Tutorial

Self-Supervised Learning for Recommender Systems

Foundations, Methods and Prospects

Presenter: **Junliang Yu, Tong Chen, Hongzhi Yin**

The University Of Queensland AUSTRALIA

ARC Training Centre for Information Resilience

Tutorial website: https://dasfaa-ssr.github.io/

CONTENTS

- **Background and History**
- **Definition and Taxonomy**
- **Data Augmentation**
- **Methodology**
  - Contrastive, Generative, Predictive, and Hybrid
- **Our Work -** Simple Contrastive Recommendation
- **Library and Findings**
- **Future Research**

THE WEB CONFERENCE ACM

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

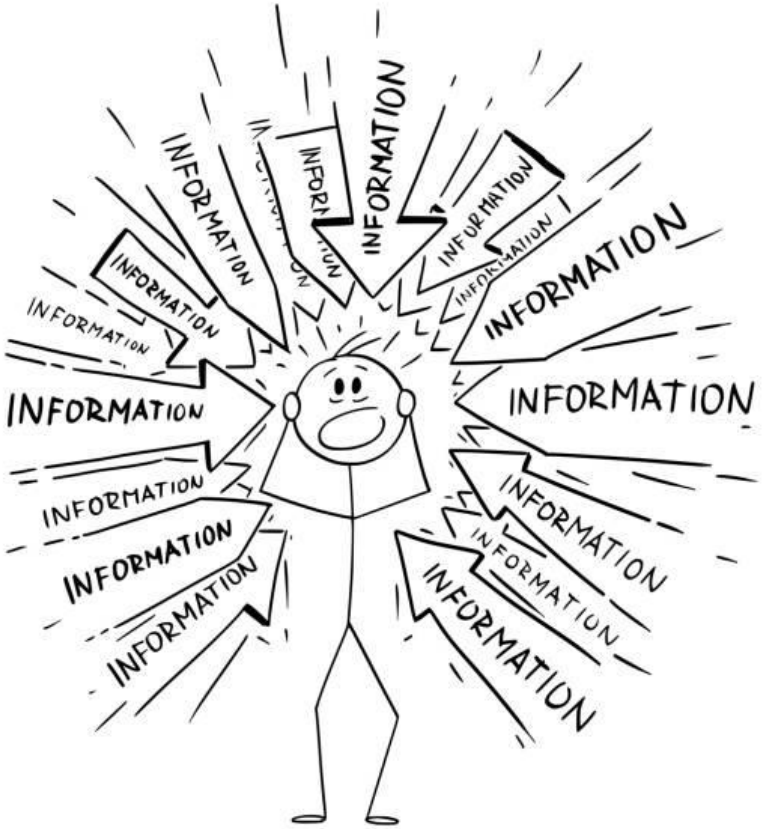
01

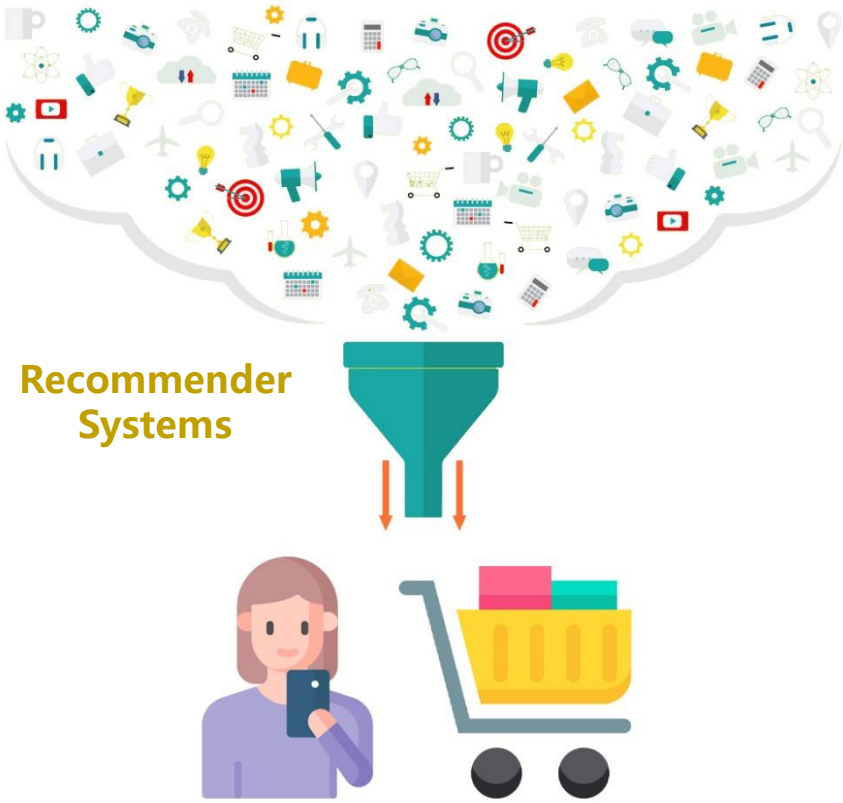
Background and History

Tutorial

Recommender Systems

We are now living in an age of information explosion.



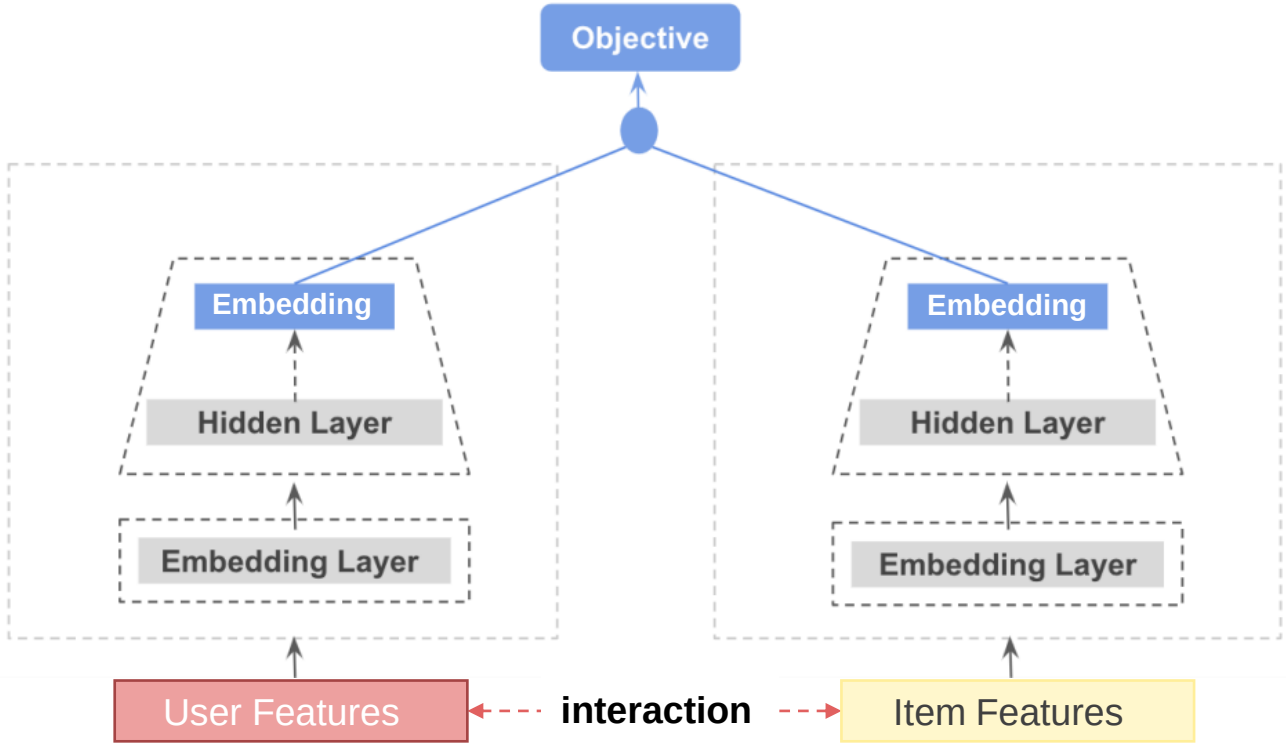


Recommender systems can discover users' interests and ease the way of decision-making.

Recommender Systems

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

A general neural-architecture for recommendation.



(gender, age, behaviors, etc.)

(category, price, brand, etc.)

Use the historical interaction/feature data to compute item scores and recommend.

Problem of Data Sparsity

The University Of Queensland AUSTRALIA

Most users generally only consume a tiny fraction of numerous items.

How to generate satisfactory recommendations with very sparse interactions?

Self-Supervised Learning (SSL)

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

What is it?

Self-supervised learning that enables algorithms to learn from large amounts of **unlabeled data** by discovering patterns and similarities in the data itself.

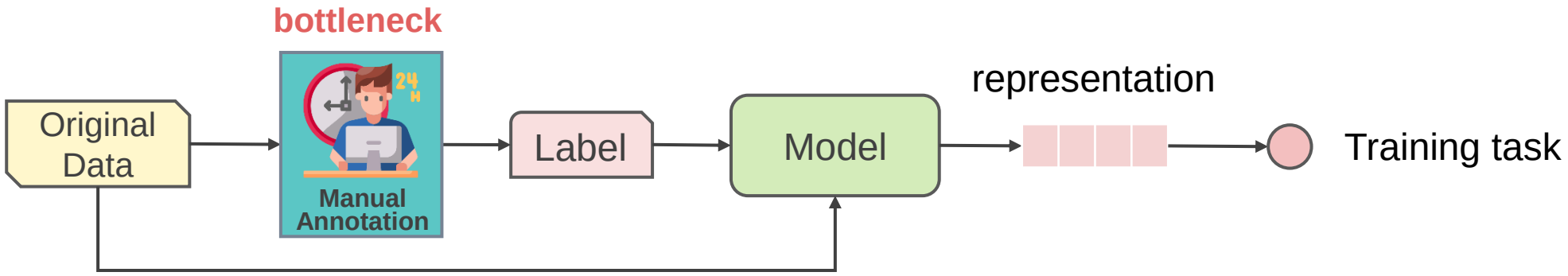
Why do we need it in recommender systems?

Labeled data may be scarce or expensive to obtain in recommender systems, particularly when dealing with new or niche products.

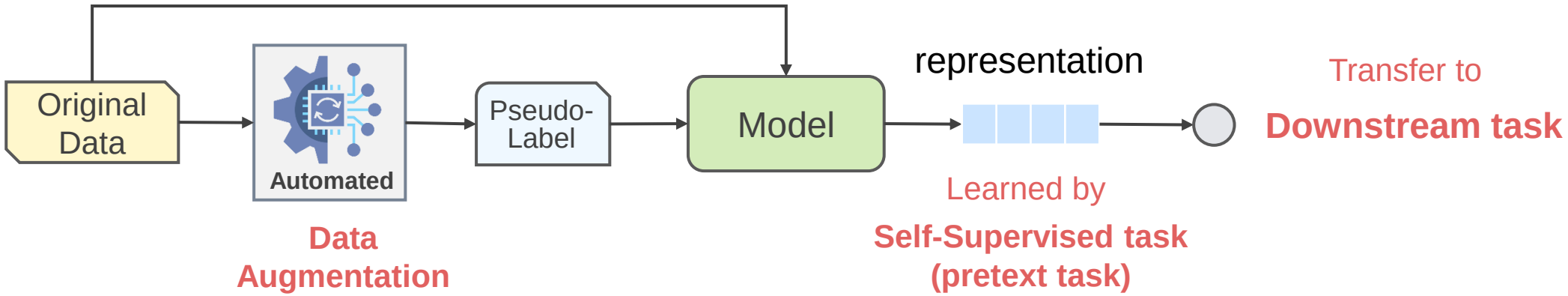
Self-Supervised Learning (SSL)

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Supervised learning workflow



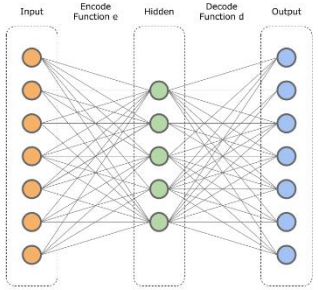
Self-Supervised learning workflow



Development of SSR

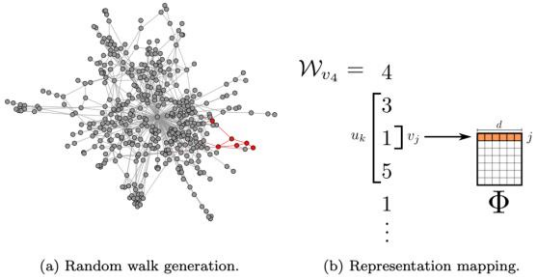
THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Autoencoder-Based



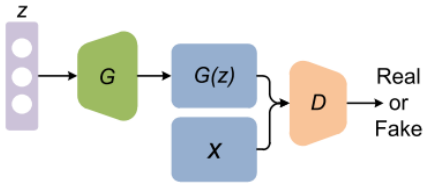
2016

Network-Embedding-Based



2017

GAN-Based

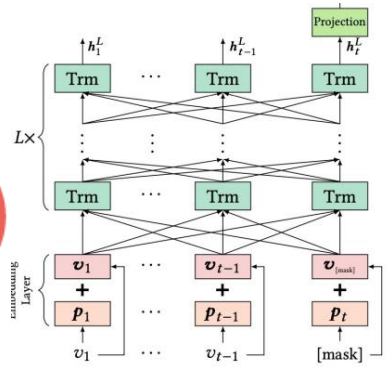


2018

2019

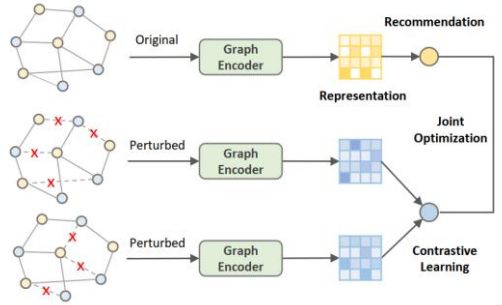
Self-Supervised Learning becomes an independent concept

BERT-Based



2020

Contrastive Learning-Based



Early prototypes and implementations

Becoming Diverse

THE WEB CONFERENCE ACM

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

02

Definition and Taxonomy

Tutorial

Self-Supervised Recommendation (SSR)

The University Of Queensland AUSTRALIA

Self-supervised recommendation has following essential features[1]:

- Semi-automatically exploiting the raw data itself to obtain more supervision signals.
- Incorporating a self-supervised task(s) to (pre-)train the recommendation model using augmented data.
- The self-supervised task is designed to enhance recommendation performance, rather than being an end goal.

