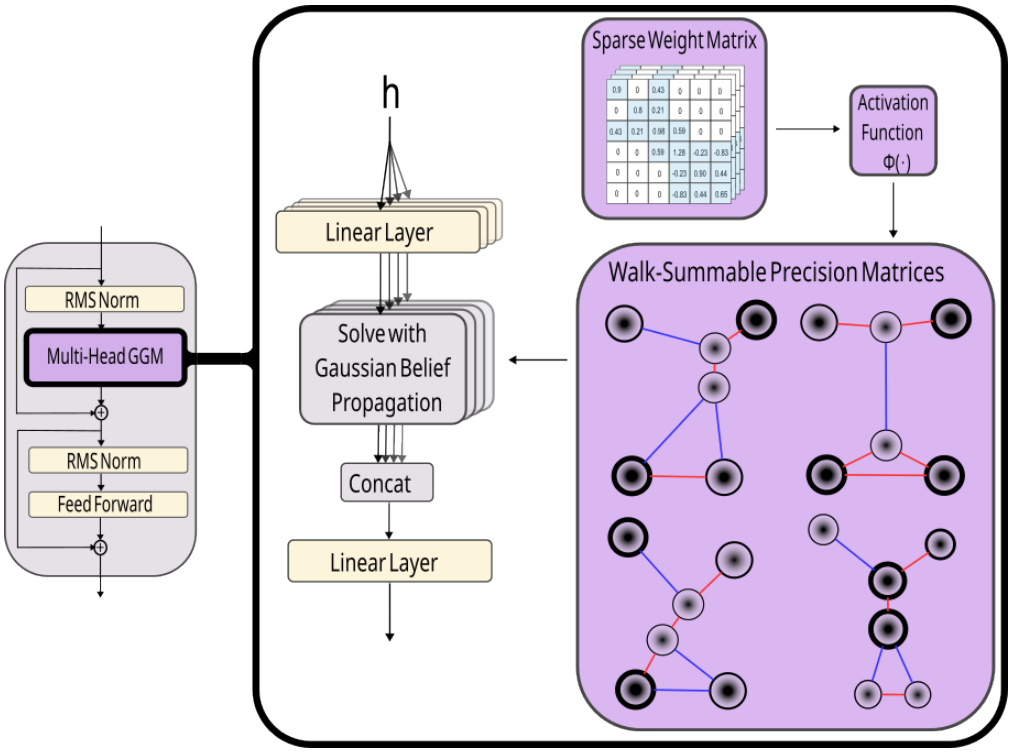

Figure 1: Gaussian Graphical Neural Network Block Architecture.

On graphs containing cycles, messages may revisit the same variables through multiple paths, so no finite message-passing schedule exists. Instead, messages are updated iteratively until convergence.[1] For the resulting iterative procedure to converge, the cumulative influence of these walks must remain finite. Walk-summability formalizes this requirement. Letting

$$\tilde{J} = D^{-1/2} J D^{-1/2}, \qquad \rho\Big(|I - \tilde{J}|\Big) < 1,$$

where D contains the diagonal entries of J, and $\rho(\cdot)$ denotes the spectral radius.

3 Gaussian Graphical Neural Network

The Gaussian Graphical Neural Network embeds Gaussian graphical inference within a GNN. To make this possible, we learn both a sparse weight matrix of the graph for the GGM along with node embeddings. A block of our model, illustrated in Figure 1, consists of a residual GGM subblock followed by a node-wise residual feed-forward network, each preceded by RMS normalization (Zhang & Sennrich, 2019). For each residual connection, we scale it by the inverse square root of the total number of blocks.

3.1 Information Vector Parameterization

To obtain the local evidence h, we apply RMS normalization followed by a learned linear projection to the node embeddings:

$$h = \mathrm{RMSNorm}(x)W, \qquad W \in \mathbb{R}^{d \times d}.$$

3.2 Precision Matrix Parameterization

Parameterizing the precision matrix is considerably more difficult than parameterizing the information vector. In principle, a classical GGM over multidimensional node embeddings must determine

[1] These iterations may be performed using either synchronous or asynchronous message-passing schedules.

the graph topology, the strengths of the interactions, and the matrix-valued dependencies between neighboring embeddings. In the fixed-graph, multi-observation setting considered here, the topology is known. Thus, we use the graph to predefine the sparsity pattern of the precision matrix and learn only the values of its permitted entries. To avoid matrix-valued interactions, we model each embedding dimension as an independent univariate GGM governed by the same sparse precision matrix. This allows inference to scale linearly with the embedding dimension. The remaining challenge is to constrain the learned precision matrix so that it is walk-summable. We, therefore, consider three such parameterizations.

Pairwise Normal Pairwise-normalizable GGMs span the same class of precision matrices as walk-summable models (Malioutov et al., 2006). This allows walk-summability to be enforced through local edge constraints. Specifically, each edge $e = (i, j)$ contributes a local precision block

$$J^{(ij)} = \begin{pmatrix} J_{ii}^{(ij)} & J_{ij} \\ J_{ij} & J_{jj}^{(ij)} \end{pmatrix}, \qquad J_{ii}^{(ij)} J_{jj}^{(ij)} > J_{ij}^2,$$

where $J_{ii}^{(ij)}$ and $J_{jj}^{(ij)}$ represent the contributions of edge (i, j) to the self-precisions of nodes i and j, respectively, while J_{ij} encodes their mutual coupling.

We parameterize each local block so that this constraint is satisfied by construction. The two self-precision contributions are constrained to be positive, while the coupling is bounded by their geometric mean:

$$J_{ii}^{(ij)} = \mathrm{Softplus}(\theta_{ii}^{(ij)}), \qquad J_{jj}^{(ij)} = \mathrm{Softplus}(\theta_{jj}^{(ij)}), \qquad J_{ij} = \gamma \sqrt{J_{ii}^{(ij)} J_{jj}^{(ij)}} \tanh(\theta_{ij}),$$

where $\theta_{ii}^{(ij)}$, $\theta_{jj}^{(ij)}$, and θ_{ij} are learned unconstrained parameters and $\gamma = 0.99$. Since $|J_{ij}| < \sqrt{J_{ii}^{(ij)} J_{jj}^{(ij)}}$, each local block satisfies the required constraint. The diagonal of the full precision matrix is then formed by aggregating the self-precision contributions from the edges incident to each node, together with an additional learned positive node precision:

$$J_{ii} = 1 + \mathrm{Softplus}(\theta_i) + \sum_{j \in \mathcal{N}(i)} J_{ii}^{(ij)},$$

where θ_i is a learned node-specific parameter. The resulting precision matrix is pairwise-normalizable and therefore walk-summable.

Diagonally Dominant Strictly diagonally dominant matrices are a subset of walk-summable matrices (Bickson, 2008). Such matrices are characterized by having each diagonal entry larger in magnitude than the sum of the absolute value of the off-diagonal entries in the same row. We enforce this condition by learning the off-diagonal entries directly and enforcing strict diagonal dominance by constructing

$$J_{ii} = 1 + \mathrm{Softplus}(\theta_i) + \sum_{j \neq i} |J_{ij}|.$$

Laplacian Laplacian-based matrices are not necessarily walk-summable because the eigenvalues of the walk matrix can lie in the range $[-1, 1]$. To enforce walk-summability, we introduce node-specific diagonal scaling and edge-specific interaction strengths and define

$$L = \mathrm{diag}(\beta) - \alpha \odot \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right), \qquad \lambda \in \left[- \max_{(i,j) \in \mathcal{E}} \frac{|\alpha_{ij}|}{\sqrt{\beta_i \beta_j}}, \max_{(i,j) \in \mathcal{E}} \frac{|\alpha_{ij}|}{\sqrt{\beta_i \beta_j}} \right],$$

where A is the weighted adjacency matrix, D is its corresponding degree matrix, and λ denotes an eigenvalue of the corresponding walk matrix. By ensuring $\beta_i \beta_j > \alpha_{ij}^2$ for every edge (i, j), the resulting precision matrix is walk-summable.

In practice, we parameterize the normalized interactions so that this constraint is satisfied by construction. We learn nonnegative edge weights together with bounded signed interaction coefficients:

$$w_{ij} = \sigma(\theta_{ij}), \qquad \alpha_{ij} = \gamma \sqrt{J_{ii} J_{jj}} \tanh(\psi_{ij}), \qquad d_i = \sum_{j \in \mathcal{N}(i)} w_{ij},$$

where θ_{ij} and ψ_{ij} are learned unconstrained edge parameters and $\gamma = 0.99$. The corresponding precision matrix entries are

$$J_{ii} = 1 + \mathrm{Softplus}(\beta_i), \qquad J_{ij} = -\alpha_{ij}\frac{w_{ij}}{\sqrt{d_i d_j}}.$$

Since $J_{ii} > 1$ and $|\alpha_{ij}| < \gamma < 1$, the required bound is satisfied by construction.

Precision Matrix Initialization. To avoid imposing an arbitrary and overly strong dependency structure, we initialize each precision matrix close to the identity. Diagonal entries are initialized near one and off-diagonal entries near zero, with small zero-mean Gaussian perturbations added to all entries to break symmetry.

