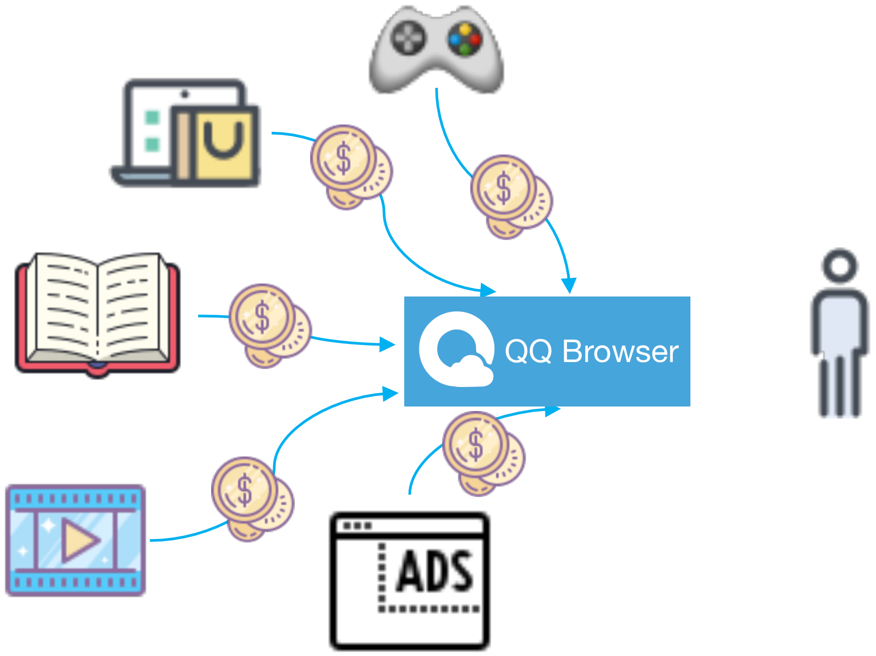
Learning Reliable User Representations from Volatile and Sparse Data to Accurately Predict Customer Lifetime Value

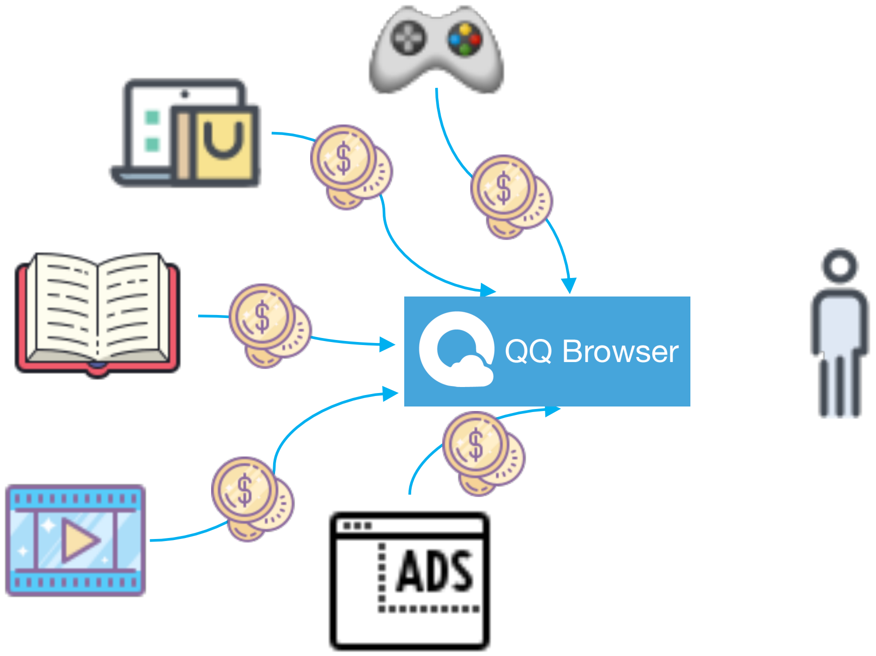
Authors: Mingzhe Xing, Shuqing Bian, Wayne Xin Zhao, Zhen Xiao,

Xinji Luo, Cunxiang Yin, Jing Cai, Yancheng He

LTV Prediction



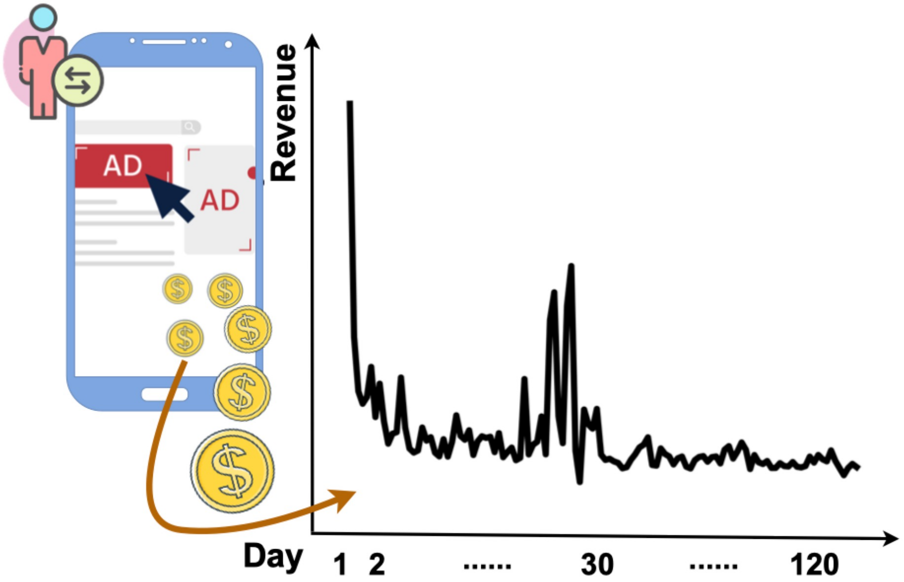
LTV Prediction



- Customer LifeTime Value (LTV):
  - measures the **value of a user** during the lifetime of using an application;
  - help **reduce user churn** and **increase retention for** user-centric companies.