Self-supervised recommendation $\neq$ Pretraining-based Recommendation

Self-supervised recommendation $\neq$ Contrastive Learning-based Recommendation

[1]. Yu J, Yin H, Xia X, et al. Self-Supervised Learning for Recommender Systems: A Survey[J]. arXiv preprint arXiv:2203.15876, 2022.

Architecture of SSR

Architecture: **Encoder** + **Projection Head**

(GNNs, Transformer, etc.) (nonlinear/linear mapping, identical mapping, etc.)

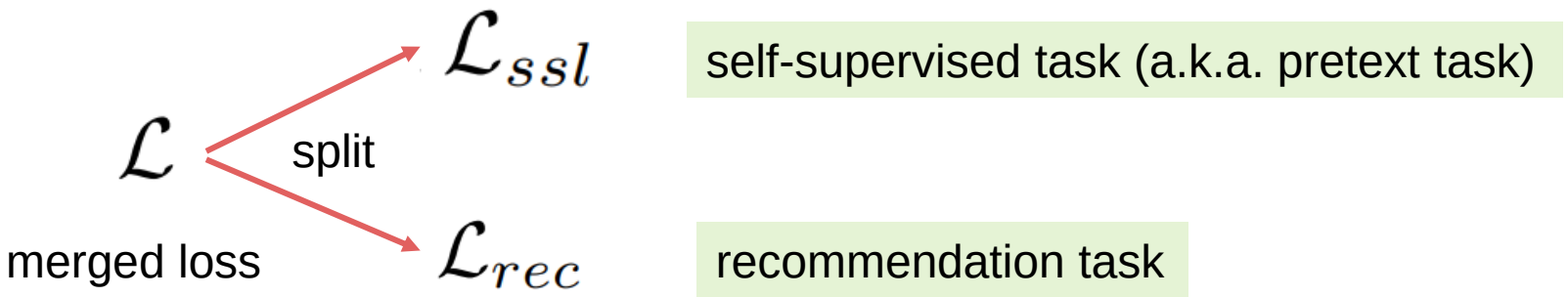
$$f_{\theta^*}, g_{\phi^*}, \mathbf{H}^* = \arg\min_{f_\theta, g_\phi} \mathcal{L}\left(g_\phi(f_\theta(\mathcal{D}, \tilde{\mathcal{D}}))\right)$$

representation; projection-head; encoder; original data; augmented data

$$\tilde{\mathcal{D}} \sim \mathcal{T}(\mathcal{D})$$

augmentation module

Multi-Task Learning:



merged loss

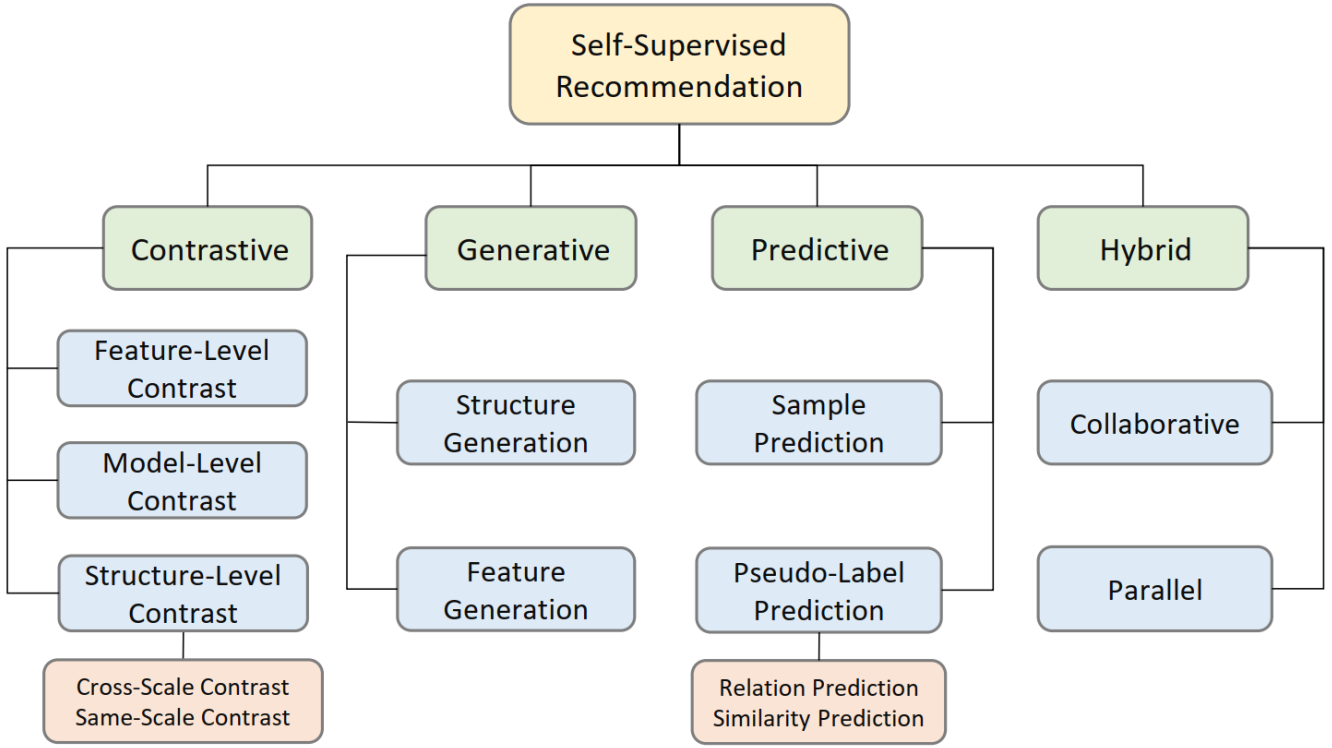
self-supervised task (a.k.a. pretext task)

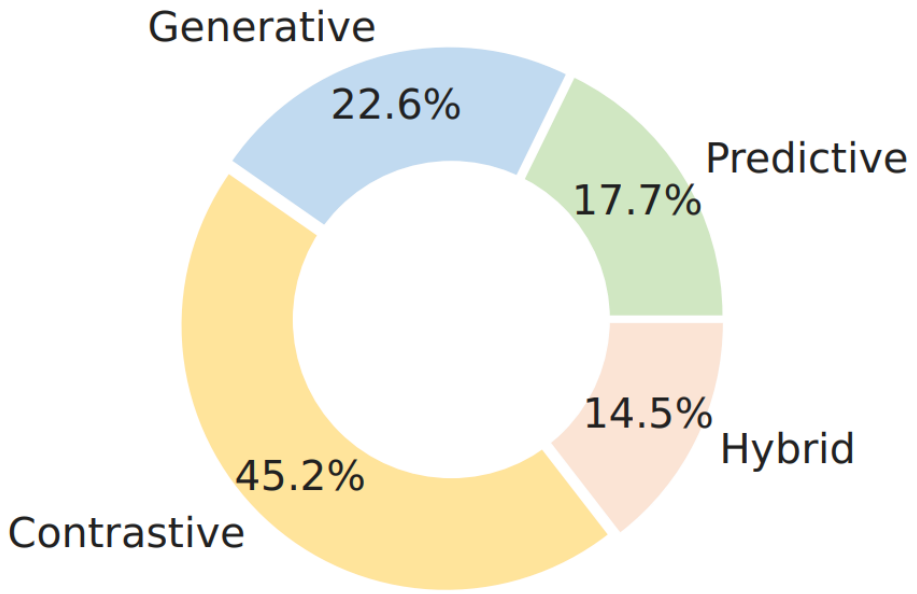
recommendation task

Taxonomy of SSR

The University Of Queensland AUSTRALIA

According to the characteristics of pretext tasks, existing SSR methods can be divided into four categories: **contrastive, predictive, generative, and hybrid**.

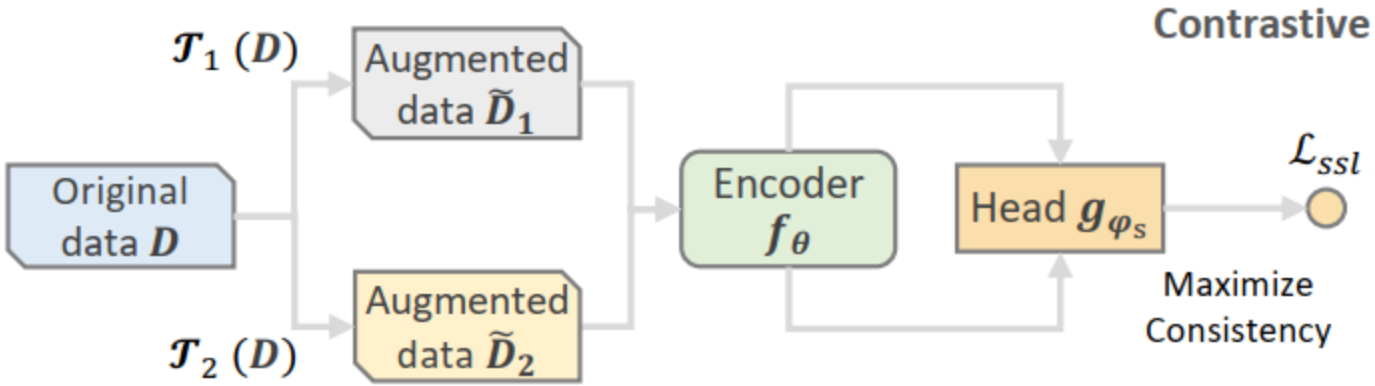




Taxonomy of SSR

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Contrastive Methods



$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}(f_\theta(\tilde{\mathcal{D}}_1), f_\theta(\tilde{\mathcal{D}}_2))\right)$$

Estimates the mutual information between views.

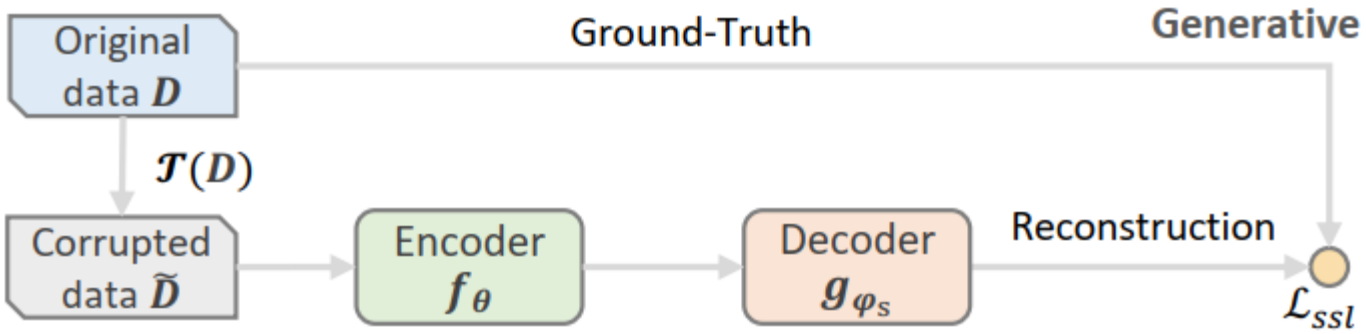
Views are created by imposing different transformations on the original data.

To treat every instance (e.g., user/item/sequence) as a class, and then pull views of the same instance closer in the embedding space, and push views of different instances apart.

Taxonomy of SSR

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Generative Methods



$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}\left(f_\theta(\tilde{\mathcal{D}})\right), \mathcal{D}\right)$$

Learn to predict a portion of the available data from the rest.