3.3 Spectral Bias and Depth

A natural concern is whether GGNN is inherently biased toward smoothing. We show that neither walk-summability nor depth forces such a bias (Proof in Appendix A.5).

Theorem 1 (No forced spectral bias at a single layer)**.** *Let $q = \rho(|I - \widetilde{J}|) < 1$. Walk-summability bounds the gain of every spectral mode to $[\frac{1}{1+q}, \frac{1}{1-q}]$ without fixing how these gains are ordered across frequencies. Consequently, walk-summable precision matrices can realize low-pass, spectrally flat, or high-pass responses.*

Corollary 1 (Depth does not force smoothing)**.** *For every depth L, the class of walk-summable L-layer GGNNs contains low-pass, spectrally flat, and high-pass configurations. Thus, depth alone does not impose any fixed spectral preference.*

3.4 Multi-Head GGMs

Modeling each latent dimension independently and using a shared graph constrains the range of dependency structures that can be represented. To increase model capacity, we introduce multi-head GGMs, in which each head defines an independent graphical model over the shared graph structure. Each head learns its own walk-summable precision matrix J^k and information vector h^k, performs GaBP, and produces a head-specific latent representation in parallel. The resulting representations are concatenated and projected back to the original feature dimension.

3.5 Memory Efficient Training

Because GaBP is an iterative inference procedure, naively training through it requires storing the intermediate messages produced at each iteration. This can be memory-intensive, especially when many iterations are required for convergence. To address this, we adopt an implicit differentiation strategy that computes gradients at convergence without unrolling the iterations. At convergence, the GGM satisfies $\mu = J^{-1}h$. Applying reverse-mode implicit differentiation and using the symmetry of J gives the adjoint system

$$J\,\frac{d\mathcal{L}}{dh} = \frac{d\mathcal{L}}{d\mu}.$$

The backward pass therefore requires solving the same linear system over the same graph with a different right-hand side, allowing us to reuse the forward GaBP solver without storing the full inference trajectory. This reduces the memory complexity from $\mathcal{O}(T(E+N)d)$ for T unrolled iterations to $\mathcal{O}((E+N)d)$, at the cost of an additional GaBP solve during the backward pass.

4 Experiments

To study the effectiveness of this method, we consider two tasks where graph edges can be considered persistent: spatial reconstruction under extreme observation sparsity and forecasting on a river network. For each task, we compare different ways of utilizing the graph, ranging from models that ignore the graph to models whose computation is constrained by or supplemented with its structure. Specifically, we evaluate a Transformer as a fully connected non-graph baseline, Graph Convolutional Networks (GCNs) as a local aggregation method, a modified GCN with learnable edge weights



(a) USCRN (b) LamaH-CE

Figure 2: Spatial structure of the benchmark graphs. USCRN edges connect geographically nearby weather stations, while LamaH-CE edges encode downstream hydrological connectivity.

(GCN-E), Implicit Graph Neural Networks (IGNNs) as equilibrium-based propagation, Generalized PageRank Graph Neural Networks (GPR-GNNs) as learnable diffusion, and GraphGPS as a hybrid method combining graph-local propagation with global attention. Moreover, we use the same residual architecture and vary only the graph operation to isolate the effect of how graph structure is used. Each method is given a comparable parameter budget and trained using mean squared error. Details and hyperparameters are provided in Section A.3.

4.1 USCRN Spatial Temperature Imputation.

4.1.1 Experiment Overview

The first task is a spatial temperature imputation task derived from the United States Climate Reference Network (USCRN) Diamond et al. (2013). The graph consists of weather stations distributed throughout the continental United States, where nodes correspond to stations and edges encode spatial relationships between nearby stations. In total, the graph contains 131 stations and 348 unique edges as seen in Figure 2a.

At each timestep, every station is associated with a temperature measurement and a binary observation mask indicating whether that measurement is available. The objective is to reconstruct temperatures at unobserved stations from the subset of observed stations and the graph structure. We compare two input representations: one consisting only of temperature observations and observation masks, and another that also contains node coordinates as inputs.

The model is trained using a single masking strategy in which 70% of stations are randomly masked at each training example. Generalization is evaluated under three masking configurations designed to test both interpolation and extrapolation capabilities: 1) Random masking where 70% of stations are randomly masked. 2) Western-region masking where all stations located from the mid line to the western United States are masked. 3) Eastern-region masking where all stations located from the mid line to the eastern United States are masked. More details about the construction of this dataset can be found in Appendix A.1.

We additionally compare against a standalone GGM in observation space to isolate the contribution of learned representations beyond structured probabilistic inference alone (Meinshausen & Bühlmann, 2010). Since the USCRN graph contains cycles, GGNN inference is performed using the simultaneous GaBP schedule. Effective convergence under the learned precision matrices is achieved using a fixed budget of 125 message-passing iterations for all GGNN variants during both training and evaluation.

4.1.2 Experiment Results

The USCRN results are reported in Table 1. GCN, IGNN, and GPR-GNN perform similarly across all masking settings despite utilizing different graph-constrained propagation mechanisms. In contrast, GCN-E yields a large reduction in error; this suggests that learning persistent edge-specific relationships is considerably more important than changing the propagation mechanism alone. GGNN improves further: under random masking, the best GGNN variants reduce MAE from 1.38 for GCN-

Table 1: USCRN spatial infilling performance with and without positional encodings. We report the mean and standard deviation over 10 runs. Best results are shown in bold.

(a) Without Positional Encodings

	Predict 70% from 30%		Predict East from West		Predict West from East	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE
GCN	8.08 ± 0.04	9.94 ± 0.02	8.15 ± 0.08	9.74 ± 0.07	8.51 ± 0.04	10.78 ± 0.07
IGNN	8.13 ± 0.02	9.97 ± 0.01	8.16 ± 0.09	9.74 ± 0.08	8.50 ± 0.02	10.76 ± 0.04
GPR-GNN	8.08 ± 0.02	9.95 ± 0.02	8.15 ± 0.05	9.74 ± 0.04	8.51 ± 0.02	10.78 ± 0.04
GCN-E	1.38 ± 0.01	2.15 ± 0.01	4.54 ± 0.12	6.11 ± 0.13	4.36 ± 0.09	5.91 ± 0.14
Transformer	5.38 ± 0.00	6.82 ± 0.00	5.91 ± 0.03	7.54 ± 0.04	6.61 ± 0.03	8.34 ± 0.04
GraphGPS	1.60 ± 0.01	2.38 ± 0.01	3.66 ± 0.13	4.99 ± 0.12	3.56 ± 0.10	4.81 ± 0.13
GGM	3.28 ± 0.00	4.37 ± 0.00	6.38 ± 0.00	7.65 ± 0.00	6.67 ± 0.00	8.07 ± 0.00
Ours						
GGNN-PN	1.33 ± 0.01	2.09 ± 0.01	3.33 ± 0.16	4.48 ± 0.18	3.43 ± 0.13	4.60 ± 0.17
GGNN-DD	$\mathbf{1.32 \pm 0.01}$	$\mathbf{2.07 \pm 0.01}$	$\mathbf{3.24 \pm 0.16}$	$\mathbf{4.39 \pm 0.26}$	$\mathbf{3.37 \pm 0.10}$	$\mathbf{4.51 \pm 0.12}$
GGNN-L	$\mathbf{1.32 \pm 0.01}$	$\mathbf{2.07 \pm 0.01}$	3.25 ± 0.11	4.43 ± 0.17	3.41 ± 0.14	4.60 ± 0.21

(b) With Positional Encodings

	Predict 70% from 30%		Predict East from West		Predict West from East	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE
GCN	7.36 ± 0.03	8.88 ± 0.02	7.11 ± 0.05	8.54 ± 0.02	7.76 ± 0.03	9.40 ± 0.04
IGNN	7.40 ± 0.03	8.91 ± 0.02	7.10 ± 0.04	8.53 ± 0.02	7.74 ± 0.02	9.37 ± 0.02
GPR-GNN	7.42 ± 0.02	8.95 ± 0.01	7.11 ± 0.02	8.55 ± 0.02	7.77 ± 0.02	9.42 ± 0.10
GCN-E	1.38 ± 0.01	2.14 ± 0.01	4.58 ± 0.12	6.14 ± 0.12	4.36 ± 0.03	5.89 ± 0.09
Transformer	1.36 ± 0.01	2.11 ± 0.01	3.78 ± 0.04	5.33 ± 0.09	3.84 ± 0.06	5.17 ± 0.08
GraphGPS	$\mathbf{1.31 \pm 0.01}$	$\mathbf{2.06 \pm 0.01}$	$\mathbf{2.98 \pm 0.11}$	$\mathbf{4.00 \pm 0.10}$	$\mathbf{3.13 \pm 0.07}$	$\mathbf{4.18 \pm 0.08}$
Ours						
GGNN-PN	1.33 ± 0.01	2.08 ± 0.01	3.35 ± 0.09	4.51 ± 0.11	3.57 ± 0.16	4.78 ± 0.22
GGNN-DD	$\mathbf{1.31 \pm 0.01}$	$\mathbf{2.06 \pm 0.01}$	3.25 ± 0.10	4.39 ± 0.11	3.43 ± 0.11	4.60 ± 0.16
GGNN-L	1.32 ± 0.01	2.07 ± 0.01	3.24 ± 0.17	4.38 ± 0.21	3.48 ± 0.15	4.67 ± 0.17

E to 1.32, and for extrapolation tasks, predicting East from West, GGNN reduces GCN-E's error from 4.58 to 3.25 MAE, a 29% relative reduction. These results suggest that graphical inference provides substantial benefit beyond learning edge-specific weights.

