Motivation of using diffusion models:

1. Uncertainty prediction
2. Multi-mode problem
3. Recent huge success in image generation. (time series and image are both high dimensional data)

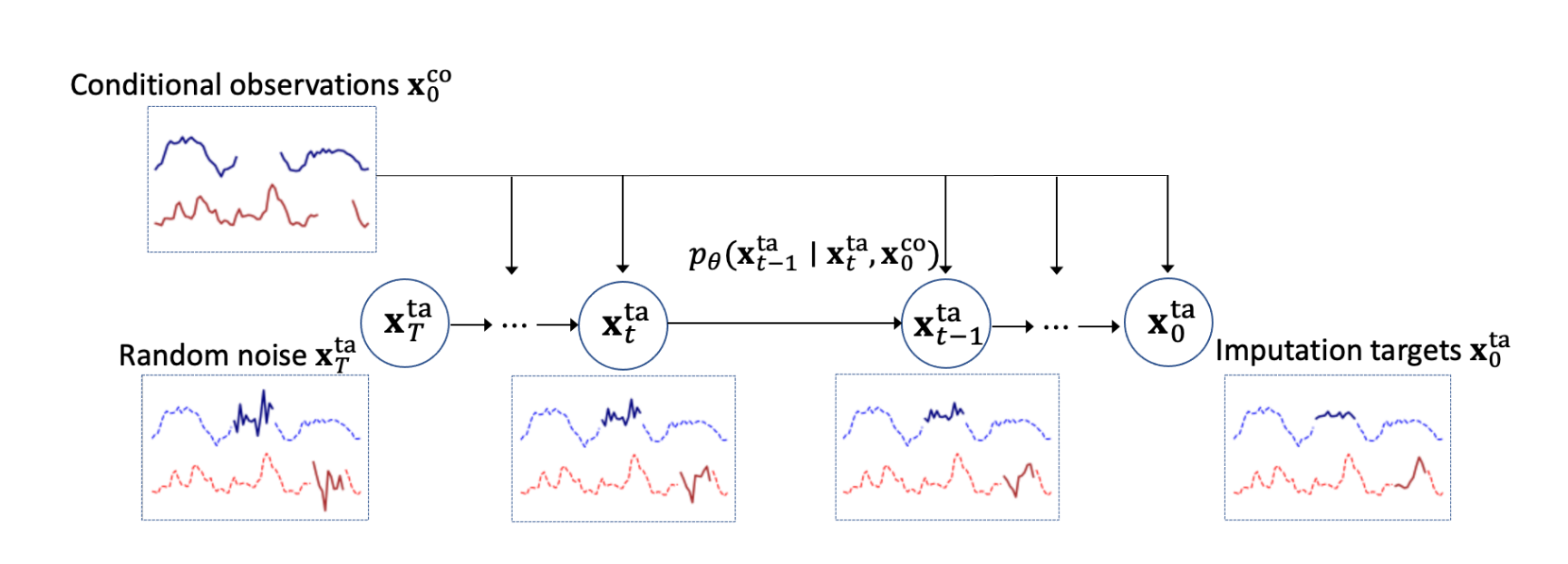
Chart, waterfall chart

Description automatically generated

Graphical user interface

Description automatically generated

CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation (published in NIPS 2021)



Diagram

Description automatically generated

Diagram

Description automatically generated

One Idea: replace the temporal transformer layer with the wave net (by deepmind in 2016), and replace the feature transformer layer with GRIN (a graph neural network that can directly use spatial structure, published in ICLR 2022). Both ideas impose stronger inductive bias, which can potentially work well on our soil moisture dataset.

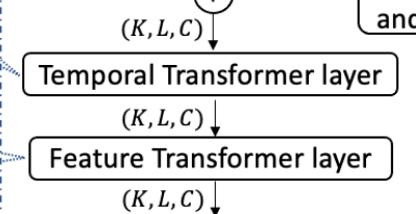
Another idea: replace the temporal and feature transformer layer with our Bidirectional LSTM.

A picture containing chart

Description automatically generated

Diagram

Description automatically generated with medium confidence



**Temporal encoder:**

Simple models, which treat time series as locally dependent, including Recurrent Neural Networks (RNNs), Bidirectional RNNs, Gated Recurrent Units (GRUs), and Long Short-Term Memory (LSTM) networks. These autoregressive models process input data sequentially, with each model offering specific advantages. RNNs are the most basic form but can suffer from vanishing or exploding gradient issues. Bidirectional RNNs enhance the basic RNN by processing input data in both forward and backward directions, allowing for a richer context. GRUs, a variant of RNNs, address the vanishing gradient problem with a gating mechanism. LSTMs, another RNN variant, also employ a gating mechanism but with a more complex structure, making them particularly effective in capturing long-term dependencies. However, these models' sequential nature can result in lengthy computation times.

Temporal Convolution Networks (TCNs) represent an alternative approach by employing Convolutional Neural Networks (CNNs) in the temporal domain. This allows for parallelized computation, which can significantly reduce processing time. TCNs utilize dilated convolutions to expand the receptive field, effectively capturing long-range dependencies while maintaining a manageable number of parameters.

