Efficient User Sequence Learning for Online Services via Compressed Graph Neural Networks

Yucheng Wu[12], Liyue Chen[12], Yu Cheng[3], Shuai Chen[3], Jinyu Xu[3], Leye Wang[12]

[1]Key Lab of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing, China
[2]School of Computer Science, Peking University, Beijing, China
[3]Alipay (Hangzhou) Information & Technology Co. Ltd., Hangzhou, China

Background

- **User sequences data** record user online activities over time
  - An example of user behavior sequence for online shopping:

- **Sequence representation learning**
  - Step 1: map user sequences into embedding vectors using deep learning models
  - Step 2: conduct predictions
- rely solely on an individual user's historical behavior sequences
- overlook information inherent in other relevant sequences

- **GNN-based sequence representation learning**
  - leverage similar sequences from other users and model their correlations

Motivation

- **Efficiency and scalability challenges** for deploying GNN-based models to ***online services***:
  - **Training**: the number of user sequences produced by online applications is often immense, potentially escalating to the magnitude of millions → large-scale graphs incur substantial computational and memory burdens
  - **Inference**: online services typically require rapid response → puts stringent demands on the algorithms' inference efficiency.

Basic idea

- we question the necessity of modeling all user sequences as nodes in GNNs
- → select a representative node subset
- **Solution**: compress the graph by reducing nodes and edges prior to GNN model training, benefiting computational efficiency