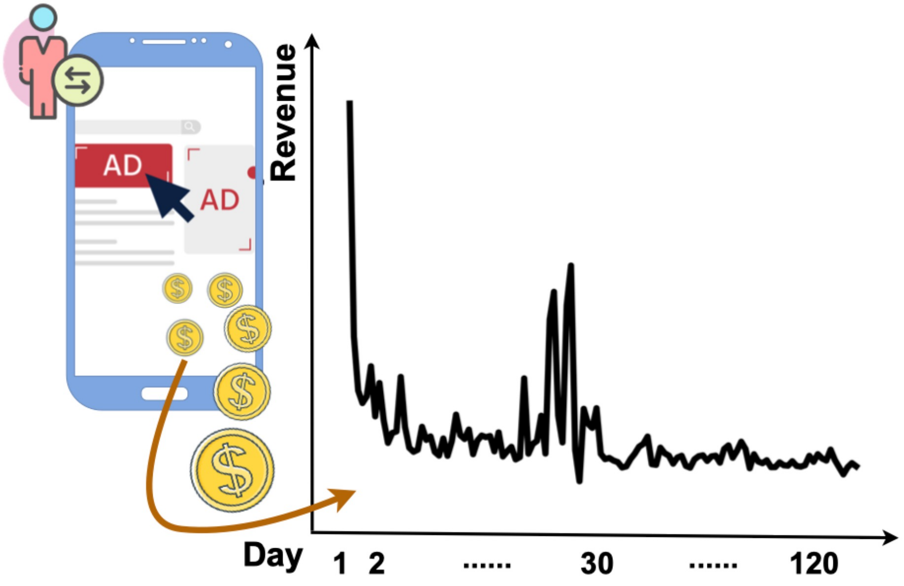
Challenges

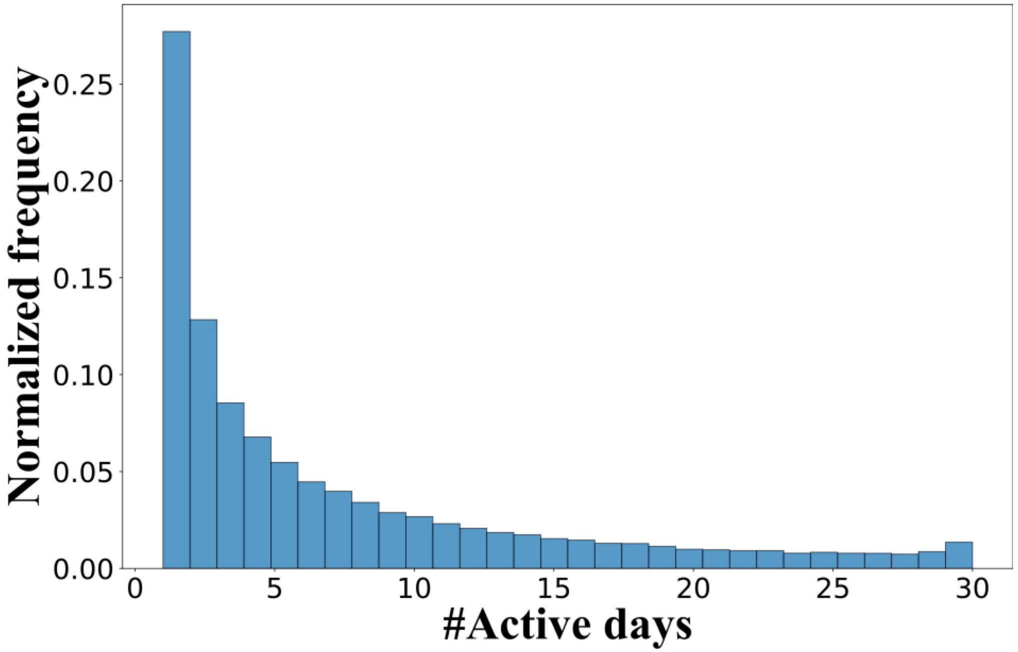
- Revenue sequences are usually **volatile** and **sparse**.



Challenges

- Revenue sequences are usually **volatile** and **sparse**.



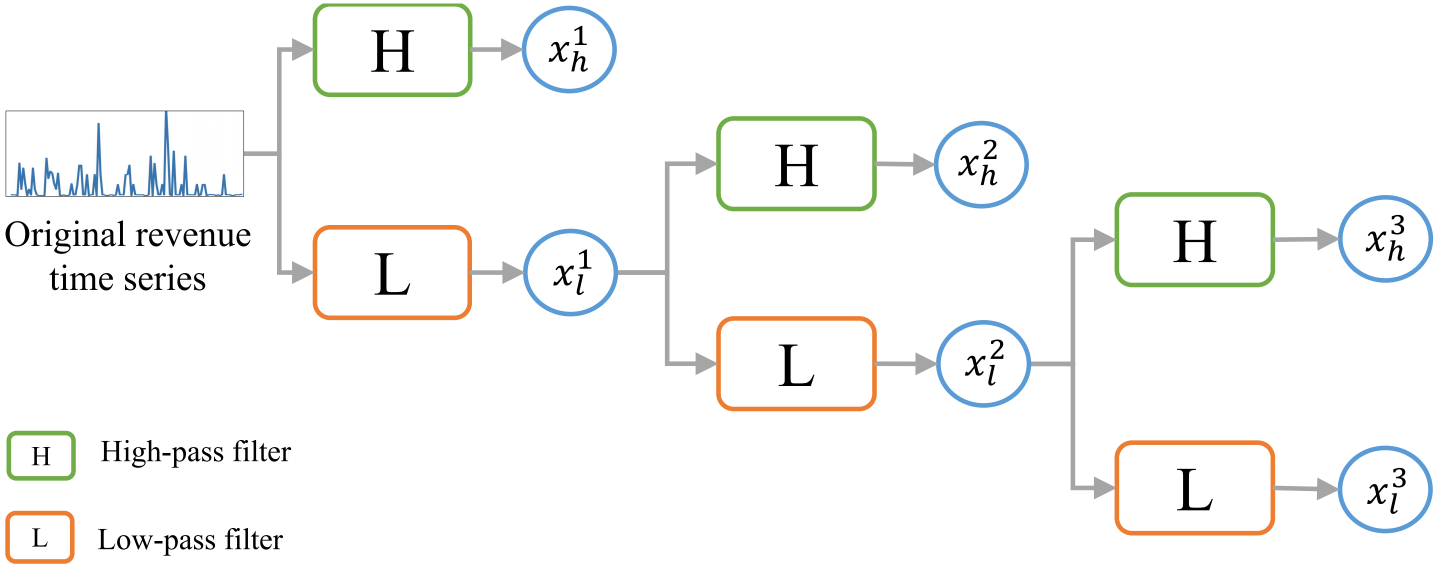


Our Solution

- Volatility Issue
  - Solution: incorporate the **wavelet transform technique** to reduce the influence of volatile data.

Our Solution

- Volatility Issue
  - Solution: incorporate the **wavelet transform technique** to reduce the influence of volatile data.



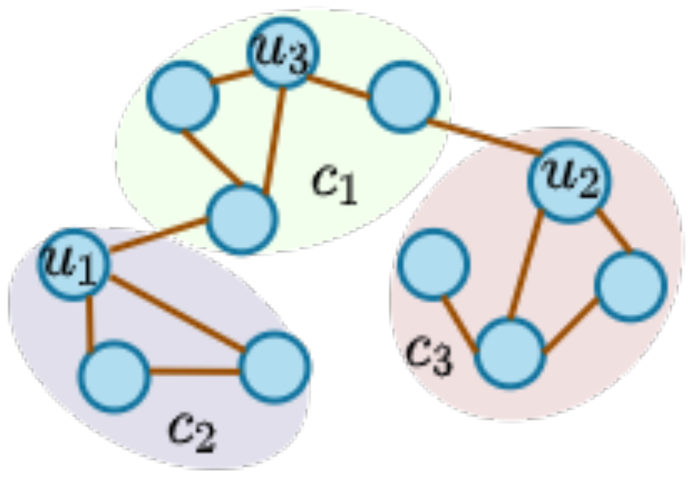
$$\mathrm{x}_h^n = \sum_{t=0}^{T-1} h[2T-1] x_h^{n-1}[t]$$

$$\mathrm{x}_l^n = \sum_{t=0}^{T-1} l[2T-1] x_l^{n-1}[t]$$

Discrete Wavelet Transform

Our Solution

- Volatility Issue
  - Solution: incorporate the **wavelet transform technique** to reduce the influence of volatile data.
- Sparsity Issue
  - Solution: learn **structural user representations** with an **attribute similarity graph** to enhance temporal user representations.



Our Solution

- Volatility Issue
  - Solution: incorporate the **wavelet transform technique** to reduce the influence of volatile data.
- Sparsity Issue
  - Solution: learn **structural user representations** with an **attribute similarity graph** to enhance temporal user representations.
- Regularization and Fusion
  - Cluster-alignment regularization to **reduce the divergence** in the two kinds of user representations.
  - Associate temporal and structural representations in the low-pass representation space, which is also useful to prevent the data noise from being transferred across different views.

Problem Definition

- For a user u from a user set U , we have two kinds of data input:
  - r_u is the revenue sequence of user u.
  - e_u denotes the feature vector for u consisting of user attributes, *e.g.*, age and activity degree.

