Ising-Traffic: Using Ising Machine Learning to Predict Traffic Congestion under Uncertainty

Zhenyu Pan[1], Anshujit Sharma[1], Jerry Yao-Chieh Hu[2], Zhuo Liu[1], Ang Li[3], Han Liu[2], Michael Huang[1], Tong Geng[1]

[1]University of Rochester,
[2] Northwestern University,
[3] Pacific Northwest National Laboratory
{zhenyupan, anshujit.sharma, zhuo.liu, michael.huang, tong.geng}@rochester.edu,
{jhu, hanliu}@northwestern.edu, ang.li@pnnl.gov

Abstract

This paper addresses the challenges in accurate and real-time traffic congestion prediction under uncertainty by proposing Ising-Traffic, a dual-model Ising-model-based traffic prediction framework that delivers higher accuracy and lower latency than SOTA solutions. While traditional solutions face the dilemma from the trade-off between algorithm complexity and computational efficiency, our Ising-model-based method breaks away from the trade-off leveraging the Ising model's strong expressivity and the Ising machine's strong computation power. In particular, Ising-Traffic formulates traffic prediction under uncertainty into two Ising models: Reconstruct-Ising and Predict-Ising. Reconstruct-Ising is mapped onto modern Ising machines and handles uncertainty in traffic accurately with negligible latency and energy consumption, while Predict-Ising is mapped onto traditional processors and predicts future congestion precisely with only at most 1.8% computational demands of existing solutions. Our evaluation shows Ising-Traffic delivers on average 98× speedups and 5% accuracy improvement over SOTA.

1 Introduction

With rapid economic development, urbanization, and an increase in personal vehicle ownership in the past decade, traffic congestion has become a significant problem in metropolitan centers around the world, resulting in more accidents, higher fuel consumption, greater emission of greenhouse gas and pollutants, health hazards, and lower productivity at the workplace. One of the best approaches to avoiding congestion is to predict it accurately and promptly. However, the uncertainty from unobserved traffic information due to the lack of sensors or cameras at certain road segments and external shocks like weather, accidents, and unexpected road conditions makes accurate prediction very challenging. As wrong or delayed prediction can result in worse traffic performance and more accidents (Cheng et al. 2022), effective and efficient traffic congestion prediction is in urgent demand.

There have been many studies on predicting traffic congestion in academia and industry (Kumar and Raubal 2021). Traditionally, researchers use one or multiple measurements among traffic speed, travel time, and queue length of a road segmentation to define a complex threshold and predict whether congestion will take place (Akhtar and Moridpour 2021). In the past decade, deep learning methods like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Graph Neural Networks (GNNs) are also used in traffic congestion prediction in the real world (Ranjan et al. 2020). Some of these methods provide high accuracy (Guo et al. 2021a) while others deliver faster-than-real-time prediction (Liu and Wu 2017; Peng et al. 2022; Zeng et al. 2023). However, the common challenge existing studies face is the trade-off between algorithm complexity and computational efficiency. Namely, the increase in accuracy generally comes from the increase in algorithmic complexity, which normally leads to higher prediction latency. Furthermore, as the end of Moore's law is approaching, the increase of hardware computational capability is inevitably slowing down, making it increasingly challenging to improve accuracy from this trade-off. Figure 1 compares current state-of-the-art methods' accuracy and computational efficiency. It shows that the traditional methods following the trade-off fail to deliver both accurate and fast solutions that should stand on the right side of the Pareto-frontier. New methods that breaks away from the trade-off are needed.

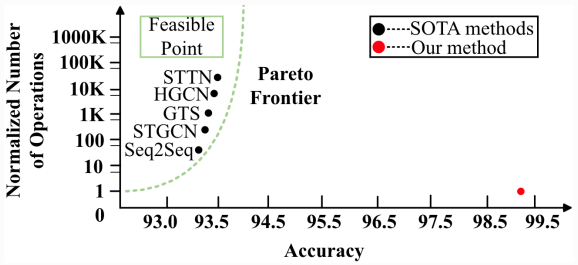

Figure 1: Accuracy vs computational demands of Ising-Traffic and SOTA methods.

We observe that recent studies on the Ising model have demonstrated that, leveraging the Ising machine's unique capability of finding the low-energy states of a system, an mW-level CMOS-based Ising model implementation (Afoakwa