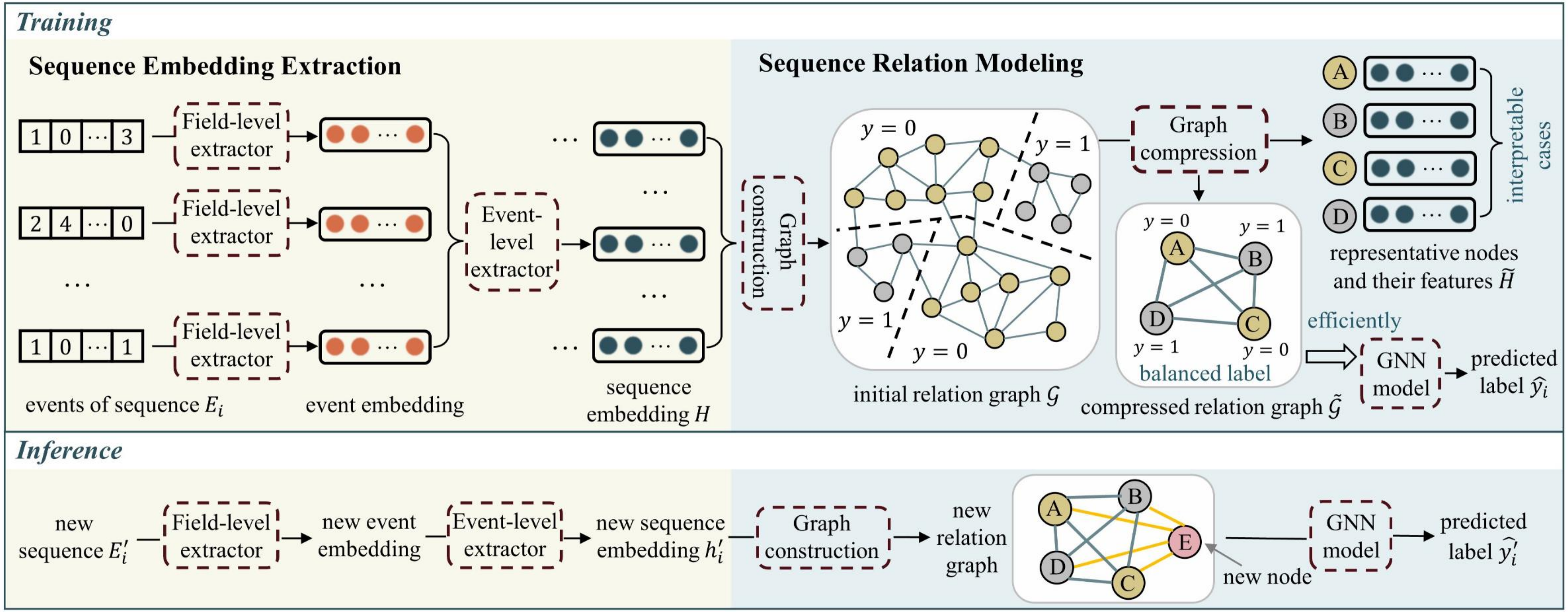
Contributions

PEKING UNIVERSITY

- We propose ECSeq, a unified user sequence learning framework for online services, to *incorporate the relations between target and similar sequences*. ECSeq enhances efficiency and scalability via **graph compression**, thus resolving the dilemma of relation modeling on large-scale sequence data and online inference with low latency.

- We compare and adapt suitable graph compression techniques for ECSeq, meeting **efficiency**, **interpretability**, and **sample balancing** demands simultaneously. Besides, ECSeq is designed to hold *plug-and-play* characteristics, seamlessly augmenting pre-trained sequence representation models in existing systems without the need to modify these models.
  - Interpretability: Graph compression provides representative sequence prototypes, offering interpretable cases of model outputs.
  - Sample Balancing: In biased distributions (e.g., fraud detection), compression can balance categories by setting similar compressed node counts.

Framework

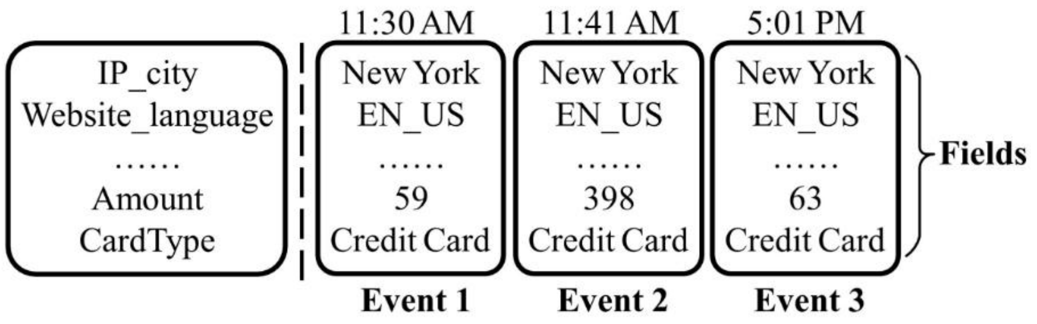


- Firstly, the sequence embedding extraction module transforms sequence information into a one-dimensional feature vector.
- Then, the sequence relation modeling module explores and leverages relationships among sequences to enhance the sequence representation, employing an appropriate graph compression technique to mitigate computational overhead and improve inference efficiency.