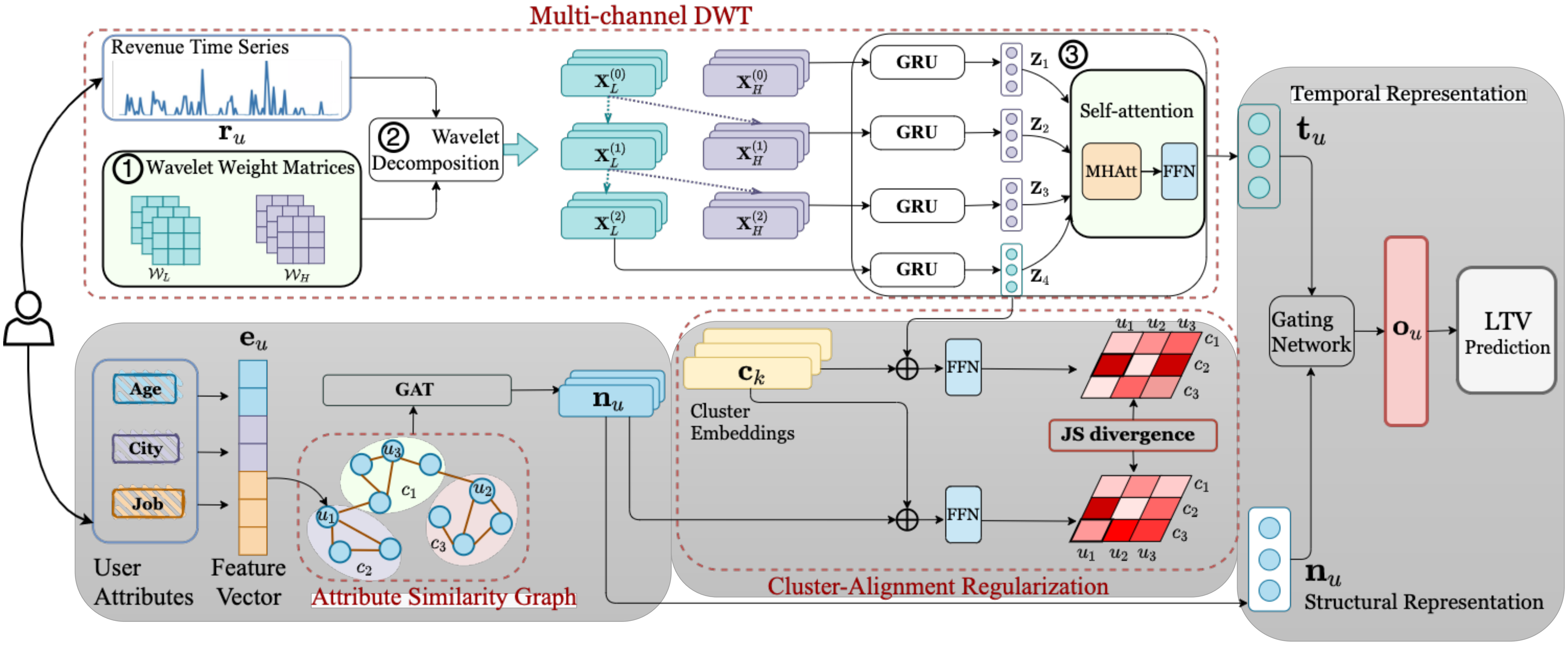
Problem Definition

- For a user u from a user set U , we have two kinds of data input:
  - r_u is the revenue sequence of user u.
  - e_u denotes the feature vector for u consisting of user attributes, *e.g.*, age and activity degree.
- Build attribute similarity graph
  - Similarity: $s_{u,v} = \exp(-\gamma||\mathrm{e}_u - e_v||^2)$
  - Connect top-K most similar neighbors

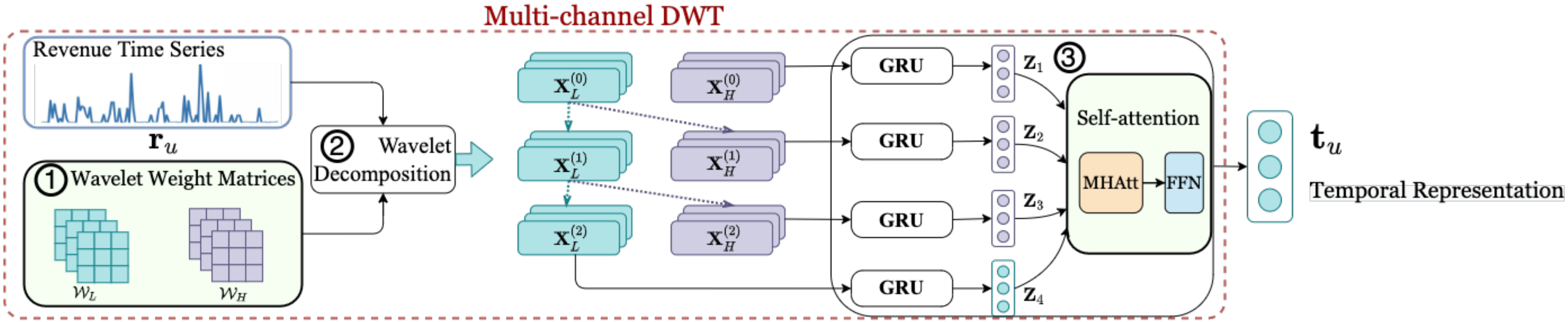
Problem Definition

- For a user u from a user set U , we have two kinds of data input:
  - r_u is the revenue sequence of user u.
  - e_u denotes the feature vector for u consisting of user attributes, *e.g.*, age and activity degree.
- Build attribute similarity graph
  - Similarity: $s_{u,v} = \exp(-\gamma||\mathrm{e}_u - e_v||^2)$
  - Connect top-K most similar neighbors
- Goal: predict accumulated LTV for the future Δm days

Methodology (Temporal View)



Temporal Trend Encoder

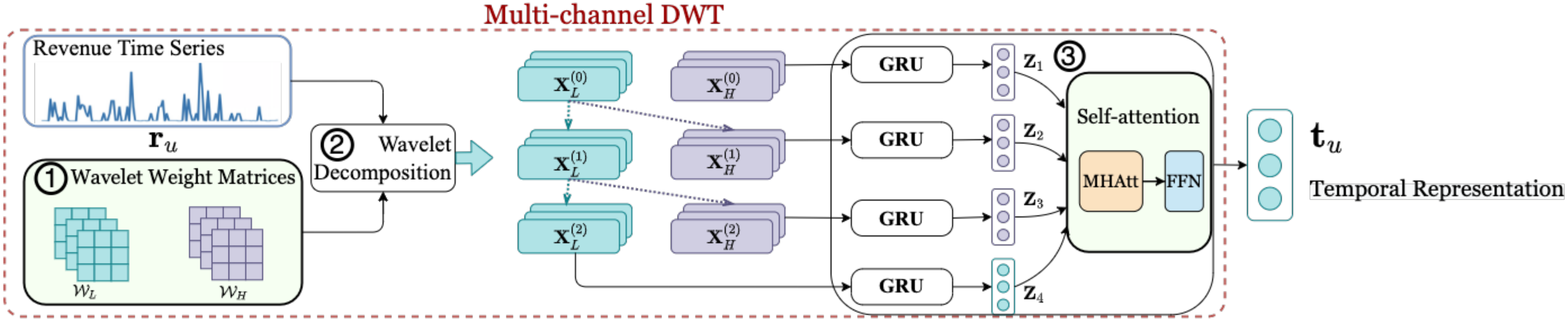


Step 1.
Multi-Channel Trainable Wavelet Filters

$$\mathbf{W}_L[i, i+j] = \mathbf{l}[j]$$
$$\mathbf{W}_H[i, i+j] = \mathbf{h}[j]$$

$$\mathcal{W}_L = [\mathbf{W}_{L,1}; \mathbf{W}_{L,2}, \cdots, \mathbf{W}_{L,C}]$$
$$\mathcal{W}_H = [\mathbf{W}_{H,1}; \mathbf{W}_{H,2}, \cdots, \mathbf{W}_{H,C}]$$