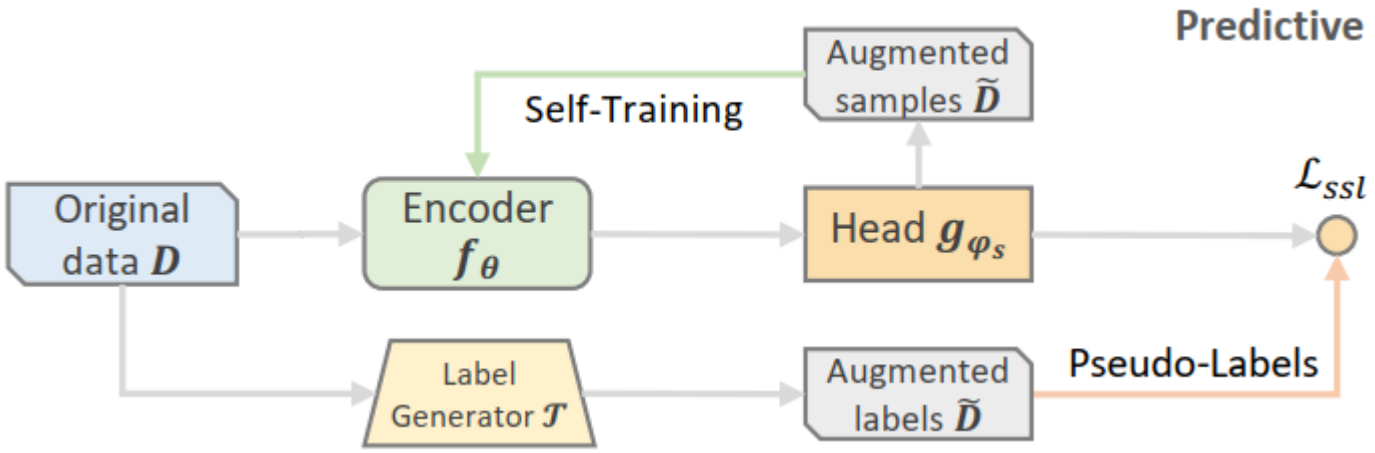
Corrupted data

To reconstruct the original user/item profile with its corrupted versions.

Example: BERT-based Models

Taxonomy of SSR

Predictive Methods



$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}\left(f_\theta(\mathcal{D})\right), \tilde{\mathcal{D}}\right)$$

Predict the augmented pseudo-labels.

Augmented pseudo-labels

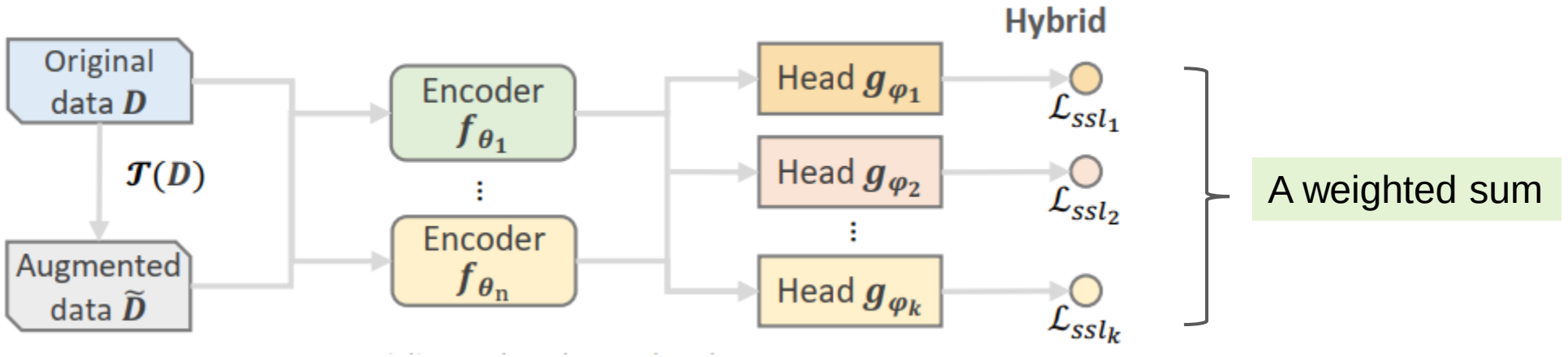
To generate new samples or labels from the original data in order to guide the pretext task.

Taxonomy of SSR

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Hybrid Methods

More than one encoder and projection head are needed.

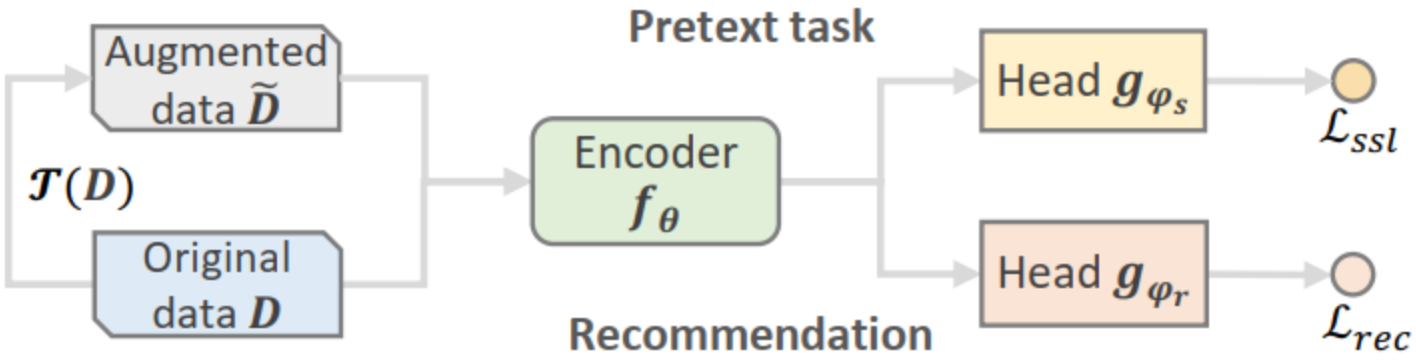


To obtain comprehensive self-supervision by combining different pretext tasks and integrating them into one recommendation model.

Typical Training Schemes

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Joint Learning (JL) mostly used in contrastive methods.



The pretext task and the recommendation task are jointly optimized with a shared encoder.

$$f_{\theta^*}, g_{\phi_r^*}, \mathbf{H}^* = \arg\min_{f_\theta, g_\phi} [\mathcal{L}_{rec}(g_{\phi_r}(f_\theta(\mathcal{D}))) + \alpha \mathcal{L}_{ssl}(g_{\phi_s}(f_\theta(\tilde{\mathcal{D}})))]$$

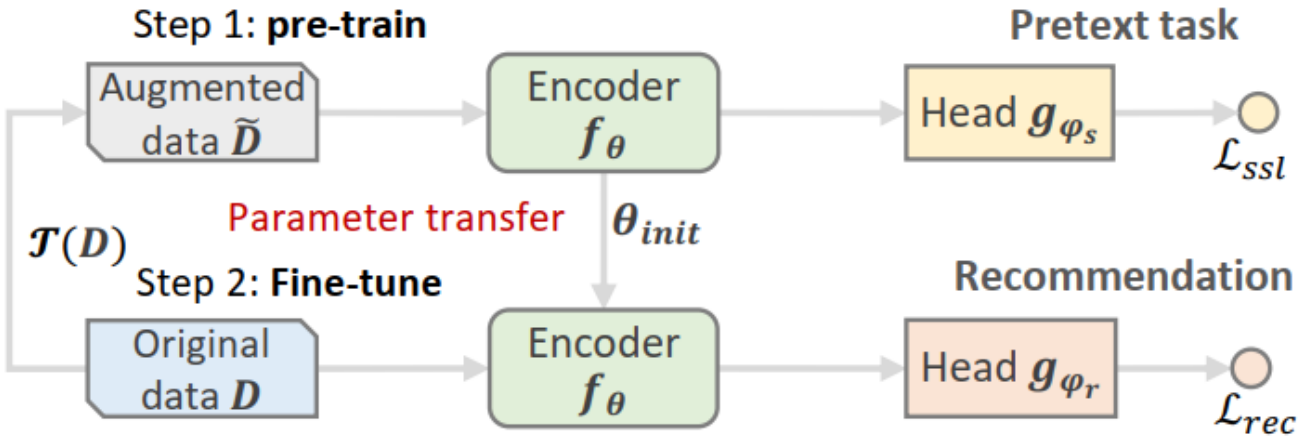
Recommendation: primary task

SSL: auxiliary task, not concerned

control the magnitude of self-supervision

Typical Training Schemes

Pre-training and Fine-tuning (PF) used in most of generative methods and a few contrastive methods.



The encoder is firstly pretrained with pretext tasks on the augmented data for a good initialization of the encoder's parameters. It is then is fine-tuned on the original data for recommendation

$$f_{\theta_{init}} = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}(f_\theta(\tilde{\mathcal{D}}), \mathcal{D})\right)$$

Pre-train the encoder

$$f_{\theta^*}, g_{\phi_r^*}, \mathbf{H}^* = \arg\min_{f_{\theta_{init}}, g_{\phi_r}} \mathcal{L}_{rec}\left(g_{\phi_r}(f_\theta(\mathcal{D}))\right)$$

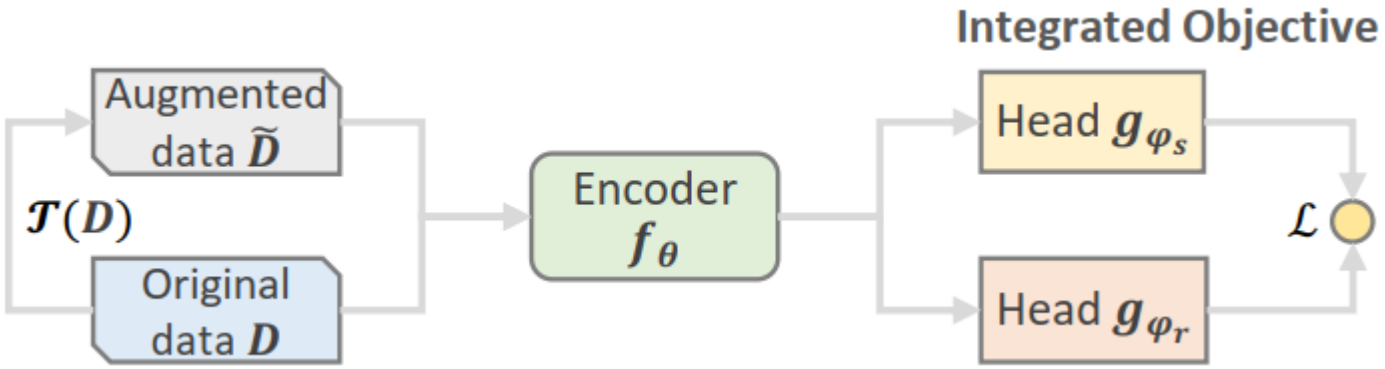
Fine-tune the encoder for recommendation

Typical Training Schemes

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Integrated Learning (IL)

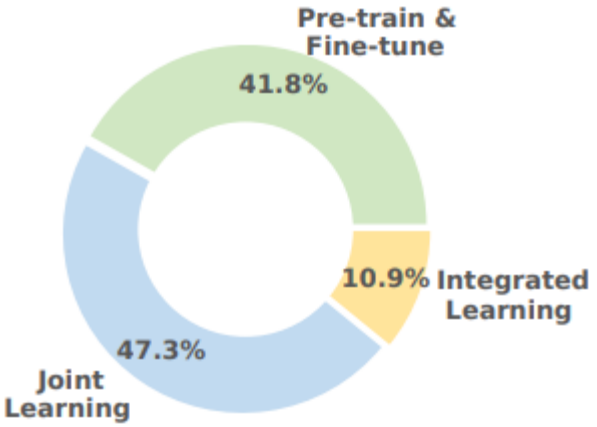
mainly used by the pseudo-labels-based predictive methods and a few contrastive methods.



The pretext task and the recommendation task are well-aligned under this setting and they are unified into an integrated objective.

$$f_{\theta^*}, g_{\phi_r^*}, \mathbf{H}^* = \arg\min_{f_\theta, g_\phi} \mathcal{L}\left(g_{\phi_r}(f_\theta(\mathcal{D})), g_{\phi_s}(f_\theta(\tilde{\mathcal{D}}))\right)$$

Integrated loss



THE WEB CONFERENCE ACM

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

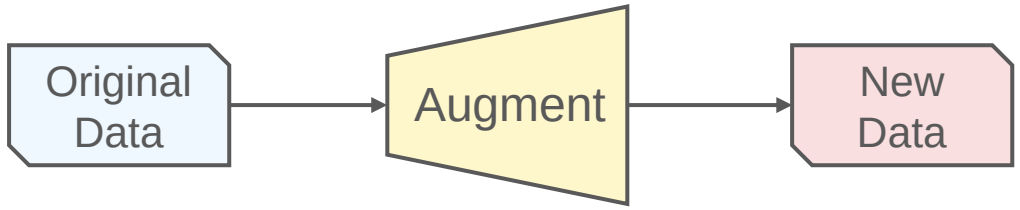
03

Data Augmentation

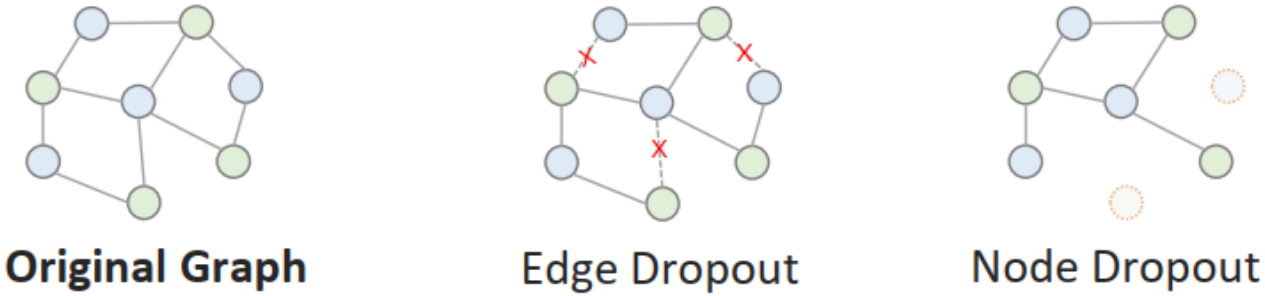
Tutorial

Data Augmentation

- Data augmentation increases the amount of training data by creating plausible variations of existing data.
- It plays a pivotal role in learning quality and generalizable representations.
- In SSR, the most commonly used data augmentation approaches can be divided into three categories: **graph-based**, **sequence-based**, and **feature-based**.



