Memformer

A Memory Guided Transformer for Time Series Forecasting

Yunyao Cheng, Chenjuan Guo, Bin Yang, Haomin Yu, Kai Zhao, Christian S. Jensen

February 2025

Proceedings of the VLDB Endowment, Volume 18, Issue 2

Presented by **Andreas Gottschalk Krath**

1. Introduction

Forecasting

- Predicting the future
  - Allows preparation

1.1 Motivation

Forecasting

- Predicting the future
  - Allows preparation
- Long term forecasting?
  - Obviously more difficult than short term
  - Time constrained tasks

Long Term Forecasting

- What defines long term?

Long Term Forecasting

- What defines long term?
- Historical horizon
- Forecasting horizon

Long Term Forecasting

- What defines long term?
- Historical horizon
- Forecasting horizon
- Both exceed 96 time steps
  - Hourly time step $\longrightarrow$ 4 days

Variable Correlation

- Complex systems have many variables
  - These relate to each other
- These impact forecasting accuracy
  - Patterns in the data

Dynamic Correlations

- Are variable correlations stable over time?
  - No

Dynamic Correlations

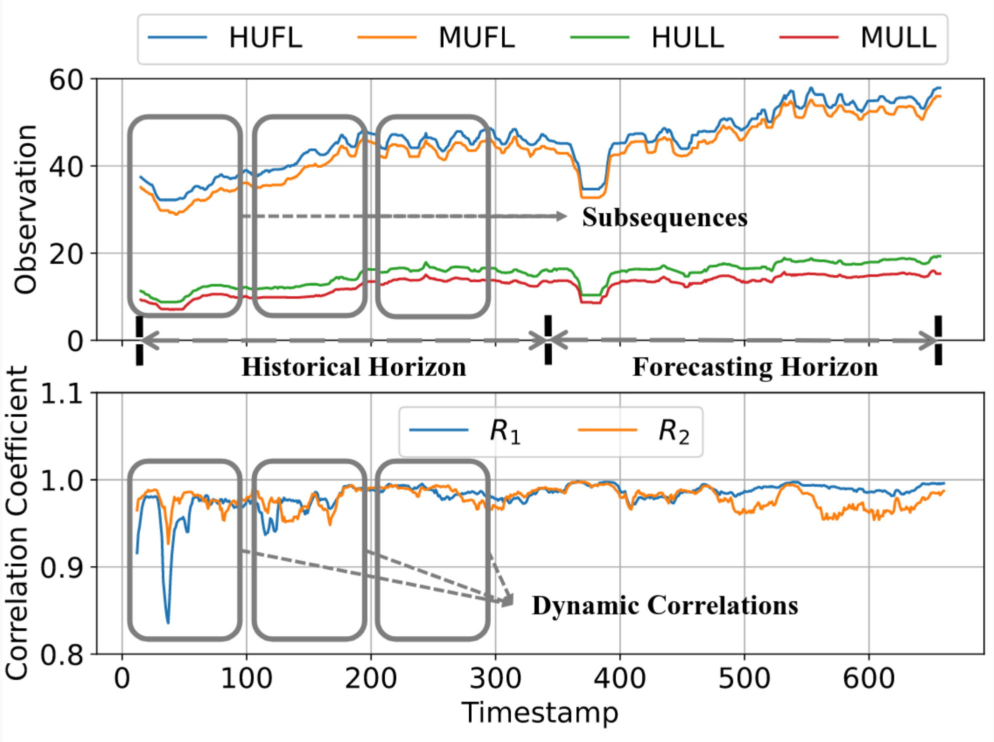
- Are variable correlations stable over time?
  - No
- Correlations are dynamic over time
  - Seasons
  - Sensor drift

Dynamic Correlations

- Are variable correlations stable over time?
  - No
- Correlations are dynamic over time
  - Seasons
  - Sensor drift
- We often consider average
  - Especially hurtful in time series
  - Predictions are bad in periods

Dynamic Correlations

- Are variable correlations stable over time?
  - No
- Correlations are dynamic over time
  - Seasons
  - Sensor drift
- We often consider average
  - Especially hurtful in time series
  - Predictions are bad in periods



(a) Dynamic correlations. The Average $R_1 = 0.995$ and $R_2 = 0.990$.

1.1 Motivation

Disrupted Correlations

- System errors
- External influence

Disrupted Correlations

- System errors
- External influence
- What happens with outliers?
  - Affect correlation $\longrightarrow$ accuracy

1.1 Motivation

Disrupted Correlations

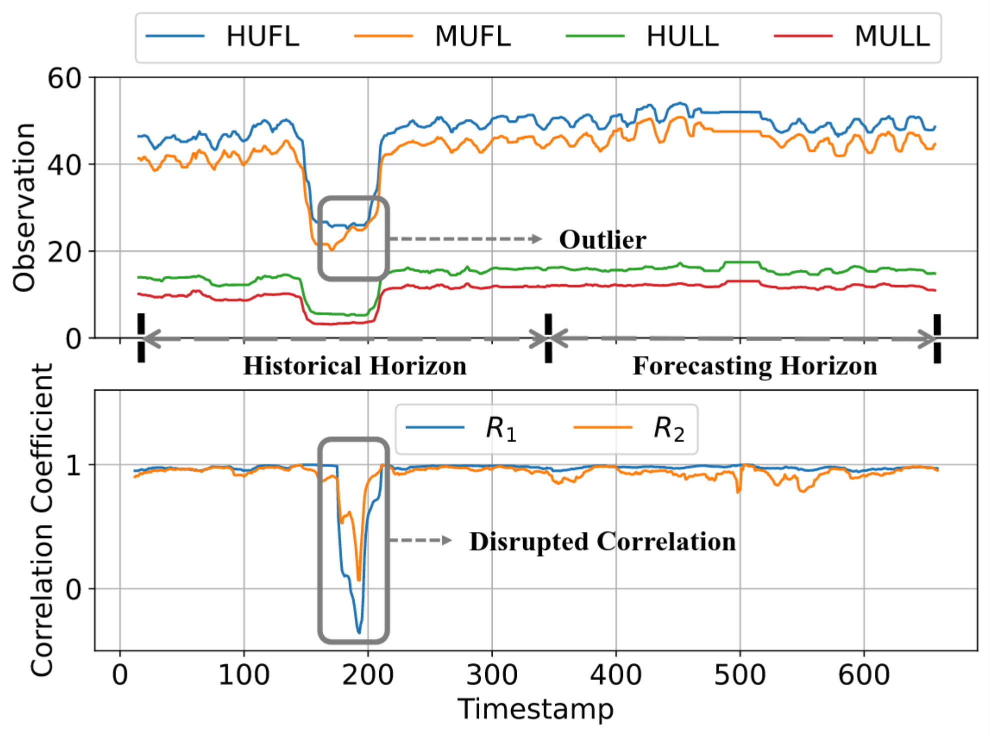
- System errors
- External influence
- What happens with outliers?
  - Affect correlation $\longrightarrow$ accuracy
- Many models are sensitive to outliers
  - Numeric difference dominates training

Disrupted Correlations

- System errors
- External influence
- What happens with outliers?
  - Affect correlation $\longrightarrow$ accuracy
- Many models are sensitive to outliers
  - Numeric difference dominates training
  - Reason for a lot of preprocessing
    - Normalization
    - Clipping
    - Pruning

Disrupted Correlations

- System errors
- External influence
- What happens with outliers?
  - Affect correlation ⟶ accuracy
- Many models are sensitive to outliers
  - Numeric difference dominates training
  - Reason for a lot of preprocessing
    - Normalization
    - Clipping
    - Pruning



(b) Disrupted correlation. The Average $R_1 = 0.908$ and $R_2 = 0.963$.

Challenge 1

- Capture dynamic correlations
- Mitigate disrupted correlations
- Existing solutions struggle with the latter
  - Capture dynamic and disrupted
  - Reduces model robustness

1.2 Problem

Challenge 1

- Capture dynamic correlations
- Mitigate disrupted correlations
- Existing solutions struggle with the latter
  - Capture dynamic and disrupted
  - Reduces model robustness

Challenge 2

- Local information 🤝 global information
- Global information is *all* local information
- Local information *affects* global information
- Existing solutions struggle with combining
  - Only local
  - Only global

1.3 Contributions

Memformer

- Transformer
- Patch-wise recurrent graph learning
  - Captures dynamic correlations
- Global attention
  - Mitigates disrupted correlations
- Adresses challenge 1

Memformer

- Transformer
- Patch-wise recurrent graph learning
    - Captures dynamic correlations
- Global attention
    - Mitigates disrupted correlations
- Adresses challenge 1

Alternating Memory Enhancer

- Memory network
- Associates local and global information
- Adresses challenge 2

1.3 Contributions

Memformer

- Transformer
- Patch-wise recurrent graph learning
  - Captures dynamic correlations
- Global attention
  - Mitigates disrupted correlations
- Adresses challenge 1

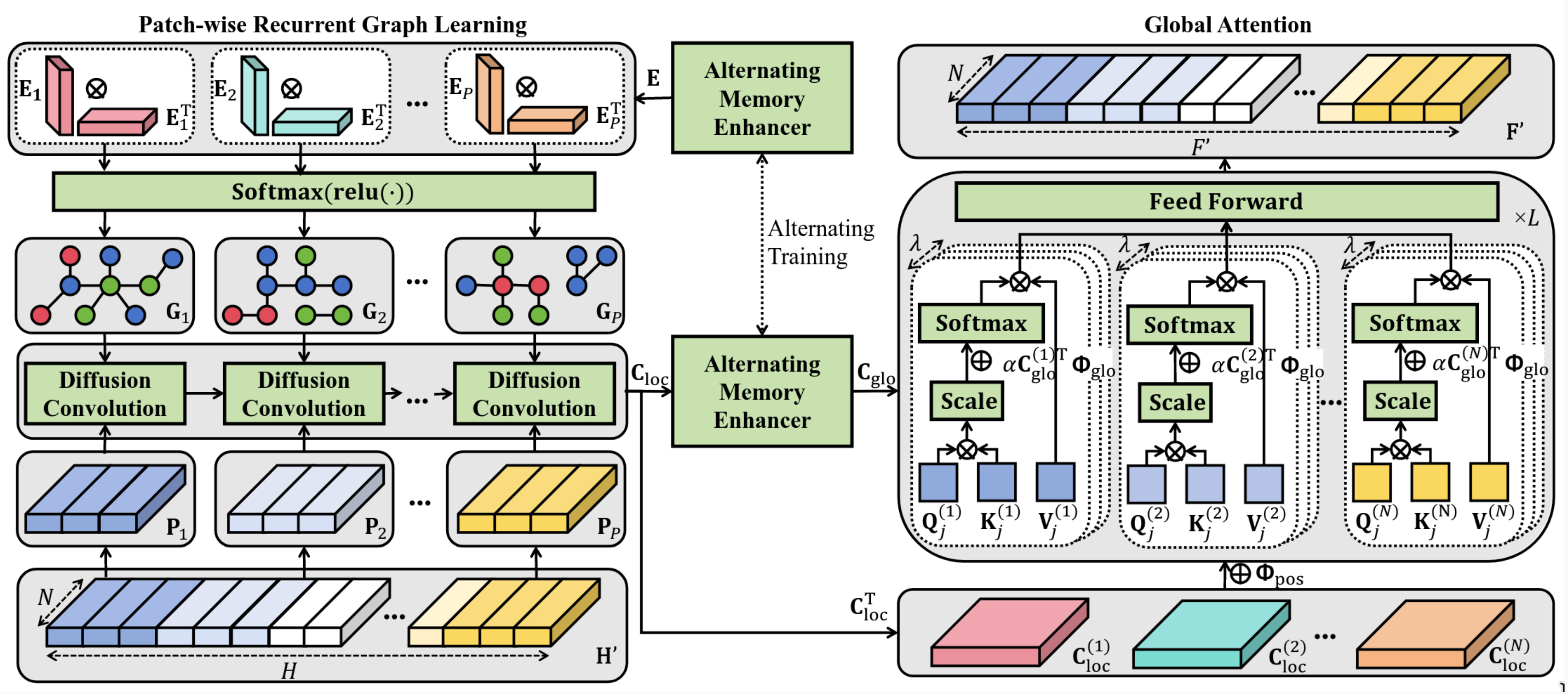
Alternating Memory Enhancer

- Memory network
- Associates local and global information
- Adresses challenge 2