Temporal Trend Encoder



Step 1.
Multi-Channel Trainable Wavelet Filters

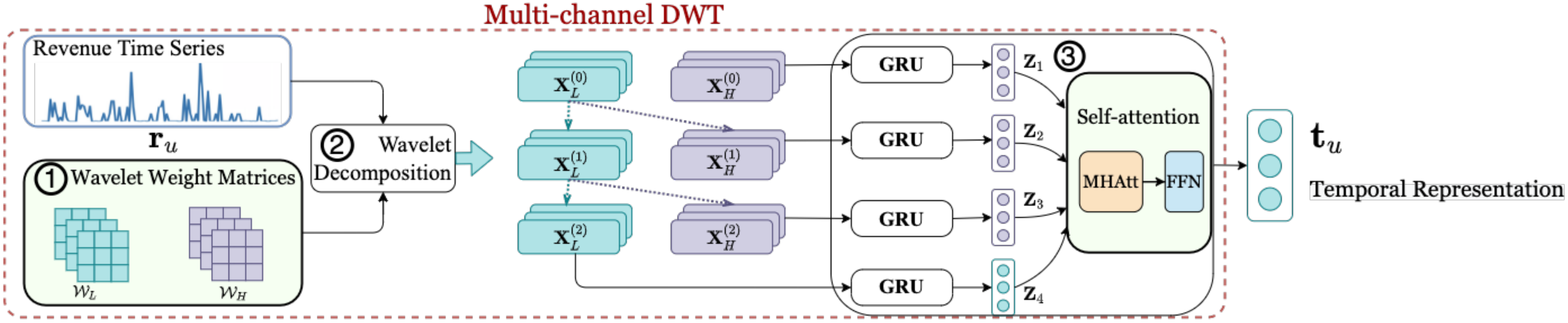
$$\mathbf{W}_L[i, i+j] = \mathbf{l}[j]$$
$$\mathbf{W}_H[i, i+j] = \mathbf{h}[j]$$

$$\mathcal{W}_L = [\mathbf{W}_{L,1}; \mathbf{W}_{L,2}, \cdots, \mathbf{W}_{L,C}]$$
$$\mathcal{W}_H = [\mathbf{W}_{H,1}; \mathbf{W}_{H,2}, \cdots, \mathbf{W}_{H,C}]$$

Step 2.
Multi-Channel Wavelet Decomposition

$$\mathbf{X}_L^{(d)} = \mathrm{AvgPool}\big(\sigma(\mathcal{W}_L \mathbf{X}_L^{(d-1)} + \mathbf{B}_L)\big)$$
$$\mathbf{X}_H^{(d)} = \mathrm{AvgPool}\big(\sigma(\mathcal{W}_H \mathbf{X}_L^{(d-1)} + \mathbf{B}_H)\big)$$

Temporal Trend Encoder



Step 1.
Multi-Channel Trainable Wavelet Filters

$$\mathbf{W}_L[i, i+j] = \mathbf{l}[j]$$
$$\mathbf{W}_H[i, i+j] = \mathbf{h}[j]$$

$$\mathcal{W}_L = [\mathbf{W}_{L,1}; \mathbf{W}_{L,2}, \cdots, \mathbf{W}_{L,C}]$$
$$\mathcal{W}_H = [\mathbf{W}_{H,1}; \mathbf{W}_{H,2}, \cdots, \mathbf{W}_{H,C}]$$

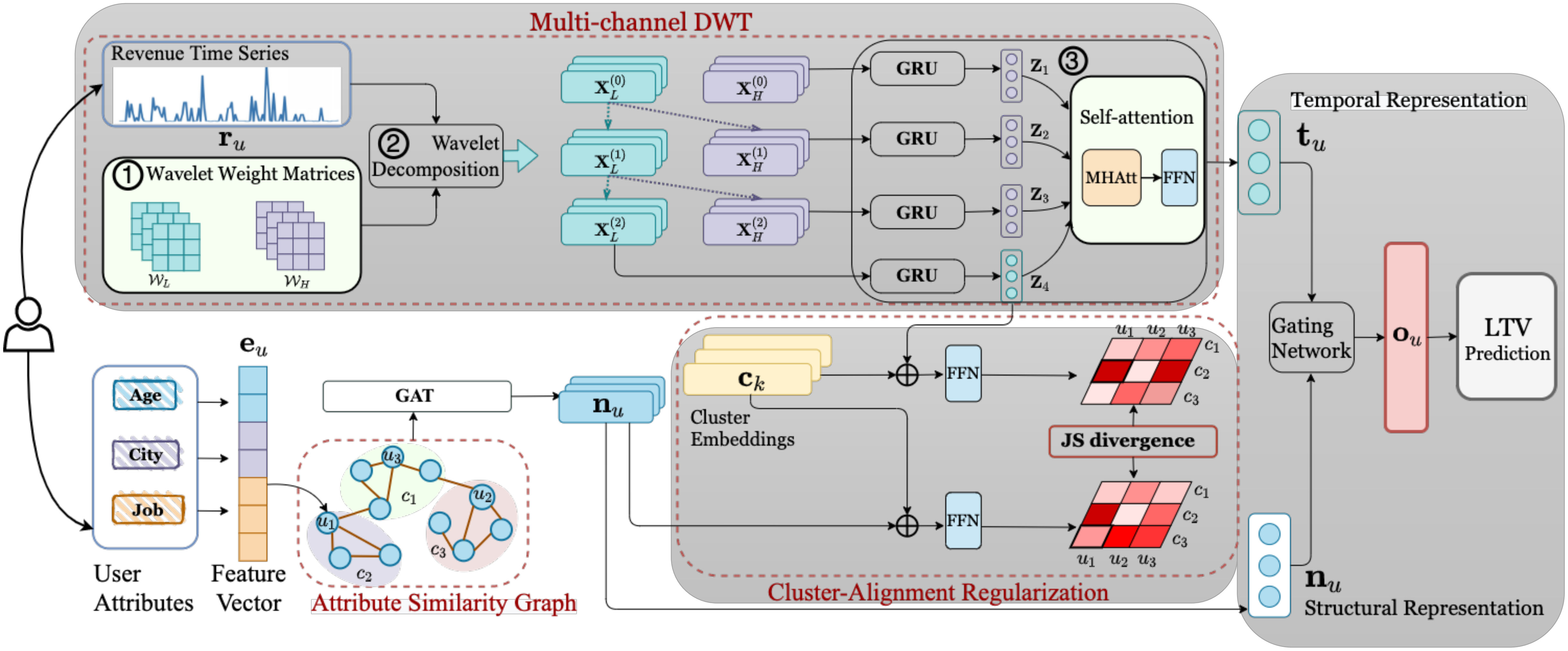
Step 2.
Multi-Channel Wavelet Decomposition

$$\mathbf{X}_L^{(d)} = \mathrm{AvgPool}\big(\sigma(\mathcal{W}_L \mathbf{X}_L^{(d-1)} + \mathbf{B}_L)\big)$$
$$\mathbf{X}_H^{(d)} = \mathrm{AvgPool}\big(\sigma(\mathcal{W}_H \mathbf{X}_L^{(d-1)} + \mathbf{B}_H)\big)$$