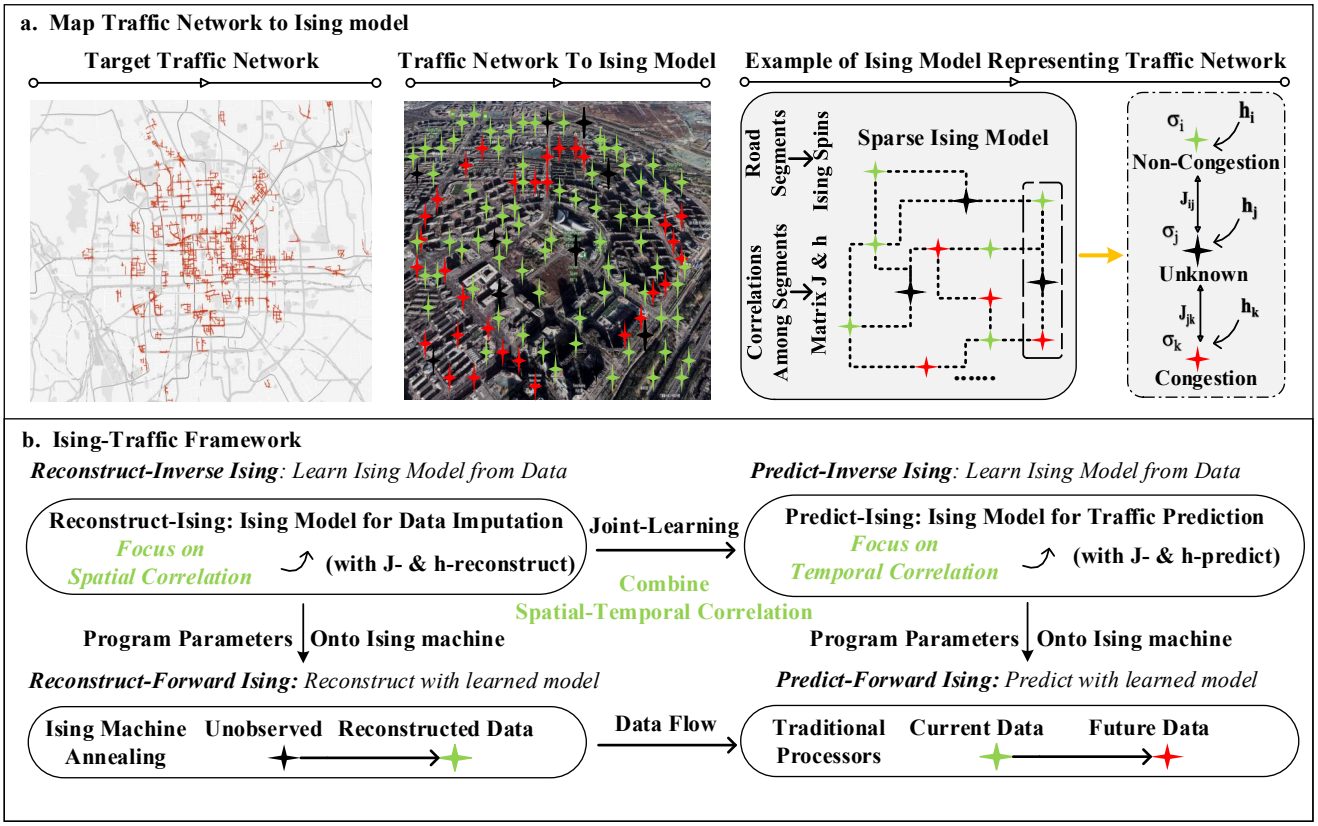

Figure 2: Ising-Traffic framework overview. Red/green/black stars represent congestion/non-congestion/unobserved segments.

et al. 2021; Sharma et al. 2022) can solve NP-complete problems such as graph max/min cuts with orders of magnitude speedups (μs-level vs. ms-level) over traditional processors like CPU and GPU with over 100W power consumption. This observation motivates us to investigate whether the Ising model, as a near-future quantum solution, can be used in traffic congestion prediction and fundamentally address the aforementioned challenge. For the convenience of readers, we first briefly introduce the Ising model

The Ising model is a probabilistic graphical model rooted in physics that has been widely used to describe complex systems with discrete degrees of freedom. As it naturally favors the spin configuration that minimizes the energy (Hamiltonian) of systems by design, the Ising model can be used as a surrogate for a variety of NP-complete optimization problems as long as they can be represented using Ising formulations with specified structural coupling parameters matching the problems of interest (Cipra 2000).

By expressing the traffic congestion prediction problem under uncertainty from unobserved data and external shocks in Ising formulations, we propose, *Ising-Traffic*, a novel dual-model **Ising Machine Learning (IML)**-based traffic prediction framework that delivers both higher accuracy and lower latency than SOTA solutions. Figure 2 presents the overview of the proposed framework where traffic prediction under uncertainty is divided into two separate Ising model problems (with their corresponding models): ***Reconstruct-Ising*** and ***Predict-Ising***.

First, both models' coupling parameters are learned from historical traffic data; this learning process is called *Inverse Ising*. After the Ising models are trained, a forward problem-solving step is performed sequentially with the learned parameters for both models; this step is called *Forward Ising*. During Forward Ising, Forward Reconstruct-Ising is used to handle the uncertainty in traffic and impute the missing data based on the observed information in the same time slice, hence mainly learning the spatial correlations among road segments; while Forward Predict-Ising is in charge of predicting the future congestion based on the imputed traffic data in the past time slices from Reconstruct-Ising, hence mainly learning the temporal correlations among segments.

In order to achieve high accuracy, we propose the following domain-specific optimizations and apply them in Inverse Ising: (1) *robust learning with different levels of uncertainty severity*: we inject different levels of uncertainty during training to improve the Ising model's robustness to random uncertainties from the real world. (2) *geographical information embedding*: we incorporate traffic's geographical information in the spatial correlation learning; (3) *message passing with dynamic attention*: we propose a dynamic message passing method with online-updated attention to better propagate the influence from the neighboring segments in the Ising model;

In order to achieve high performance, Ising-traffic (1)

conducts sparse Inverse Ising training to create the Ising model with high sparsity for reduced computational complexity and (2) performs Forward Ising with heterogeneous hardware platforms for better performance. In particular, we use an Ising machine to conduct Forward Reconstruct-Ising, which delivers precise graph reconstruction with over $10^6\times$ speedups over traditional methods. For the execution of Forward Predict-Ising, we still use traditional processors considering the mechanism to apply the Ising model to temporal information analysis is yet to be verified. Note that, although not being mapped directly to an Ising machine, Forward Predict-Ising only requires at most 1.8% arithmetic operations of current SOTA solutions.

To the best of our knowledge, Ising-Traffic is the first work that demonstrates that Ising methods can outperform traditional solutions to real-world applications. Our contributions are summarized as follows:

- We propose Ising-Traffic, a novel and efficient dual-model Ising-based traffic congestion prediction framework with the support of uncertainty handling.
- We propose domain-specific optimizations for Inverse Ising, including robust learning with different levels of uncertainty severity, geographical information embedding, and dynamic message passing, to improve the accuracy of Ising-Traffic.
- We propose to use heterogeneous platforms with CMOS-based Ising machines and traditional processors to accelerate Ising-based spatial and temporal information analysis, respectively, to achieve high performance.
- Experimental results demonstrate that compared with 7 traditional SOTA methods, Ising-Traffic delivers on average $98\times$ speedups with 5% accuracy improvement.

2 Background

2.1 Ising Model

The Ising model is a foundation for describing statistical physics systems of correlated binary spin variables $\sigma_i \in \{\pm 1\}_{i=1}^N$ up to quadratic interactions. Formally, an Ising model of N spins is an exponential family model for binary N-spin data $s \equiv \{\sigma_1, \cdots, \sigma_N\} \in \{\pm 1\}^N$ up to quadratic sufficient statistic taking the Boltzmann form:

$$\begin{aligned} P(s) &= \frac{1}{\mathcal{Z}} \exp\left\{-\beta \mathcal{H}(s)\right\} \\ &= \frac{1}{\mathcal{Z}} \exp\left\{-\beta \left(\sum_{i,j=1}^{N} J_{ij}\sigma_i\sigma_j + \sum_{i=1}^{N} h_i\sigma_i \right)\right\}, \end{aligned} \tag{1}$$

where data s is the configuration of N spins $\sigma_i \in \{\pm 1\}$, β is the inverse temperature, and $\mathcal{Z} \equiv \sum_s e^{-\beta\mathcal{H}(s)}$ is the partition function. The graphical structure of the system of interest is encoded into the *Ising energy function* $\mathcal{H}(s)$ through the symmetric $N \times N$ *interaction strength* matrix $\boldsymbol{J} \equiv \{J_{ij}\}_{i,j=1}^N$ with zeros on the diagonals, and the *external field* vector $\boldsymbol{h} \equiv \{h_i\}_{i=1}^N$. In this paper, since we do not consider any thermal change effects, we simply set the inverse temperature $\beta = 1$ and absorb it into $(\boldsymbol{J}, \boldsymbol{h})$. Further, for later convenience, we denote an Ising model by an Ising parameter matrix $\boldsymbol{I} \equiv (\boldsymbol{J}, \boldsymbol{h})$, where $\boldsymbol{h}$ is embedded into $\boldsymbol{J}$ as its diagonal elements, see Figure 3.