Experiments

- Proof

2. Methodology

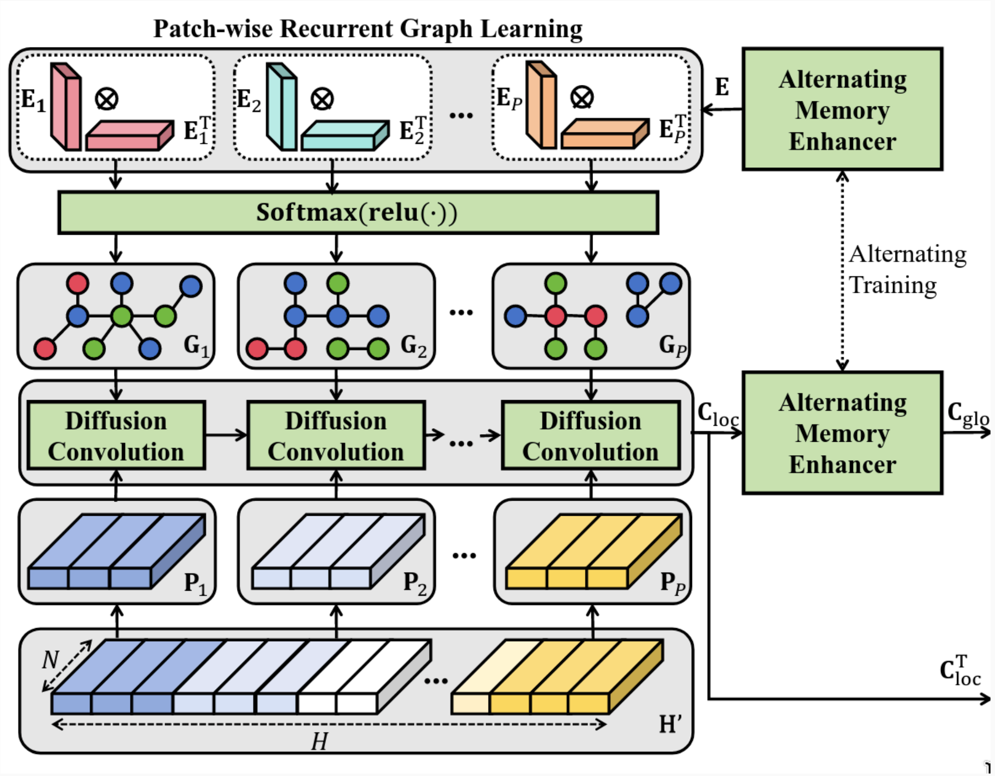

Patch-wise Recurrent Graph Learning
Global Attention
Alternating Memory Enhancer
Softmax(relu(·))
Diffusion Convolution
Alternating Training
Alternating Memory Enhancer
Feed Forward
Softmax
Scale
H'
F'

Architecture

Upper part → dynamic correlation

Lower part → normalized data

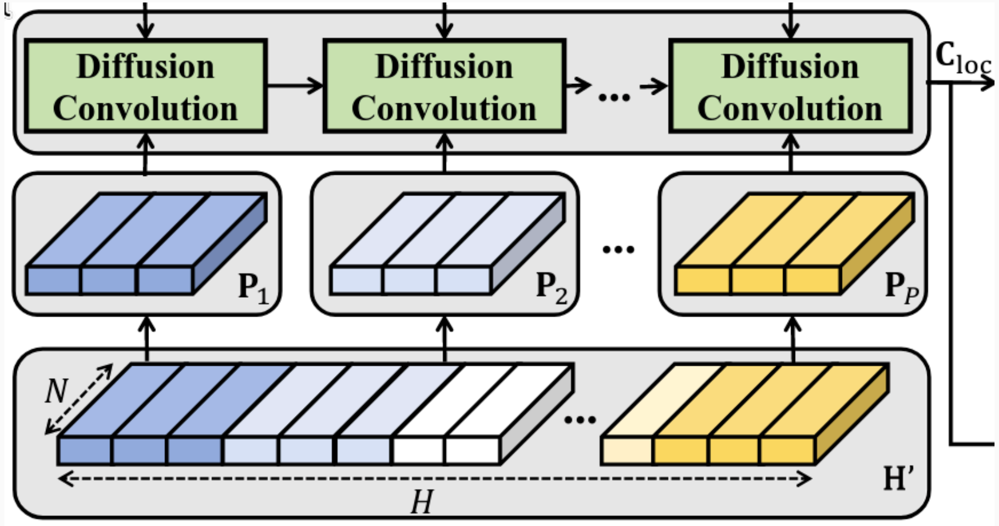
Output → enriched input features



2.2 Patch-wise Recurrent Graph Learning

Normalized Data

- Instance normalization

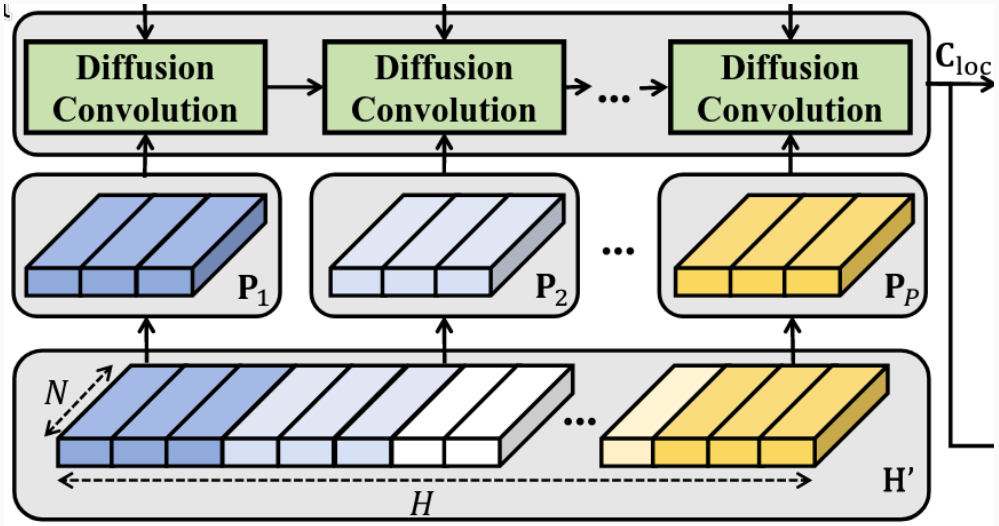


Normalized Data

- Instance normalization

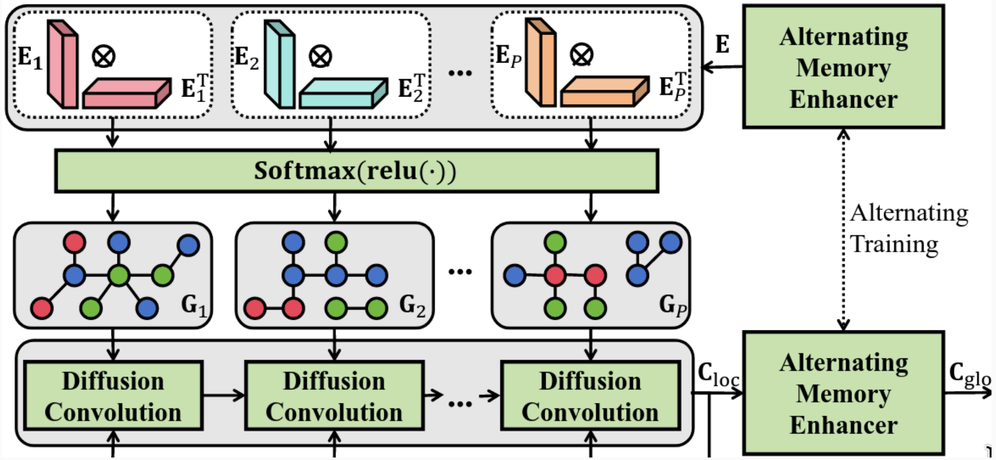
Patches

- H' is split into p patches
- Group temporally related data



AME

- Provides local information
  - These are learnable parameters
- Consistant local information for patch P_i
- Matrix product of $E_i \otimes E_i^T$
  - Similarity matrix for variables in P_i

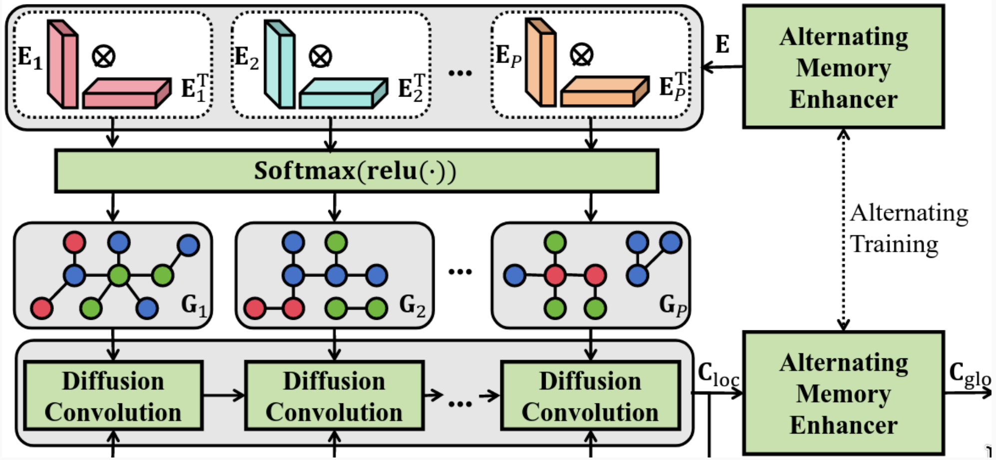


AME

- Provides local information
  - These are learnable parameters
- Consistant local information for patch P_i
- Matrix product of $E_i \otimes E_i^T$
  - Similarity matrix for variables in P_i

ReLU + Softmax

- ReLU eliminates negative values
  - Removes negative correlations
- Softmax scales into influence scores



AME

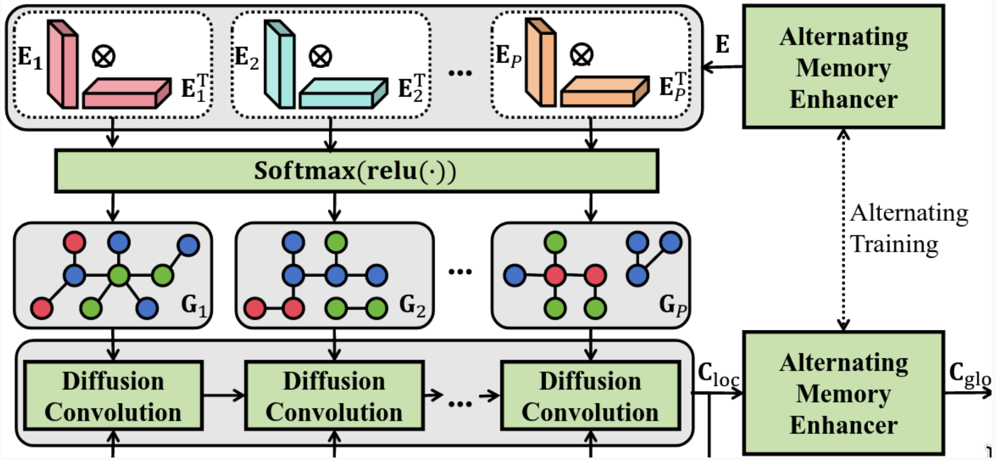
- Provides local information
  - These are learnable parameters
- Consistant local information for patch P_i
- Matrix product of $E_i \otimes E_i^T$
  - Similarity matrix for variables in P_i