Sequence Embedding Extraction

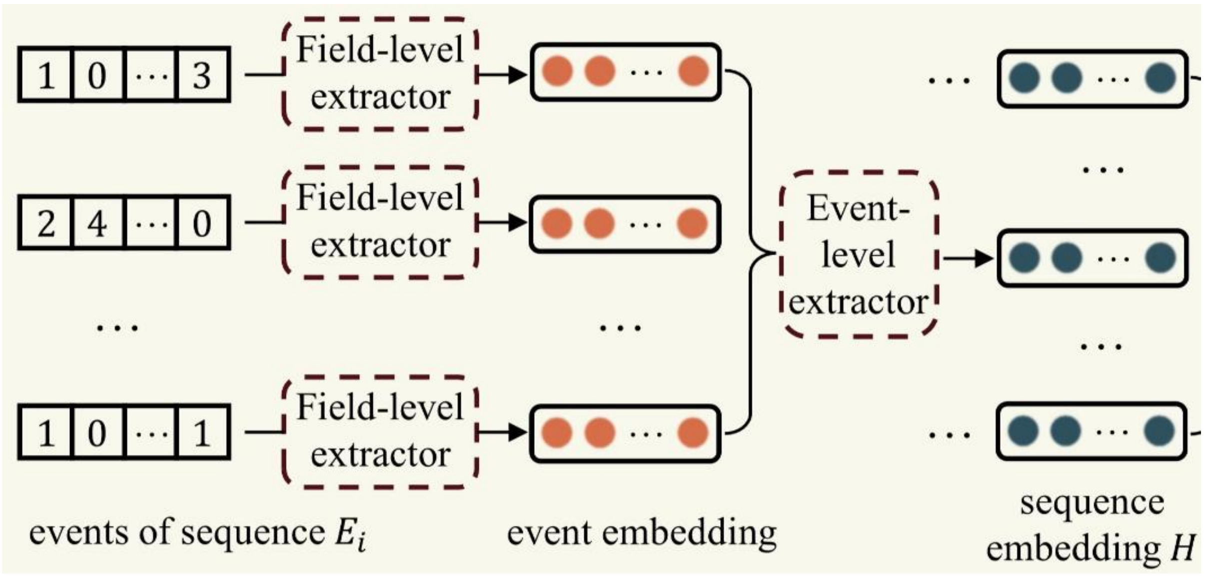
Field, Event, and Sequence

- A user behavior sequence example consists of three events and each event has several fields.