With positional encodings, GGNN's advantage over GraphGPS narrows and depends on the masking method. Under random masking, GGNN-DD matches GraphGPS (1.31 vs. 1.31 MAE); under the structured geographic masks, GraphGPS pulls ahead (2.98 vs. 3.25 MAE predicting East from West, a 9% relative gap). Notably however, GraphGPS's contains unconstrained global attention, which can draw on any station's coordinates directly, whereas GGNN must rely on long propagation along the fixed graph. This is a particular disadvantage under regional masking, where a station's graph neighbors provide no evidence for an extended period. Even so, on the random masking it was trained on, GGNN remains competitive with global reasoning despite operating under a fixed local graph.

4.2 LamaH-CE Hydrological Forecasting.

4.2.1 Experiment Overview

The second task is hydrological forecasting derived from the LamaH-CE dataset Klingler et al. (2021). Further dataset details, preprocessing, and acknowledgments of the underlying data sources are provided in Appendix A.2. The dataset consists of hourly streamflow measurements collected at river gauges throughout Central Europe. We represent the river network as a tree graph in which nodes correspond to gauges and edges encode upstream-downstream connectivity as seen in Figure 2b. The resulting graph contains 608 nodes, a maximum depth of 25, a mean depth of 11.25, and a maximum diameter of 29. The forecasting task is defined as follows. Given the previous 24 hours of streamflow observations at all gauges, the model must predict streamflow 6, 12, 24, 48, 72, and

Table 2: LamaH-CE forecasting performance across representative prediction horizons with and without positional encodings. We report the mean and standard deviation over 10 runs. Best results are shown in bold. Full results, including 72- and 96-hour horizons, are reported in Appendix A.6.

(a) Without Positional Encodings

	6 hours		12 hours		24 hours		48 hours	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
GCN	2.76 ± 0.09	16.51 ± 0.55	3.86 ± 0.14	25.47 ± 0.54	5.24 ± 0.10	36.93 ± 0.34	7.32 ± 0.10	50.64 ± 0.27
IGNN	2.73 ± 0.09	16.25 ± 0.32	3.82 ± 0.11	25.28 ± 0.29	5.26 ± 0.06	36.84 ± 0.24	7.35 ± 0.06	50.34 ± 0.18
GPR-GNN	2.83 ± 0.17	17.30 ± 1.60	3.81 ± 0.18	25.64 ± 1.41	5.27 ± 0.14	37.25 ± 0.74	7.29 ± 0.09	50.47 ± 0.32
GCN-E	2.26 ± 0.06	13.30 ± 0.39	3.17 ± 0.07	20.04 ± 0.51	4.39 ± 0.08	30.67 ± 0.33	6.47 ± 0.08	46.04 ± 0.27
Transformer	2.49 ± 0.17	13.73 ± 1.05	3.17 ± 0.07	18.64 ± 0.52	4.35 ± 0.15	27.34 ± 0.48	6.60 ± 0.23	44.37 ± 1.23
GraphGPS	2.53 ± 0.21	14.91 ± 1.18	3.23 ± 0.19	19.55 ± 0.93	4.46 ± 0.31	28.18 ± 1.47	6.55 ± 0.21	$\mathbf{43.54 \pm 0.51}$
Ours								
GGNN-PN	2.29 ± 0.18	13.22 ± 0.88	3.05 ± 0.18	18.57 ± 1.13	4.30 ± 0.16	28.46 ± 0.88	6.46 ± 0.17	44.66 ± 0.54
GGNN-DD	$\mathbf{2.23 \pm 0.08}$	$\mathbf{12.84 \pm 0.47}$	3.06 ± 0.14	18.19 ± 0.84	4.21 ± 0.17	27.71 ± 0.83	6.42 ± 0.25	44.12 ± 0.69
GGNN-L	2.25 ± 0.18	12.98 ± 0.89	$\mathbf{2.98 \pm 0.18}$	$\mathbf{17.76 \pm 0.81}$	$\mathbf{4.07 \pm 0.13}$	$\mathbf{26.86 \pm 1.04}$	$\mathbf{6.23 \pm 0.23}$	43.61 ± 0.99

(b) With Positional Encodings

	6 hours		12 hours		24 hours		48 hours	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
GCN	2.72 ± 0.13	16.41 ± 0.77	3.72 ± 0.09	25.24 ± 0.69	5.21 ± 0.12	37.20 ± 0.37	7.22 ± 0.13	50.47 ± 0.49
IGNN	2.73 ± 0.12	16.25 ± 0.88	3.75 ± 0.10	25.20 ± 0.55	5.25 ± 0.13	36.92 ± 0.36	7.19 ± 0.16	50.03 ± 0.31
GPR-GNN	2.74 ± 0.14	16.81 ± 0.92	3.78 ± 0.13	25.79 ± 0.81	5.19 ± 0.13	37.44 ± 0.60	7.16 ± 0.12	50.51 ± 0.39
GCN-E	2.36 ± 0.08	13.72 ± 0.43	3.18 ± 0.12	19.93 ± 0.52	4.56 ± 0.22	31.07 ± 0.57	6.63 ± 0.18	46.05 ± 0.33
Transformer	2.56 ± 0.46	14.51 ± 2.89	3.12 ± 0.37	$\mathbf{18.19 \pm 2.12}$	4.29 ± 0.35	$\mathbf{26.71 \pm 1.61}$	6.55 ± 0.18	$\mathbf{42.46 \pm 0.73}$
GraphGPS	2.55 ± 0.23	15.47 ± 2.11	3.34 ± 0.23	20.16 ± 2.78	4.34 ± 0.26	27.50 ± 1.95	6.42 ± 0.29	43.02 ± 1.16
Ours								
GGNN-PN	$\mathbf{2.26 \pm 0.15}$	$\mathbf{13.27 \pm 0.69}$	$\mathbf{3.06 \pm 0.13}$	18.75 ± 0.63	4.40 ± 0.14	28.85 ± 0.55	6.50 ± 0.21	44.80 ± 0.49
GGNN-DD	2.32 ± 0.23	13.56 ± 1.21	$\mathbf{3.06 \pm 0.14}$	18.65 ± 0.93	$\mathbf{4.27 \pm 0.19}$	27.92 ± 0.77	$\mathbf{6.40 \pm 0.23}$	44.67 ± 1.21
GGNN-L	2.48 ± 0.22	14.75 ± 1.38	3.15 ± 0.16	19.33 ± 1.48	4.35 ± 0.19	28.23 ± 1.72	6.61 ± 0.42	44.79 ± 1.75

96 hours into the future. Additionally, we perform two runs on this dataset, one with latitude and longitude as inputs and one without. Performance is evaluated independently at each forecasting horizon.

4.2.2 Experiment Results

The forecasting results are reported in Table 2. Consistent with the USCRN experiments, GCN, IGNN, and GPR-GNN perform similarly across forecasting horizons despite using different propagation mechanisms. GCN-E substantially reduces error relative to these conventional graph baselines. GGNN improves further on most horizons and metrics. Without positional encodings, GGNN achieves the lowest MAE at all four reported horizons and the lowest RMSE at 6, 12, and 24 hours, while GraphGPS obtains a slightly lower RMSE at 48 hours (43.54 vs. 43.61). At 24 hours, for example, MAE decreases from approximately 5.25 for the conventional graph baselines to 4.39 for GCN-E, 4.35 for the Transformer, 4.46 for GraphGPS, and 4.07 for GGNN-L.

With positional encodings, GGNN's advantage over the strongest global-attention baselines narrows and depends on the metric and forecasting horizon. On MAE, GGNN remains the best model at every reported horizon (6, 12, 24, and 48 hours). On RMSE, GGNN performs best at the shortest horizon (6 hours), while the Transformer takes the lead at 12, 24, and 48 hours. Nevertheless, GGNN remains competitive with unconstrained global attention despite operating under a fixed graph.

4.3 How much Inference is Necessary?

	6 hours		12 hours		24 hours		48 hours	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
GGNN-L (Tree)	2.25 ± 0.18	12.98 ± 0.89	2.98 ± 0.18	17.76 ± 0.81	4.07 ± 0.13	26.86 ± 1.04	6.23 ± 0.23	43.61 ± 0.99
GGNN-L (20)	2.23 ± 0.08	13.13 ± 0.67	2.94 ± 0.11	17.85 ± 1.12	4.08 ± 0.13	27.04 ± 1.14	6.16 ± 0.23	43.58 ± 0.84
GGNN-L (10)	2.39 ± 0.11	14.24 ± 0.64	3.17 ± 0.16	19.34 ± 1.08	4.32 ± 0.12	28.69 ± 0.57	6.60 ± 0.31	45.86 ± 1.48
GGNN-L (5)	2.42 ± 0.15	14.30 ± 0.88	3.18 ± 0.19	19.90 ± 1.15	4.40 ± 0.13	29.62 ± 0.79	6.51 ± 0.22	45.43 ± 0.69
GGNN-L (2)	2.42 ± 0.11	14.37 ± 0.41	3.30 ± 0.12	21.03 ± 0.22	4.59 ± 0.12	31.55 ± 0.33	6.62 ± 0.17	46.56 ± 1.75

Table 3: Forecasting performance across prediction horizons with different convergences. We report the mean and standard deviation over 10 runs.