Complex models assume global dependencies within time series data and include Transformer-based models and Longformer. Transformer-based models excel at capturing long-time dependencies through a self-attention mechanism, which allows them to model relationships between any pair of points in the input sequence. However, the quadratic computational complexity of the self-attention mechanism can be problematic for long-range time series, resulting in inefficiencies.

Longformer, an extension of the Transformer model, addresses these computational challenges by using a combination of sliding window attention and dilated attention patterns. This approach allows Longformer to handle long-range dependencies more efficiently while maintaining the advantages of the self-attention mechanism. The Longformer model is particularly well-suited for large-scale time series analysis, offering improved performance in both computation time and memory usage.

**Spatial Encoder**

Simple models that encode spatial information include Graph Neural Networks (GNNs) and Graph Convolution Networks (GCNs). GNNs are a class of neural networks specifically designed to operate on graph-structured data. They are capable of learning meaningful representations of the nodes in a graph by aggregating information from their local neighborhood. GCNs, a popular variant of GNNs, extend traditional convolutional neural networks to graph-structured data. By leveraging the graph's structure and node features, GCNs effectively capture local patterns and relationships within the graph.

Complex models for encoding spatial information comprise Diffusion-Convolutional Neural Networks (DCNNs) and Transformer-based models. DCNNs are a class of deep learning models that capture both local and global structural patterns in graph data. They achieve this by simulating a diffusion process over the graph, allowing the models to consider not only immediate neighbors but also more distant nodes. This approach enables DCNNs to capture long-range dependencies in the graph, making them suitable for a wide range of graph-based tasks.

Transformer-based models, originally designed for natural language processing, have been adapted for graph-structured data. These models leverage self-attention mechanisms to capture relationships between nodes, regardless of their distance in the graph. The self-attention mechanism allows for parallel computation, which can result in improved efficiency compared to sequential processing models. Transformer-based models for graph data, such as Graph Attention Networks (GATs) and Graph Transformers, have shown strong performance in various tasks, including node classification, graph classification, and link prediction.

I’ve already implemented the following models:

Time:

Bidirection-LSTM, Temporal Convolutional Network, Temporal transformer network, longformer (in progress).

Space:

GNN, Diffusion-Convolutional Neural Network (DCNN).

Experient results:

Air quality imputation with covariates, T = 36, 50% missing, 100 epochs.

With no temporal transformer and no spatial transformer:

validation loss = 0.35

With temporal transformer, with positional encoding before feeding into the temporal transformer:

Validation loss = 0.20

With temporal transformer and spatial transformer:

Validation loss = 0.17, testing mae=17.4

With tcn and spatial transformer:

Validation loss = 0.2, testing mae = 37

Air quality imputation with covariates, T = 108, 50% missing, 100 epochs:

With temporal and spatial transformer:

Validation loss = 0.237

Spatio-temporal imputation techniques are used to fill in missing values in datasets with spatial and temporal dimensions. These techniques take advantage of the spatial and temporal correlations in the data to predict missing values. Here are some statistical spatio-temporal imputation methods:

1. Kriging-based methods:
   1. Spatio-Temporal Kriging (STK): This method extends the traditional spatial kriging technique to incorporate the temporal dimension.
   2. Bayesian Spatio-Temporal Kriging (BTK): This method incorporates a Bayesian framework to estimate missing data, accounting for uncertainty in the model.
2. Space-time autoregressive integrated moving average (STARIMA) models
   1. STARIMA models are an extension of ARIMA models used for time series data, incorporating both spatial and temporal autoregressive components.
3. Spatio-temporal regression models:
   1. Generalized Linear Mixed Models (GLMM): These models allow for the inclusion of both fixed and random effects, capturing spatial and temporal dependencies in the data.
   2. Generalized Additive Mixed Models (GAMM): GAMMs extend GLMMs by incorporating smooth functions of spatial and temporal covariates.
4. Dynamic Linear Models (DLM) or State-Space Models: These models use a state equation to describe the underlying data-generating process and an observation equation to describe the observed data. They can be applied to spatio-temporal data by incorporating spatial and temporal dependencies.
5. Tensor Completion Methods:
   1. These methods leverage the low-rank structure of spatio-temporal data to fill in missing values, treating the data as a three-dimensional tensor (with dimensions for spatial locations, time points, and attributes).
6. Matrix Factorization Techniques: By decomposing the spatio-temporal data matrix into lower-dimensional matrices, these methods can impute missing values by reconstructing the original matrix. Examples include Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF).
7. Multiple Imputation:
   1. This approach involves generating multiple imputed datasets using different statistical models, then combining the results to create a single imputed dataset.

Note that some of these methods may require tuning parameters or selecting appropriate covariance functions to achieve optimal performance. Additionally, the choice of the best method will depend on the specific characteristics of your data, such as the degree of spatial and temporal autocorrelation and the nature of the missing data mechanism.

Memory leak issue, why my job gets killed every time?

<https://github.com/tqdm/tqdm/issues/746>

synthetic data:

spatial: 5x5

temporal: 50 days

y: temperature

static features: elevation, distance to water

varying features: wind speed

wind speed is spatial-temporally correlated

error term consists of independent error, spatially-correlated error and temporally-correlated error

missing patterns: random missing, block missing

Goal: impute the missing values