ReLU + Softmax

- ReLU eliminates negative values
  - Removes negative correlations
- Softmax scales into influence scores

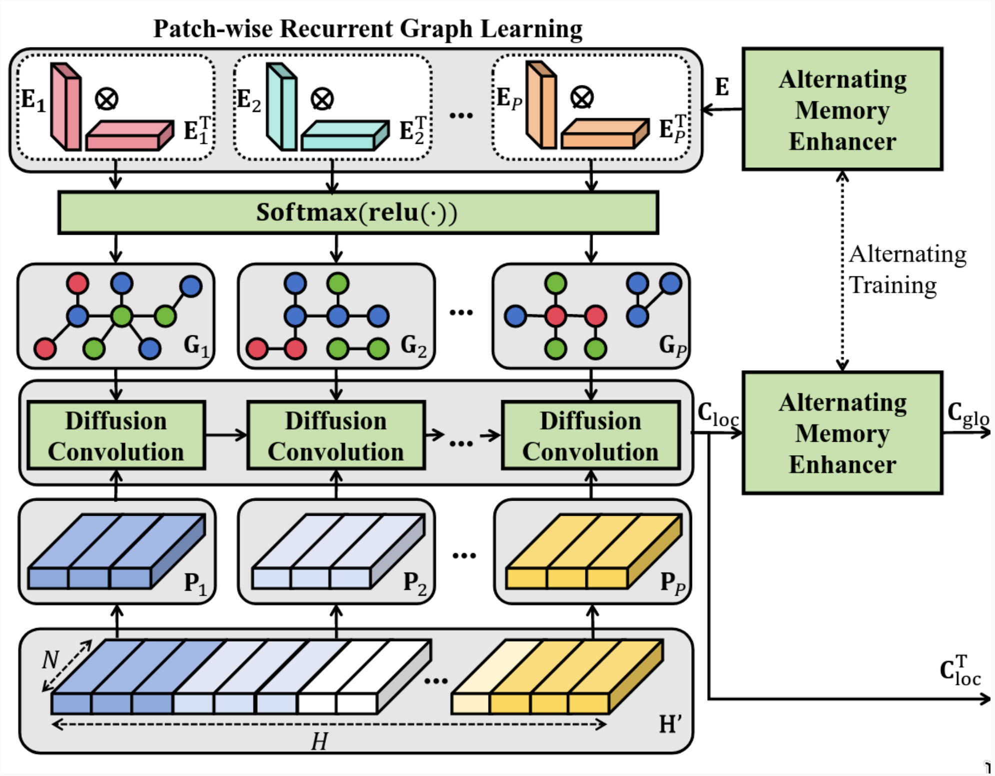
Graph

- Translates influence scores into graph
- Captures connection between variables
  - Dynamic correlations



Diffusion Convolution

- Normalized data is adjusted based on connections in graph
- Numeric values "diffuse" into neighbours
  - Not only immediate neighbours
- Spatially relates data based on connections



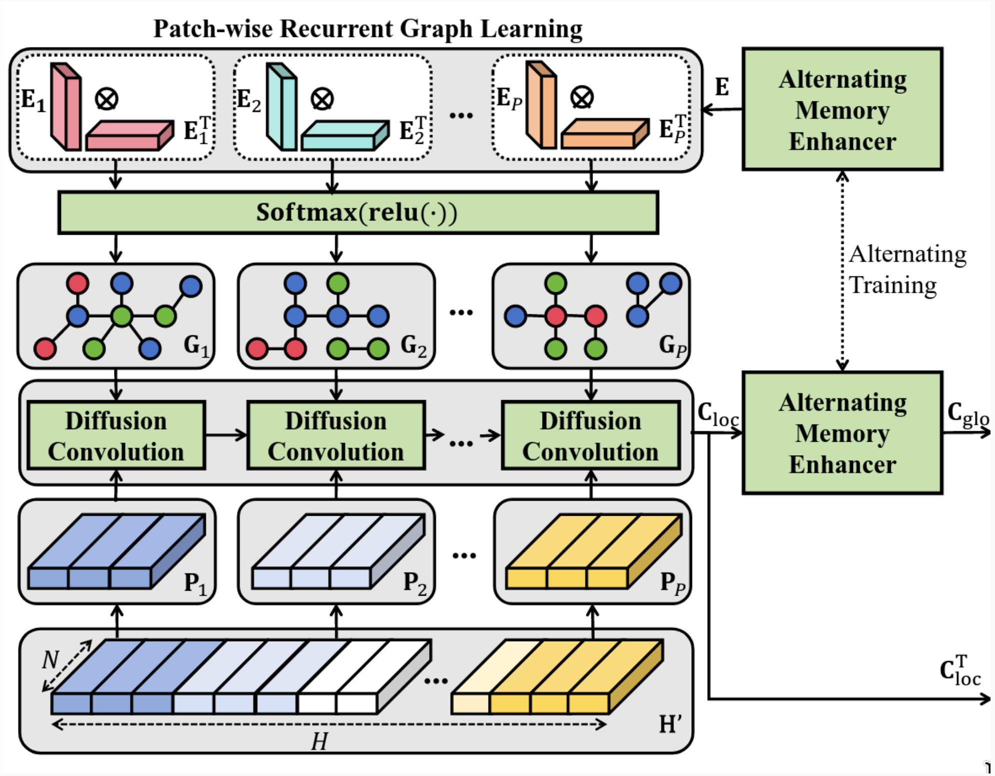
2.2 Patch-wise Recurrent Graph Learning

Diffusion Convolution

- Normalized data is adjusted based on connections in graph
- Numeric values "diffuse" into neighbours
  - Not only immediate neighbours
- Spatially relates data based on connections

Gated Recurrent Unit

- Forwards information from P_i to P_{i+1}
- Temporally relates data in a sequence



Diffusion Convolution

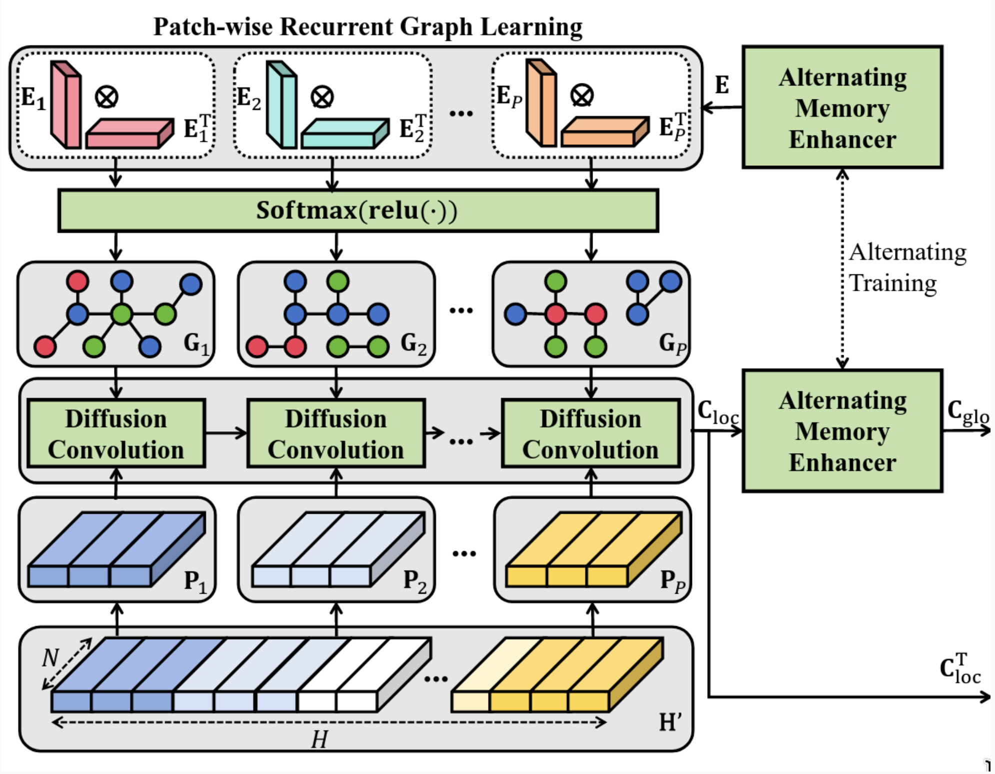
- Normalized data is adjusted based on connections in graph
- Numeric values "diffuse" into neighbours
  - Not only immediate neighbours
- Spatially relates data based on connections

Gated Recurrent Unit

- Forwards information from P_i to P_{i+1}
- Temporally relates data in a sequence

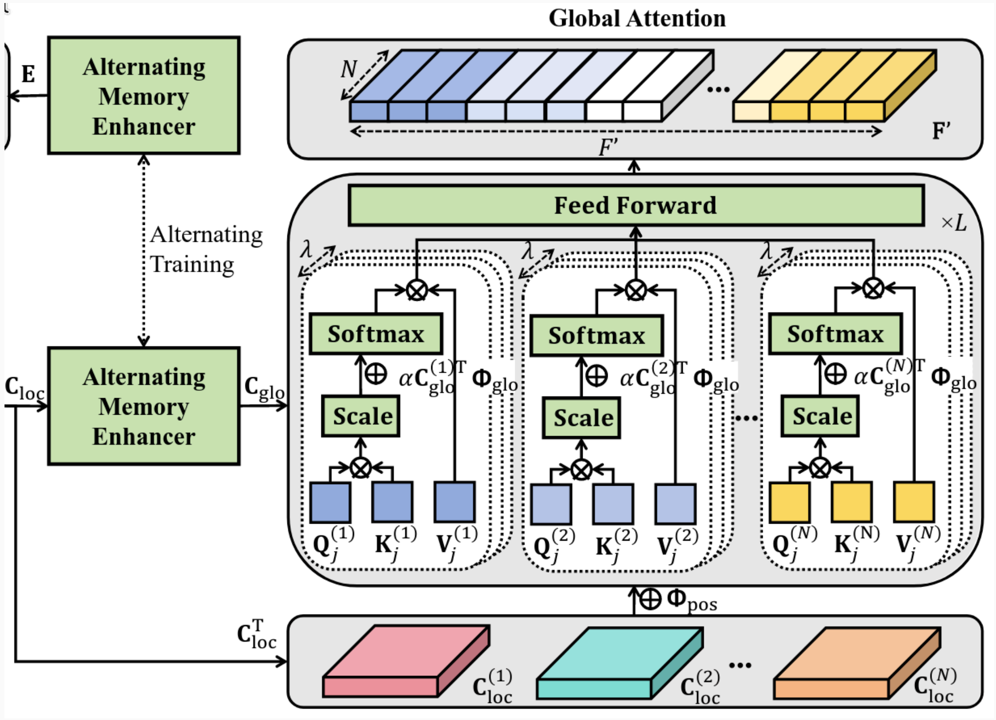
Output

- Input features enriched with local information
- Spatial $\longrightarrow$ dynamic correlations
- Temporal $\longrightarrow$ GRU