Physically, the lowest energy state (the ground state) of such a statistical physics system naturally corresponds to the most probable configuration. Therefore, an Ising machine can be utilized to find the ground state of a given Ising model with annealing methods (Inagaki et al. 2016), and many NP problems, that can be easily cast into Ising formulation (Lucas 2014), can be solved accordingly.

2.2 Ising Machine

The Ising machine is the physical system implementation of the Ising model, which naturally tends to evolve towards the state $s = \{\sigma_1, \ldots, \sigma_n\}$ with lower energy $H(s)$ and therefore can be used as a solution to optimization problems that can be represented in the Ising formulation. Many hardware Ising machines have been developed, including D-quantum Wave's annealers (King et al. 2021), Coherent Ising Machines (CIMs) (Inagaki et al. 2016), Electronic Oscillator-based Ising Machines (OIM) (Vaidya, Surya Kanthi, and Shukla 2022), and the recently developed CMOS-compatible BRIM (Afoakwa et al. 2021; Sharma et al. 2022). Among them, BRIM provides direct and physical coupling and higher-order interaction among spins and automatically and quickly finds the lowest-energy states via charging and discharging of nano-scale capacitors. The emergence of BRIM addresses many critical issues in previous Ising Substrates that hurdle the applications of Ising Machines. Therefore, Ising-Traffic uses BRIM in Forward Reconstruct-Ising.

2.3 Related Works

Traditional Methods: With the increase in the computational power of emerging computing hardware, recent studies have developed complex approaches based on neural networks for traffic congestion prediction. Yaguang Li (Li et al. 2017) proposes a diffusion Convolutional RNN that models the traffic as a diffusion process to capture spatial and temporal dependencies with sampling. In the GNNs domain, Shang Chao (Shang, Chen, and Bi 2021) proposes a graph for time series (GTS) approach, a graph probability model, to optimize desired performance on a graph probability distribution. Fanglan Chen (Chen et al. 2020) presents the Deep Kalman Filtering Network which combines two modules, the self-dependency modeling network and the neighbor dependency modeling network. However, as illustrated in Figure.1, they are all extremely computationally expensive, leading to significantly delayed prediction.

Ising Methods: As discussed above, there are not many studies on Ising methods in real-world applications due to the limitations of previous Ising hardware. Most applications discussed in Ising studies are relatively simple, e.g., max-cut and min-cut of graphs (Haribara, Utsunomiya, and Yamamoto 2016). Some researchers try to use Ising to solve complex real-world problems such as stock price prediction (Liu et al. 2023; Zhao, Bao, and Li 2018). Although they theoretically demonstrate the potential of mapping such applications onto the Ising model, these works fail to deliver com-

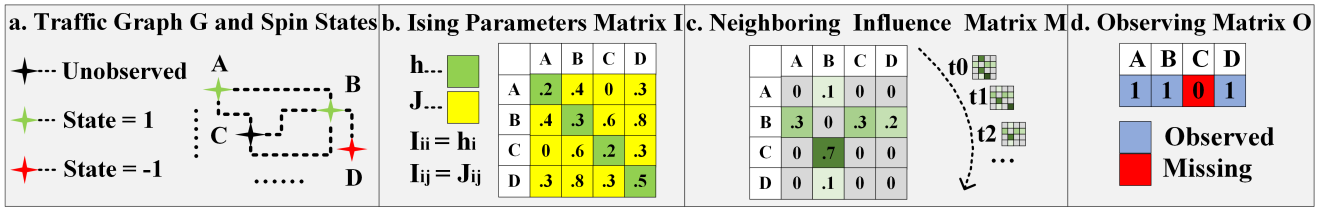

Figure 3: (a) Spin states, (b) Ising parameters matrix I (J & h), (c) Neighboring Influence Matrix, (d) Observing Matrix.

parable accuracy to traditional methods and rarely report computational performance such as latency and throughput (Son, Jeong, and Noh 2006). With the emergence of more efficient Ising machines and the approaching end of Moore's law, we contend it is time to develop end-to-end Ising-based machine learning solutions for the real world.

3 Methodology

This section introduces the design of the Ising-Traffic framework. We first provide an overview of the framework and then discuss the details of Inverse training and Forward processing of Reconstruct-Ising and Predict-Ising, respectively.

3.1 Framework Overview

As illustrated in Figure 2, the Ising-Traffic framework consists (i) Reconstruct-Ising model for data imputation and uncertainty handling in the same time slice, and (ii) Predict-Ising model for future congestion prediction based on the completed graphs from Reconstruct-Ising. In both Reconstruct-Ising and Predict-Ising, the first step is to solve an inverse problem that accurately learns their coupling parameters $(\boldsymbol{J}, \boldsymbol{h})$ reflecting road segments' spatial and temporal correlations. The local and global minima of the energy landscape shaped by the learned parameters represent the correct graph imputation and traffic congestion prediction. After Inverse Ising, the process of learning coupling parameters from data, we use a BRIM Ising machine to solve the Forward Ising problem targeting the same time slice, i.e., Reconstruct-Ising, and impute the missing data in traffic graphs. Leveraging Ising's inherent capability of accurately chasing the lowest energy states, Ising-Traffic can accurately impute graph data with even ns-level latency. Meanwhile, we use traditional processors to solve the Forward Ising problem for inter-time-slice analysis, i.e., Predict-Ising. The proposed Inverse Ising with 2-step learning guarantees that the computational demands of Forward Predict-Ising is at most 1.8% of existing SOTA solutions although it is also performed on traditional digital processors (e.g. GPUs and CPUs); hence, although Ising Machine is not used for prediction, Ising-Traffic still delivers significant speedups.

3.2 Reconstruct-Ising

This subsection introduces Reconstruct-Ising. We first describe how to map a generic traffic problem onto the Ising model. Next, we introduce our traffic-domain-specific algorithm of Inverse Ising, with which the coupling parameters of Reconstruct Ising can be trained to precisely represent the correlations among the target road segments. Finally, we discuss how to apply the trained Reconstruct-Ising model onto Ising machines to perform Forward Reconstruct-Ising and reconstruct the missing congestion data.