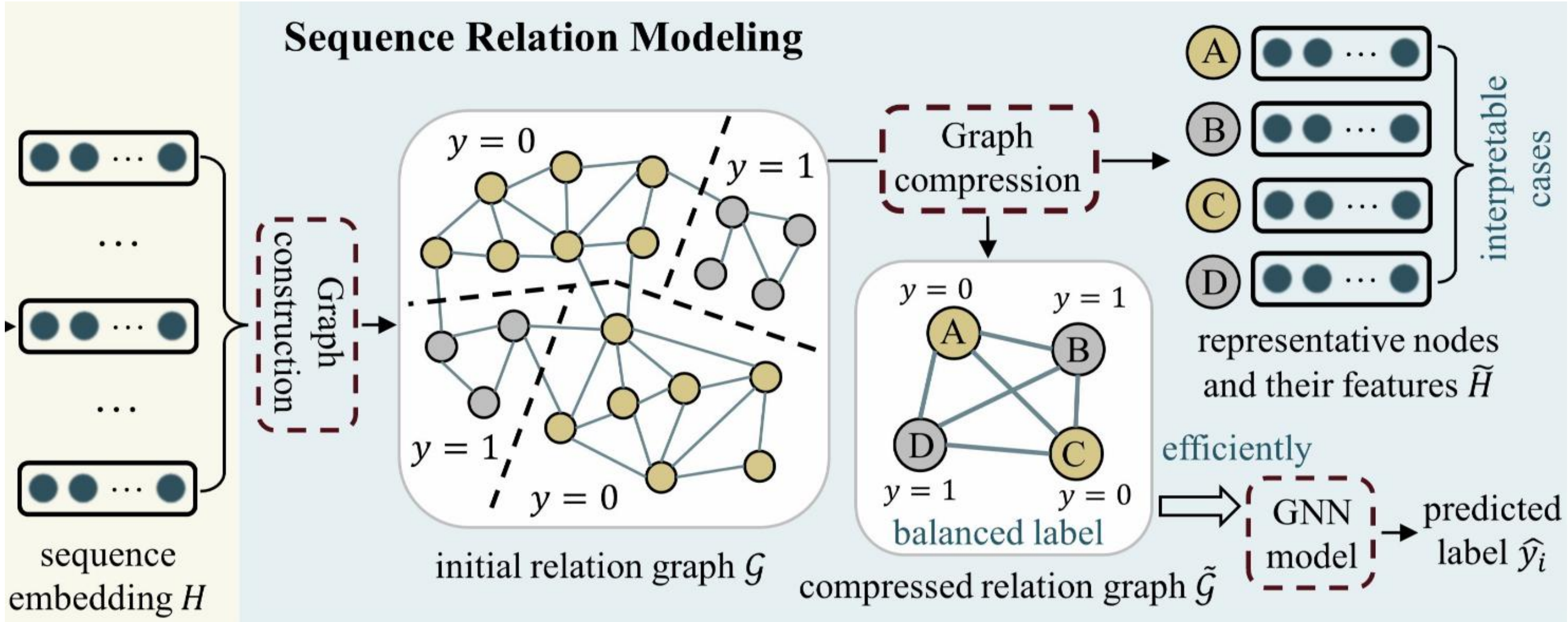
Sequence Embedding Extraction

- capture a user's sequence representation by taking into account both field-level characteristics and event-level sequential patterns over time.
- transforms sequence information into a one-dimensional feature vector.



北京大学 PEKING UNIVERSITY

Sequence Relation Modeling



Sequence Relation Modeling

- leverages relationships among sequences to enhance the sequence representation, employing an appropriate graph compression technique to mitigate computational overhead and improve inference efficiency.

1. **Graph Construction:** we regard users' sequences as nodes to construct a relationship graph, and we can infer node connectivity based on attribute similarity.
2. **Graph Compression:** we aim to reduce the number of nodes or edges while maintaining model performance.
3. **Graph Prediction:** we extract node representations of the compressed relation graph and make

Graph Compression

TABLE I: Summary of typical graph compression methods. N: number of nodes, M: number of edges, D: dimension of node features, K: number of clusters/compressed nodes, c: some absolute constant. *Traceable*: whether the source of the compressed nodes is known; *Configurable*: whether the compression method can assign separate compressed node quantities for each category.

Category	Methods	Input	*Efficiency*		*Interpretability*	*Balancing*
			#Nodes↓	**#Edges↓**	**Traceable**	**Configurable**
Coreset Selection	k-means [27]	$\mathcal{X}$	✓	✓	✓	✓
	AGC [28]	$\mathcal{A}, \mathcal{X}$	✓	✓	✓	✗
	Grain [29]	$\mathcal{A}, \mathcal{X}$	✓	✓	✓	✓
	VNG [30]	$\mathcal{A}, \mathcal{X}$	✓	✓	✓	✗
Graph Coarsening	RSA [31]	$\mathcal{A}$	✓	✓	✓	✗
	REC [32]	$\mathcal{A}$	✓	✓	✓	✗
	GOREN [22]	$\mathcal{A}$	✓	✓	✓	✗
Graph Sparsification	ApproxCut [24]	$\mathcal{A}$	✗	✓	✓	✗

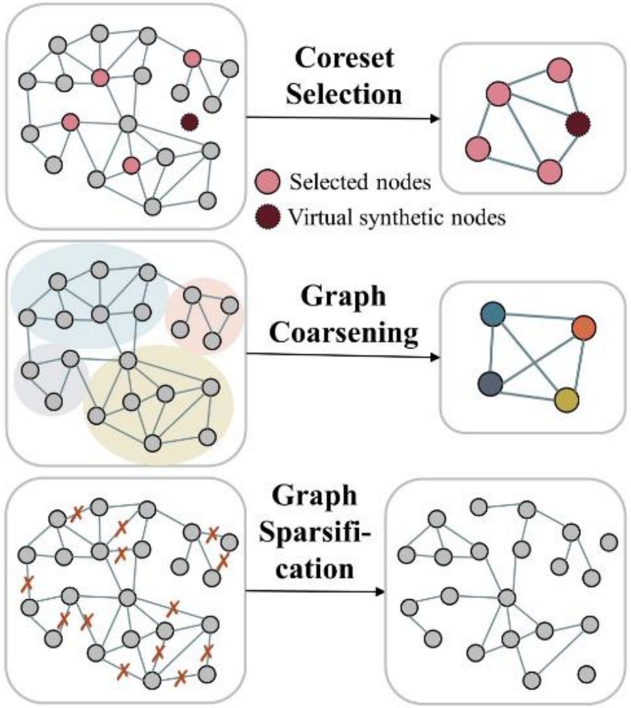


Fig. 3: Illustration of diverse graph compression methods.