Motivation

- Patch-wise correlations are sensitive
  - Outliers dominate
- Constrain locally enriched features
  - Mitigate disrupted correlations

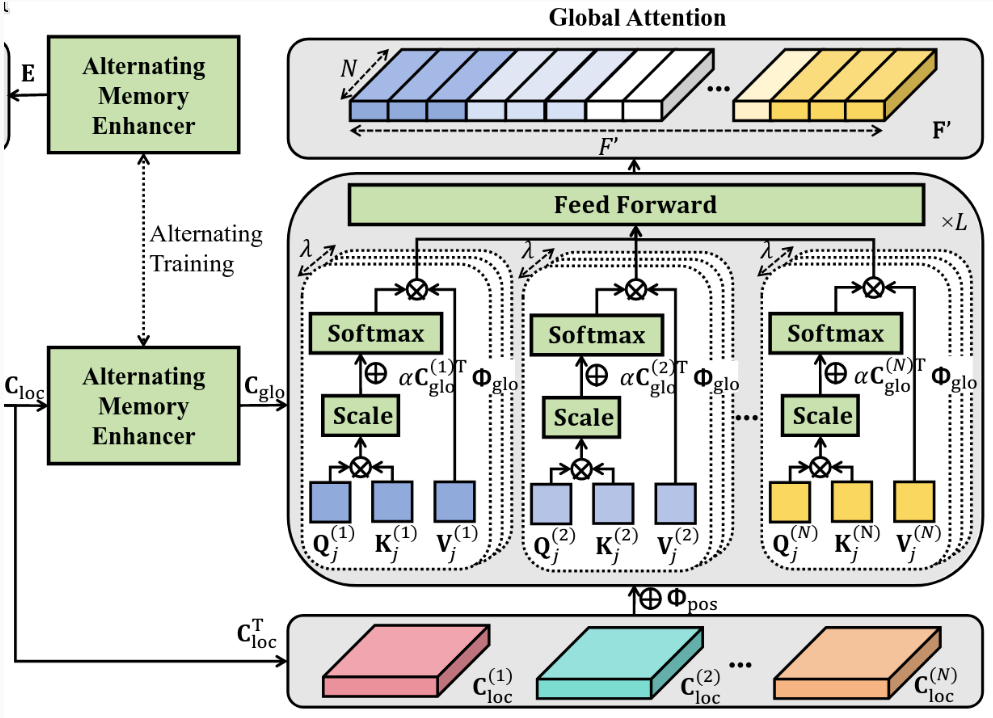


Motivation

- Patch-wise correlations are sensitive
  - Outliers dominate
- Constrain locally enriched features
  - Mitigate disrupted correlations

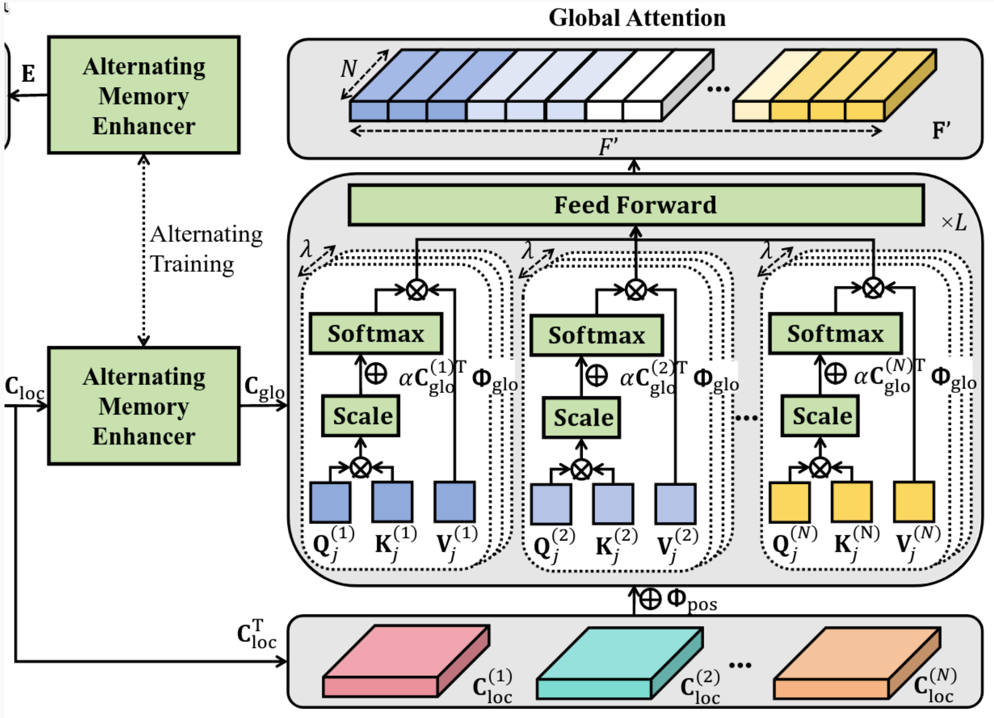
Input

- Transpose locally enriched features
  - Isolate variables
  - Diffusion earlier
- Converted to Q, K, V matrices
  - Learnable parameters



Attention

- Relatively conventional implementation
- Global information is new
- Adding global information after softmax
  - Bias probabilities

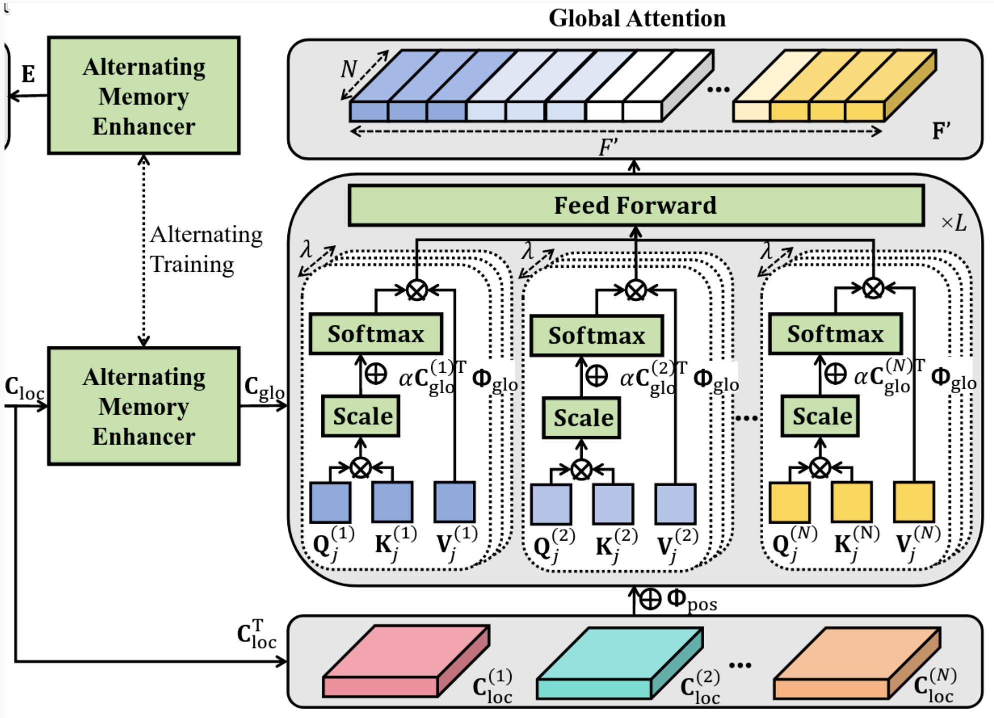


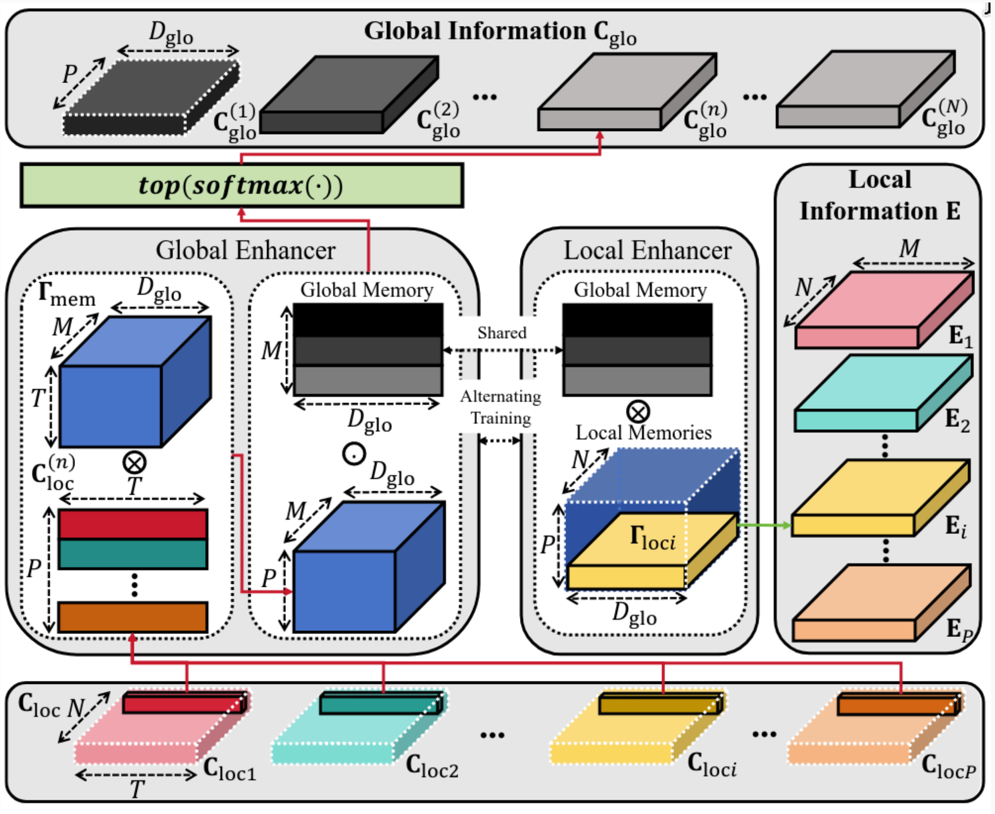
Attention

- Relatively conventional implementation
- Global information is new
- Adding global information after softmax
  - Bias probabilities

Output

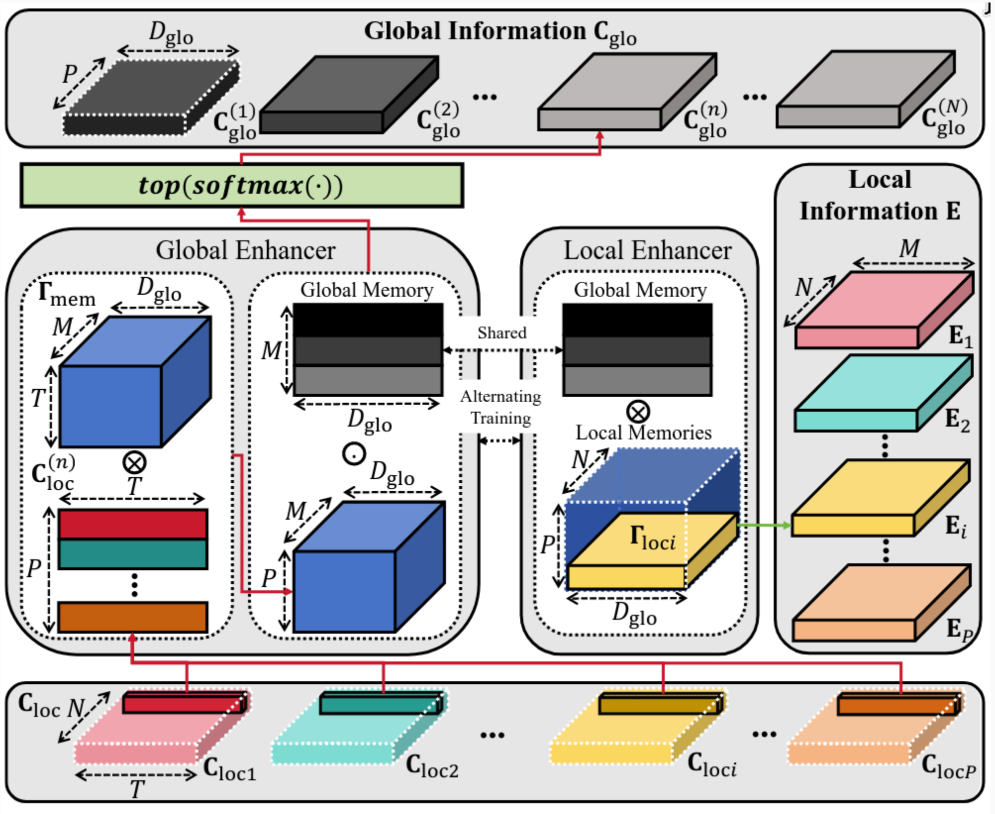
- The final “representation” of data
- **F'** is not a forecast
  - Final feature representation
- Linear layer maps to forecasting horizon