To examine how GGNN performance depends on the extent of graphical inference, we evaluate the Laplacian GGNN using both the exact tree-specific message-passing schedule and simultaneous message passing truncated after 20, 10, 5, and 2 iterations. Apart from the inference procedure, all aspects of training and evaluation remain unchanged. Table 3 reports the resulting forecasting performance. Performance generally improves as the iteration budget increases, with 20 iterations approaching the performance of exact tree inference across forecasting horizons.

4.4 Precision Matrix Analysis

Although the three precision parameterizations achieve similar predictive performance, they impose different constraints on the learned precision matrices. We therefore examine the walk-summability spectral radius $\rho\left(\left|I - \widetilde{J}\right|\right)$ of the resulting matrices after training, as shown in Figure 3, where smaller values correspond to tighter bounds on spectral-mode gain in Theorem 1. Although all three parameterizations are initialized close to the identity, they learn distinct precision structures and occupy different regions of the feasible space. For a complementary view of how the parameterizations shape the learned precision matrices, we report the distributions of their diagonal and off-diagonal entries in Appendix A.7.

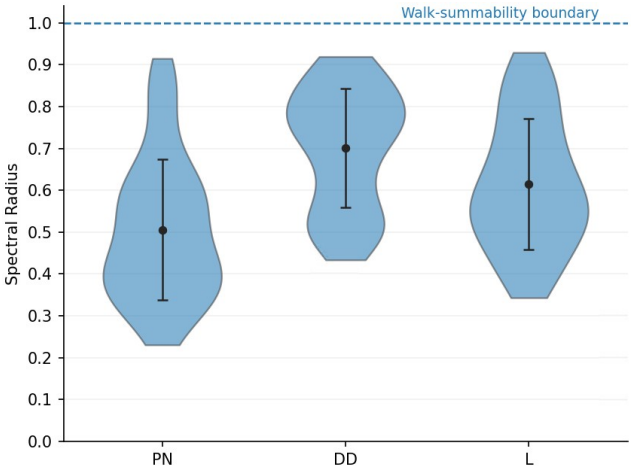

Figure 3: Walk-summability spectral radius $\rho\left(\left|I - \widetilde{J}\right|\right)$ of the learned precision matrices after training. The dashed line denotes the walk-summability boundary at 1.

4.5 Runtime Analysis

Table 4 reports the average training time per batch and peak GPU memory usage on USCRN with a batch size of 256. All measurements were obtained using an NVIDIA RTX A6000 GPU. GGNN incurs a higher computational cost than the other methods; however, established acceleration techniques, such as multigrid methods Briggs et al. (2000), could potentially reduce this overhead. Despite its higher computational cost, GGNN remains competitive in terms of peak GPU memory usage.

Model	Time/Batch	Memory
GCN	29.2	0.54
IGNN	73.9	0.50
GPR-GNN	108.5	4.13
Transformer	30.4	0.46
GGNN	160.2	0.46
GraphGPS	36.1	0.70

Table 4: Average training time per batch in milliseconds and peak GPU memory usage in gigabytes on USCRN.

5 Conclusion

We introduced Gaussian Graphical Neural Networks (GGNNs), which combine expressive neural representations with Gaussian graphical inference for settings in which repeated observations share a fixed relational graph. GGNNs learn persistent edge-specific dependencies and perform inference over latent node representations using precision-matrix parameterizations that guarantee walk-summability, while implicit differentiation enables end-to-end training without unrolling the full inference procedure.

Across spatial reconstruction and hydrological forecasting, our results show that learning persistent edge-specific relationships provides substantial gains over conventional graph propagation, and that incorporating graphical inference can improve further over learned edge-weight baselines. These results suggest that explicitly modeling persistent relational structure can provide a useful inductive bias when repeatedly reasoning over the same graph. GGNNs are currently limited to fixed graphs and Gaussian dependency models, and iterative inference introduces additional computational cost. Improving inference efficiency and extending the approach to a more general settings are promising directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools to polish the paper and verify the mathematical proofs. We have not used generative AI tools for any other tasks in this work. We have reviewed all AI-assisted work, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The code for this paper can be found at XXX.github.com. The datasets can be found at XXX.datasets.com. More details about the datasets can be found in the appendix.

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17804–17815. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ce1aad92b939420fc17005e5461e6f48-Paper.pdf.

Bavarian State Office for the Environment – Hydrographic Service. Runoff data, 2020. URL https://www.gkd.bayern.de/en/rivers/discharge/tables. Runoff data downloaded 15 September 2020.

Danny Bickson. *Gaussian Belief Propagation: Theory and Application*. PhD thesis, The Hebrew University of Jerusalem, 2008. URL https://www.cs.huji.ac.il/~dolev/pubs/thesis/phd-thesis-bickson.pdf.

William Briggs, Van Henson, and Steve McCormick. *A Multigrid Tutorial, 2nd Edition*. SIAM: Society for Industrial and Applied Mathematics, 01 2000. ISBN 978-0-89871-462-3.

Patrick D. Broxton, Xubin Zeng, William Scheftic, and Peter A. Troch. A MODIS-based global 1-km maximum green vegetation fraction dataset. *Journal of Applied Meteorology and Climatology*, 53:1996–2004, 2014. doi: 10.1175/JAMC-D-13-0356.1.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1407–1418. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/chamberlain21a.html.

Czech Hydrometeorological Institute (CHMI). Czech hydrometeorological institute, brno, czech republic, 2020. Runoff data received 14 December 2020.

Howard J. Diamond, Thomas R. Karl, Michael A. Palecki, C. Bruce Baker, Jesse E. Bell, Ronald D. Leeper, David R. Easterling, Jay H. Lawrimore, Tilden P. Meyers, Michael R. Helfert, Grant Goodge, and Peter W. Thorne. U.S. climate reference network after one decade of operations: Status and assessment. *Bulletin of the American Meteorological Society*, 94(4):485 – 498, 2013. doi: 10.1175/BAMS-D-12-00170.1. URL https://journals.ametsoc.org/view/journals/bams/94/4/bams-d-12-00170.1.xml.

European Commission and European Soil Bureau Network. The european soil database distribution version 2.0, 2004. Dataset, CD-ROM, EUR 19945 EN.

European Environment Agency. CORINE land cover 2012, 2012. Dataset.

European Environment Agency. EU-Hydro – river network database, version 1.2, 2019. Dataset produced under the framework of the Copernicus programme.

Tom G. Farr, Paul A. Rosen, Edward Caro, Robert Crippen, Riley Duren, Scott Hensley, Michael Kobrick, Mimi Paller, Ernesto Rodriguez, Ladislav Roth, David Seal, Scott Shaffer, Joanne Shimada, Jeffrey Umland, Marian Werner, Michael Oskin, Douglas Burbank, and Douglas E. Alsdorf. The shuttle radar topography mission. *Reviews of Geophysics*, 45:RG2004, 2007. doi: 10.1029/2005RG000183.

Federal Ministry of Agriculture, Regions and Tourism – Hydrographic Central Office. Hydrological atlas of austria (dighao), 3rd delivery, 2007. Dataset.

Federal Ministry of Agriculture, Regions and Tourism – Hydrographic Central Office. Hydrographic yearbook of austria 2013, 2013.

Federal Ministry of Agriculture, Regions and Tourism – Hydrographic Central Office. Runoff data, 2020. Runoff data received 8 September 2020.

Federal Office for the Environment (BAFU). Hydrology division, bern, switzerland, 2020. Runoff data received 23 September 2020.

Mark Friedl and Damien Sulla-Menashe. MCD12Q1 MODIS/Terra+Aqua Land Cover Type Yearly L3 Global 500m SIN Grid V006, 2019.

Víctor Garcia Satorras and Max Welling. Neural enhanced belief propagation on factor graphs. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 685–693. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/garcia-satorras21a.html.

Johannes Gasteiger, Stefan Weiß enberger, and Stephan Günnemann. Diffusion improves graph learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/23c894276a2c5a16470e6a31f4618d73-Paper.pdf.

Tom Gleeson, Nils Moosdorf, Jens Hartmann, and Ludovic P. H. van Beek. A glimpse beneath earth's surface: GLobal HYdrogeology MaPS (GLHYMPS) of permeability and porosity. *Geophysical Research Letters*, 41:3891–3898, 2014. doi: 10.1002/2014GL059856.