Mapping Traffic Graphs onto Reconstruct-Ising:

As traffic networks and the Ising model can be embedded as graphs, we define an undirected graph representation $G = (N, V, E, C, D)$ to represent a traffic network and its corresponding Ising model. Each node in the set V represents a road segment in the traffic network and a spin in the Ising model. Each edge e_{ij} in the set E represents a physical or logical connection between two segments (two spins in the Ising model) with the coupling J_{ij} as its weight. N is the number of nodes (spins); the attribute of each node represents the congestion state of a segment as well as the $\pm$ of the spin; C is the set of such attributes over nodes. Also, edges have a set of attributes D determined by the physical distances between neighboring segments.

This graph directly maps a traffic network to an Ising model with a probability distribution over 2^N possible configurations given by (1).

Inverse Reconstruct-Ising:

Inverse Reconstruct-Ising is the process of learning the precise couplings J_{ij} and external fields h_i using training samples which are historical congestion states of road segments, i.e., configurations of spins in the Ising model, observed in the same time slice (therefore, samples are also called **observations**). With a large number of observations $s_1, s_2, ..., s_K$, we try to estimate the parameters of the Ising model that maximize the likelihood of $p(s_1, s_2, ..., s_K | J_{ij}, h_i)$, which can be formulated as:

$$\{J_{ij}, h_i\} = \underset{J_{ij}, h_i}{\operatorname{argmax}} \left\{ \prod_{k=1}^{K} P(s_k | J_{ij}, h_i) \right\}. \qquad (2)$$

However, maximum likelihood estimation is computationally intractable because of the intractable Ising partition function $\mathcal{Z}$. Therefore, we adopt a neighborhood pursuit algorithm (Zhao, Roeder, and Liu 2012) to recover the graph and the parameters instead. Neighborhood pursuit not only provides accurate enough estimations for the problem of interest but also leads to significantly reduced algorithmic

complexity. Starting from Equation 1, we have

$$\begin{aligned} P(s) &= P(\sigma_i, \sigma_{\setminus i}) \\ &= \frac{1}{\mathcal{Z}} \exp\left\{ -\left(\sum_{i,j}^{N} J_{ij}\sigma_i\sigma_j + \sum_i h_i\sigma_i \right) \right\} \\ &= \frac{1}{\mathcal{Z}} \exp\left\{ -\left(h_i\sigma_i + \sum_{j}^{N} J_{ij}\sigma_i\sigma_j + \alpha \right) \right\}, \end{aligned}$$

with $\alpha \equiv \sum_j^N h_j\sigma_j + \sum_{j<k}^N \sigma_j\sigma_k$. Then the probability of a certain spin conditioned on the status of all other spins is formulated as follows,

$$\begin{aligned} P(\sigma_1 = 1|\sigma_{\setminus 1}) &= \frac{P(\sigma_1 = 1, \sigma_{\setminus 1})}{P(\sigma_1 = 1, \sigma_{\setminus 1}) + P(\sigma_1 = -1, \sigma_{\setminus 1})} \\ &= \frac{\frac{e^{-\left(h_1 + \sum_j^N J_{1j}\sigma_j + \alpha\right)}}{\mathcal{Z}}}{\frac{e^{-\left(h_1 + \sum_j^N J_{1j}\sigma_j + \alpha\right)}}{\mathcal{Z}} + \frac{e^{+\left(h_1 + \sum_j^N J_{1j}\sigma_j + \alpha\right)}}{\mathcal{Z}}} \\ &= \frac{1}{1 + e^{+2\left(h_1 + \sum_{j=2}^N J_{1j}\sigma_j\right)}}, \end{aligned} \tag{3}$$

where the last line takes a logistic regression formulation. The conditional probability of the target spin only depends on the status of the spins that are directly connected with it, hence the *neighborhood pursuit*.

With Equantion (3), the conditional probability of any $spin_i$ can be realistically computed with the neighboring spins' status. To further improve the precision of the traffic Ising model, we propose the following domain-specific optimization methods for traffic problems.

(a) Message Passing with Dynamic Attention: Inspired by message passing and attention-based aggregation in GNNs (Geng et al. 2020, 2021; You et al. 2022; Zhang et al. 2021), we propose to use a Neighboring Influence Matrix $\mathbf{M^i}$ to represent the severity of congestion in each segment's neighborhood in the certain time slice. This Neighboring Influence Matrix can be used as attention in training making the resulting Reconstruct-Ising model better take in information from the propagation of congestion from neighborhoods. Note that as each segment may have different congestion status in different time slices leading to an evolving neighborhood congestion map, the Neighboring Influence Matrix is dynamic. Figure 3 illustrates how to create the Neighboring Influence Matrix using a simplified traffic network as a motivational example. In particular, the neighboring influence from Segment A to B is determined by the congestion ratio of A's directly physically connected neighbors, excluding B. If two segments are not connected, their neighboring influence is 0. During training, this Neighboring Influence Matrix will be online updated upon the change of time slices. With the neighboring influence data enabled as attentions, the training process focuses more on the impact of the congestion from the neighborhood that can be timely propagated through the physical traffic network. More importantly, with the proposed Neighboring Influence Matrix enabled, the congestion messages of the indirectly-connected neighbors of a certain segment are first aggregated as neighborhood-level messages and then are efficiently passed onto the the target node, which improves Reconstruct-Ising's capability to represent the cascading effects of traffic congestion.

(b) Geographical Information Embedding: With the Ising parameter matrix I_i, we propose a domain-specific $\ell 1$ regularizer, $\Lambda_i^{(k)} \equiv \lambda \left\| M_i^{(k)} \cdot \mathsf{D}_i \cdot I_i \right\|_1$, which uses two traffic matrices including: for the ith spin, (i) the ith row of the dynamic neighboring influence matrix ,M_i, labeled by sample index k and (ii) the ith row of the distance matrix D_i that records the geometric information between the i'th spin and its physically connected neighbors. Using this norm to regularize the coupling parameters, Reconstruct-Ising efficiently learns the delay of congestion message propagation among the physically connected segments. Specifically, the ℓ's impact increases with shorter distance, and vice versa.

(c) Robust Learning with Different Levels of Uncertainty Severity: We inject different levels of uncertainty in training and average their loss values to update the parameters, making the regression more robust to the traffic graph reconstruction tasks with random uncertainty from the real world. We use training data with different missing rates to represent different levels of uncertainty. We use an Observing Matrix $\mathbf{O^t}$ to record whether the traffic state of each segment is missing/unobserved. Figure 3(d) gives an example of Observing Matrix. During training, this matrix is used as a mask to filter out the information of unobserved spins.