Global Information $\mathbf{C}_{\text{glo}}$
D_{glo}
P
$\mathbf{C}^{(1)}_{\text{glo}}$
$\mathbf{C}^{(2)}_{\text{glo}}$
$\mathbf{C}^{(n)}_{\text{glo}}$
$\mathbf{C}^{(N)}_{\text{glo}}$
$top(softmax(\cdot))$
Local Information E
Global Enhancer
Γ_{mem}
$\mathbf{C}^{(n)}_{\text{loc}}$
Global Memory
Shared
Alternating Training
Local Enhancer
Global Memory
Local Memories
$\Gamma_{\text{loc}i}$
$\mathbf{E}_1$
$\mathbf{E}_2$
$\mathbf{E}_i$
$\mathbf{E}_P$
$\mathbf{C}_{\text{loc}}$
$\mathbf{C}_{\text{loc}1}$
$\mathbf{C}_{\text{loc}2}$
$\mathbf{C}_{\text{loc}i}$
$\mathbf{C}_{\text{loc}P}$

Overview

- Input
  - Locally correlated features
- Outputs
  - Local information E
  - Global information C_{glo}
- Shared global memory

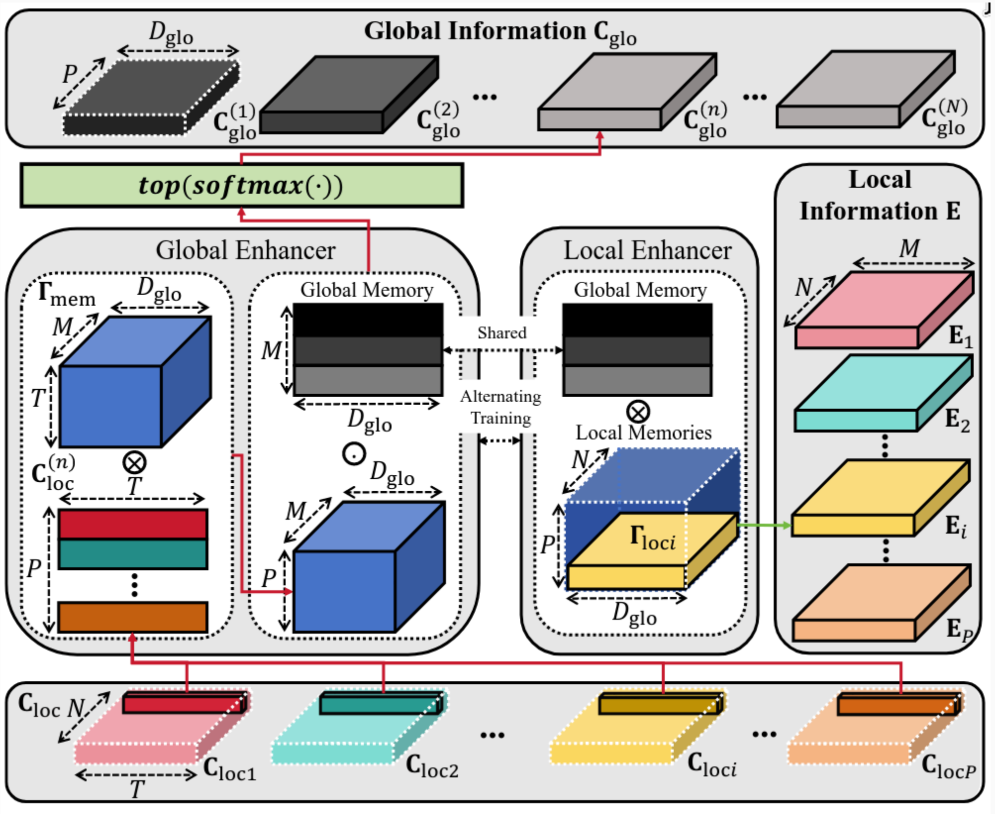


Overview

- Input
  - Locally correlated features
- Outputs
  - Local information E
  - Global information C_{glo}
- Shared global memory

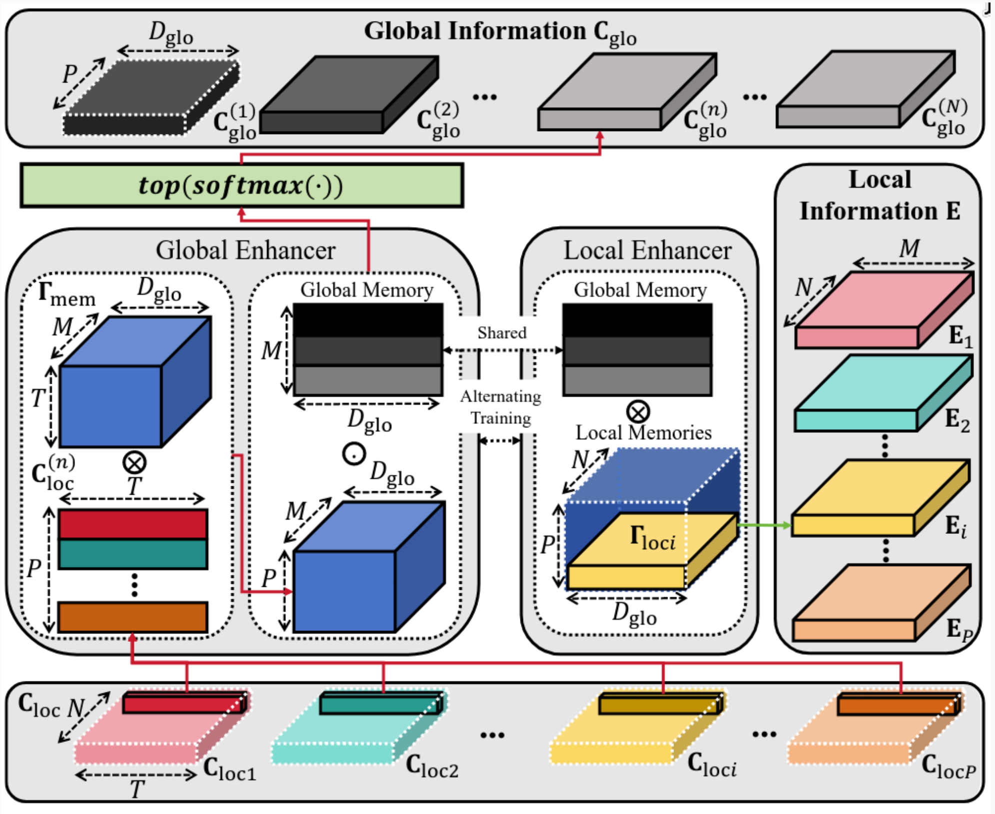
Hyperparameters

- $M \longrightarrow$ number of high level patterns
  - Spikes, seasons, stable
- $D_{\text{glo}} \longrightarrow$ richness of patterns



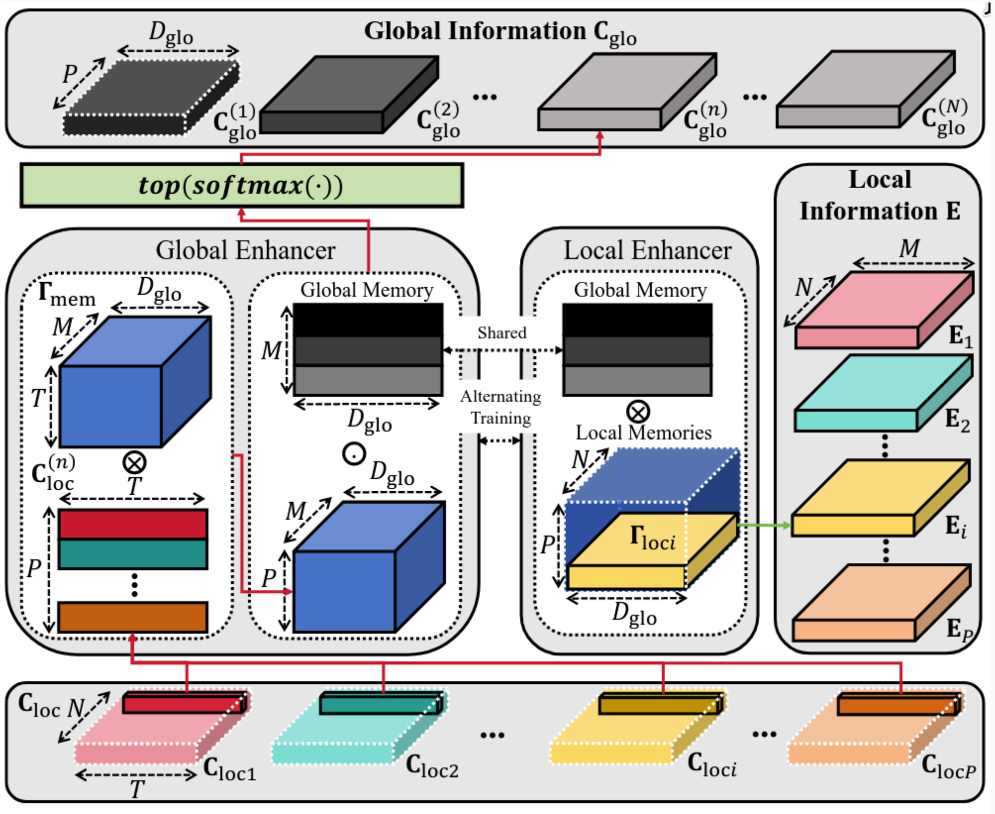
Local Enhancer

- Local memory regions $\Gamma_{\text{loc}i}$
  - One for each patch
- $P_i \longleftrightarrow \Gamma_{\text{loc}i} \longrightarrow \Gamma_{\text{loc}i} \longleftrightarrow E_i$
- Not directly identical
  - E contains global memories influence
  - Defined by C_{loc}
- Memories are **not** information