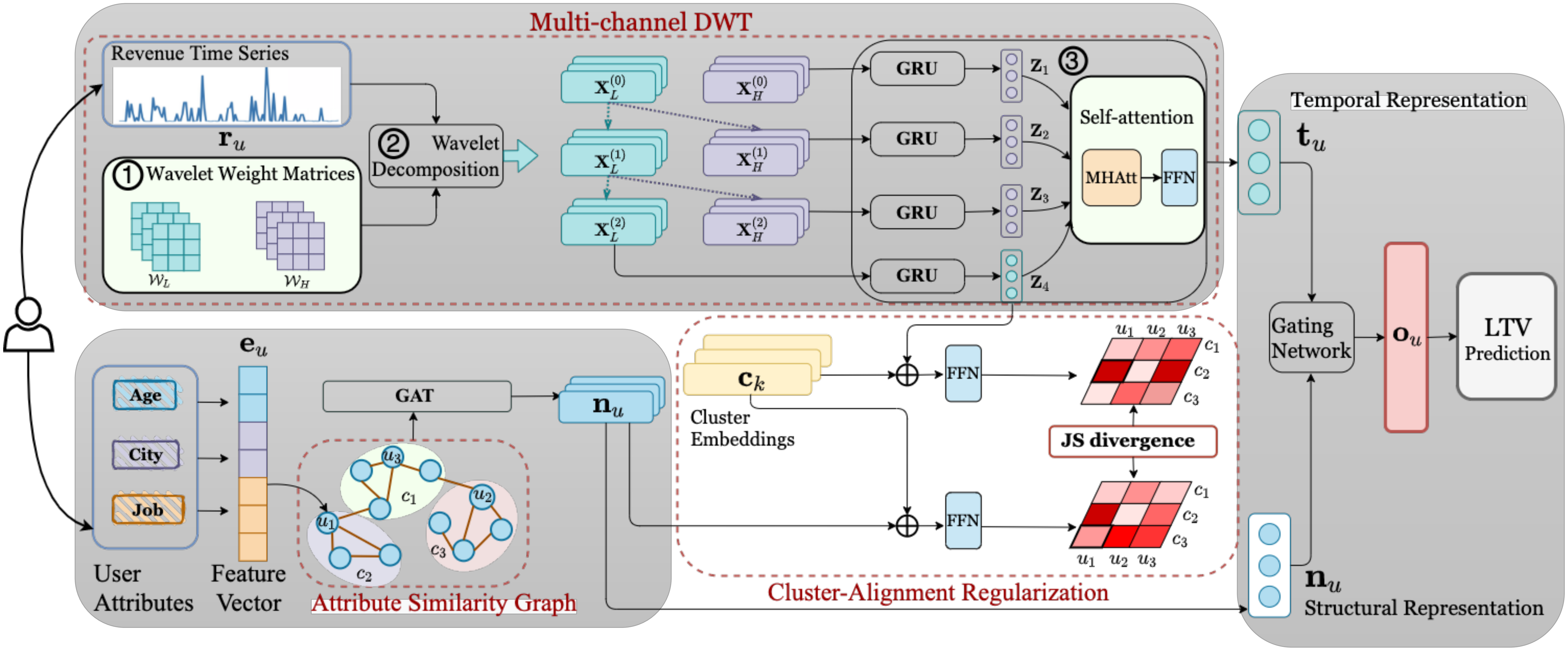
Step 3.
Self-attentive Channels and Frequency Components

$$\mathbf{z} = \mathrm{GRU}(\mathbf{z}^{(d)})$$
$$\mathbf{Z} = [\mathbf{z}_1; \mathbf{z}_2; \cdots; \mathbf{z}_D; \mathbf{z}_{D+1}]$$
$$\mathbf{F} = \mathrm{MHA}(\mathbf{Z})$$
$$\mathbf{t}_u = \mathrm{AvgPool}(\mathbf{F})$$

Methodology (Structural View)



Methodology (Regularization)

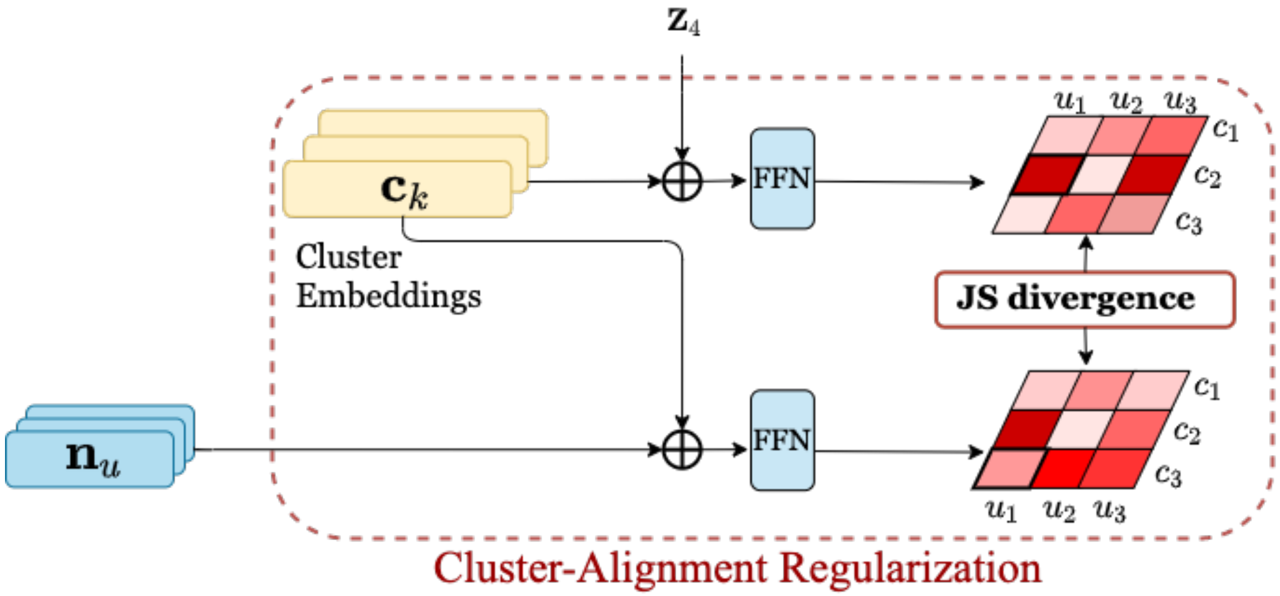


Cluster-Alignment Regularization

Step 1. Distance with cluster centroids

$$d_{u,k}^{(1)} = \tanh(\mathbf{W}_1[\mathbf{z}_{D+1}; \mathbf{c}_k] + b_1)$$

$$d_{u,k}^{(2)} = \tanh(\mathbf{W}_2[\mathbf{n}_u; \mathbf{c}_k] + b_2)$$



Cluster-Alignment Regularization

Step 1. Distance with cluster centroids

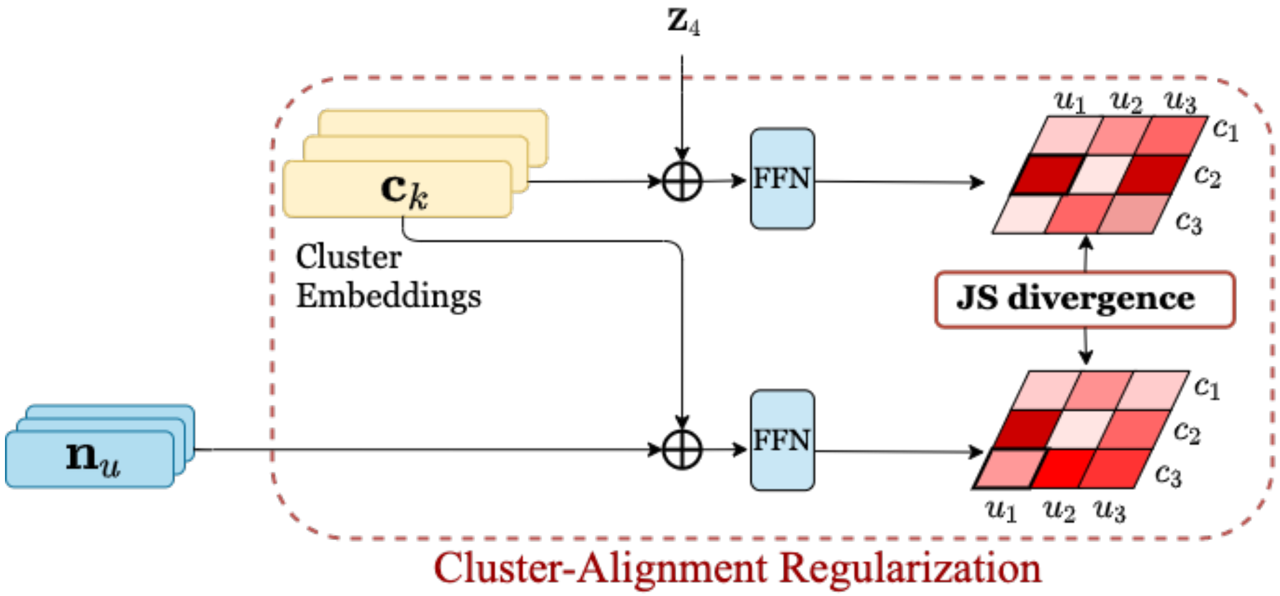
$$d_{u,k}^{(1)} = \tanh(\mathbf{W}_1[\mathbf{z}_{D+1}; \mathbf{c}_k] + b_1)$$