Fangda Gu, Heng Chang, Wenwu Zhu, Somayeh Sojoudi, and Laurent El Ghaoui. Implicit graph neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11984–11995. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8b5c8441a8ff8e151b191c53c1842a38-Paper.pdf.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

Jens Hartmann and Nils Moosdorf. The new global lithological map database GLiM: A representation of rock properties at the earth surface. *Geochemistry, Geophysics, Geosystems*, 13:1–37, 2012. doi: 10.1029/2012GC004370.

Roland Hiederer. Mapping soil properties for europe – spatial representation of soil database attributes. Technical Report EUR26082EN, Publications Office of the European Union, Luxembourg, 2013a.

Roland Hiederer. Mapping soil typologies – spatial decision support applied to european soil database. Technical Report EUR25932EN, Publications Office of the European Union, Luxembourg, 2013b.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

C. Klingler, K. Schulz, and M. Herrnegger. LamaH-CE: Large-sample data for hydrology and environmental sciences for central europe. *Earth System Science Data*, 13(9):4529–4565, 2021. doi: 10.5194/essd-13-4529-2021. URL https://essd.copernicus.org/articles/13/4529/2021/.

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between gaussian fields and gaussian markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011. doi: 10.1111/j.1467-9868.2011.00777.x.

Simon Linke, Bernhard Lehner, Camille Ouellet Dallaire, Jemima Ariwi, Günther Grill, Meghna Anand, Paul Beames, Victor Burchard-Levine, Sean Maxwell, Harsha Moidu, Felicia Tan, and Michele Thieme. Global hydro-environmental sub-basin and river reach characteristics at high spatial resolution. *Scientific Data*, 6:283, 2019. doi: 10.1038/s41597-019-0300-6.

Juncheng Liu, Kenji Kawaguchi, Bryan Hooi, Yiwei Wang, and Xiaokui Xiao. Eignn: Efficient infinite-depth graph neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18762–18773. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/9bd5ee6fe55aaeb673025dbcb8f939c1-Paper.pdf.

Dmitry M. Malioutov, Jason K. Johnson, and Alan S. Willsky. Walk-sums and belief propagation in gaussian graphical models. *Journal of Machine Learning Research*, 7(73):2031–2064, 2006. URL http://jmlr.org/papers/v7/malioutov06a.html.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010. doi: 10.1111/j.1467-9868.2010.00740.x.

Joaquín Muñoz Sabater. ERA5-Land hourly data from 1981 to present, version cy45r1, 2019.

Joaquín Muñoz-Sabater, Emanuel Dutra, Anna Agustí-Panareda, Clément Albergel, Gabriele Arduini, Gianpaolo Balsamo, Souhail Boussetta, Margarita Choulga, Shaun Harrigan, Hans Hersbach, Brecht Martens, Diego G. Miralles, Maria Piles, Nemesio J. Rodríguez-Fernández, Ervin Zsoter, Carlo Buontempo, and Jean-Noël Thépaut. ERA5-Land: a state-of-the-art global reanalysis dataset for land applications. *Earth System Science Data*, 13:4349–4383, 2021. doi: 10.5194/essd-13-4349-2021.

Ranga Myneni, Yuri Knyazikhin, and Taejin Park. MCD15A3H MODIS/Terra+Aqua Leaf Area Index/FPAR 4-Day L4 Global 500m SIN Grid V006, 2015.

Seth Nabarro, Mark Van Der Wilk, and Andrew Davison. Learning in deep factor graphs with Gaussian belief propagation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37141–37163. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/nabarro24a.html.

Joseph Ortiz, Talfan Evans, and Andrew J. Davison. A visual introduction to gaussian belief propagation, 2021. URL https://arxiv.org/abs/2107.02308.

Joel Oskarsson, Per Sidén, and Fredrik Lindsten. Scalable deep Gaussian Markov random fields for general graphs. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17117–17137. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/oskarsson22a.html.

Jon D. Pelletier, Patrick D. Broxton, Pieter Hazenberg, Xubin Zeng, Peter A. Troch, Guo-Yue Niu, Zachary C. Williams, Michael A. Brunke, and David Gochis. Global 1-km gridded thickness of soil, regolith, and sedimentary deposit layers, 2016.

Meng Qu, Yoshua Bengio, and Jian Tang. GMNN: Graph Markov neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5241–5250. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/qu19a.html.

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 14501–14515. Curran Associates, Inc., 2022. doi: 10.52202/068431-1054. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/5d4834a159f1547b267a05a4e2b7cf5e-Paper-Conference.pdf.

Håvard Rue and Leonhard Held. *Gaussian Markov Random Fields: Theory and Applications*. Chapman and Hall/CRC, 2005.

Per Sidén and Fredrik Lindsten. Deep Gaussian Markov random fields. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8916–8926. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/siden20a.html.

State Agency for the Environment Baden-Württemberg – Hydrographic Service. Runoff data, 2020. URL http://udo.lubw.baden-wuerttemberg.de/public/p/pegel_messwerte_leer. Runoff data downloaded 4 September 2020.

Brigitta Toth, Melanie Weynants, Laszlo Pasztor, and Tomislav Hengl. 3D soil hydraulic database of europe at 250 m resolution. *Hydrological Processes*, 31:2662–2666, 2017. doi: 10.1002/hyp.11203.

Antonio Trabucco and Robert Zomer. Global aridity index and potential evapotranspiration (ET0) climate database v2, 2019.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018*. OpenReview.net, 2018.

Eric Vermote. MOD09Q1 MODIS/Terra Surface Reflectance 8-Day L3 Global 250m SIN Grid V006, 2015.

Yu Wang, Yuesong Shen, and Daniel Cremers. Explicit pairwise factorized graph neural network for semi-supervised node classification. In Cassio de Campos and Marloes H. Maathuis (eds.), *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1979–1987. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/wang21d.html.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 1907–1913. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/264. URL https://doi.org/10.24963/ijcai.2019/264.

Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 28877–28888. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f1c1592588411002af340cbaedd6fc33-Paper.pdf.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems 32*, Vancouver, Canada, 2019. URL https://openreview.net/references/pdf?id=S1qBAf6rr.

A Appendix

A.1 Dataset Creation – USCRN

We construct the USCRN dataset from average daily temperature observations recorded by stations in the United States Climate Reference Network (USCRN) between 2003 and 2025. We retain stations located in the contiguous United States, as well as one station in Ontario, Canada that is included in the USCRN data. Because some stations were operational for only a small fraction of the study period, we exclude stations that contain observations for fewer than 50% of the timesteps. After filtering, the dataset contains 131 stations.

Observations and temporal split. We split the dataset chronologically, using observations from 2003–2020 for training, 2021–2022 for validation, and 2023–2025 for testing. Temperature measurements are standardized using statistics computed from the training split. Each station at each timestep is represented by two input features: the standardized temperature measurement and a binary missingness indicator. For missing observations, the temperature input is set to zero. The indicator is zero for observed measurements and one for missing measurements.

Graph construction. USCRN provides station observations and geographic coordinates but does not define a graph over the stations. We therefore construct a sparse spatial graph from the station locations. For each station, we connect up to four nearest stations within the same state and up to two nearest stations outside the state, subject to a maximum geographic distance of 325 miles. We then symmetrize the resulting adjacency matrix, yielding an undirected graph with 131 nodes and 348 unique edges. The same fixed graph is used for all observations.

Evaluation masks. We evaluate spatial reconstruction using three masking configurations: random masking and two structured geographic masks. To reduce variability due to the particular realization of the random mask, each test sample is evaluated under 10 independently sampled masks. Consequently, the random-mask evaluation contains ten times as many masked instances as each structured masking configuration. The two structured masks are deterministic and are visualized in Figures 4 and 5 with red being observed and white being masked values.

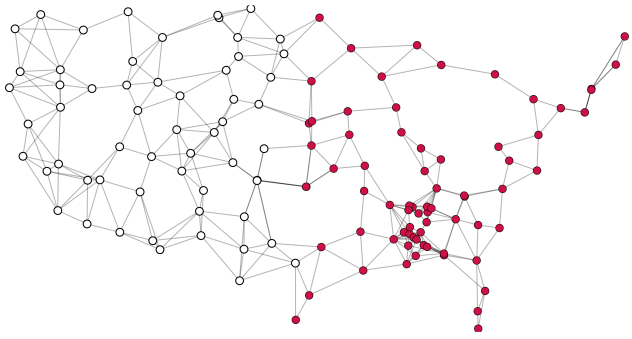

Figure 4: Predict East From West Mask.