Graph-Based Augmentation



- **Edge/Node Dropout**: With the probability ρ, each edge/node would be removed from the graph.

Idea behind

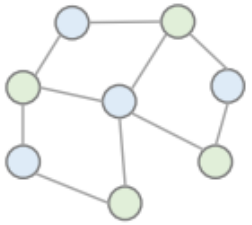
Only partial connections/nodes are informative to learn quality node representations.

Formulation

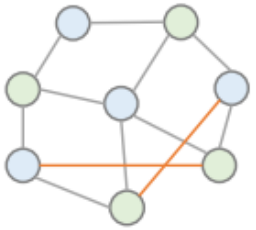
$$\tilde{\mathcal{G}}, \tilde{\mathbf{A}} = \mathcal{T}_{\text{E-dropout}}(\mathcal{G}) = (\mathcal{V}, \mathbf{m} \odot \mathcal{E}) \qquad \mathbf{m} \in (0,1)^{|\mathcal{E}|}$$

$$\tilde{\mathcal{G}}, \tilde{\mathbf{A}} = \mathcal{T}_{\text{N-dropout}}(\mathcal{G}) = (\mathcal{V} \odot \mathbf{m}, \mathcal{E} \odot \mathbf{m}') \qquad \mathbf{m} \in (0,1)^{|\mathcal{V}|}$$

Graph-Based Augmentation



Original Graph



Diffusion

- **Graph Diffusion**: Opposite to the dropout-based method, the diffusion-based augmentation adds edges into the graph to create views.

Idea behind

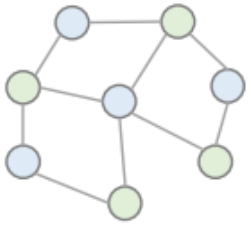
The missing user behaviors include unknown positive preferences which can be represented with weighted user-item edges.

Formulation

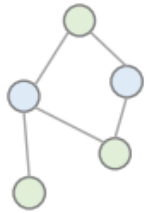
$$\tilde{\mathcal{G}}, \tilde{\mathbf{A}} = \mathcal{T}_{\text{diffusion}}(\mathcal{G}) = (\mathcal{V}, \mathcal{E} + \tilde{\mathcal{E}})$$

Discover the possible edges by calculating the similarities of the user and item representations and retain the edges with top-*K* similarities.

Graph-Based Augmentation



Original Graph



Subgraph Sampling

- **Subgraph Sampling**: samples a portion of nodes and edges by rules to form subgraphs.

Idea behind

The sampled graph reflects the local connectivity and semantics.

Formulation

$$\tilde{\mathcal{G}}, \tilde{\mathbf{A}} = \mathcal{T}_{\text{sampling}}(\mathcal{G}) = (\mathcal{Z} \in \mathcal{V}, \mathbf{A}[\mathcal{Z}, \mathcal{Z}]) \qquad \mathcal{Z} \text{ the sampled node set}$$

A lot of approaches can be used to induce subgraphs like meta-path guided random walks and the ego-network sampling.

Sequence-Based Augmentation

- **Item Masking**: randomly masks a portion of items and replaces them with special tokens [mask].

Idea behind

A user's intention is relatively stable during a period of time. Though part of items are masked, the primary intent is still retained in the rest.

Formulation

$$\tilde{S} = \mathcal{T}_{\text{masking}}(S) = [\tilde{i}_1, \tilde{i}_2, ..., \tilde{i}_k], \tilde{i}_t = \begin{cases} i_t, & t \notin \mathcal{M} \\ [\text{mask}], & t \in \mathcal{M} \end{cases}$$

$\mathcal{M}$ denotes the indices of the masked items.

Sequence-Based Augmentation

- **Item Cropping**: a sub-sequence with a certain length is randomly cropped.

Idea behind

The selected sub-sequence is expected to endow the model with the ability to learn generalized representations without the comprehensive user profile.

Formulation

$$\tilde{S} = \mathcal{T}_{\text{cropping}}(S) = [\tilde{i}_c, \tilde{i}_{c+1}, \ldots, \tilde{i}_{c+L_c-1}]$$

where $L_c = \lfloor \eta * |S| \rfloor$ and $\eta \in (0, 1)$

Sequence-Based Augmentation

- **Item Shuffling**: shuffle a continuous sub-sequence to create sequence augmentations.

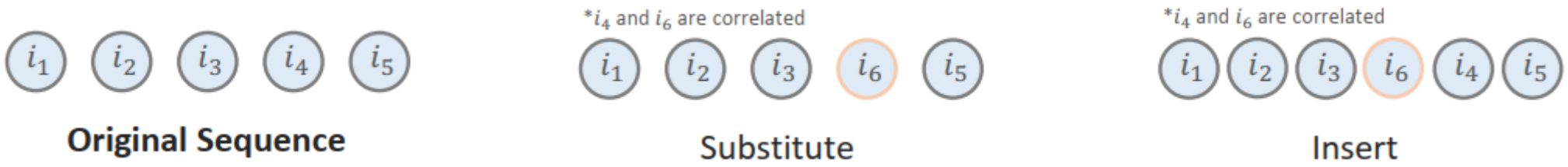
Idea behind

The item order in a sequence is not strict, and different item orders may actually correspond to the same user intent.

Formulation

$$\tilde{S} = \mathcal{T}_{\text{reordering}}(S) = [i_1, \ldots, \tilde{i}_r, \tilde{i}_{r+1}, \ldots, \tilde{i}_{r+L_r-1}, \ldots, i_k]$$

Sequence-Based Augmentation



- **Item Substitution/Insertion**: substitute/insert items in(to) short sequences with highly correlated items

Idea behind

Random item cropping and masking could exaggerate the data sparsity issue in short sequences. Substituting/Inserting highly-correlated items injects less corruptions.

Formulation

$$\tilde{S} = \mathcal{T}_{\text{substitution}}(S) = [\tilde{i}_1, \tilde{i}_2, ..., \tilde{i}_k], \quad \tilde{i}_t = \begin{cases} i_t, & t \notin \mathcal{Z} \\ \text{item correlated to } i_t, & t \in \mathcal{Z} \end{cases}$$

$$\tilde{S} = \mathcal{T}_{\text{insert}}(S) = [i_1, ..., \tilde{i}_{id_1}, i_{id_1}, ..., \tilde{i}_{id_l}, i_{id_l} ..., i_k]$$

Feature-Based Augmentation

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

- **Feature Dropout**: similar to the item masking and edge dropout, which randomly masks/drops a small portion of features.

Idea behind: The whole profile/information can be inferred by a portion of features.

Formulation: $\tilde{\mathbf{X}} = \mathcal{T}_{\text{F-dropout}}(\mathbf{X}) = \mathbf{X} \odot \mathbf{M}$ The matrix $\mathbf{M}$ is generated by Bernoulli distribution.

- **Feature Shuffling**: switches rows and columns in the feature matrix.

Idea behind: By randomly changing the contextual information, the feature matrix is corrupted to yield augmentations.

Formulation: $\tilde{\mathbf{X}} = \mathcal{T}_{\text{shuffling}}(\mathbf{X}) = \mathbf{P}_r \mathbf{X} \mathbf{P}_c$ $\mathbf{P}_r$ and $\mathbf{P}_r$ are permutation matrices

Feature-Based Augmentation

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

- **Feature Mixing**: mix the original features with features from other users/items or previous versions to synthesize informative negative/positive examples .

Idea behind: Mixing related features can generate hard examples.

Formulation:

$$\tilde{\mathbf{x}}_i = \mathcal{T}_{\text{mixing}}(\mathbf{x}_i) = \alpha \mathbf{x}_i + (1-\alpha)\mathbf{x}_j', \quad \alpha \in [0,1]$$

- **Feature Clustering**: the augmented prototype representations can be learned via clustering algorithms.

Idea behind: Assume that there are prototypes in the feature/representation space and user/item representations should be closer to their assigned prototypes.

Formulation:

$$\tilde{\mathbf{C}} = \mathcal{T}_{\text{clustering}}(\mathbf{X}) = \text{EM}(\mathbf{X}, \mathcal{C}),$$

where $\mathcal{C}$ is the presupposed clusters (prototypes) and $\tilde{\mathbf{C}}$ is the augmented prototype representations.

THE WEB CONFERENCE ACM

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

04

SSR Methods

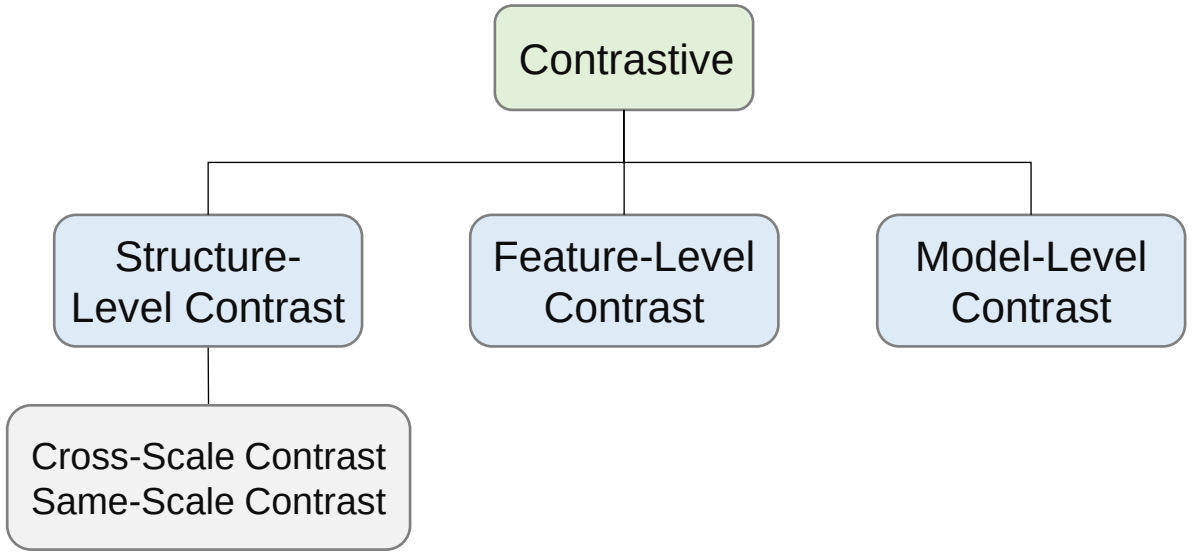
- Contrastive, Generative, Predictive, and Hybrid

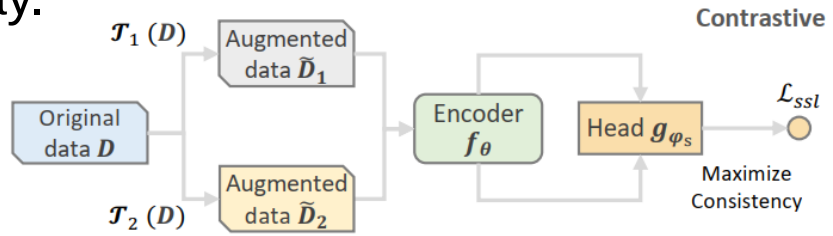
Tutorial

Contrastive SSR Methods

Contrastive Methods

Pull similar samples closer and Push dissimilar samples away.





$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}(f_\theta(\tilde{\mathcal{D}}_1), f_\theta(\tilde{\mathcal{D}}_2))\right)$$

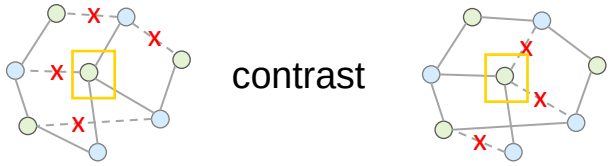
According to where the self-supervision signals come from, we divide them into three categories: **structure-level contrast**, **feature-level contrast**, and **model-level contrast**.