With all these optimizations applied, to accurately estimate the coupling parameter J_{ij} and h_i of Reconstruct-Ising model, we need to solve the ℓ_{traffic}-constrained logistic regression problem for each spin σ_i by minimizing the following loss L_i.

$$L_i = \frac{1}{7K} \sum_{k=1}^{K} \sum_{u=0\%}^{U=60\%} y_i^k \ln\left(\hat{y}_i^k\right) + \left(1 - y_i^k\right) \ln\left(1 - \hat{y}_i^k\right) + \Lambda_i^{(k)}, \tag{4}$$

where K is the number of samples, and

$$\hat{y}_i^k \equiv \frac{1}{1 + e^{-2\left[I_{ii} + \sum (I_i \cdot O_i^k \cdot (s_{\setminus i}^k) \cdot M_i^k)\right]}}, \tag{5}$$

is the probability of $\sigma_i = +1$ in the kth sample which is evaluated with observation $s_{\setminus i}^k$ using Observation Matrix O_i indicating which nodes are unobserved.

To summarize, we use the following algorithm to learn Reconstruct-Ising and estimate its parameters.

Forward Reconstruct-Ising with Ising Machine:

After Inverse Reconstruct-Ising, we conduct Forward Ising with the learnt Ising model to impute the missing data and reconstruct the graph, which will be further used for prediction. As discussed in Section 2, BRIM is used for graph reconstruction in Ising-Traffic. A simple first-principle analysis shows that it takes simulated annealing an average of about 10^5 instructions to mimic one step in the random walk of a 1000-node energy landscape. The same effect in BRIM

Algorithm 1: Inverse Reconstruct-Ising

1: **for** each node $\sigma_i, i \in [0, N]$ **do**
2: Initialize $I_i = [I_{i,1}, I_{i,2}, ..., I_{i,N}]$ where $I_{i,j} = J_{i,j}, I_{i,i} = h_i$
3: $I_i \leftarrow \text{argmin}_I\{\text{Equation (4)}\}$ through mini-batch gradient descent with learning rate decay.
4: **end for**
5: concatenate $\{I_0, I_1, ..., I_N\}$ to a matrix $I \in \mathbb{R}^{N\times N}$
6: **return** $\left(I + I^T\right)/2$

requires a single spin flip resulting from a single charge/discharge of a nano-scale capacitor (e.g., 50fF) which can happen on the order of pico-seconds. More details can be found in (Afoakwa et al. 2021; Sharma et al. 2022). The latency of traffic graph data reconstruction is evaluated in Section 4.3, demonstrating that BRIMs can reconstruct traffic graphs accurately within 100 ns.

Mapping the Ising Model to the Ising Machine: To correctly map the Ising model trained during the inverse process onto BRIM, we set resistor values as $R_{ij} = 1/J_{ij}$.

3.3 Predict-Ising

Mapping Time Sequence Prediction onto Predict-Ising:

Unlike Reconstruct-Ising, which only focuses on spatial correlations in the same time slice, Predict-Ising learns temporal correlations among segments and spins from historical congestion data across different time slices. That is, the probability function used for Reconstruct-Ising (Equation (1)) needs to be augmented to support temporal information. We derive the Predict-Ising augmentation in the following steps.

First, we transform the joint probability given by Equation (3) to

$$P(s) = \frac{1}{\mathcal{Z}} \exp\left\{-\mathcal{H}_i - \mathcal{H}_{\setminus i}(s_{\setminus i})\right\} , \tag{6}$$

where $\mathcal{H}_i \equiv h_i\sigma_i + \sum_j^N J_{ij}\sigma_i\sigma_j$ and $\mathcal{H}_{\setminus i} \equiv \sum_j^N h_j\sigma_j + \sum_{j<k}^N \sigma_j\sigma_k$. According global Markov property, we know that given an undirected graph, if a certain subset of nodes γ is a separator of two sets of remaining nodes A and B, we have $P(A, B|\gamma) = P(A|\gamma) \cdot P(B|\gamma)$. Then it can be easily proved that the probability of each spin's state depends solely on its directly-connected neighborhoods. Thus, the probability function can be recast into the following expression:

$$\begin{aligned} P(s) &= \frac{1}{\mathcal{Z}} e^{-\mathcal{H}_i} \cdot e^{-\mathcal{H}_{\setminus i}(s_{\setminus i})} \propto e^{-\mathcal{H}_i} \\ &= e^{-\left(h_i\sigma_i + \sum_j^N J_{ij}\sigma_i\sigma_j\right)} , \end{aligned} \tag{7}$$

Furthermore, we are able decompose the transition of the global congestion distribution between adjacent time slices into the transition of the congestion status of individual spin. For the transition of each spin, we use an augmented regression model as expressed in Equation (8) where H_i defined in Equation (7) is used in logit transformation.

$$\begin{aligned} P(\sigma_i^t = 1|s(t-1)) &= \frac{1}{1 + e^{-\mathcal{H}_i^{t-1}}} \\ &= \frac{1}{1 + e^{\sigma_i^{t-1}\left(h_i + \sum_j^N J_{ij}\sigma_j^{t-1}\right)}} , \end{aligned} \tag{8}$$

With this expression, a similar learning approach to Inverse Reconstruct-Ising can be used to learn Predict-Ising.

Inverse Predict-Ising:

As mentioned above, the Inverse Predict-Ising process is similar to Reconstruct-Ising. Therefore, we mainly introduce their differences. Unlike reconstruction, the prediction needs to consider temporal information in parameter estimation. Hence, we extend the loss function of Reconstruct-Ising to the following expression:

$$L_i = \frac{1}{T}\sum_{t=1}^{T} y_i^t \ln\left(\hat{y}_i^t\right) + \left(1 - y_i^t\right)\ln\left(1 - \hat{y}_i^t\right) + \Lambda^t , \tag{9}$$

where $\hat{y}_i^t$ denotes the probability of σ_i in time slice t and $\Lambda^t \equiv \lambda\left\|M_i^{t-1} \cdot \mathsf{distance}_{ij} \cdot I_i\right\|_1$. We further apply the dynamic Neighboring Influence Matrix M (Fig. 3(c)) and get:

$$\hat{y}_i^k = \frac{1}{1 + e^{\sigma_i^{t-1}\left[I_{ii} + \sum\left(I_i \cdot O_i^{t-1} \cdot (s_{\setminus i}^{t-1}) \cdot M_i^{t-1}\right)\right]}} . \tag{10}$$

The learning algorithm of Inverse Predict-Ising is similar to Algorithm 1. Note that we propose to learn Predict-Ising jointly with Reconstruct-Ising for the following reasons: (1) joint-learning alleviates the influence of cascading errors from congestion imputation to prediction; (2) joint-learning makes the models better learn the correlations between temporal and spatial relationships among segments.