$$d_{u,k}^{(2)} = \tanh(\mathbf{W}_2[\mathbf{n}_u; \mathbf{c}_k] + b_2)$$

Step 2. Soft assignment probabilities

$$\theta_{u,k} = \frac{(1 + {d_{u,k}^{(1)}}^2/t)^{-\frac{t+1}{2}}}{\sum_{k'}(1 + {d_{u,k}^{(1)}}^2/t)^{-\frac{t+1}{2}}}$$

$$\phi_{u,k} = \frac{(1 + {d_{u,k}^{(2)}}^2/t)^{-\frac{t+1}{2}}}{\sum_{k'}(1 + {d_{u,k}^{(2)}}^2/t)^{-\frac{t+1}{2}}}$$



Cluster-Alignment Regularization

Step 1. Distance with cluster centroids

$$d_{u,k}^{(1)} = \tanh(\mathbf{W}_1[\mathbf{z}_{D+1}; \mathbf{c}_k] + b_1)$$

$$d_{u,k}^{(2)} = \tanh(\mathbf{W}_2[\mathbf{n}_u; \mathbf{c}_k] + b_2)$$

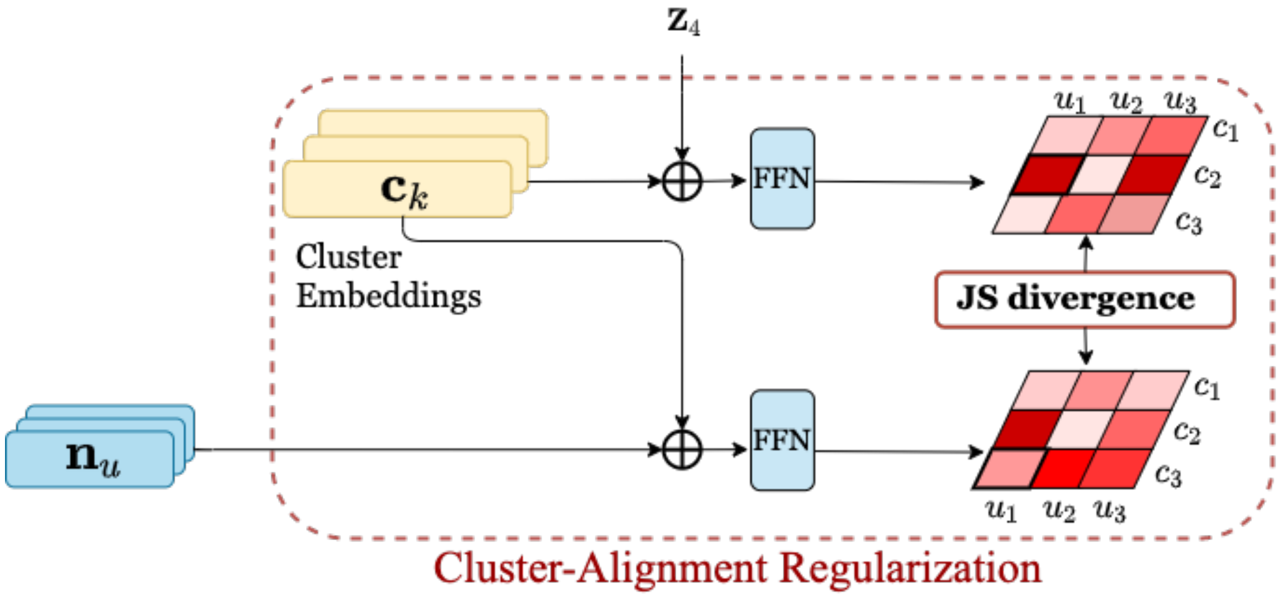
Step 2. Soft assignment probabilities

$$\theta_{u,k} = \frac{(1 + {d_{u,k}^{(1)}}^2/t)^{-\frac{t+1}{2}}}{\sum_{k'}(1 + {d_{u,k}^{(1)}}^2/t)^{-\frac{t+1}{2}}}$$

$$\phi_{u,k} = \frac{(1 + {d_{u,k}^{(2)}}^2/t)^{-\frac{t+1}{2}}}{\sum_{k'}(1 + {d_{u,k}^{(2)}}^2/t)^{-\frac{t+1}{2}}}$$

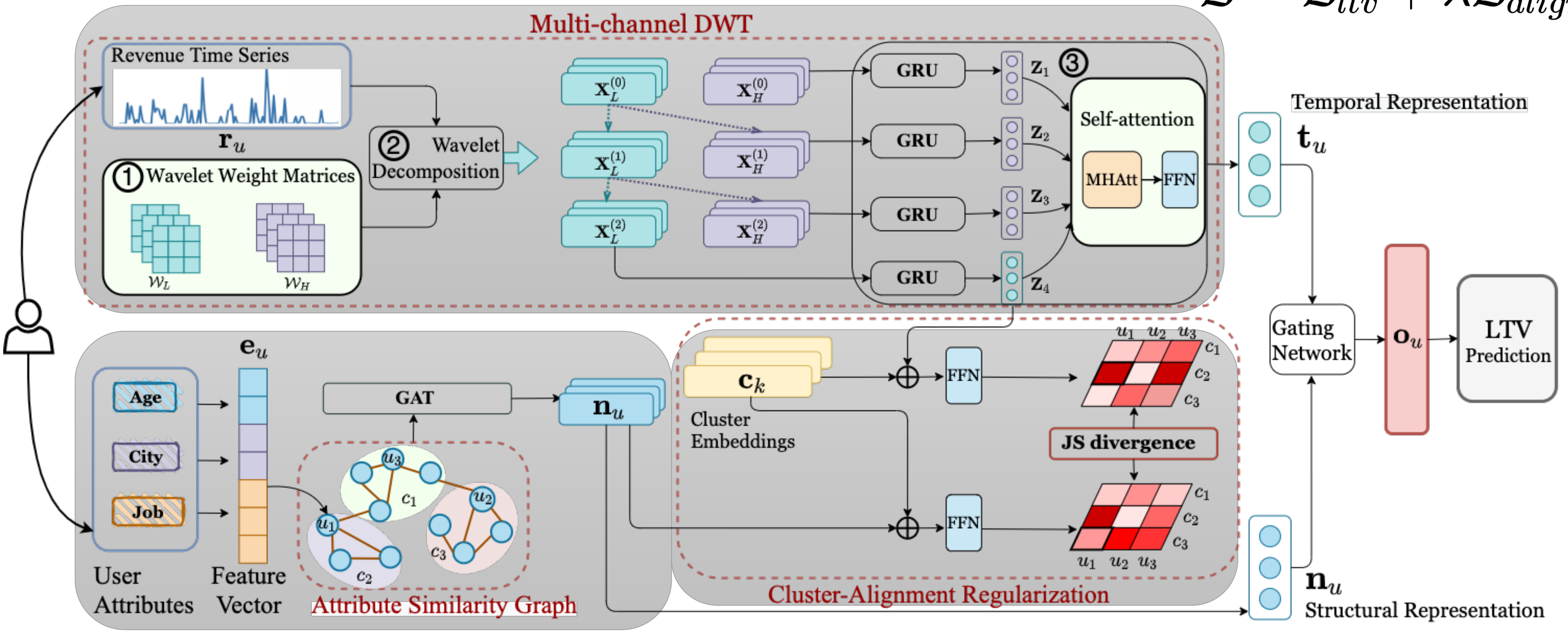
Step 3. Cluster-alignment regularization