- **Coreset Selection** expedites training by selecting or synthesizing a subset of representative nodes.
- **Graph Coarsening** combines original nodes into super-nodes and establishes their connections.
- **Graph Sparsification** reduces the number of edges in a graph by approximating its structural properties.

北京大学 PEKING UNIVERSITY

Workflow

Sequence Embedding Extraction

Algorithm 1: *ECSeq* Training Procedure

Input: Sequence set E and its label matrix Y

Output: Optimized sequence embedding extractor $(\mathcal{M}_f, \mathcal{M}_e)$, optimized relation model $(\mathcal{M}_g, \mathcal{F}_{gnn})$, and the compressed graph $\tilde{\mathcal{G}}$.

```
Initialize parameters of sequence embedding extractor M_f, M_e, and F_seq;
while stopping condition is not met do
    H = M_e(M_f(E)), Ŷ = F_seq(H);
    Compute the loss L_seq by Eq. 4;
    Update the parameters of M_f, M_e, and F_seq;
end
Treat sequences as nodes with X = H;
Construct node connections and get relation graph G = (A, X);
Compress G to get the compressed graph G̃ = (Ã, X̃) and label Ỹ;
Initialize parameters of relation model M_g and F_gnn;
while stopping condition is not met do
    Ŷ = F_gnn(M_g(Ã, X̃));
    Compute the loss L_com by Eq. 10;
    Update the parameters of M_g and F_gnn;
end
Establish connections between X and X̃, denoted as A';
while stopping condition is not met do
    Ŷ' = F_gnn(M_g(A', X ∪ X̃));
    Compute the loss L_cor by Eq. 12;
    Update the parameters of M_g and F_gnn;
end
```

Sequence Relation Modeling

Algorithm 2: *ECSeq* Inference Procedure

Input: Optimized sequence embedding extractor $(\mathcal{M}_f, \mathcal{M}_e)$, optimized relation model $(\mathcal{M}_g, \mathcal{F}_{gnn})$, the compressed graph $\tilde{\mathcal{G}}$, and a set of new sequences E''

Output: The predicted label $\hat{Y}''$ of the new sequences.

```
Get new sequence embedding H'' = M_e(M_f(E''));
Treat new sequences as nodes with X'' = H'';
Establish connections between X'' and X̃, denoted as A'';
Derive inference results Ŷ'' = F_gnn(M_g(A'', X'' ∪ X̃));
```

Our ***step-wise*** approach has advantages over end-to-end training:

- Graph compression is only needed once, mitigating instability and inefficiency;
- Both modules' training processes incorporate label information supervision, ensuring the models' stability and optimality;
- Decoupled optimization enables flexibility in using diverse models without alignment concerns.

Experiments

PEKING UNIVERSITY 北京大學

Datasets

		Fraud Detection		
Datasets	#Fields	#Events	#Sequences	#Positive Samples
FD1	236	2,130,962	245,045	24,489 (9.99%)
FD2	178	275,322	15,366	777 (5.06%)
		User Mobility		
Datasets	#Sensors	#Timesteps	Time Interval	Value Range
Bike	717	1,488	60 minutes	$0.0 \sim 108.0$
Speed	325	1,488	60 minutes	$3.1 \sim 83.2$

- FD1/FD2: consist of real-world online card transaction sequences from a global e-commerce company.
- Bike: forecasts the number of bike-sharing demands at each station.
- Speed: contains traffic speed data from the Bay Area.