Structure-Level Contrast

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Same-Scale

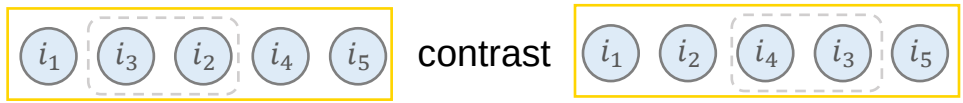
Local-Local



$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\left(g_{\phi_s}(\tilde{\mathbf{h}}_i, \tilde{\mathbf{h}}_j)\right)$$

$\tilde{\mathbf{h}}_i$ and $\tilde{\mathbf{h}}_j$ are node representations

Global-Global



$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\left(g_{\phi_s}(\mathrm{Agg}(f_\theta(\tilde{S}_i)), \mathrm{Agg}(f_\theta(\tilde{S}_j)))\right)$$

$\tilde{S}_i$ and $\tilde{S}_j$ are two sequence augmentations

Cross-Scale

Local-Global

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\left(g_{\phi_s}(\tilde{\mathbf{h}}, \mathcal{R}(f_\theta(\tilde{\mathcal{G}}, \tilde{\mathbf{A}})))\right)$$

$\mathcal{R}$ is the readout function that generates global-level graph representation.

Local-Context

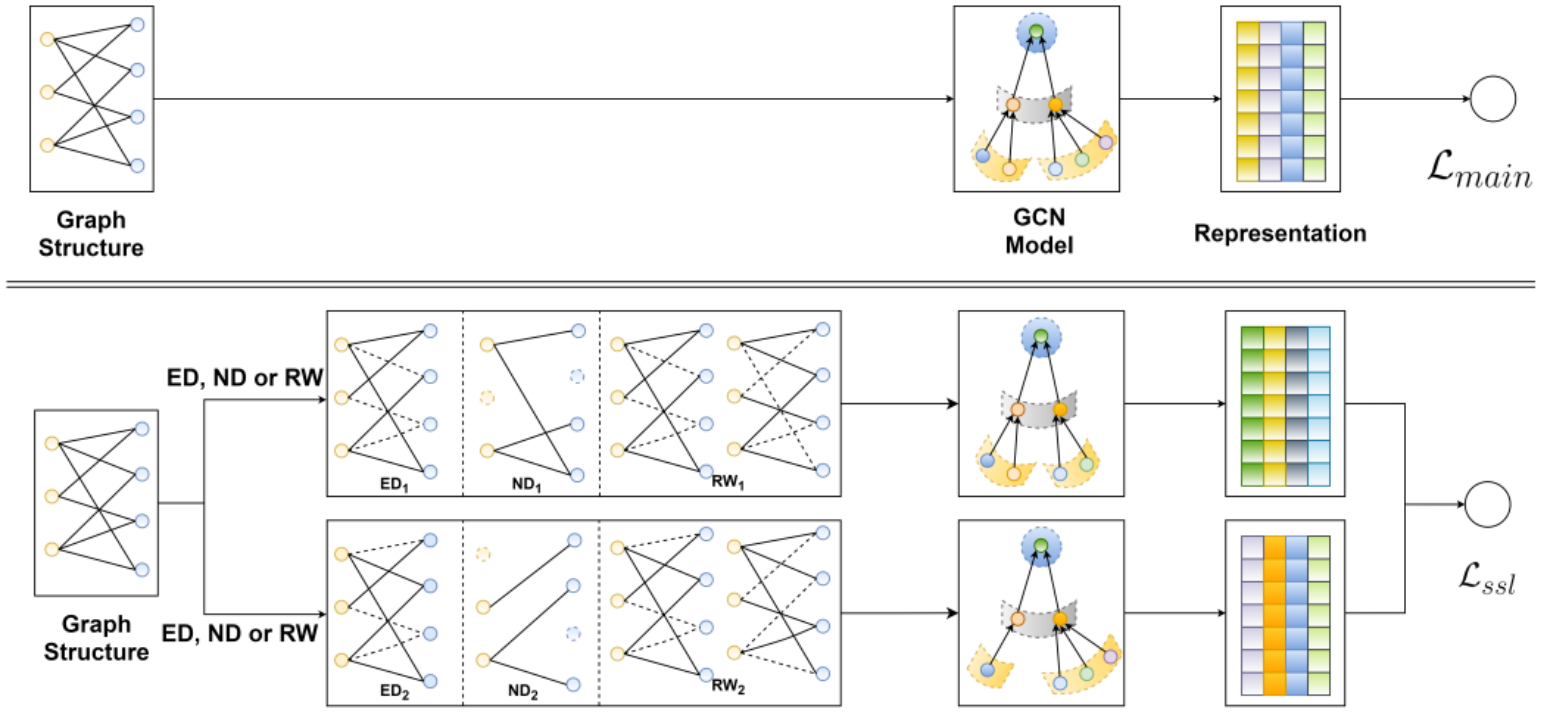
$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\left(g_{\phi_s}(\mathbf{h}_i, \mathcal{R}(f_\theta(\mathcal{C}_j)))\right)$$

$\mathcal{C}_j$ denotes the context of node (sequence) j.

Structure-Level Contrast

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Local-Local Contrast: SGL (Wu et al. 2021)



BPR Loss

$$\mathcal{L}_{main} = \sum_{(u,i,j) \in O} -\log \sigma(\hat{y}_{ui} - \hat{y}_{uj})$$

InfoNCE Loss

$$\mathcal{L}_{ssl}^{user} = \sum_{u \in \mathcal{U}} -\log \frac{\exp(s(\mathbf{z}'_u, \mathbf{z}''_u)/\tau)}{\sum_{v \in \mathcal{U}} \exp(s(\mathbf{z}'_u, \mathbf{z}''_v)/\tau)}$$

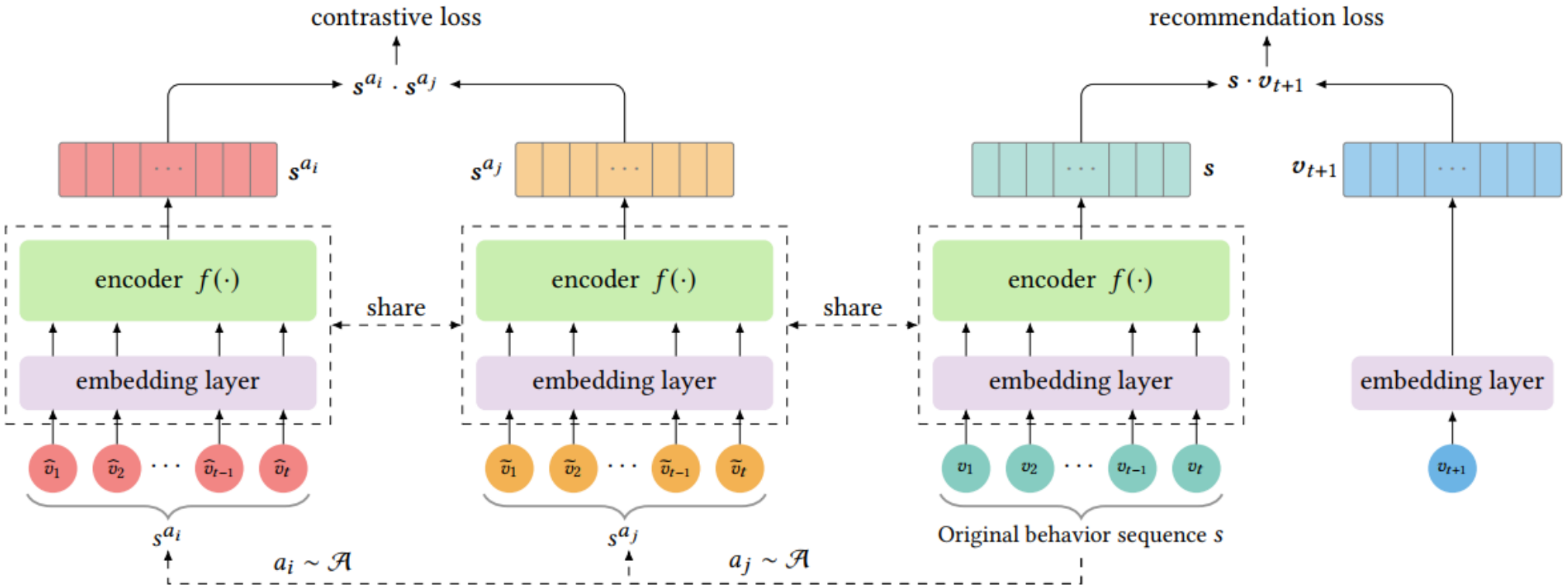
$$\mathcal{L} = \mathcal{L}_{main} + \lambda_1 \mathcal{L}_{ssl} + \lambda_2 \|\Theta\|_2^2$$

J. Wu, X Wang et al. Self-supervised Graph Learning for Recommendation. SIGIR'21

Structure-Level Contrast

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Global-Global Contrast: CL4SRec (Xie et al. 2021)



$$\mathcal{L}_{\text{cl}}(s_u^{a_i}, s_u^{a_j}) = -\log \frac{\exp\left(\text{sim}(s_u^{a_i}, s_u^{a_j})\right)}{\exp\left(\text{sim}(s_u^{a_i}, s_u^{a_j})\right) + \sum_{s^- \in S^-} \exp\left(\text{sim}(s_u^{a_i}, s^-)\right)}$$

$$\mathcal{L}_{\text{main}}(s_{u,t}) = -\log \frac{\exp\left(s_{u,t}^\top v_{t+1}^+\right)}{\exp\left(s_{u,t}^\top v_{t+1}^+\right) + \sum_{v_{t+1}^- \in \mathcal{V}^-} \exp\left(s_{u,t}^\top v_{t+1}^-\right)}$$

X, Xie et al. Contrastive Learning for Sequential Recommendation. arXiv'21

Feature-Level Contrast

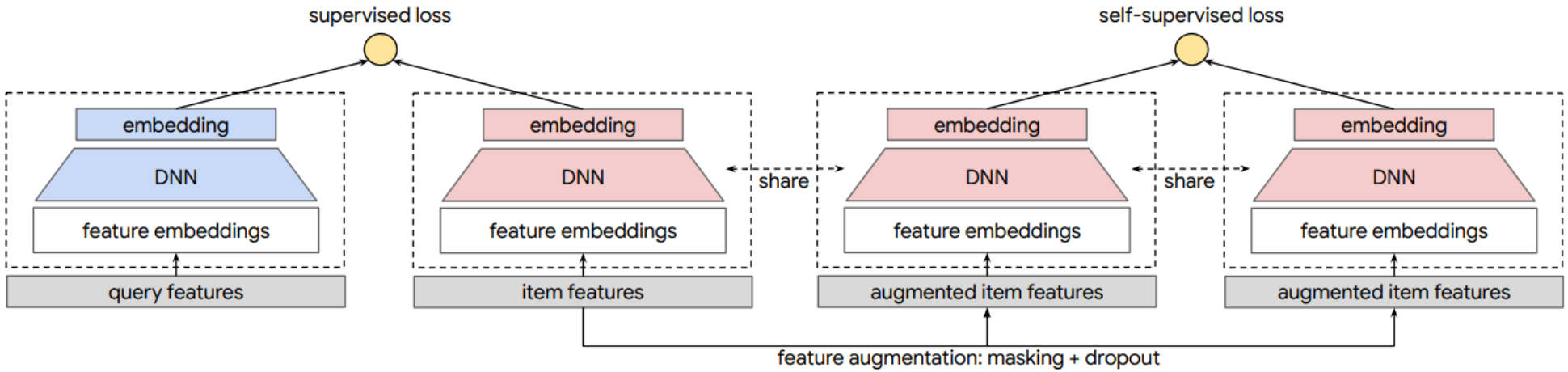
THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Formulation

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\left(g_{\phi_s}\left(f_\theta(\tilde{\mathbf{x}}_i), f_\theta(\tilde{\mathbf{x}}_j)\right)\right)$$

$\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ are feature-level augmentations which can be obtained by modifying the input feature

SSL4ItemRec (Yao et al. 2022)



T, Yao et al. Self-supervised Learning for Large-scale Item Recommendations. CIKM'21

Model-Level Contrast

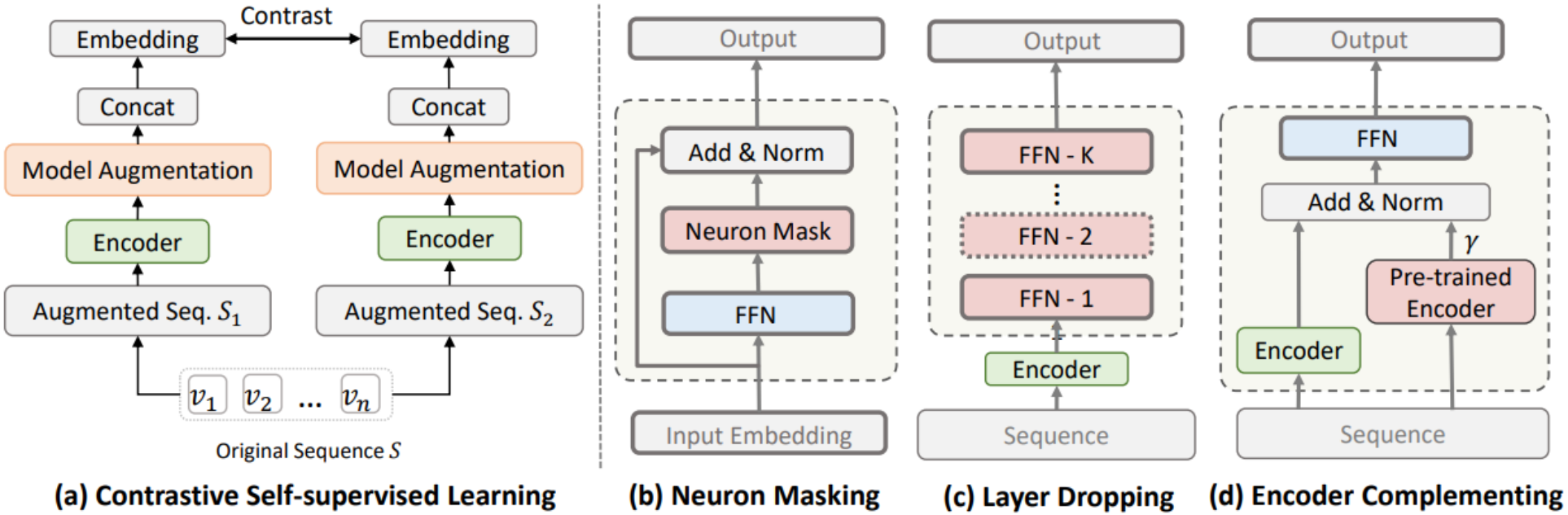
THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Formulation

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\left(g_{\phi_s}\left(f_{\theta'}(\mathcal{D}), f_{\theta''}(\mathcal{D})\right)\right)$$

$f_{\theta'}$ and $f_{\theta''}$ are perturbed versions of f_θ.