$$\mathcal{L}_{align} = \frac{1}{2}KL(\Theta||\frac{\Phi + \Theta}{2}) + \frac{1}{2}KL(\Phi||\frac{\Phi + \Theta}{2})$$
$$= \frac{1}{2}\sum_{u \in \mathcal{U}}\sum_{k=1}^{K}\left(\theta_{u,k}\log\frac{2\theta_{u,k}}{\phi_{u,k} + \theta_{u,k}} + \phi_{u,k}\log\frac{2\phi_{u,k}}{\phi_{u,k} + \theta_{u,k}}\right)$$



Methodology (Fusion)

$$\mathcal{L} = \mathcal{L}_{ltv} + \lambda \mathcal{L}_{align}$$



Datasets

- Two datastes:
  - PI: pre-installation on new mobile phones
  - AS: download in app stores
- Training / Validation / Test set sizes = 8 : 1 : 1

Table 1: The statistics of our datasets.

Dataset	#users	average consumption frequency	average LTV
PI	33,505	14.46	2.01
AS	36,264	14.35	2.00

Compared Methods

- Four categories of baselines
  - LTV prediction
    - **Two-stage XGBoost**
    - **Group RandomForest**
    - **WhalesDetector**
  - Time series forecasting
    - **DSANet**
    - **LSTNet**
    - **Nbeats**
  - Graph neural network
    - **GAT**
    - **GraphSAGE**
    - **Graph WaveNet**
  - User behavior model
    - **TiSSA**

Performance Comparison

Methods	PI				AS			
	30-day		90-day		30-day		90-day	
	NRMSE	NMAE	NRMSE	NMAE	NRMSE	NMAE	NRMSE	NMAE
Two-stage XGBoost	0.8786	0.5709	1.0386	0.6237	0.9012	0.5834	1.0422	0.6275
Group RandomForest	0.6681	0.4625	0.8910	0.5984	0.6853	0.4777	0.8978	0.6107
WhalesDetector	<u>0.5396</u>	<u>0.3167</u>	<u>0.8456</u>	0.4681	<u>0.5467</u>	<u>0.3256</u>	<u>0.8915</u>	0.4935
DSANet	0.7248	0.3619	0.9916	0.5889	0.7273	0.3436	1.0168	0.6302
LSTNet	0.6671	0.3265	0.8860	0.5821	0.7251	0.4075	0.9685	0.6559
NBeats	0.5843	0.3513	0.8834	0.5211	0.5489	0.3392	0.9245	0.5403
GraphSAGE	0.7868	0.5271	0.9886	0.6328	0.7499	0.5101	1.0397	0.6437
Graph WaveNet	0.6266	0.3306	0.9599	<u>0.4482</u>	0.7343	0.4378	0.9582	<u>0.4830</u>
TiSSA	0.7521	0.5478	0.9949	0.7333	0.7756	0.5744	1.0141	0.7311
TSUR (our method)	**0.4274**	**0.2464**	**0.7193**	**0.4220**	**0.4432**	**0.2542**	**0.6863**	**0.3915**

Ablation study

- Four variants are compared:
  - **T** : use only the **temporal representation** to predict LTV;
  - **S** : use only the **structural representation** to predict LTV;
  - **TS** : **directly fusing** the two representations to predict LTV;
  - **TSC** : our **complete model**.

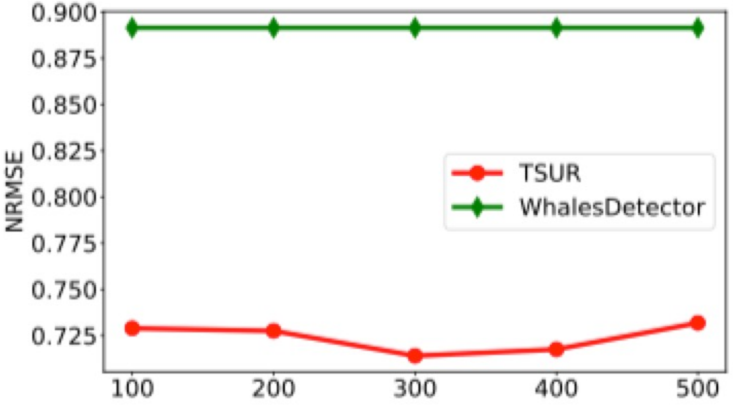
Ablation study

- Four variants are compared:
  - **T** : use only the **temporal representation** to predict LTV;
  - **S** : use only the **structural representation** to predict LTV;
  - **TS** : **directly fusing** the two representations to predict LTV;
  - **TSC** : our **complete model**.

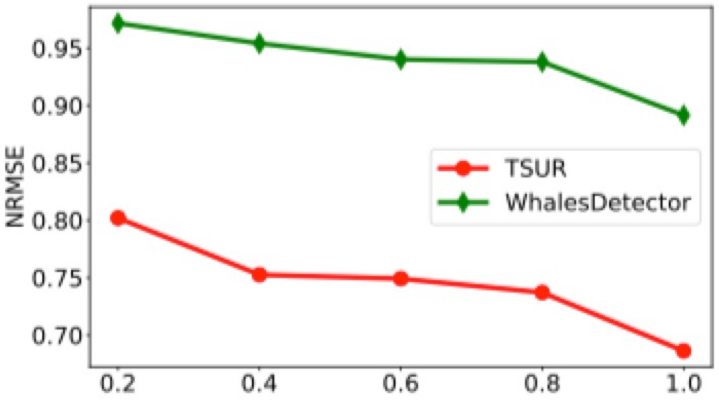
Future horizon	Variant	PI		AS	
		NRMSE	NMAE	NRMSE	NMAE
30 days	T	0.4660	0.2655	0.4853	0.2792
	S	0.7242	0.4817	0.7304	0.4715
	TS	0.4379	0.2504	0.4594	0.2611
	TSC	**0.4274**	**0.2464**	**0.4432**	**0.2542**
90 days	T	0.7501	0.4434	0.7511	0.4404
	S	0.9847	0.6009	1.0208	0.6957
	TS	0.7448	0.4241	0.7119	0.4026
	TSC	**0.7193**	**0.4220**	**0.6863**	**0.3915**