Forward Predict-Ising with Traditional Processors:

Instead of using the Ising Machine for annealing to find the ground states of Predict-Ising model as prediction results, Forward Predict-Ising directly runs the transition probability function (Equation (8)) on traditional processors to predict the configurations of spins in the next time slice ($s(t+1)$) based on the current states ($s(t)$). As discussed in Section 1, although we do not use an Ising machine for the Forward Predict-Ising, the Forward Predict-Ising performed on traditional processors can still provide significant speedups as it only requires 1.8% arithmetic operations of SOTA methods.

4 Experimental Results

This section first evaluates the accuracy of Ising-Traffic, including the accuracy of graph reconstruction only, prediction only, and reconstruction + prediction and then evaluates the prediction latency of Ising-Traffic.

4.1 Experimental Setup

Datasets: We use four real-world datasets (*Q-traffic, PEMS4, PEMS8, PEMS-BAY*). ***Q-traffic*** contains traffic speed per 15-minutes of 15,073 city road segments in Beijing, including 5856 time slices.

Missing Rate	LRTC-TNN (Baseline)	Ising-1	Ising-2	**Ising-final (Final Design)**
10%	82.20/97.01/96.05/95.42	89.73/87.43/83.06/82.46	94.69/93.77/96.71/95.15	**96.75/99.45/99.23/99.74**
20%	82.18/96.95/95.93/94.98	89.78/83.26/74.96/77.63	94.63/92.34/95.47/93.75	**96.66/99.21/99.07/99.63**
30%	82.17/96.88/95.89/94.75	89.39/80.03/70.17/74.20	94.4/90.54/93.40/91.48	**96.41/98.98/98.83/99.49**
40%	82.16/96.81/95.86/94.67	89.40/76.98/64.72/71.34	94.15/88.25/89.01/89.02	**96.19/98.69/98.68/99.12**
50%	82.05/96.66/95.67/94.42	89.72/74.40/63.06/68.73	93.51/85.47/85.82/85.56	**95.85/98.43/98.50/98.97**
60%	81.56/96.41/95.39/94.21	88.81/70.84/56.71/66.24	93.35/81.91/80.29/81.83	**95.13/98.21/98.43/98.92**

Table 1: Accuracy of Reconstruct-Ising & Baseline. Q-traffic, PEMS8, PEMS4, PEMS-BAY are separated by dashes.

	P4(%)	P8(%)	PB(%)
Ising-Joint	**99.72**	**99.62**	**99.48**
HA	93.56	92.44	91.91
ARIMA	95.63	96.79	95.30
TGCLSTM	97.52	96.61	97.55
Seq2Seq	97.51	97.99	97.62
STTN	98.52	97.94	97.75
DKFN	97.76	96.78	97.49
GTS	97.69	97.77	97.73
STGCN	97.89	98.52	98.01
HGCN	96.93	97.77	98.05

Table 2: Accuracy comparison of Predict-Ising & baselines.

PEMS4 (P4) contains speed data in San Francisco Area from 01/01 to 02/28 in 2018 with 307 freeway segments. ***PEMS8 (P8)*** contains speed data in San Bernardino from 06/01 to 08/31 in 2016 with 170 freeway segments. ***PEMS-BAY (PB)*** contains speed data in Bay Area from 01/01 to 05/31 in 2017 with 325 freeway segments.

Baselines: This research compares the proposed approach with carefully selected SOTA works in congestion prediction as baselines for the prediction stage including Historical Average (HA), Autoregressive Integrated Moving Average (ARIMA), TGCLSTM (Cui et al. 2019), Seq2Seq (Sutskever, Vinyals, and Le 2014), STTN (Xu et al. 2020), DKFN (Chen et al. 2020), GTS (Shang, Chen, and Bi 2021), STGCN (Yu, Yin, and Zhu 2017), and HGCN (Guo et al. 2021b). These works use different methodologies as follows. ***TGCLSTM*** is based on Convolutional Long Short-Term Memory Neural Network. ***Seq2Seq*** uses the encoder-decoder architecture with gated recurrent units. ***STTN*** is a spatial-temporal network using Transformer to learn spatial-temporal correlations. ***DKFN*** uses Kalman filtering network. ***GTS*** learns a graph structure among multiple time slices and predicts simultaneously. ***STGCN*** combines the gated temporal convolution and graph convolutions. ***HGCN*** is a hierarchical Graph Convolution Network that divides traffic graphs into micro and macro parts. Moreover, we select the current SOTA traffic reconstruction model, ***LRTC-TNN***, as the baseline for reconstruction.

Platforms: The Forward Reconstruct-Ising of Ising-Traffic is performed on a simulated BRIM system (Afoakwa et al. 2021). The GPU and CPU used to evaluate the latency of baseline solutions and perform Forward Predict-Ising are Nvidia A100-40GB and Intel Xeon Gold 6330.

4.2 Evaluation of Accuracy

Reconstruction Only: We first evaluate the accuracy of reconstruction. Table 1 compares the accuracy of Reconstruct-Ising with different optimization choices with the baseline, LRTC-TNN, at different levels of uncertainty. **Ising-1** is the basic version of Reconstruct-Ising learned with gradient descent and the lasso for sparsity but without the proposed domain-specific algorithmic optimizations discussed in Section 3. **Ising-2** is trained with the support of robust training with different levels of uncertainty. **Ising-final** is the final design with all optimizations enabled and is used for the rest of the evaluation. Our results demonstrate that the proposed optimizations can improve data reconstruction accuracy significantly. In particular, Ising-1 provides 85.67% accuracy on average; Ising-2 improves the accuracy to 95.08%; Ising-final further improves the accuracy to 98.79%. Compared to the baseline, the proposed Reconstruct-Ising delivers 6.12% improvement in accuracy.

Prediction Only: Table 2 compares the prediction-only accuracy of Predict-Ising and baseline models without uncertainty. In another word, the input traffic graphs include complete and accurate congestion information of previous time slices. The evaluation shows that the proposed Predict-Ising, which only requires at most 1.8% operations of baselines, is able to deliver over 99.4% prediction accuracy, outperforming all 9 baselines. Note that it only takes 10 epochs to train Predict-Ising to achieve high accuracy , but it takes over 100 epochs to train the baselines to reach acceptable accuracy.