SRMA (Liu et al. 2022)



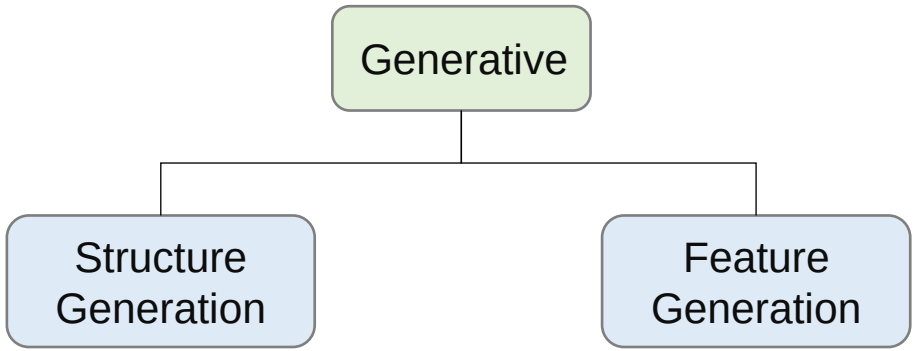
Z, Liu et al. Improving Contrastive Learning With Model Augmentation. arXiv'22

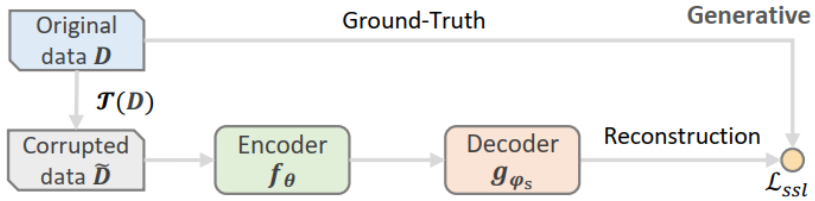
Generative SSR Methods

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Generative Methods

Reconstructing the original input with its corrupted version.





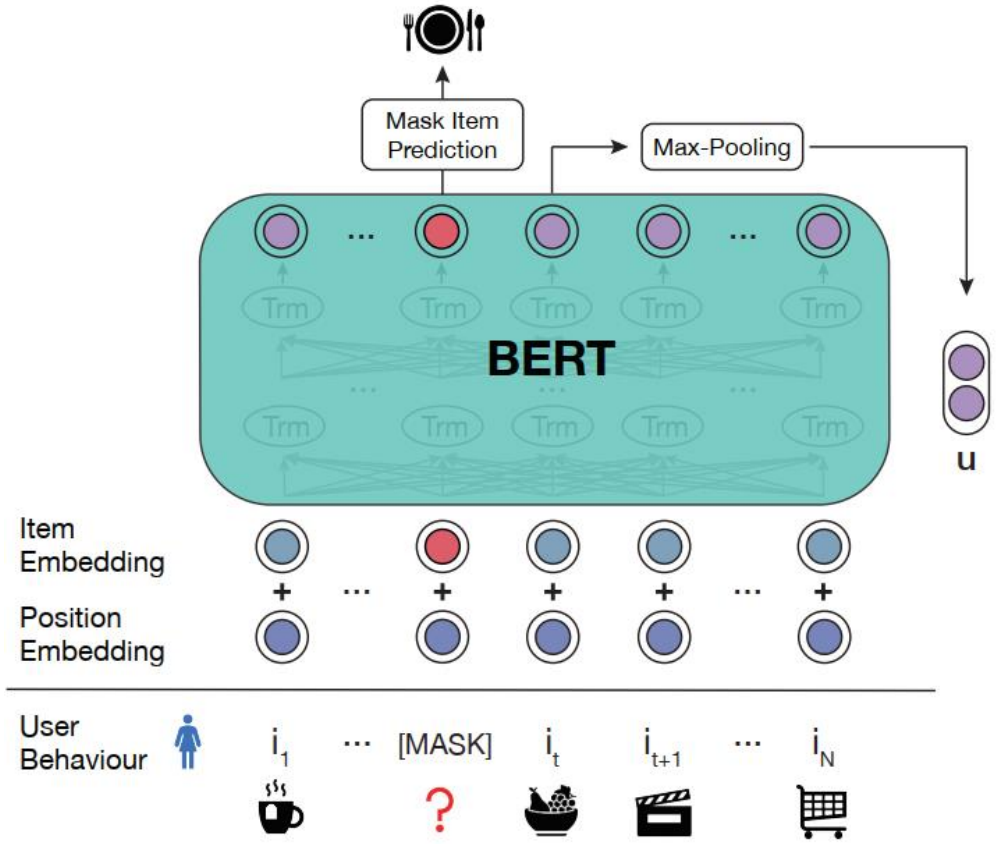
$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}\left(f_\theta(\tilde{\mathcal{D}})\right), \mathcal{D}\right)$$

According to the reconstruction objectives, generative SSR methods are divided into two categories: **Structure Generation** and **Feature Generation**.

Structure Generation

Sequence Reconstruction

UPRec (Xiao et al. 2022)



Pre-trained with the masked-item-prediction method.

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}\left(f_\theta(\tilde{S})\right), S\right)$$

$$\mathcal{L} = \frac{1}{|S_u^m|} \sum_{i_m \in S_u^m} -\log P\left(i_m = i_m^* \mid \tilde{S}_u\right)$$

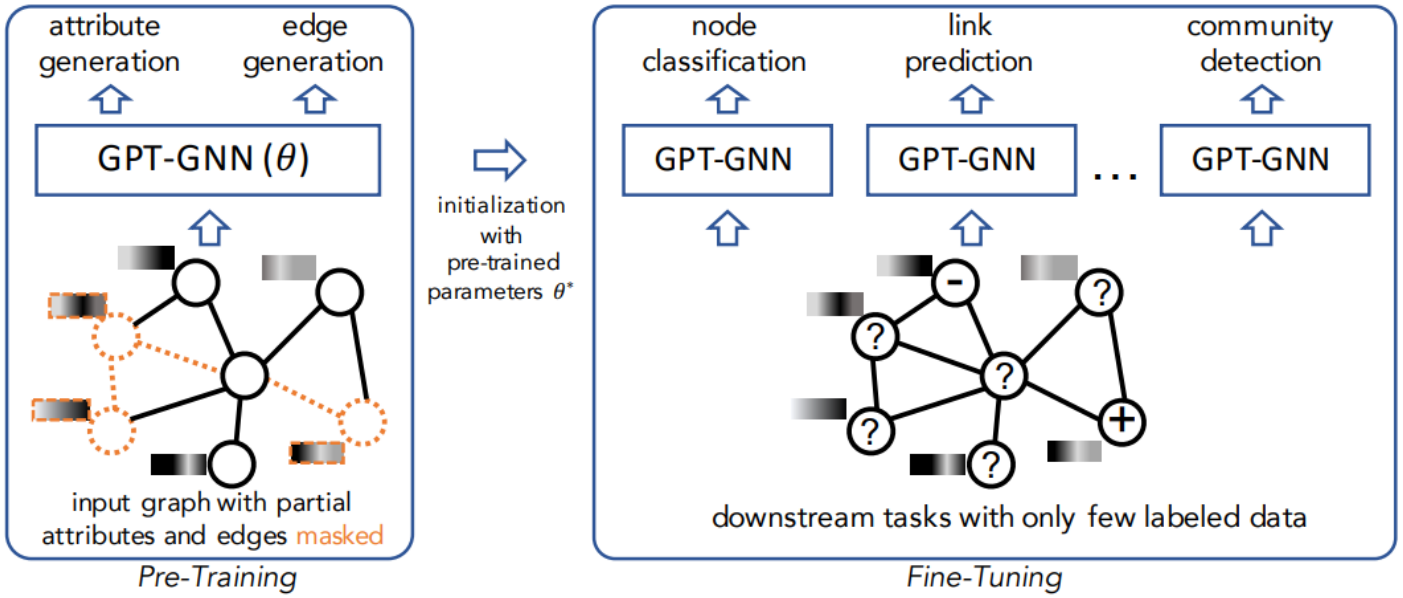
Fine-tuned by predicting the last item of each sequence.

C, Xiao et al. UPRec: User-Aware Pre-training for Recommender Systems. arXiv'21

Structure Generation

Graph Reconstruction

GPT-GNN (Hu et al. 2020)



$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}\left(f_\theta(\tilde{\mathcal{G}})\right), \mathbf{A}\right)$$

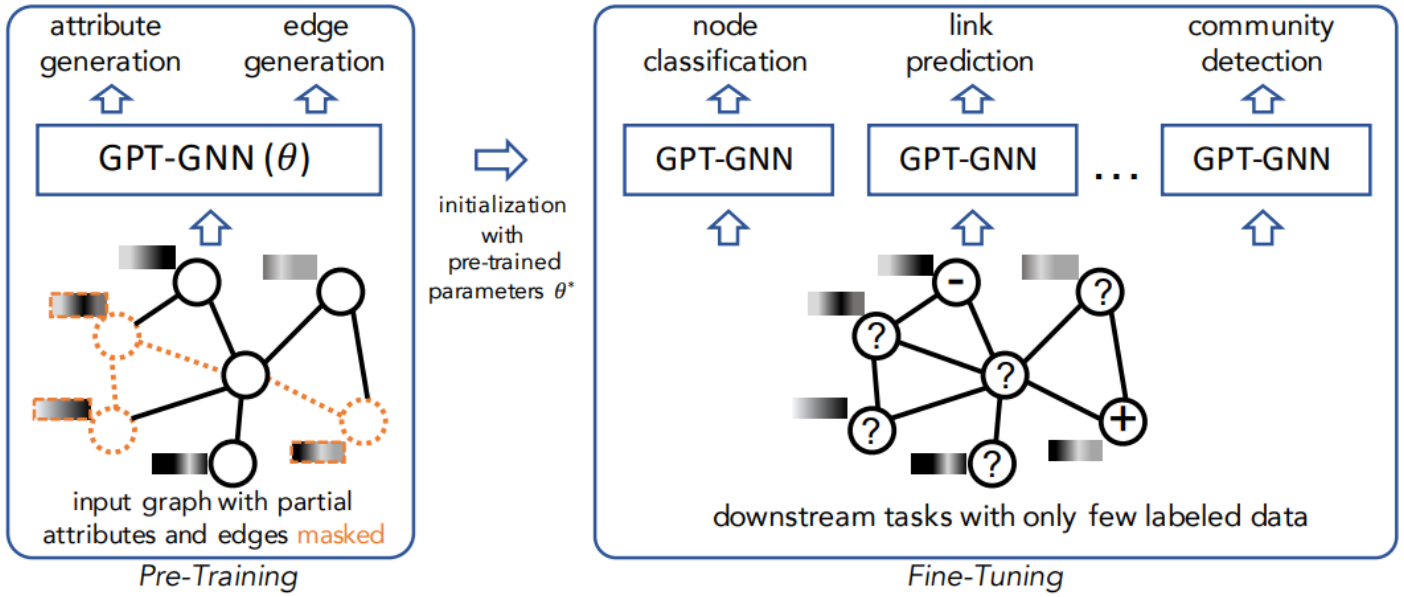
A GNN is pre-trained with the self-supervised generative task — structure generations. Second, the pretrained model and its parameters are then used to initialize models for downstream tasks on the input graph or graphs of the same domain such as recommendation.