S < T < TS < TSC

Performance Tuning



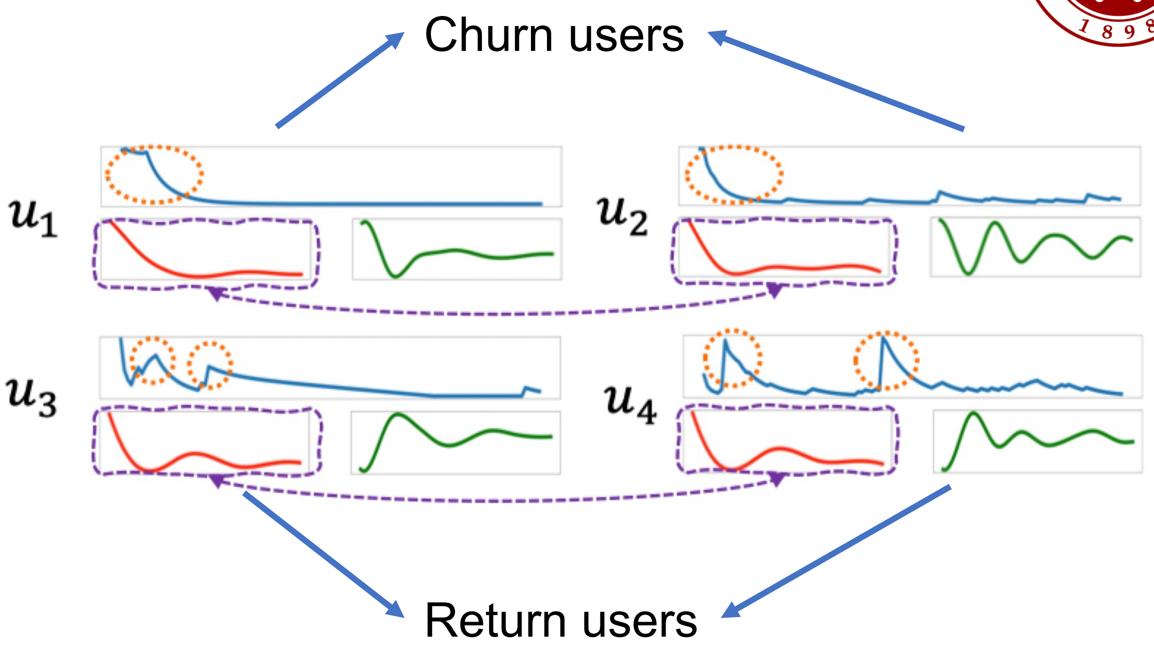
(a) The number of clusters.



(b) The ratio of training data.

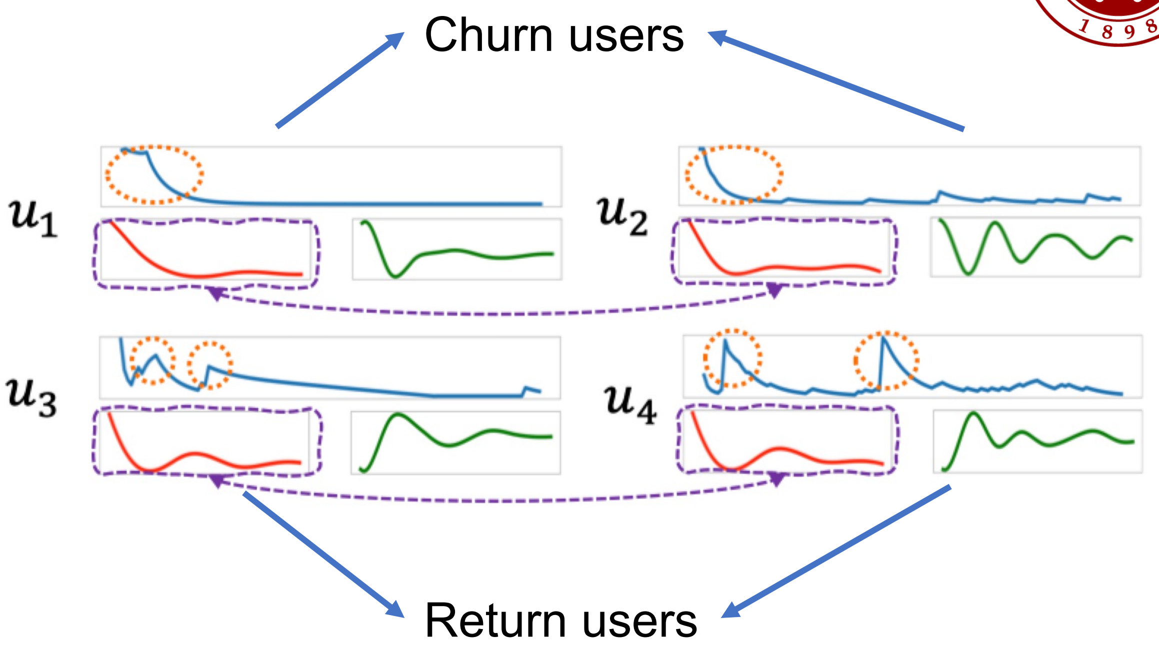
Case Study

- The raw time series is in blue and the decomposed low- and high-frequency components are in red and green.

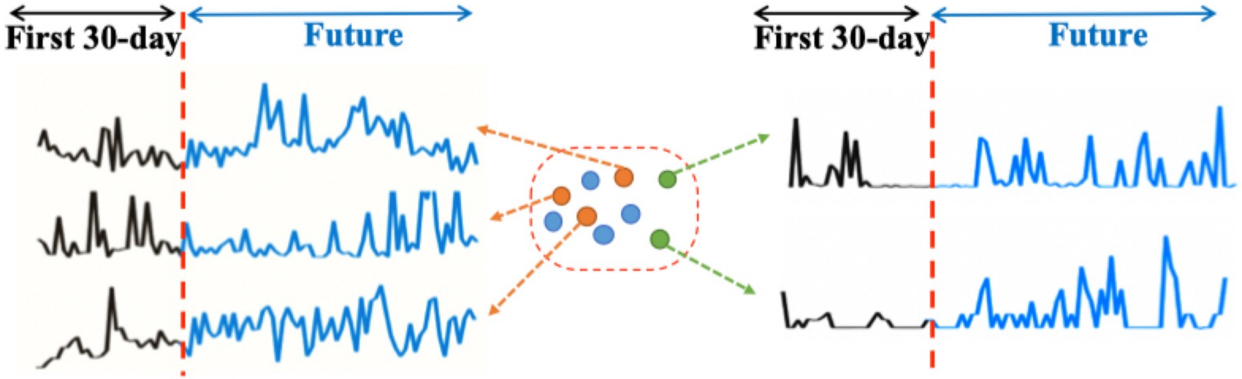


Case Study

- The raw time series is in blue and the decomposed low- and high-frequency components are in red and green.



- The first 30-day and future sequence are in black and blue, respectively.



Online A/B Test

- Return on Investment (ROI)

$$ROI = \frac{Net\ Return\ on\ Investment}{Cost\ of\ Investment}$$

Methods	ROI-10	ROI-20
WhalesDetector	0.1420	0.3571
TSUR	**0.1636**	**0.3699**

Conclusion

- We proposed a **Temporal-Structural user representation model** for LTV prediction.

Conclusion

- We proposed a **Temporal-Structural user representation model** for LTV prediction.
- For temporal trend encoder, we developed an improved **multi-channel DWT** to learn more reliable temporal user representations;

Conclusion

- We proposed a **Temporal-Structural user representation model** for LTV prediction.
- For temporal trend encoder, we developed an improved **multi-channel DWT** to learn more reliable temporal user representations;
- For structural encoder, we leveraged GAT to learn **structural user representations** over **attribute similarity graph**.

Conclusion

- We proposed a **Temporal-Structural user representation model** for LTV prediction.
- For temporal trend encoder, we developed an improved **multi-channel DWT** to learn more reliable temporal user representations;
- For structural encoder, we leveraged GAT to learn **structural user representations** over **attribute similarity graph**.
- Cluster-alignment regularization technique was proposed to **align the two kinds of user representations**.

Conclusion

- We proposed a **Temporal-Structural user representation model** for LTV prediction.
- For temporal trend encoder, we developed an improved **multi-channel DWT** to learn more reliable temporal user representations;
- For structural encoder, we leveraged GAT to learn **structural user representations** over **attribute similarity graph**.
- Cluster-alignment regularization technique was proposed to **align the two kinds of user representations**.
- Future work
  - incorporate other influencing factors such as bursty social events;
  - leverage other kinds of user correlation data such as social graphs to learn better structural user representations.

Thank you