A.2 Dataset Creation – LamaH-CE

We construct the hydrological forecasting dataset from LamaH-CE using the gauge and river-network information from delineation B. We gratefully acknowledge the data providers and sources

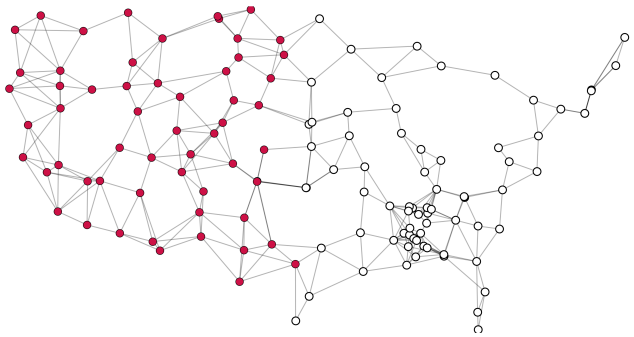

Figure 5: Predict West From East Mask.

compiled as part of the original LamaH-CE dataset (Federal Office for the Environment (BAFU), 2020; Czech Hydrometeorological Institute (CHMI), 2020; Bavarian State Office for the Environment – Hydrographic Service, 2020; Federal Ministry of Agriculture, Regions and Tourism – Hydrographic Central Office, 2020; State Agency for the Environment Baden-Württemberg – Hydrographic Service, 2020; Federal Ministry of Agriculture, Regions and Tourism – Hydrographic Central Office, 2013; Broxton et al., 2014; European Environment Agency, 2012; 2019; European Commission and European Soil Bureau Network, 2004; Farr et al., 2007; Friedl & Sulla-Menashe, 2019; Gleeson et al., 2014; Federal Ministry of Agriculture, Regions and Tourism – Hydrographic Central Office, 2007; Hartmann & Moosdorf, 2012; Hiederer, 2013a;b; Linke et al., 2019; Muñoz-Sabater et al., 2021; Muñoz Sabater, 2019; Myneni et al., 2015; Pelletier et al., 2016; Toth et al., 2017; Trabucco & Zomer, 2019; Vermote, 2015).

River network construction. In LamaH-CE, delineation B contains 859 gauging stations organized according to the upstream–downstream structure of the river network. The network contains multiple disconnected river systems. We select the largest connected component, which contains 608 of the 859 gauges, and use this component for all experiments.

Each gauge is represented as a node, with edges defined by the upstream–downstream connectivity provided by LamaH-CE. Although the underlying river network is directed, we represent each connection as an undirected edge for graph inference by symmetrizing the adjacency matrix. The resulting graph is a tree with 608 nodes and 607 undirected edges, a maximum depth of 25, a mean depth of 11.25, and a diameter of 29.

Observations and temporal split. The dataset contains hourly streamflow observations from 1981 through 2017. Because individual gauges were not necessarily operational throughout the entire observation period, measurements may be missing at some nodes and timesteps. We split the observations chronologically, using 1981–2012 for training, 2013–2014 for validation, and 2015–2017 for testing.

Preprocessing and missing observations. We first transform streamflow measurements using $\log(1 + x)$ and then standardize the transformed values using statistics computed from the training split. Each gauge at each timestep is represented by two input features: the standardized streamflow measurement and a binary missingness indicator. For missing observations, the streamflow input is set to zero. The indicator is zero for observed measurements and one for missing measurements.

A.3 HYPERPARAMETERS

All neural networks were trained using the Adam optimizer with a learning rate of 10^{-3}, a batch size of 256, and early stopping with a patience of 150 epochs. They used a hidden dimension of 64. The depth of each architecture was adjusted to obtain approximately 100k trainable parameters. Transformer, GraphGPS, and GGNN models used 8 heads per layer.

Table 5: Model sizes used in the experiments.

Model	Layers	Parameters
GCN	5	104,577
IGNN	4	100,097
GPR-GNN	6	100,547
Transformer	3	100,289
GraphGPS	2	100,225
GGNN-PN (USCRN)	3	94,793
GGNN-DD (USCRN)	3	86,441
GGNN-L (USCRN)	3	94,793
GGNN-PN (LamaH)	3	122,070
GGNN-DD (LamaH)	3	107,502
GGNN-L (LamaH)	3	122,070

The number of trainable parameters in GGNN depends on the underlying graph because the precision matrix is parameterized over the graph structure. Consequently, datasets with different numbers of nodes and edges result in different parameter counts even though the same number of layers is used.

A.4 GAUSSIAN BELIEF PROPAGATION

For completeness, we provide the Gaussian Belief Propagation updates used for inference in our model. In the univariate setting, each directed edge $i \to j$ is associated with a scalar Gaussian message. Although J is symmetric, the messages $i \to j$ and $j \to i$ are distinct because they encode directional information exchange between nodes. Messages are represented by a precision parameter $\pi_{i\to j}$ and an information parameter $\eta_{i\to j}$. At each iteration, node i computes a message to neighbor j by combining its local evidence with all incoming messages except the one received from j. Let $\mathcal{N}(i) \setminus j$ denote this set, and define

$$\alpha_{i\setminus j} = J_{ii} + \sum_{k\in\mathcal{N}(i)\setminus j} \pi_{k\to i}, \qquad \beta_{i\setminus j} = h_i + \sum_{k\in\mathcal{N}(i)\setminus j} \eta_{k\to i}.$$

The message updates are then given by

$$\pi_{i\to j} = -\frac{J_{ij}^2}{\alpha_{i\setminus j}}, \qquad \eta_{i\to j} = -\frac{J_{ij}}{\alpha_{i\setminus j}}\beta_{i\setminus j}.$$

After convergence, each node computes its local marginal parameters by aggregating all incoming messages,

$$\pi_i = J_{ii} + \sum_{k\in\mathcal{N}(i)} \pi_{k\to i}, \qquad \eta_i = h_i + \sum_{k\in\mathcal{N}(i)} \eta_{k\to i},$$

yielding the mean estimate

$$\mu_i = \frac{\eta_i}{\pi_i}.$$

While not used in our paper, loopy GaBP message damping can be beneficial for improving convergence stability. Damping replaces each message update with a convex combination of its previous value and the newly computed value, effectively acting as a moving average over iterations. This is particularly helpful on frustrated graphs, where competing edge interactions cannot be simultaneously satisfied and undamped message updates may oscillate.

Algorithm 1 Gaussian Belief Propagation

Input: $J, h, \varepsilon, \lambda$
Initialize $\pi_{i\to j} = 0$, $\eta_{i\to j} = 0$ for all $(i, j) \in E$
while not converged ($\Delta > \varepsilon$) **do**
  $\alpha_{i\setminus j} = J_{ii} + \sum_{k\in N(i)\setminus j} \pi_{k\to i}$
  $\beta_{i\setminus j} = h_i + \sum_{k\in N(i)\setminus j} \eta_{k\to i}$
  $\pi^{\text{new}}_{i\to j} = -J_{ij}^2/\alpha_{i\setminus j}$
  $\eta^{\text{new}}_{i\to j} = -J_{ij}\,\beta_{i\setminus j}/\alpha_{i\setminus j}$
  $\Delta\pi_{i\to j} = \pi^{\text{new}}_{i\to j} - \pi_{i\to j}$
  $\Delta\eta_{i\to j} = \eta^{\text{new}}_{i\to j} - \eta_{i\to j}$
  $\pi_{i\to j} \leftarrow (1-\lambda)\,\pi_{i\to j} + \lambda\,\pi^{\text{new}}_{i\to j}$
  $\eta_{i\to j} \leftarrow (1-\lambda)\,\eta_{i\to j} + \lambda\,\eta^{\text{new}}_{i\to j}$
  $\Delta = \max_{(i,j)\in E} \left(\max\left(|\Delta\pi_{i\to j}|,\ |\Delta\eta_{i\to j}|\right)\right)$
end while
Output: $\mu_i = \eta_i/\pi_i$

A.5 Proofs of Theorem 1 and Corollary 1

Proof of Theorem 1. Let

$$\widetilde{J} = D^{-1/2} J D^{-1/2} = I - R,$$

with J walk-summable, so that

$$q := \rho(|R|) < 1.$$

Writing

$$\bar{h} = D^{-1/2} h, \qquad \bar{\mu} = D^{1/2}\mu,$$

exact inference satisfies

$$\bar{\mu} = (I - R)^{-1}\bar{h}.$$

Since R is symmetric, let $Ru_i = r_i u_i$. Moreover,

$$|r_i| \le \rho(R) \le \rho(|R|) = q < 1.$$

Therefore,

$$(I - R)^{-1} u_i = \frac{1}{1 - r_i} u_i,$$

and hence

$$\frac{1}{1+q} \le \frac{1}{1-r_i} \le \frac{1}{1-q}.$$

Thus, walk-summability bounds the gain of every mode but does not determine how these gains are ordered across graph frequencies.

To show that no spectral preference is forced, let S be a symmetric, zero-diagonal graph shift with $\rho(|S|) = 1$ and

$$Su_i = \lambda_i u_i.$$

Consider

$$\widetilde{J}_\alpha = I - \alpha S, \qquad |\alpha| < 1.$$

This family is walk-summable because

$$\rho\Big(|I - \widetilde{J}_\alpha|\Big) = \rho(|\alpha S|) = |\alpha| < 1.$$

Its response along an eigenmode of S is

$$g_\alpha(\lambda) = \frac{1}{1 - \alpha\lambda},$$

with derivative

$$\frac{d}{d\lambda} g_\alpha(\lambda) = \frac{\alpha}{(1 - \alpha\lambda)^2}.$$

Under the convention that larger λ corresponds to smoother graph modes, $\alpha > 0$ yields a low-pass response, $\alpha = 0$ yields a spectrally flat response, and $\alpha < 0$ yields a high-pass response. Hence, walk-summability permits low-pass, spectrally flat, and high-pass responses.