Baselines:

- *Methods with only features of the target event:* Regression and GBDT take features extracted by the field-level extractor of the target event as inputs to train a machine-learning classifier.
- *Methods with deep neural networks to extract historical information:* LSTM can capture long-term dependencies for sequential data, then we give the prediction by MLP layers, while R-Transformer combines RNNs and the multi-head attention mechanism.
- *Methods with GNN to capture sequence relationship:* GRASP enhances representation learning by leveraging knowledge extracted from similar users within the same batch, which is originally proposed for healthcare sequence classification problems.

Experimental Results -- Effectiveness

北京大學 PEKING UNIVERSITY

TABLE IV: Experimental results on fraud detection and user mobility tasks. The best results are highlighted in bold. While *R-Transformer* [48] and *GRASP* [7] are primarily intended for classification tasks, they do not show comparable performance on user mobility tasks.

Methods	FD1		FD2		Bike		Speed	
	AUPRC (↑)	R@P$_{0.9}$ (↑)	AUPRC (↑)	R@P$_{0.9}$ (↑)	RMSE (↓)	sMAPE (↓)	RMSE (↓)	sMAPE (↓)
Non-Graph Methods								
Regression	0.7685±0.0000	0.4890±0.0000	0.5271±0.0000	0.3052±0.0000	2.8169±0.0000	0.2148±0.0000	6.3994±0.0000	0.0672±0.0000
GBDT	0.7742±0.0000	0.5244±0.0000	0.6147±0.0006	0.3766±0.0000	2.7596±0.0077	0.2063±0.0008	6.2964±0.0575	0.0632±0.0005
LSTM	0.8332±0.0047	0.5840±0.0132	0.7124±0.0076	0.6987±0.0078	1.5463±0.0504	0.1782±0.0040	4.8383±0.1600	0.0561±0.0011
R-Transformer	0.8338±0.0040	0.5847±0.0289	0.7064±0.0149	0.5403±0.1951	–	–	–	–
Graph Methods								
GRASP	0.8362±0.0037	0.6049±0.0230	0.7138±0.0353	0.6776±0.0420	–	–	–	–
ECSeq	**0.8383±0.0018**	**0.6153±0.0079**	**0.7249±0.0112**	**0.7039±0.0032**	**1.4832±0.0209**	**0.1766±0.0038**	**4.4362±0.1015**	**0.0542±0.0012**

- Compared to Regression and GBDT that only use features of target event, LSTM and R-Transformer extract historical information of sequences, and have significant improvement in all datasets.
- ECSeq further enhances LSTM by exploring and utilizing the relationships among sequences, and ECSeq gains the best performance in all of the four datasets.

Experimental Results -- Efficiency and Scalability

TABLE V: Fraud detection performance and computation time on *FD1*, whose training set contains 145,236 sequences, *i.e.*, 145,236 original nodes when modeling the relation. We train the GNN for 50 epochs.

Methods	#Nodes	AUPRC (↑)	$R@P_{0.9}$ (↑)	Compression Time (s) (↓)	GNN Training Time (s) (↓)	Inference Time (10^{-4} s/sample) (↓)	GPU Memory Usage (GB) (↓)
LSTM	–	0.8332±0.0047	0.5840±0.0132	–	–	**0.286**	–
ECSeq	100	0.8377±0.0023	0.6111±0.0077	**5.318**	**9.950**	0.614	**1.306**
	500	**0.8383±0.0018**	**0.6153±0.0079**	15.444	9.955	0.618	1.664
	1,000	0.8372±0.0021	0.6115±0.0083	36.686	10.480	0.622	2.283
	5,000	0.8343±0.0013	0.6056±0.0055	137.570	28.930	0.630	7.037
batch GNN	145,236 (1,000/batch)	0.8335±0.0017	0.5861±0.0157	–	10.675	2.243	6.414
full graph GNN	145,236	Out-of-Memory					

- Training GNN on the full graph would result in an out-of-memory issue on our GPU with 11 GB RAM.
- However, by using graph compression, we can achieve satisfactory performance improvement (~ 5% increase in $R@P_{0.9}$) while using only around 1GB of GPU RAM (100 compressed nodes).
- When compressing the original graph to 100 nodes using k-means, the time consumption, including both compression and GNN training, is only 15.3 seconds. The inference time consumption is still at the 10^{-5} second-scale per sequence, similar to LSTM.
- These results demonstrate the practicality of using ECSeq in real systems.