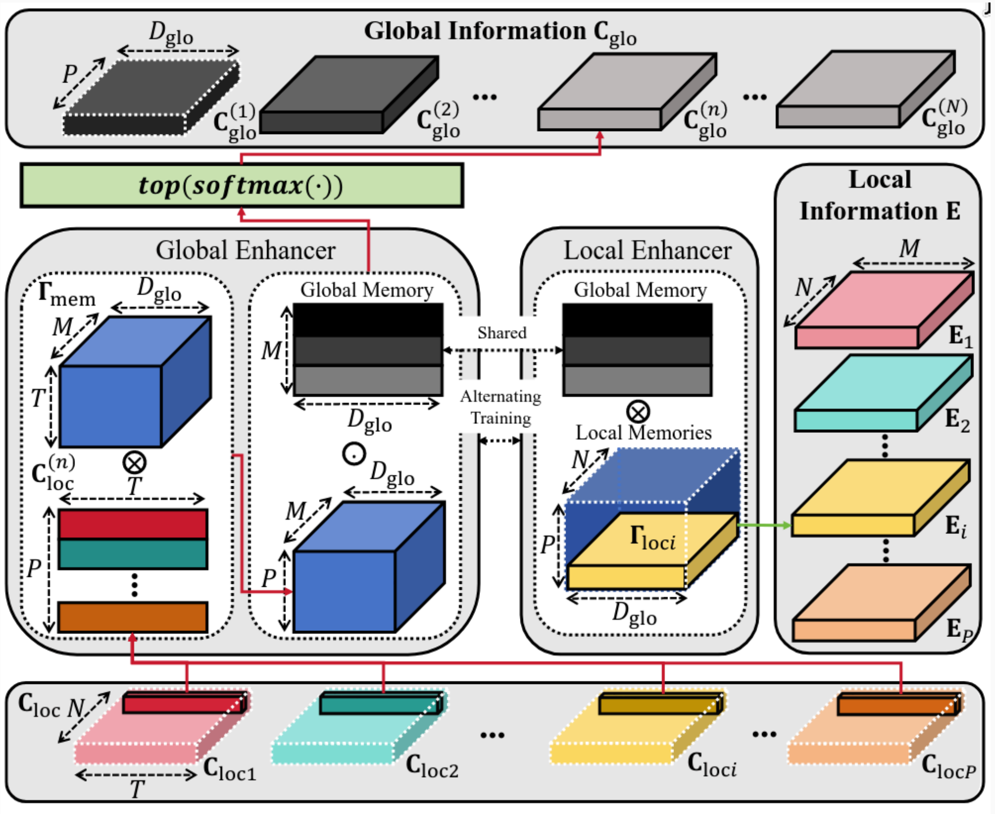
Global Enhancer

- Learns from locally correlated features
- Γ_{mem} is a large trainable tensor
  - Produces inquiry tensor
  - Recognizes patterns in data
    - The M high level patterns



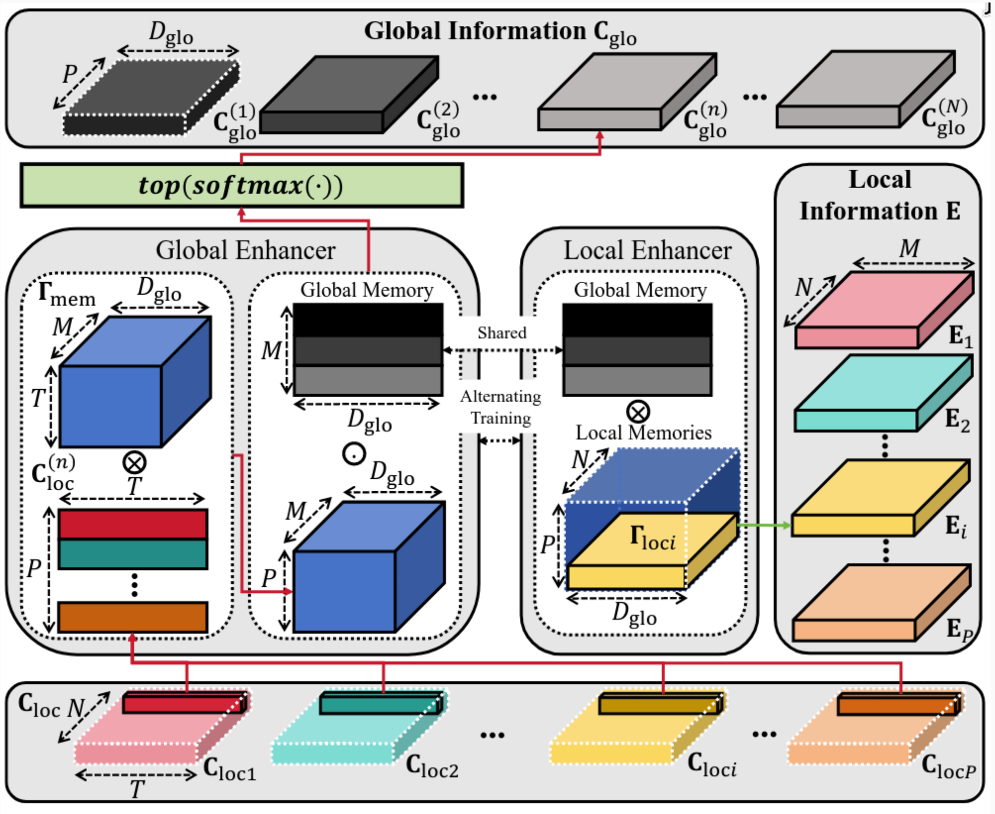
Global Enhancer

- Learns from locally correlated features
- Γ_{mem} is a large trainable tensor
  - Produces inquiry tensor
  - Recognizes patterns in data
    - The M high level patterns
- Inquiry tensor
  - Prevalence of patterns in local data
  - Similarity scores with global memory



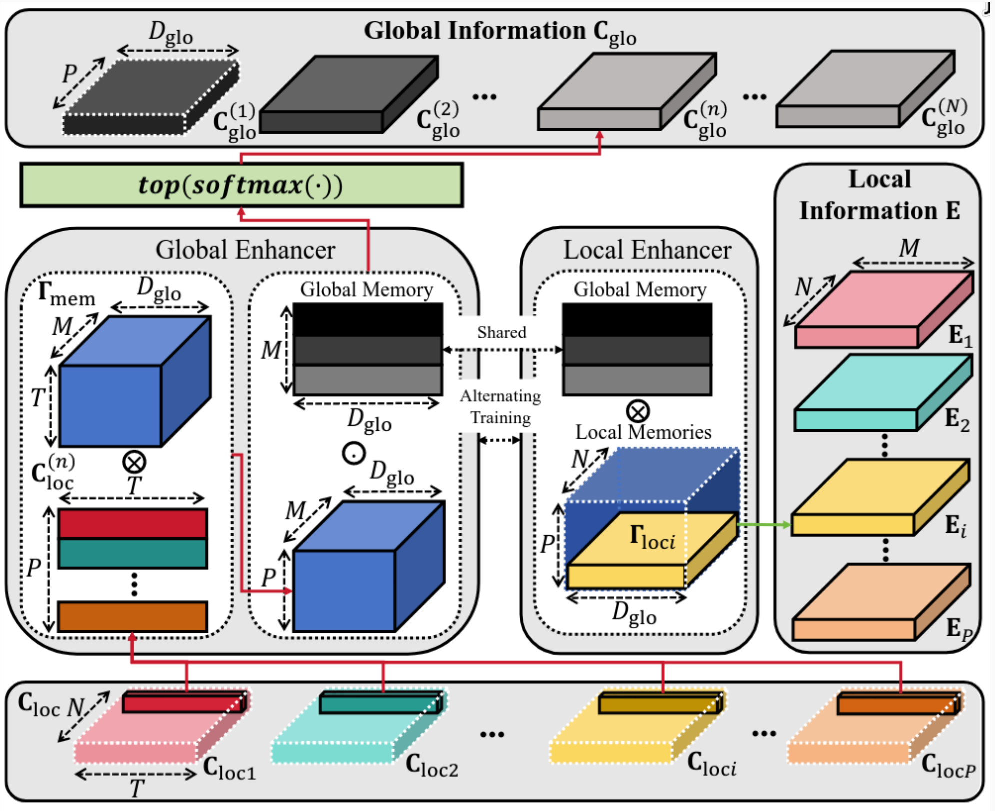
Global Enhancer

- Learns from locally correlated features
- Γ_{mem} is a large trainable tensor
  - Produces inquiry tensor
  - Recognizes patterns in data
    - The M high level patterns
- Inquiry tensor
  - Prevalence of patterns in local data
  - Similarity scores with global memory
- Probability distribution
  - Importance of pattern



Global Enhancer

- Learns from locally correlated features
- Γ_{mem} is a large trainable tensor
  - Produces inquiry tensor
  - Recognizes patterns in data
    - The M high level patterns
- Inquiry tensor
  - Prevalence of patterns in local data
  - Similarity scores with global memory
- Probability distribution
  - Importance of pattern
- Top k most important patterns
  - Stored in C_{glo}
  - Scaled based on importance
    - Weighted sum



Alternating Training

- Local information E requires
  - Local memories
  - Global memories
- Updating both simultaneously
  - Unstable training
  - Issues converging
- LE and GE alternate training
  - Split adjustment of memories

2.4 Alternating Memory Enhancer

Alternating Training

- Local information E requires
  - Local memories
  - Global memories
- Updating both simultaneously
  - Unstable training
  - Issues converging
- LE and GE alternate training
  - Split adjustment of memories

LE Training

- Local memories > global memories
  - More parameters $\longrightarrow$ longer convergence
- Balance convergence
  - Different learning rates
  - LE training more

Alternating Training

- Local information E requires
  - Local memories
  - Global memories
- Updating both simultaneously
  - Unstable training
  - Issues converging
- LE and GE alternate training
  - Split adjustment of memories

LE Training

- Local memories > global memories
  - More parameters ⟶ longer convergence
- Balance convergence
  - Different learning rates
  - LE training more

Algorithm 1 AME alternating training

Input: Historical horizon and ground truth $\mathbf{H}$, $\mathbf{F}$; local and global memories $\boldsymbol{\Gamma}_{\text{loc}}$, $\boldsymbol{\Gamma}_{\text{glo}}$; local training step ϵ; learning rates η_{loc}, η_{glo} for local and global enhancers

Output: Local and global information $\mathbf{E}$, $\mathbf{C}_{\text{glo}}$; learned local and global memories $\boldsymbol{\Gamma}_{\text{loc}}$, $\boldsymbol{\Gamma}_{\text{glo}}$, tensor $\boldsymbol{\Gamma}_{\text{mem}}$, and bias $\mathbf{b}_{\text{mem}}$