Proof of Corollary 1. Fix any depth $L \geq 1$. To establish existence, it is sufficient to consider the subclass in which all L layers use the same shift S and coefficient $\alpha \in (-1, 1)$. Each layer is walk-summable by the construction above, and the composed response along an eigenmode of S is

$$g_\alpha^{(L)}(\lambda) = \left(\frac{1}{1-\alpha\lambda}\right)^L = (1-\alpha\lambda)^{-L}.$$

Its derivative is

$$\frac{d}{d\lambda} g_\alpha^{(L)}(\lambda) = L\alpha(1-\alpha\lambda)^{-L-1}.$$

Therefore, for arbitrary $L \geq 1$, choosing $\alpha > 0$ gives a low-pass response, $\alpha = 0$ gives a spectrally flat response, and $\alpha < 0$ gives a high-pass response.

Since this construction holds for every $L \geq 1$, walk-summable GGNNs admit all three spectral preferences at every depth. Thus, depth alone does not force smoothing or any other fixed spectral preference, rather the resulting spectral preference is determined by the learned precision matrices.

A.6 Expanded Results

Table 6: Full LamaH-CE forecasting performance across all prediction horizons with and without positional encodings. We report the mean and standard deviation over 10 runs.

(a) Without Positional Encodings

	6 hours		12 hours		24 hours		48 hours		72 hours		96 hours	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
GCN	2.76 ± 0.09	16.51 ± 0.55	3.86 ± 0.14	25.47 ± 0.54	5.24 ± 0.10	36.93 ± 0.34	7.32 ± 0.10	50.64 ± 0.27	8.54 ± 0.11	57.81 ± 0.28	9.36 ± 0.11	61.58 ± 0.31
IGNN	2.73 ± 0.09	16.25 ± 0.32	3.82 ± 0.11	25.28 ± 0.29	5.26 ± 0.06	36.84 ± 0.24	7.35 ± 0.06	50.34 ± 0.18	8.57 ± 0.15	57.58 ± 0.18	9.37 ± 0.20	61.44 ± 0.43
GPR-GNN	2.83 ± 0.17	17.30 ± 1.60	3.81 ± 0.18	25.64 ± 1.41	5.27 ± 0.14	37.25 ± 0.74	7.29 ± 0.09	50.47 ± 0.32	8.49 ± 0.09	57.55 ± 0.32	9.26 ± 0.08	61.19 ± 0.31
GCN-E	2.26 ± 0.06	13.30 ± 0.39	3.17 ± 0.07	20.04 ± 0.51	4.39 ± 0.08	30.67 ± 0.33	6.47 ± 0.08	46.04 ± 0.27	7.72 ± 0.08	54.15 ± 0.28	8.52 ± 0.06	58.44 ± 0.27
Transformer	2.49 ± 0.17	13.73 ± 1.05	3.17 ± 0.07	18.64 ± 0.52	4.35 ± 0.15	27.34 ± 0.48	6.60 ± 0.23	44.37 ± 1.23	8.26 ± 0.35	55.27 ± 2.27	9.33 ± 0.48	61.36 ± 2.84
GraphGPS	2.53 ± 0.21	14.91 ± 1.18	3.23 ± 0.19	19.55 ± 0.93	4.46 ± 0.31	28.18 ± 1.47	6.55 ± 0.21	43.54 ± 0.51	8.09 ± 0.31	53.53 ± 1.39	9.17 ± 0.52	59.75 ± 2.65
Ours												
GGNN-PN	2.29 ± 0.18	13.22 ± 0.88	3.05 ± 0.18	18.57 ± 1.13	4.30 ± 0.16	28.46 ± 0.88	6.46 ± 0.17	44.66 ± 0.54	7.85 ± 0.18	53.48 ± 0.38	8.77 ± 0.21	58.32 ± 0.45
GGNN-DD	2.23 ± 0.08	12.84 ± 0.47	3.06 ± 0.14	18.19 ± 0.84	4.21 ± 0.17	27.71 ± 0.83	6.42 ± 0.25	44.12 ± 0.69	7.82 ± 0.59	53.17 ± 1.59	8.69 ± 0.84	58.37 ± 3.08
GGNN-L	2.25 ± 0.18	12.98 ± 0.89	2.98 ± 0.18	17.76 ± 0.81	4.07 ± 0.13	26.86 ± 1.04	6.23 ± 0.23	43.61 ± 0.99	7.83 ± 0.54	53.70 ± 2.24	8.94 ± 0.82	60.20 ± 4.48

(b) With Positional Encodings

	6 hours		12 hours		24 hours		48 hours		72 hours		96 hours	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
GCN	2.72 ± 0.13	16.41 ± 0.77	3.72 ± 0.09	25.24 ± 0.69	5.21 ± 0.12	37.20 ± 0.37	7.22 ± 0.13	50.47 ± 0.49	8.33 ± 0.13	57.26 ± 0.55	9.03 ± 0.13	60.68 ± 0.62
IGNN	2.73 ± 0.12	16.25 ± 0.88	3.75 ± 0.10	25.20 ± 0.55	5.25 ± 0.13	36.92 ± 0.36	7.19 ± 0.16	50.03 ± 0.31	8.33 ± 0.16	56.96 ± 0.35	9.05 ± 0.12	60.50 ± 0.36
GPR-GNN	2.74 ± 0.14	16.81 ± 0.92	3.78 ± 0.13	25.79 ± 0.81	5.19 ± 0.13	37.44 ± 0.60	7.16 ± 0.12	50.51 ± 0.39	8.28 ± 0.14	57.20 ± 0.37	8.98 ± 0.19	60.53 ± 0.39
GCN-E	2.36 ± 0.08	13.72 ± 0.43	3.18 ± 0.12	19.93 ± 0.52	4.56 ± 0.22	31.07 ± 0.57	6.63 ± 0.18	46.05 ± 0.33	7.88 ± 0.16	54.09 ± 0.32	8.69 ± 0.20	58.37 ± 0.33
Transformer	2.56 ± 0.46	14.51 ± 2.89	3.12 ± 0.37	18.19 ± 2.12	4.29 ± 0.35	26.71 ± 1.61	6.55 ± 0.18	42.46 ± 0.73	8.38 ± 0.48	53.12 ± 1.91	9.57 ± 0.69	59.84 ± 3.11
GraphGPS	2.55 ± 0.23	15.47 ± 2.11	3.34 ± 0.23	20.16 ± 2.78	4.34 ± 0.26	27.50 ± 1.95	6.42 ± 0.29	43.02 ± 1.16	7.92 ± 0.49	52.66 ± 1.63	8.93 ± 0.63	58.74 ± 2.41
Ours												
GGNN-PN	2.26 ± 0.15	13.27 ± 0.69	3.06 ± 0.13	18.75 ± 0.63	4.40 ± 0.14	28.85 ± 0.55	6.50 ± 0.21	44.80 ± 0.49	7.80 ± 0.20	53.44 ± 0.56	8.64 ± 0.19	58.08 ± 0.56
GGNN-DD	2.32 ± 0.23	13.56 ± 1.21	3.06 ± 0.14	18.65 ± 0.93	4.27 ± 0.19	27.92 ± 0.77	6.40 ± 0.23	44.67 ± 1.21	7.75 ± 0.21	53.43 ± 1.21	8.62 ± 0.26	57.97 ± 0.81
GGNN-L	2.48 ± 0.22	14.75 ± 1.38	3.15 ± 0.16	19.33 ± 1.48	4.35 ± 0.19	28.23 ± 1.72	6.61 ± 0.42	44.79 ± 1.75	8.17 ± 0.96	54.87 ± 3.61	9.16 ± 1.29	60.79 ± 5.07

Table 7: Forecasting performance across prediction horizons with different convergences. We report the mean and standard deviation over 10 runs.

	6 hours		12 hours		24 hours		48 hours		72 hours		96 hours	
Model	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
GGNN-L (Tree)	2.25 ± 0.18	12.98 ± 0.89	2.98 ± 0.18	17.76 ± 0.81	4.07 ± 0.13	26.86 ± 1.04	6.23 ± 0.23	43.61 ± 0.99	7.83 ± 0.54	53.70 ± 2.24	8.94 ± 0.82	60.20 ± 4.48
GGNN-L (20)	2.23 ± 0.08	13.13 ± 0.67	2.94 ± 0.11	17.85 ± 1.12	4.08 ± 0.13	27.04 ± 1.14	6.16 ± 0.23	43.58 ± 0.84	7.59 ± 0.29	52.97 ± 0.87	8.58 ± 0.42	58.74 ± 2.01
GGNN-L (10)	2.39 ± 0.11	14.24 ± 0.64	3.17 ± 0.16	19.34 ± 1.08	4.32 ± 0.12	28.69 ± 0.57	6.60 ± 0.31	45.86 ± 1.48	8.22 ± 0.73	56.21 ± 3.81	9.29 ± 1.14	62.56 ± 6.54
GGNN-L (5)	2.42 ± 0.15	14.30 ± 0.88	3.18 ± 0.19	19.90 ± 1.15	4.40 ± 0.13	29.62 ± 0.79	6.51 ± 0.22	45.43 ± 0.69	7.72 ± 0.21	53.77 ± 0.63	8.46 ± 0.26	58.25 ± 0.63
GGNN-L (2)	2.42 ± 0.11	14.37 ± 0.41	3.30 ± 0.12	21.03 ± 0.22	4.59 ± 0.12	31.55 ± 0.33	6.62 ± 0.17	46.56 ± 1.75	7.80 ± 0.18	54.39 ± 0.27	8.58 ± 0.16	58.54 ± 0.18

A.7 Precision Matrix Visualizations

We visualize the learned precision matrices from the final layer of trained GGNN models on the USCRN dataset. We additionally visualize the corresponding correlation matrices to illustrate the dependency structures learned by different heads.