Experimental Results -- Flexibility

北京大學 PEKING UNIVERSITY

TABLE VI: Performance of *ECSeq* variants on *FD2* and *Bike*. *R-Transformer* cannot converge on *Bike*, so we do not report the results.

	Sequence Embedding Extractor	Graph Compression Algorithm	GNN Model	FD2		Bike	
				AUPRC (↑)	R@P$_{0.9}$ (↑)	RMSE (↓)	sMAPE (↓)
Sequence Model w.o. Graphs	LSTM	–	–	0.7124±0.0076	0.6987±0.0078	1.5463±0.0504	0.1782±0.0040
	R-Transformer	–	–	0.7064±0.0149	0.5403±0.1951	–	–
ECSeq	LSTM	k-means	GraphSAGE (Mean)	**0.7249±0.0112**	**0.7039±0.0032**	1.4832±0.0209	0.1766±0.0038
	R-Transformer	k-means	GraphSAGE (Mean)	0.7105±0.0184	0.6991±0.0031	–	–
	LSTM	AGC	GraphSAGE (Mean)	0.7232±0.0102	0.6935±0.0095	**1.4792±0.0218**	0.1764±0.0044
	LSTM	Grain	GraphSAGE (Mean)	0.7232±0.0102	0.6870±0.0095	1.4949±0.0219	0.1812±0.0050
	LSTM	RSA	GraphSAGE (Mean)	0.7201±0.0091	0.6922±0.0120	1.4812±0.0190	**0.1761±0.0040**
	LSTM	k-means	GraphSAGE (Max)	0.7162±0.0083	0.7026±0.0049	1.4846±0.0166	0.1768±0.0037
	LSTM	k-means	GCN	0.7212±0.0102	0.6987±0.0052	1.9226±0.0376	0.2139±0.0059
	LSTM	k-means	GAT	0.7200±0.0112	0.7026±0.0026	2.0056±0.0107	0.2383±0.0053

- We conduct experiments to evaluate the flexibility of ECSeq framework in modifying sequence embedding extractors, graph compression methods, and GNN algorithms.
- When either LSTM or R-Transformer is used as the sequence embedding extractor, applying ECSeq can greatly enhance prediction performance. The plug-and-play characteristics of ECSeq can flexibly enhance existing sequential modeling models that do not account for sequence relations.
- Results show that different variants may have varying performance on different tasks. The flexibility of ECSeq allows us to easily change the compression and GNN algorithms.

Conclusion and Limitation

Conclusion

- Our proposed framework, named ECSeq, aims to improve user behavior sequence learning by ***integrating sequence relations***, while maintaining **high efficiency and scalability** for applicability in online services.
- The framework consists of two modules: **sequence embedding extraction** and **sequence relation modeling**, which enhances training and inference efficiency and provides a plug-and-play capability.
- Specifically, with an extra training time of tens of seconds in total on 100,000+ sequences and inference time maintained within 10^{-4} seconds/sample, ECSeq enhances the prediction $R@P_{0.9}$ of the widely used LSTM by ~ 5%.

Limitation

- Currently, ECSeq has a limitation where it can only deal with one type of relation. We aim to simultaneously **incorporate more types of real-life relationships**, such as users' social networks and behavioral habits similarity, into our framework in the future.
- We believe that the integration of different relationships can greatly enhance the relation modeling module. Our next step is thus to explore how to effectively and efficiently incorporate heterogeneous relationships.

Thanks!

Contact: wuyucheng@stu.pku.edu.cn

Paper

Code