Initialisation: Initializing local and global memories $\boldsymbol{\Gamma}_{\text{loc}}$, $\boldsymbol{\Gamma}_{\text{glo}}$, tensor $\boldsymbol{\Gamma}_{\text{mem}}$, and bias $\mathbf{b}_{\text{mem}}$ randomly
while $\boldsymbol{\Gamma}_{\text{loc}}, \boldsymbol{\Gamma}_{\text{glo}}, \boldsymbol{\Gamma}_{\text{mem}}$, and $\mathbf{b}_{\text{mem}}$ are not converged **do**
  for iteration $= 0$ to ϵ **do**
    $\mathbf{H}' \leftarrow \text{Preprocessing}(\mathbf{H})$
    $\mathbf{E} \leftarrow \mathcal{A}_{\text{loc}}(\boldsymbol{\Gamma}_{\text{loc}}, \boldsymbol{\Gamma}_{\text{glo}})$
    $\mathbf{C}_{\text{loc}} \leftarrow \mathcal{G}_{\Theta}(\mathbf{H}', \mathbf{E})$
    $\mathbf{C}_{\text{glo}} \leftarrow \mathcal{A}_{\text{glo}}(\mathbf{C}_{\text{loc}}, \boldsymbol{\Gamma}_{\text{glo}})$
    $\mathbf{F}' \leftarrow \mathcal{T}_{\Phi}(\mathbf{C}_{\text{loc}}, \mathbf{C}_{\text{glo}})$
    $\hat{\mathbf{F}} \leftarrow \text{LinearHead}(\mathbf{F}')$
    $\boldsymbol{\Gamma}_{\text{loc}} \leftarrow \boldsymbol{\Gamma}_{\text{loc}} - \eta_{loc} \nabla_{\boldsymbol{\Gamma}_{\text{loc}}} \mathcal{L}(\hat{\mathbf{F}}, \mathbf{F})$
  end for
  $\mathbf{H}' \leftarrow \text{Preprocessing}(\mathbf{H})$
  $\mathbf{E} \leftarrow \mathcal{A}_{\text{loc}}(\boldsymbol{\Gamma}_{\text{loc}}, \boldsymbol{\Gamma}_{\text{glo}})$
  $\mathbf{C}_{\text{loc}} \leftarrow \mathcal{G}_{\Theta}(\mathbf{H}', \mathbf{E})$
  $\mathbf{C}_{\text{glo}} \leftarrow \mathcal{A}_{\text{glo}}(\mathbf{C}_{\text{loc}}, \boldsymbol{\Gamma}_{\text{glo}})$
  $\mathbf{F}' \leftarrow \mathcal{T}_{\Phi}(\mathbf{C}_{\text{loc}}, \mathbf{C}_{\text{glo}})$
  $\hat{\mathbf{F}} \leftarrow \text{LinearHead}(\mathbf{F}')$
  $\boldsymbol{\Gamma}_{\text{glo}} \leftarrow \boldsymbol{\Gamma}_{\text{glo}} - \eta_{glo} \nabla_{\boldsymbol{\Gamma}_{\text{glo}}} \mathcal{L}(\hat{\mathbf{F}}, \mathbf{F})$
  $\boldsymbol{\Gamma}_{\text{mem}} \leftarrow \boldsymbol{\Gamma}_{\text{mem}} - \eta_{glo} \nabla_{\boldsymbol{\Gamma}_{\text{mem}}} \mathcal{L}(\hat{\mathbf{F}}, \mathbf{F})$,
  $\mathbf{b}_{\text{mem}} \leftarrow \mathbf{b}_{\text{mem}} - \eta_{glo} \nabla_{\mathbf{b}_{\text{mem}}} \mathcal{L}(\hat{\mathbf{F}}, \mathbf{F})$
end while

3. Experiments

3.1 Noteworthy Details

Datasets

- 7 in total
  - 4 are variants of the same
- 7, 21, 321, and 862 variables
- $H = 336$
- $F = [96, 192, 336, 720]$

Datasets

- 7 in total
  - 4 are variants of the same
- 7, 21, 321, and 862 variables
- $H = 336$
- $F = [96, 192, 336, 720]$

Comparisons

- Multiple different model architectures
  - Channel independent models
  - Linear models
  - Attention models

Results

- Compare on MSE and MAE
- Bold is best, underline is second best
- Almost always best performance
  - Loses on MSE for low F in one dataset

Models		Memformer		ModernTCN		PatchTST		NLinear		DLinear		iTransformer		CARD		Crossformer		MTGNN	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	0.151	**0.185**	0.155	0.201	0.152	0.199	0.182	0.232	0.176	0.237	0.174	0.214	<u>0.150</u>	<u>0.188</u>	**0.145**	0.211	0.342	0.385
	192	<u>0.197</u>	**0.231**	0.198	0.245	<u>0.197</u>	0.243	0.225	0.269	0.220	0.282	0.221	0.254	0.202	<u>0.238</u>	**0.190**	0.259	0.427	0.445
	336	**0.247**	**0.274**	0.251	0.286	<u>0.249</u>	0.283	0.271	0.301	0.265	0.319	0.278	0.296	0.260	<u>0.282</u>	0.259	0.326	0.506	0.523
	720	**0.318**	**0.326**	0.321	0.336	<u>0.320</u>	<u>0.335</u>	0.338	0.348	0.323	0.362	0.358	0.347	0.343	0.353	0.332	0.382	0.510	0.527
Traffic	96	**0.361**	**0.230**	0.368	0.253	<u>0.367</u>	<u>0.251</u>	0.410	0.279	0.410	0.282	0.395	0.268	0.419	0.269	0.511	0.292	0.516	0.308
	192	**0.381**	**0.239**	<u>0.384</u>	0.261	0.385	<u>0.259</u>	0.423	0.284	0.423	0.287	0.417	0.276	0.443	0.276	0.523	0.311	0.534	0.324
	336	**0.394**	**0.245**	<u>0.397</u>	0.270	0.398	<u>0.265</u>	0.435	0.290	0.436	0.296	0.433	0.283	0.460	0.283	0.530	0.300	0.540	0.335
	720	**0.432**	**0.267**	0.440	0.296	<u>0.434</u>	<u>0.287</u>	0.464	0.307	0.466	0.315	0.467	0.302	0.490	0.299	0.573	0.313	0.557	0.343
Electricity	96	**0.130**	**0.217**	<u>0.131</u>	0.228	**0.130**	<u>0.222</u>	0.141	0.237	0.140	0.237	0.132	0.228	0.141	0.233	0.186	0.281	0.202	0.314
	192	**0.147**	**0.232**	0.150	0.242	<u>0.148</u>	<u>0.240</u>	0.154	0.248	0.153	0.249	0.154	0.249	0.160	0.250	0.208	0.300	0.266	0.349
	336	**0.162**	**0.249**	0.171	0.265	<u>0.167</u>	<u>0.261</u>	0.171	0.265	0.169	0.267	0.172	0.267	0.173	0.263	0.323	0.369	0.328	0.373
	720	**0.199**	**0.281**	0.203	0.294	0.202	0.291	0.210	0.297	0.203	0.301	0.204	0.296	<u>0.197</u>	<u>0.284</u>	0.404	0.423	0.422	0.410
ETTh1	96	**0.362**	**0.385**	0.382	0.401	0.375	0.399	<u>0.374</u>	0.394	0.375	0.399	0.386	0.405	0.383	<u>0.391</u>	0.377	0.419	0.401	0.442
	192	**0.386**	**0.404**	0.420	0.424	0.414	0.421	0.408	<u>0.415</u>	<u>0.405</u>	0.416	0.441	0.436	0.435	0.420	0.410	0.439	0.587	0.601
	336	**0.402**	**0.421**	<u>0.427</u>	0.434	0.431	0.436	0.429	<u>0.427</u>	0.439	0.443	0.487	0.458	0.479	0.442	0.440	0.461	0.736	0.643
	720	**0.436**	**0.452**	0.450	0.461	0.449	0.466	<u>0.440</u>	<u>0.453</u>	0.472	0.490	0.503	0.491	0.471	0.461	0.519	0.524	0.916	0.750
ETTh2	96	**0.264**	**0.321**	0.276	0.342	<u>0.274</u>	0.336	0.277	0.338	0.289	0.353	0.297	0.349	0.281	<u>0.330</u>	0.770	0.529	0.735	0.643
	192	**0.314**	**0.358**	0.340	0.381	<u>0.339</u>	<u>0.379</u>	0.344	0.381	0.383	0.418	0.380	0.400	0.363	0.381	0.848	0.657	0.859	0.717
	336	**0.312**	**0.364**	<u>0.329</u>	<u>0.378</u>	0.331	0.380	0.357	0.400	0.448	0.465	0.428	0.432	0.411	0.418	0.859	0.674	1.050	0.849
	720	**0.374**	**0.410**	0.392	0.433	<u>0.379</u>	<u>0.422</u>	0.394	0.436	0.605	0.551	0.427	0.445	0.416	0.431	1.221	0.825	1.336	0.963
ETTm1	96	**0.285**	**0.336**	0.292	0.346	<u>0.290</u>	<u>0.342</u>	0.306	0.348	0.299	0.343	0.334	0.368	0.316	0.347	0.320	0.373	0.428	0.446
	192	**0.323**	**0.358**	<u>0.332</u>	0.368	<u>0.332</u>	0.369	0.349	0.375	0.335	<u>0.365</u>	0.377	0.391	0.363	0.370	0.372	0.411	0.551	0.505
	336	**0.365**	**0.381**	0.367	0.393	<u>0.366</u>	0.392	0.375	0.388	0.369	<u>0.386</u>	0.426	0.420	0.392	0.390	0.429	0.441	0.706	0.622
	720	**0.419**	**0.409**	0.422	0.429	<u>0.420</u>	0.424	0.433	0.422	0.425	<u>0.421</u>	0.491	0.459	0.458	0.425	0.573	0.531	0.982	0.764
ETTm2	96	**0.160**	**0.245**	0.166	0.256	<u>0.165</u>	0.255	0.167	0.255	0.167	0.260	0.180	0.264	0.169	<u>0.248</u>	0.254	0.348	0.442	0.483
	192	**0.215**	**0.285**	0.222	0.293	<u>0.220</u>	<u>0.292</u>	0.221	0.293	0.224	0.303	0.250	0.309	0.234	<u>0.292</u>	0.370	0.433	0.642	0.570
	336	**0.263**	**0.317**	0.276	<u>0.327</u>	0.278	0.329	<u>0.274</u>	<u>0.327</u>	0.281	0.342	0.311	0.348	0.294	0.339	0.511	0.527	0.726	0.658
	720	**0.350**	**0.372**	<u>0.365</u>	<u>0.383</u>	0.367	0.385	0.368	0.384	0.397	0.421	0.412	0.407	0.390	0.388	0.901	0.689	1.139	0.862

Overview

- Are components contributing?
- Experiment without
  - Graph learning
  - GRU
  - Local
  - Global
  - Sharing
  - Alternating

Overview

- Are components contributing?
- Experiment without
  - Graph learning
  - GRU
  - Local
  - Global
  - Sharing
  - Alternating

Models		Memformer		w/o Graph		w/o Recurrent		w/o Local		w/o Global		w/o Sharing		w/o Alternating	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.151**	**0.185**	0.152	0.197	0.155	0.194	0.159	0.204	0.153	0.195	0.151	0.187	0.155	0.199
	192	**0.197**	**0.231**	0.200	0.235	0.204	0.248	0.199	0.255	0.202	0.238	0.199	0.234	0.202	0.235
	336	**0.247**	**0.274**	0.252	0.279	0.254	0.298	0.257	0.307	0.251	0.284	0.252	0.279	0.255	0.286
	720	**0.318**	**0.326**	0.334	0.341	0.332	0.355	0.364	0.380	0.323	0.333	0.324	0.335	0.334	0.360
Electricity	96	**0.130**	**0.217**	0.133	0.226	0.132	0.224	0.132	0.243	0.131	0.223	0.131	0.223	0.138	0.235
	192	**0.147**	**0.232**	0.154	0.245	0.155	0.248	0.153	0.250	0.150	0.238	0.152	0.241	0.158	0.252
	336	**0.162**	**0.249**	0.169	0.260	0.174	0.268	0.179	0.270	0.169	0.258	0.170	0.261	0.181	0.274
	720	**0.199**	**0.281**	0.208	0.299	0.220	0.317	0.231	0.341	0.205	0.293	0.210	0.300	0.229	0.339
ETTh2	96	**0.264**	**0.321**	0.271	0.329	0.269	0.326	0.322	0.369	0.266	0.324	0.266	0.324	0.294	0.347
	192	**0.314**	**0.358**	0.328	0.365	0.325	0.362	0.458	0.478	0.320	0.364	0.318	0.361	0.372	0.401
	336	**0.312**	**0.364**	0.329	0.376	0.334	0.381	0.530	0.517	0.317	0.370	0.319	0.370	0.380	0.419
	720	**0.374**	**0.410**	0.379	0.421	0.401	0.437	0.705	0.627	0.385	0.422	0.388	0.425	0.437	0.463