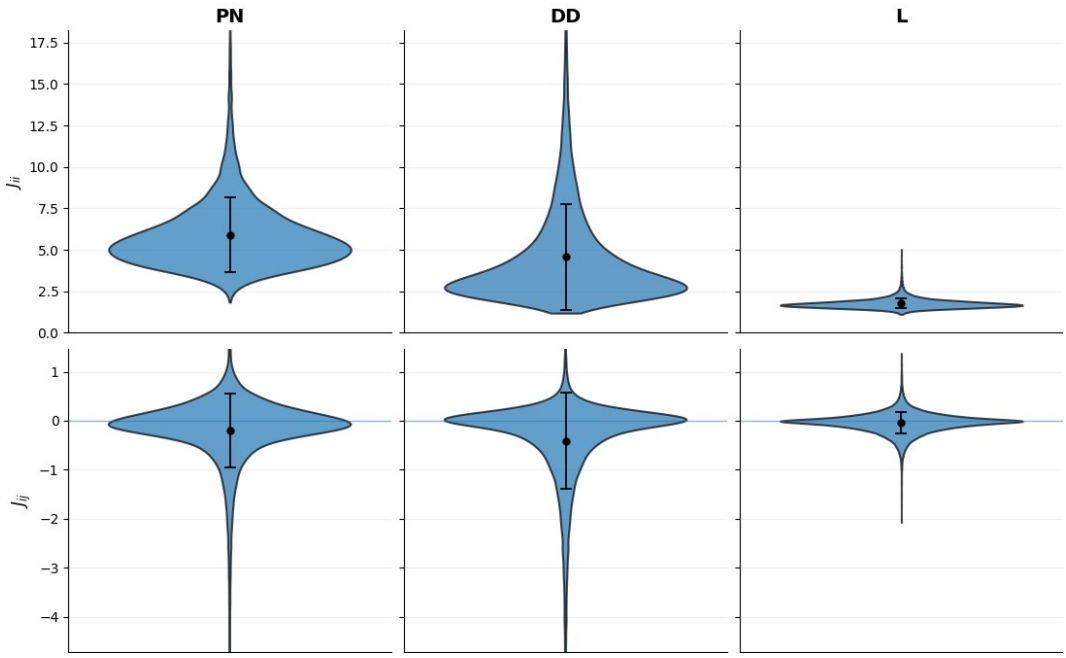

Figure 6: Distribution of learned precision-matrix entries in the final GGNN layer after training on USCRN. We show the distributions of diagonal (top row) and off-diagonal (bottom row) entries for each precision-matrix parameterization.

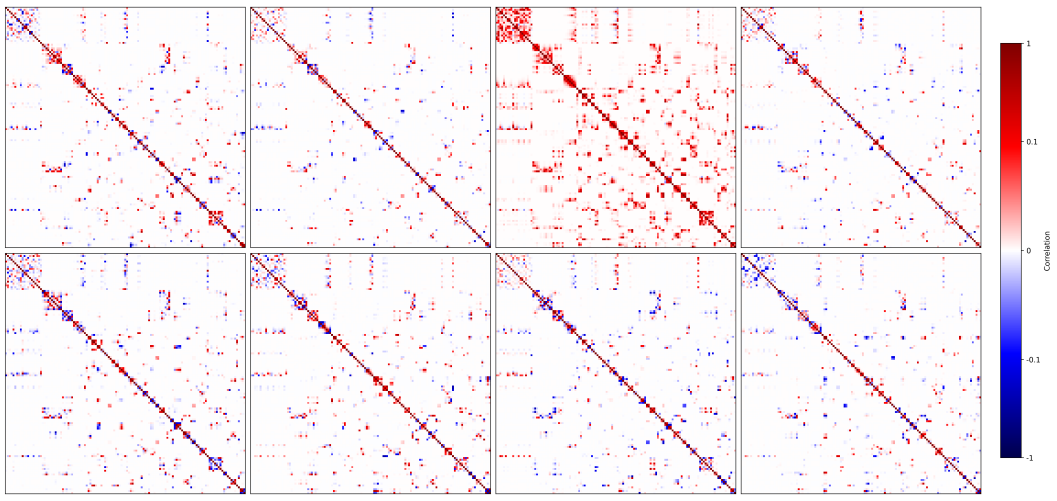

Figure 7: Correlation matrices implied by the learned pairwise-normalizable precision matrices. Different heads learn distinct correlation patterns, capturing a mixture of local and long-range dependencies. Positive and negative values indicate positive and negative correlations, respectively.

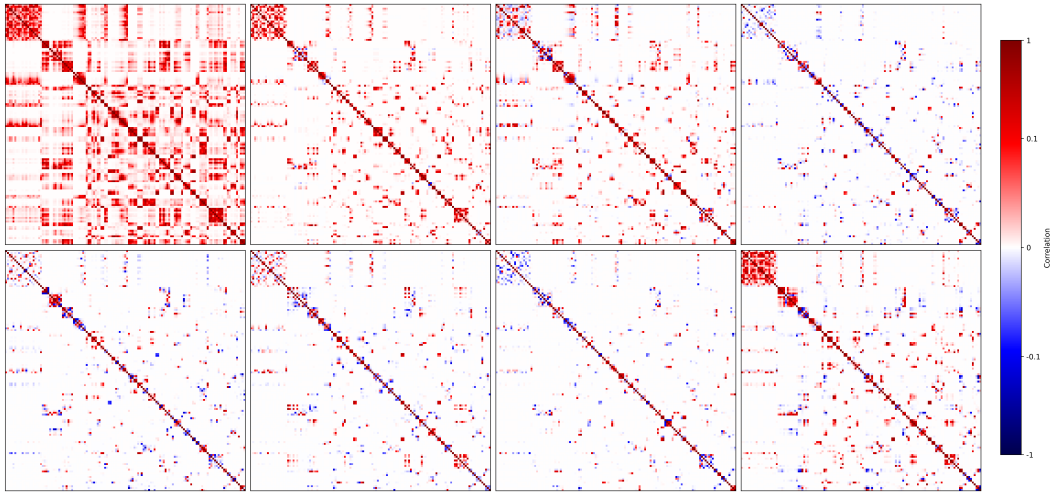

Figure 8: Correlation matrices implied by the learned diagonally dominant precision matrices. Different heads learn distinct correlation patterns, capturing a mixture of local and long-range dependencies. Positive and negative values indicate positive and negative correlations, respectively.

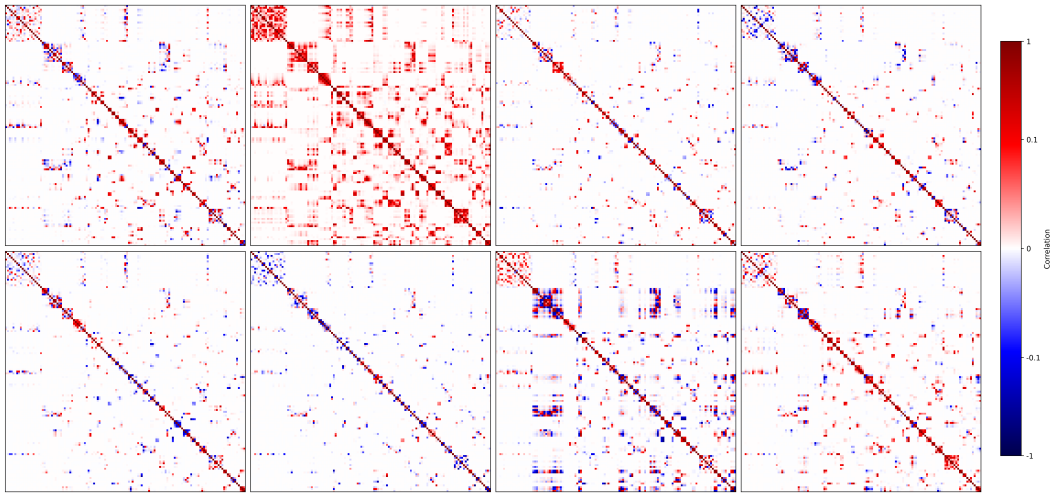

Figure 9: Correlation matrices implied by the learned Laplacian precision matrices. Different heads learn distinct correlation patterns, capturing a mixture of local and long-range dependencies. Positive and negative values indicate positive and negative correlations, respectively.