Reconstruction + Prediction: Table 3 compares the overall accuracy of the proposed Ising-Traffic framework and baseline models with different levels of uncertainty. In particular, the input graphs only include partial congestion information of an entire traffic system. The missing information is first imputed through Reconstruct-Ising for Ising-Traffic and LRTC-TNN for baseline models, and then the resulting imputed graphs are used for congestion prediction. The results from our method without imputation (Ising-N) show that uncertainty from input graphs results in significant accuracy degradation. In the case of a 50% missing rate, the accuracy is decreased by over 22%. With Ising-Traffic (Ising-F), the prediction accuracy with 50% uncertainty is still over 99%, only 0.4% lower than the ones without uncertainty reported in Table 2 and 5% higher than baselines.

4.3 Evaluation of Latency

This subsection evaluates the computational efficiency of Traffic-Prediction systematically.

	Ising-N	**Ising-F**	TGCLSTM	Seq2Seq	STTN	DKFN	GTS	STGCN	HGCN
P8	train / test	train / test	train / test	train / test	train / test	train / test	train / test	train / test	train / test
10%	95.4 / 95.1	**99.7 / 99.6**	95.6 / 93.7	97.1 / 95.5	96.9 / 95.0	95.8 / 93.8	96.8 / 94.8	96.9 / 94.9	96.8 / 94.8
50%	72.2 / 71.6	**99.7 / 99.3**	94.4 / 93.3	95.8 / 94.7	95.8 / 94.6	94.6 / 93.5	95.6 / 94.5	95.7 / 94.6	95.6 / 94.5
P4	train / test	train / test	train / test	train / test	train / test	train / test	train / test	train / test	train / test
10%	95.5 / 95.2	**99.3 / 99.2**	95.5 / 93.6	95.5 / 93.6	96.5 / 94.6	95.8 / 93.8	95.7 / 93.8	96.5 / 94.6	95.0 / 93.1
50%	73.4 / 71.3	**99.1 / 99.0**	94.4 / 93.2	94.4 / 93.2	95.3 / 94.2	94.6 / 93.5	94.6 / 93.4	95.3 / 94.2	93.8 / 92.7
PB	train / test	train / test	train / test	train / test	train / test	train / test	train / test	train / test	train / test
10%	95.5 / 95.0	**99.4 / 99.4**	95.6 / 93.0	95.6 / 93.1	95.8 / 93.2	95.5 / 93.0	95.7 / 93.2	96.0 / 93.5	96.0 / 93.5
50%	72.0 / 71.6	**99.2 / 99.1**	93.2 / 92.1	93.3 / 92.1	93.4 / 92.2	93.2 / 92.0	93.4 / 92.2	93.7 / 92.5	93.7 / 92.5

Table 3: Overall Accuracy Comparison of Ising-Traffic & 7 Baselines with 10% & 50% unobserved data from 3 datasets.

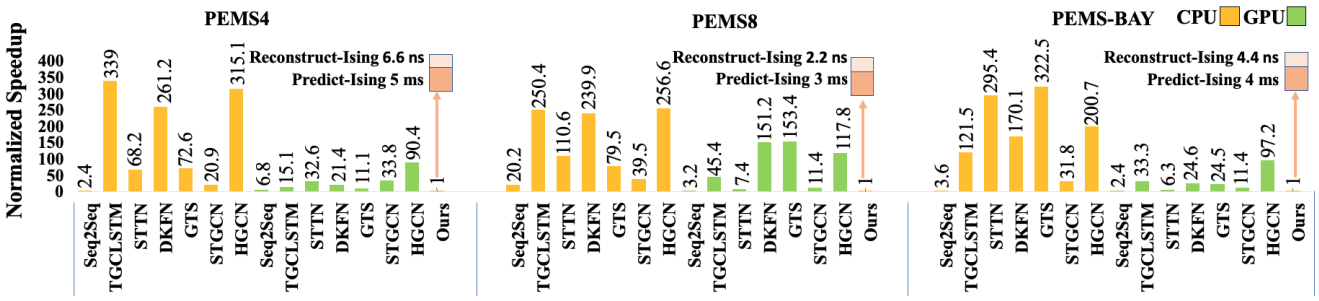

Figure 4: Overall latency comparision of Ising-Traffic and baselines with PEMS4, PEMS8, and PEMS-BAY datasets.

Missing Rate	**Ising Machine (ns)**	LRTC-TNN (ms)
10%	**6.6/2.2/4.4**	3/6/4
20%	**6.6/2.2/4.4**	3/3/4
30%	**6.6/2.2/4.4**	7/4/4
40%	**6.6/2.2/4.4**	2/5/5
50%	**6.6/2.2/4.4**	4/5/5

Table 4: Reconstruction latency of Ising-Traffic & baseline

We first compare the latency of Forward Reconstruct-Ising on BRIMs and LRTC-TNN on Xeon Gold 6330 CPU for data imputation. Table 4 demonstrates that the latency of annealing on BRIMs, i.e., finding the lowe-energy states representing the reconstruction results with 98.7% accuracy, is $10^6\times$ faster and hence negligible compared with LRTC-TNN. Besides the annealing latency, it also takes tens of microseconds to program Ising Machine, which is a one-time cost hence excluded from the results listed in the Table. Note that even if the programming latency is counted, Forward Reconstruct-Ising is still $10^3\times$ faster than the baseline.

Figure.4 compares the overall latency of Ising-Traffic, including the forward process of both Ising models and seven baselines with 10% uncertainty. The total latency of prediction with Ising-Traffic consists of two parts: the Ising machine's annealing for Reconstruction and prediction using traditional processors, i.e., Xeon Gold 6330. Obviously, the proposed Ising-Traffic framework can predict congestion with uncertainty more accurately and, on average, 98× faster than other traditional methods.

5 Conclusion

This paper tackles the challenges in accurate and real-time traffic congestion prediction with uncertainty by proposing Ising-Traffic, an efficient dual-model IML-based traffic prediction framework. Leveraging the Ising model and the Ising machine's inherent and unique capability of finding the low-energy state of a dynamic system and applying it to traffic prediction, Ising-Traffic delivers on average 98× speedups and 5% accuracy improvement over SOTA methods with real-world traffic datasets.

Acknowledgements

This work was supported, in part, by NYS center of excellence; by NSF through grant 2231036, TRIPODS1740735, and CAREER1841569; by the NIH through award TRIPODS1740735; and by faculty research award of META. This research was also partially supported by the U.S. DOE Office of Science, Office of Advanced Scientific Computing Research, under award 66150: "CENATE- Center for Advanced Architecture Evaluation."