Results

- All component are contributing

3.4 Challenge 1

Disrupted Correlations

- Robustness
- Introduce outliers
  - Different amounts
  - Independent
  - Dependent

Disrupted Correlations

- Robustness
- Introduce outliers
  - Different amounts
  - Independent
  - Dependent
- Results
  - Performed the best
  - Mitigate both types of outliers

Disrupted Correlations

- Robustness
- Introduce outliers
  - Different amounts
  - Independent
  - Dependent
- Results
  - Performed the best
  - Mitigate both types of outliers

Dynamic Correlations

- Introduce dynamic correlations
  - Different amounts

Disrupted Correlations

- Robustness
- Introduce outliers
  - Different amounts
  - Independent
  - Dependent
- Results
  - Performed the best
  - Mitigate both types of outliers

Dynamic Correlations

- Introduce dynamic correlations
  - Different amounts
- Results
  - Performed the best

4. Critique

Instance normalization

- Normalize within historical horizon only
- Mitigates the issue of internal covariate shift
- Allows model to effectively grasp the intricate temporal dynamics inherent in time series

$H' = (H - \mu)/\sqrt{(\sigma^2 + c)}$, where

H is the historical horizon

μ is the mean

σ is the variance

c ensures numerical stability

Instance normalization

- Normalize within historical horizon only
- Mitigates the issue of internal covariate shift
- Allows model to effectively grasp the intricate temporal dynamics inherent in time series

$$H' = (H - \mu)/\sqrt{(\sigma^2 + c)}, \text{where}$$

H is the historical horizon

μ is the mean

σ is the variance

c ensures numerical stability

poral dynamics inherent in time series. Instance normalization is defined as $\mathbf{H'} = (\mathbf{H} - \mu)/\sqrt{(\sigma^2 + \text{constant})}$, where $\mathbf{H'}$ denotes the preprocessed feature, μ and σ denote the mean and variance of the sample, respectively, and "constant" is a small positive real number included to ensure numerical stability.

4.1 Preprocessing

Instance normalization

- Normalize within historical horizon only
- Mitigates the issue of internal covariate shift
- Allows model to effectively grasp the intricate temporal dynamics inherent in time series

$H' = (H - \mu)/\sqrt{(\sigma^2 + c)}$, where

H is the historical horizon

μ is the mean

σ is the variance

c ensures numerical stability

poral dynamics inherent in time series. Instance normalization is defined as $\mathbf{H}' = (\mathbf{H} - \mu)/\sqrt{(\sigma^2 + \text{constant})}$, where $\mathbf{H}'$ denotes the preprocessed feature, μ and σ denote the mean and variance of the sample, respectively, and "constant" is a small positive real number included to ensure numerical stability.

- Mistake in variance?
  - σ is conventional notation for standard deviation
  - σ^2 is conventional notation for variance

What is going on?

What is going on?

- Explored code to find answer
- `data_provider/data_loader.py`
  - Only place anything related to loading data happens
  - `Dataset_ETT_hour`, `Dataset_ETT_minute`, `Dataset_Custom`, `Dataset_Pred`

What is going on?

- Explored code to find answer
- `data_provider/data_loader.py`
  - Only place anything related to loading data happens
  - `Dataset_ETT_hour`, `Dataset_ETT_minute`, `Dataset_Custom`, `Dataset_Pred`

```
from sklearn.preprocessing import StandardScaler
class ...:
    def __read_data__(self):
        self.scalar = StandardScaler()
        self.scaler.fit(train_data.values)
        data = self.scaler.transform(df_data.values)
```

What is going on?

- Explored code to find answer
- `data_provider/data_loader.py`
  - Only place anything related to loading data happens
  - `Dataset_ETT_hour`, `Dataset_ETT_minute`, `Dataset_Custom`, `Dataset_Pred`

```python
from sklearn.preprocessing import StandardScaler
class ...:
    def __read_data__(self):
        self.scalar = StandardScaler()
        self.scaler.fit(train_data.values)
        data = self.scaler.transform(df_data.values)
```

- They fit on training data
- Normalize entire dataset with μ and σ from training data

What are they actually doing?

Preprocessing

$H' = (H - \mu)/\sqrt{(\sigma^2 + c)}$, where

H is the historical horizon

μ is the mean

σ is the variance

c ensures numerical stability

StandardScaler

$z = (x - \mu)/\sigma$, where

x is the sample

μ is the mean

σ is the standard deviation

What are they actually doing?

Preprocessing

$H' = (H - \mu)/\sqrt{(\sigma^2 + c)}$, where

H is the historical horizon

μ is the mean

σ is the variance

c ensures numerical stability

StandardScaler

$z = (x - \mu)/\sigma$, where

x is the sample

μ is the mean

σ is the standard deviation

- We know that $\sqrt{\sigma^2} = \sigma$

What are they actually doing?

Preprocessing

$H' = (H - \mu)/\sqrt{(\sigma^2 + c)}$, where

H is the historical horizon

μ is the mean

σ is the variance

c ensures numerical stability

StandardScaler

$z = (x - \mu)/\sigma$, where

x is the sample

μ is the mean

σ is the standard deviation

- We know that $\sqrt{\sigma^2} = \sigma$
- Essentially same formula, except constant

What are they actually doing?

Preprocessing

$$H' = (H - \mu)/\sqrt{(\sigma^2 + c)}, \text{where}$$

H is the historical horizon

μ is the mean

σ is the variance

c ensures numerical stability

StandardScaler

$$z = (x - \mu)/\sigma, \text{where}$$

x is the sample

μ is the mean

σ is the standard deviation

- We know that $\sqrt{\sigma^2} = \sigma$
- Essentially same formula, except constant
- Fit on training data, normalize entire dataset $\longrightarrow$ global normalization

What are they actually doing?

Preprocessing

$H' = (H - \mu)/\sqrt{(\sigma^2 + c)}$, where

H is the historical horizon

μ is the mean

σ is the variance

c ensures numerical stability

StandardScaler

$z = (x - \mu)/\sigma$, where

x is the sample

μ is the mean

σ is the standard deviation

- We know that $\sqrt{\sigma^2} = \sigma$
- Essentially same formula, except constant
- Fit on training data, normalize entire dataset $\longrightarrow$ global normalization
- None of the stated benefits of instance normalization
  - Mitigate internal covariate shift
  - Grasp intricate temporal dynamics in TS

Inconsistencies

- C_{glo} is global memory
- C_{loc} is locally correlated features
- E is local memory

4.2 Notation

Inconsistencies

- C_{glo} is global memory
- C_{loc} is locally correlated features
- E is local memory

Symbol Reuse

- $\mathbf{F}$ is the ground truth
- F is the dimensionality of $\mathbf{F}$
- $\mathbf{F'}$ is the encoding output
- F' is the dimensionality of $\mathbf{F'}$
- Confusing statements and diagrams

4.2 Notation

Inconsistencies

- C_{glo} is global memory
- C_{loc} is locally correlated features
- E is local memory

Symbol Reuse

- $\mathbf{F}$ is the ground truth
- F is the dimensionality of $\mathbf{F}$
- $\mathbf{F'}$ is the encoding output
- F' is the dimensionality of $\mathbf{F'}$
- Confusing statements and diagrams

$\mathbf{F'} \in \mathbb{R}^{F' \times N}$, where F' is the temporal dimension of the representation.

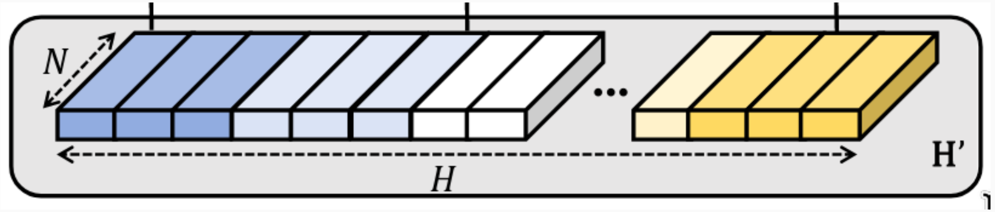
Inconsistencies

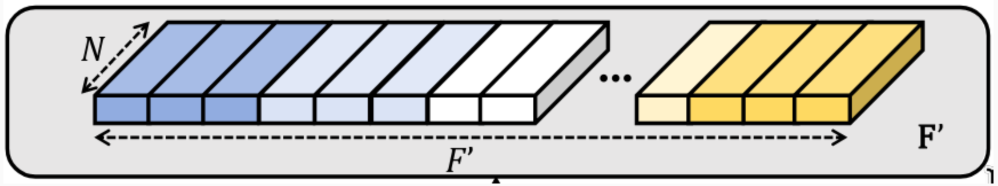
- C_{glo} is global memory
- C_{loc} is locally correlated features
- E is local memory

Symbol Reuse

- $\mathbf{F}$ is the ground truth
- F is the dimensionality of $\mathbf{F}$
- $\mathbf{F'}$ is the encoding output
- F' is the dimensionality of $\mathbf{F'}$
- Confusing statements and diagrams

$\mathbf{F'} \in \mathbb{R}^{F' \times N}$, where F' is the temporal dimension of the representation.

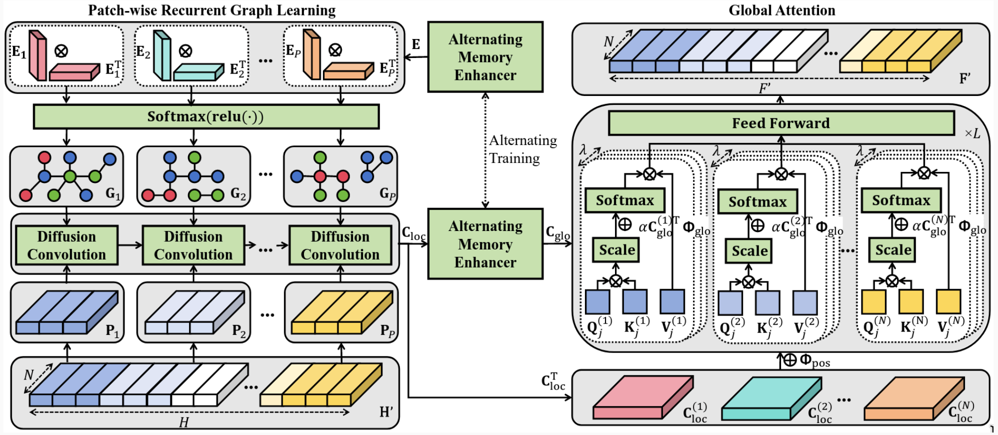


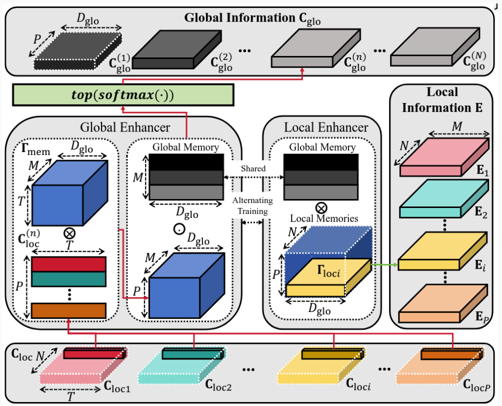


5. Praise

Colors

- Help understanding and data flow
  - Preprocessing ⟶ final encoding
  - Minor inconsistencies
    - Attention





Colors

- Help understanding and data flow
  - Preprocessing ⟶ final encoding
  - Minor inconsistencies
    - Attention

Dimensionality

- Squares ⟶ 2-dimensional
- Cubes ⟶ 3-dimensional
- Transposed ⟶ lying down
- Slices of shapes
  - M slices of global memory
  - P slices of local memory