Z, Hu et al. GPT-GNN: Generative Pre-Training of Graph Neural Networks. KDD'20

Feature Generation

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

GPT-GNN (Hu et al. 2020)



$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \left\| g_{\phi_s}\left(f_\theta(\tilde{\mathcal{D}})\right) - \mathbf{X} \right\|^2$$

A GNN is pre-trained with the self-supervised generative task — feature generations. Second, the pretrained model and its parameters are then used to initialize models for downstream tasks on the input graph or graphs of the same domain such as recommendation.

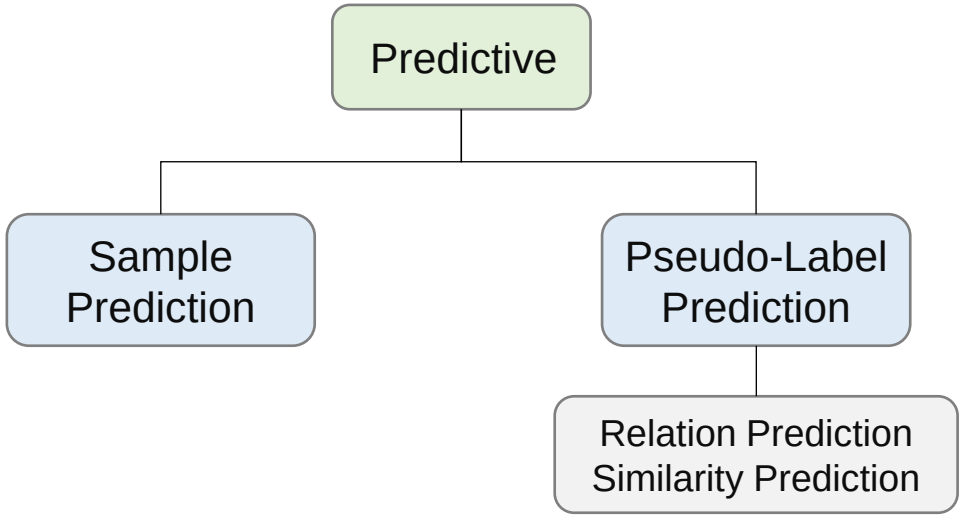
Z, Hu et al. GPT-GNN: Generative Pre-Training of Graph Neural Networks. KDD'20

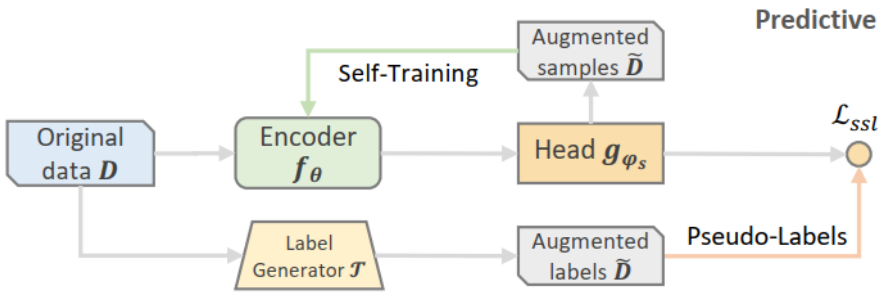
Predictive SSR Methods

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Predictive Methods

Deal with the self-generated supervisory signals obtained from the complete original data.



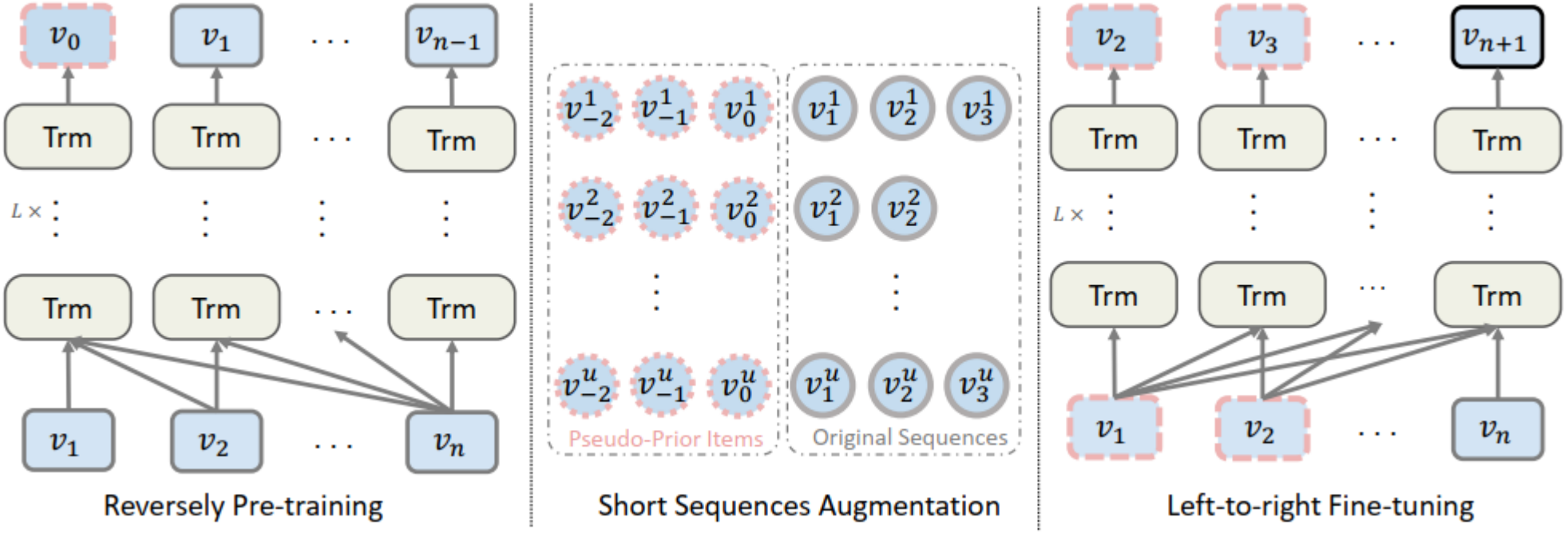


$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}\left(f_\theta(\mathcal{D})\right), \tilde{\mathcal{D}}\right)$$

According to what the predictive pretext task predicts, predictive methods can be categorized into two branches: **Sample Prediction** and **Pseudo-Labels Prediction**.

Sample Prediction

ASReP (Liu et al. 2021)



Firstly pre-train a transformer in a reverse direction to predict prior items.

Use this transformer to generate fabricated historical items at the beginning of short sequences

Fine-tune the transformer using these augmented sequences from the time order to predict the next item.

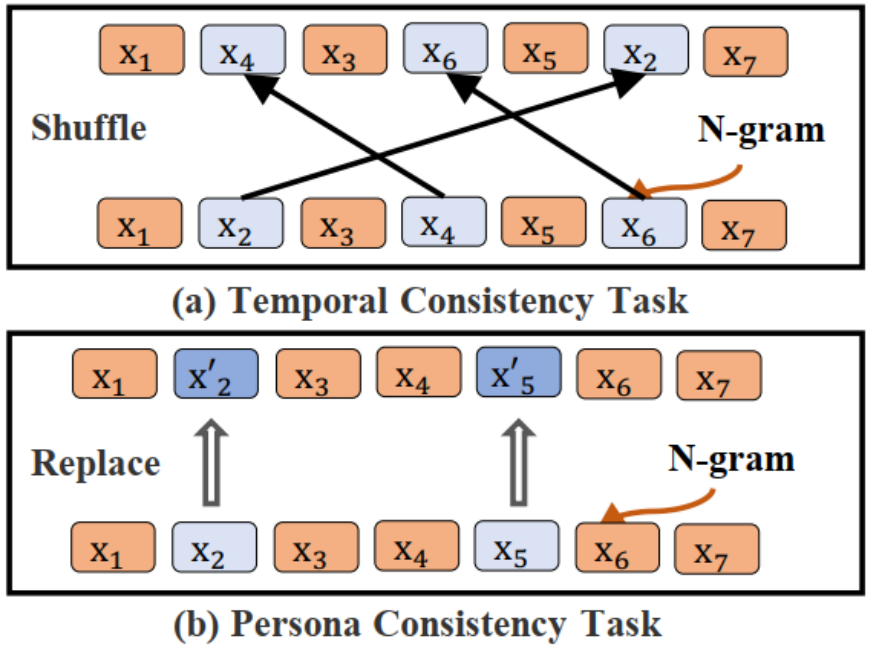
Z, Liu et al. Augmenting Sequential Recommendation with Pseudo-Prior Items via Reversely Pre-training Transformer. SIGIR'21

Pseudo-Label Prediction

Relation Prediction

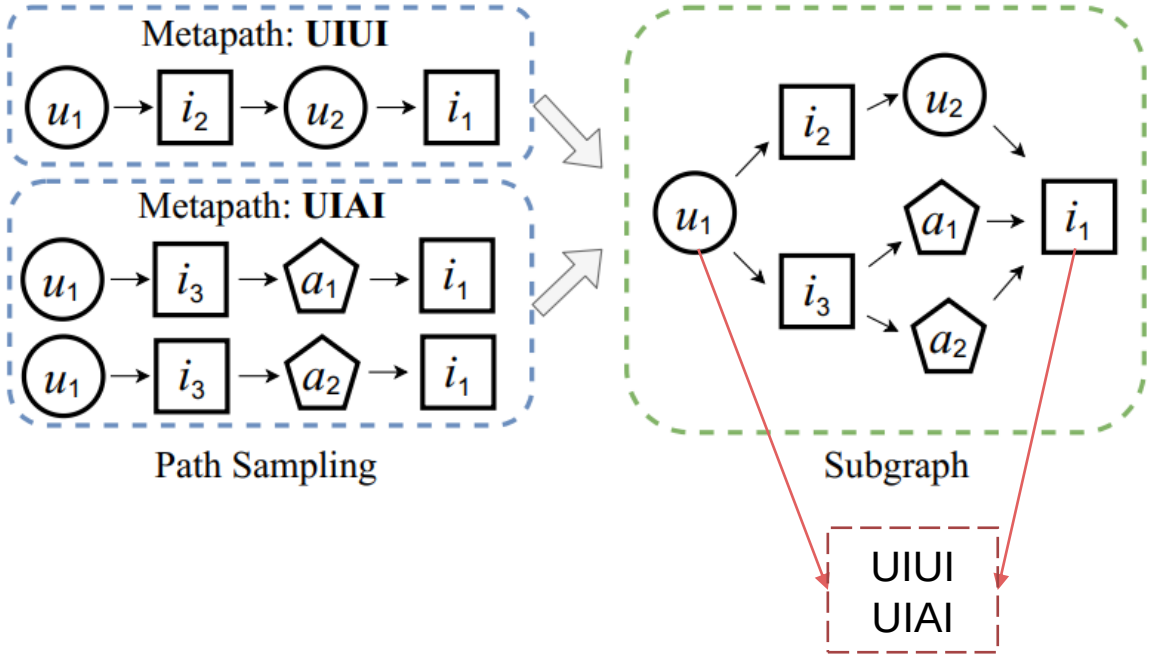
THE UNIVERSITY OF QUEENSLAND AUSTRALIA

SSI (Song et al. 2021)



Shuffle/replace a portion of items in a given sequence, and then predicts if the modified sequence is in the original order/from the same user.

CHEST (Wang et al. 2021)



Predict if there exists a path instance of a specific meta-path between a user-item pair.

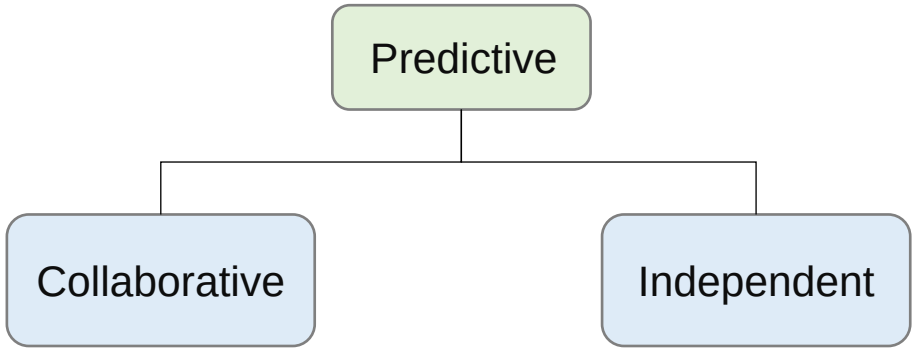
Y, Song et al. Improving Sequential Recommendation Consistency with Self-Supervised Imitation. IJCAI'21
H, Wang et al. Curriculum Pre-Training Heterogeneous Subgraph Transformer for Top-*N* Recommendation. arXiv'21

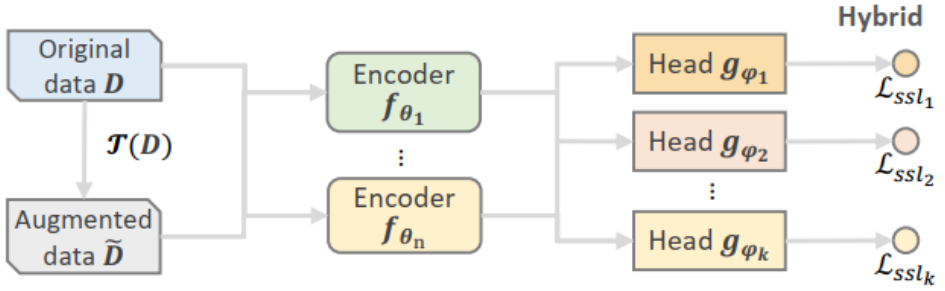
Hybrid SSR Methods

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Hybrid Methods

Assemble multiple types of pretext tasks to take advantage of different self-supervision signals.