References

Afoakwa, R.; Zhang, Y.; Vengalam, U. K. R.; Ignjatovic, Z.; and Huang, M. 2021. Brim: Bistable resistively-coupled Ising machine. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 749–760. IEEE.

Akhtar, M.; and Moridpour, S. 2021. A review of traffic congestion prediction using artificial intelligence. *Journal of Advanced Transportation*, 2021.

Chen, F.; Chen, Z.; Biswas, S.; Lei, S.; Ramakrishnan, N.; and Lu, C.-T. 2020. Graph convolutional networks with

kalman filtering for traffic prediction. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, 135–138.

Cheng, W.; Li, J.-l.; Xiao, H.-C.; and Ji, L.-n. 2022. Combination predicting model of traffic congestion index in weekdays based on LightGBM-GRU. *Scientific reports*, 12(1): 1–13.

Cipra, B. A. 2000. The Ising model is NP-complete. *SIAM News*, 33(6): 1–3.

Cui, Z.; Henrickson, K.; Ke, R.; and Wang, Y. 2019. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11): 4883–4894.

Geng, T.; Li, A.; Shi, R.; Wu, C.; Wang, T.; Li, Y.; Haghi, P.; Tumeo, A.; Che, S.; Reinhardt, S.; et al. 2020. AWB-GCN: A graph convolutional network accelerator with runtime workload rebalancing. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 922–936. IEEE.

Geng, T.; Wu, C.; Zhang, Y.; Tan, C.; Xie, C.; You, H.; Herbordt, M.; Lin, Y.; and Li, A. 2021. I-GCN: A graph convolutional network accelerator with runtime locality enhancement through islandization. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, 1051–1063.

Guo, J.; Liu, Y.; Yang, Q.; Wang, Y.; and Fang, S. 2021a. GPS-based citywide traffic congestion forecasting using CNN-RNN and C3D hybrid model. *Transportmetrica A: transport science*, 17(2): 190–211.

Guo, K.; Hu, Y.; Sun, Y.; Qian, S.; Gao, J.; and Yin, B. 2021b. Hierarchical Graph Convolution Network for Traffic Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 151–159.

Haribara, Y.; Utsunomiya, S.; and Yamamoto, Y. 2016. A coherent Ising machine for MAX-CUT problems: Performance evaluation against semidefinite programming and simulated annealing. In *Principles and Methods of Quantum Information Technologies*, 251–262. Springer.

Inagaki, T.; Haribara, Y.; Igarashi, K.; Sonobe, T.; Tamate, S.; Honjo, T.; Marandi, A.; McMahon, P. L.; Umeki, T.; Enbutsu, K.; et al. 2016. A coherent Ising machine for 2000-node optimization problems. *Science*, 354(6312): 603–606.

King, A. D.; Raymond, J.; Lanting, T.; Isakov, S. V.; Mohseni, M.; Poulin-Lamarre, G.; Ejtemaee, S.; Bernoudy, W.; Ozfidan, I.; Smirnov, A. Y.; et al. 2021. Scaling advantage over path-integral Monte Carlo in quantum simulation of geometrically frustrated magnets. *Nature communications*, 12(1): 1–6.

Kumar, N.; and Raubal, M. 2021. Applications of deep learning in congestion detection, prediction and alleviation: A survey. *Transportation Research Part C: Emerging Technologies*, 133: 103432.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

Liu, Y.; and Wu, H. 2017. Prediction of road traffic congestion based on random forest. In *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, 361–364. IEEE.

Liu, Z.; Yang, Y.; Pan, Z.; Sharma, A.; Hasan, A.; Ding, C.; Li, A.; Huang, M.; and Geng, T. 2023. Ising-CF: A Pathbreaking Collaborative Filtering Method Through Efficient Ising Machine Learning. In *Proceedings of the 60th ACM/IEEE Design Automation Conference. of DAC*.

Lucas, A. 2014. Ising formulations of many NP problems. *Frontiers in physics*, 5.

Peng, H.; Huang, S.; Chen, S.; Li, B.; Geng, T.; Li, A.; Jiang, W.; Wen, W.; Bi, J.; Liu, H.; et al. 2022. A length adaptive algorithm-hardware co-design of transformer on fpga through sparse attention and dynamic pipelining. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 1135–1140.

Ranjan, N.; Bhandari, S.; Zhao, H. P.; Kim, H.; and Khan, P. 2020. City-wide traffic congestion prediction based on CNN, LSTM and transpose CNN. *IEEE Access*, 8: 81606–81620.

Shang, C.; Chen, J.; and Bi, J. 2021. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*.

Sharma, A.; Afoakwa, R.; Ignjatovic, Z.; and Huang, M. 2022. Increasing Ising Machine Capacity with Multi-Chip Architectures. In *2022 International Symposium on Computer Architecture (ISCA '22)*.

Son, S.-W.; Jeong, H.; and Noh, J. D. 2006. Random field Ising model and community structure in complex networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 50(3): 431–437.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Vaidya, J.; Surya Kanthi, R.; and Shukla, N. 2022. Creating electronic oscillator-based Ising machines without external injection locking. *Scientific Reports*, 12(1): 1–8.

Xu, M.; Dai, W.; Liu, C.; Gao, X.; Lin, W.; Qi, G.-J.; and Xiong, H. 2020. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*.

You, H.; Geng, T.; Zhang, Y.; Li, A.; and Lin, Y. 2022. Gcod: Graph convolutional network acceleration via dedicated algorithm and accelerator co-design. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, 460–474. IEEE.

Yu, B.; Yin, H.; and Zhu, Z. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

Zeng, D.; Liu, W.; Chen, W.; Zhou, L.; Zhang, M.; and Qu, H. 2023. Substructure Aware Graph Neural Networks. In *Proc. of AAAI*.

Zhang, Y.; You, H.; Fu, Y.; Geng, T.; Li, A.; and Lin, Y. 2021. G-CoS: Gnn-accelerator co-search towards both better accuracy and efficiency. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 1–9. IEEE.

Zhao, L.; Bao, W.; and Li, W. 2018. The stock market learned as Ising model. In *Journal of Physics: Conference Series*, volume 1113, 012009. IOP Publishing.

Zhao, T.; Roeder, K.; and Liu, H. 2012. Smooth-projected neighborhood pursuit for high-dimensional nonparanormal graph estimation. *Advances in Neural Information Processing Systems*, 25.