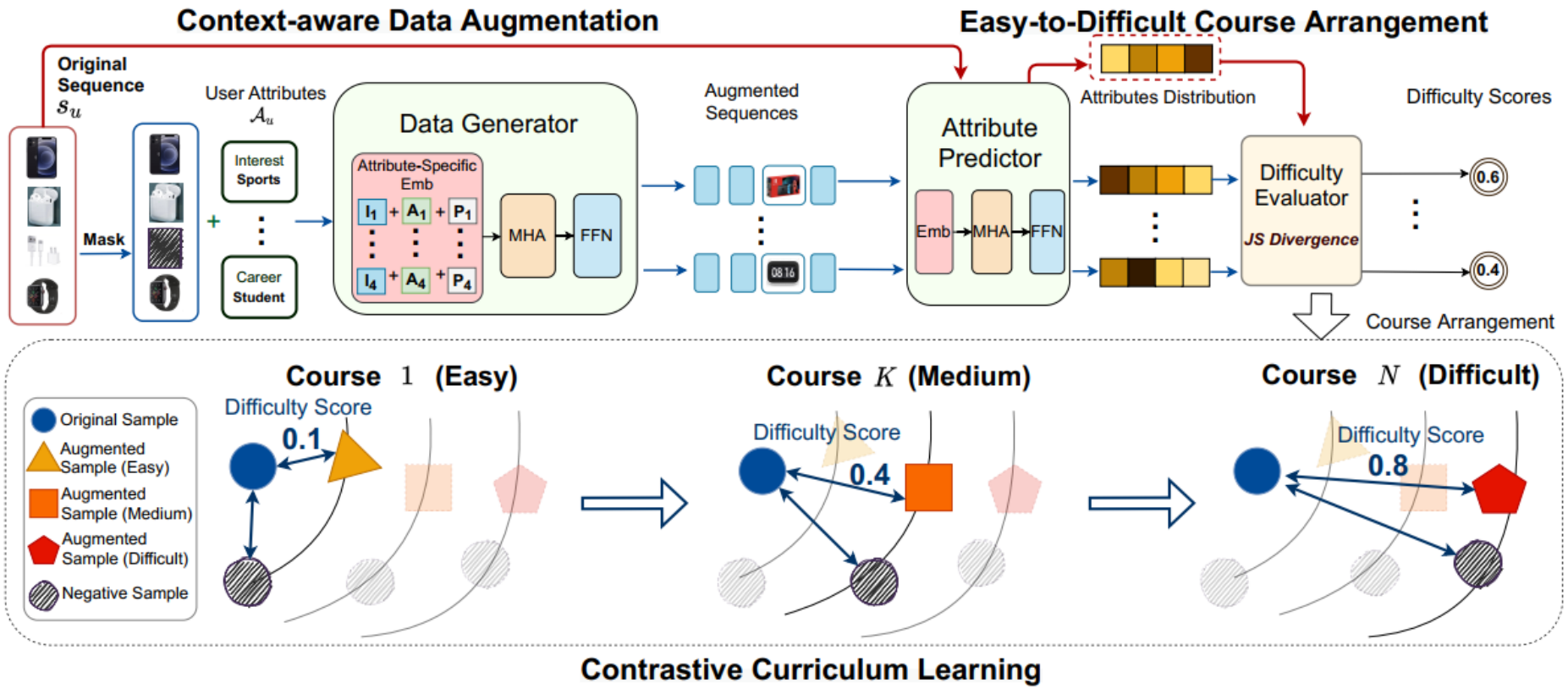


Collaborative: Different pretext tasks collaborate in a way to serve one primary pretext task.

Independent: There are no correlations between different pretext tasks and they work independently.

Collaborative Methods

CCL (Bian et al. 2021)



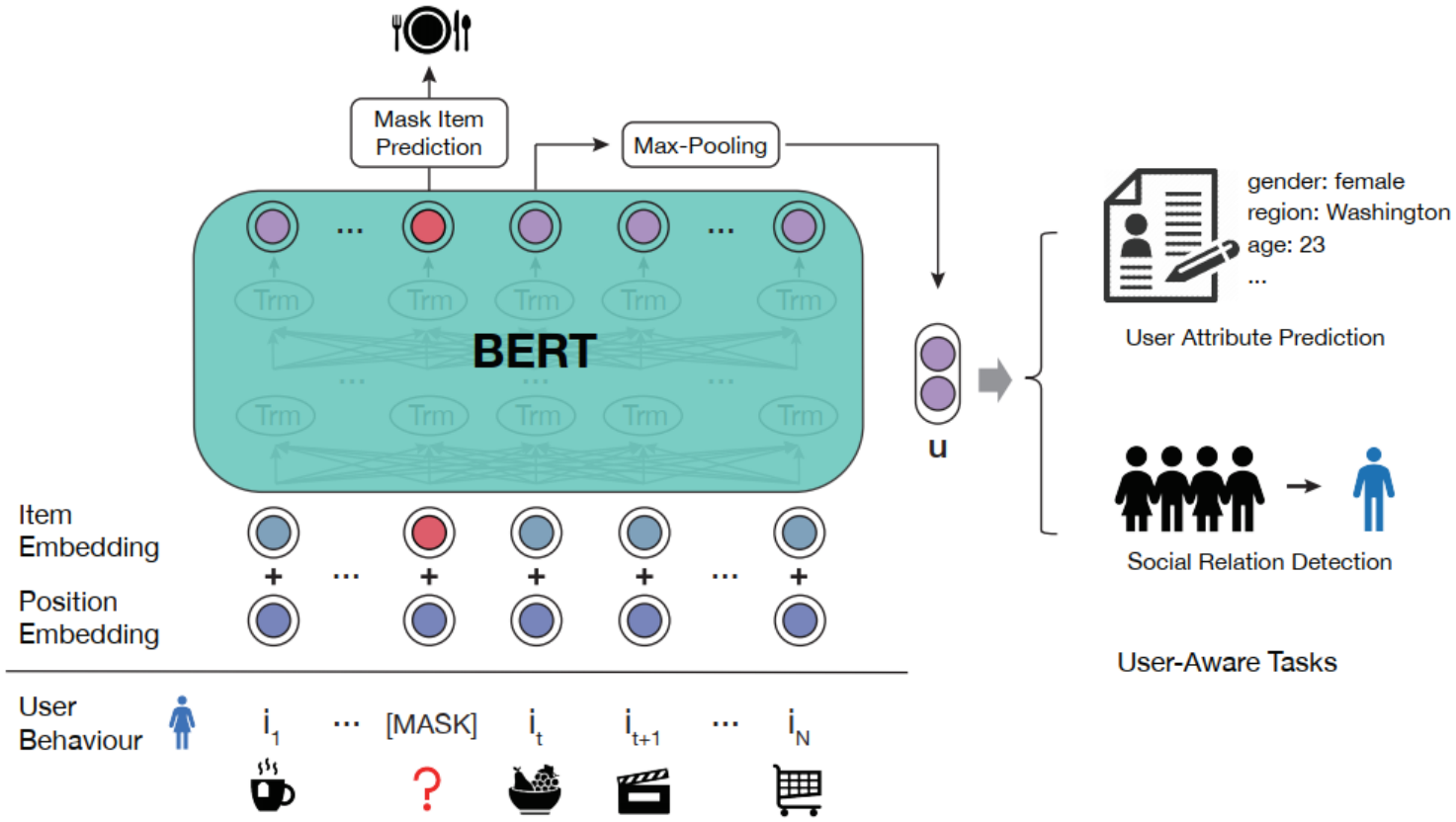
The generative task serves the contrastive task.

S, Bian et al. Contrastive Curriculum Learning for Sequential User Behavior Modeling via Data Augmentation. CIKM'21

Parallel Methods

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

UPRec (Xiao et al. 2022)



One Generative Task
Two Predictive Tasks

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{MIP}} + \lambda_2 \mathcal{L}_{\text{UAP}} + \lambda_3 \mathcal{L}_{\text{SRD}}$$

C, Xiao et al. UPRec: User-Aware Pre-training for Recommender Systems. arXiv'21

Pros and Cons

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Contrastive

Pros

Flexible to design augmentations and pretext tasks.

Cons

Often compromised by low-quality augmentations.

Generative

Pros

Can follow the successful experience for training masked language models

Cons

Confronted with heavy computation when building general-purpose models.

Predictive

Pros

Acquire samples and pseudo-labels in more dynamic and flexible ways.

Cons

Collect the pseudo-labels based on heuristics, without assessing how relevant these labels and predictive tasks are to recommendation.

Hybrid

Pros

Can get enhanced and comprehensive self-supervision.

Cons

Confronted with the problem of coordinating multiple pretext tasks.

THE WEB CONFERENCE ACM

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

05

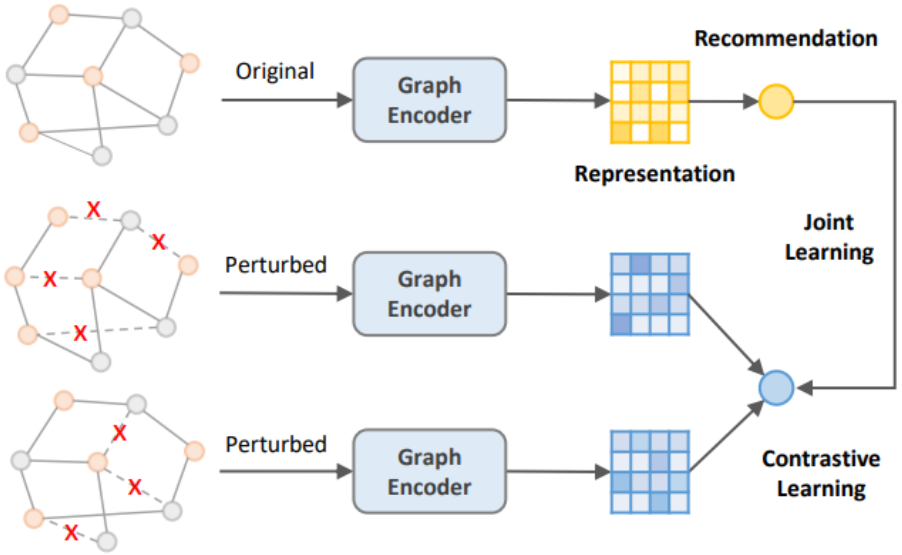
Our Work

- Towards Simple Graph Contrastive Recommendation

Tutorial

Preliminaries

Graph Contrastive Learning (GCL)



A typical way to apply CL to recommendation

Step 1. Augment the user-item graph with structure perturbation.

Step 2. Maximize the consistency of representations under different views.

Optimization Setting. The CL task acts as the auxiliary task, and is jointly optimized with the recommendation task.

Motivation

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Why is CL effective?

It is assumed that contrasting different graph augmentations can capture the essential information existing in the original user-item interactions.

However, it is reported that even extremely sparse graph augmentations in CL can bring desired performance gains.

A meaningful question arises: ***Do we really need graph augmentations when integrating CL with recommendation?***

Revisit Graph CL in Recommendation

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Model SGL (Wu et al. 2021)

SGL employs an effective graph encoder LightGCN (He et al. 2020) as its backbone, and performs node and edge dropout to augment the original graph and adopts InfoNCE (Oord et al. 2018) to optimize the CL task.

Joint learning scheme in SGL,

$$\mathcal{L}_{joint} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{cl}$$

InfoNCE in SGL is formulated as

$$\mathcal{L}_{cl} = \sum_{i \in \mathcal{B}} -\log \frac{\exp(\mathbf{z}_i'^{\top} \mathbf{z}_i'' / \tau)}{\sum_{j \in \mathcal{B}} \exp(\mathbf{z}_i'^{\top} \mathbf{z}_j'' / \tau)}$$

[1]. Wu et al. Self-Supervised Graph Learning for Recommendation, SIGIR 2021.
[2]. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation, SIGIR 2020
[3]. Oord et al. Representation learning with contrastive predictive coding, 2018

Revisit Graph CL in Recommendation

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Necessity of Graph Augmentation

Four variants of SGL are compared. SGL-ND (node dropout), SGL-ED (edge dropout), SGL-RW (random walk), and **SGL-WA** (without augmentation)

Method	Yelp2018		Amazon-Book	
	Recall@20	NDCG@20	Recall@20	NDCG@20
LightGCN	0.0639	0.0525	0.0410	0.0318
SGL-ND	0.0644	0.0528	0.0440	0.0346
SGL-ED	**0.0675**	**0.0555**	**0.0478**	**0.0379**
SGL-RW	0.0667	0.0547	0.0457	0.0356
SGL-WA	0.0671	0.0550	0.0466	0.0373
CL Only	0.0245	0.0190	0.0314	0.0258

For SGL-WA, the CL loss is:

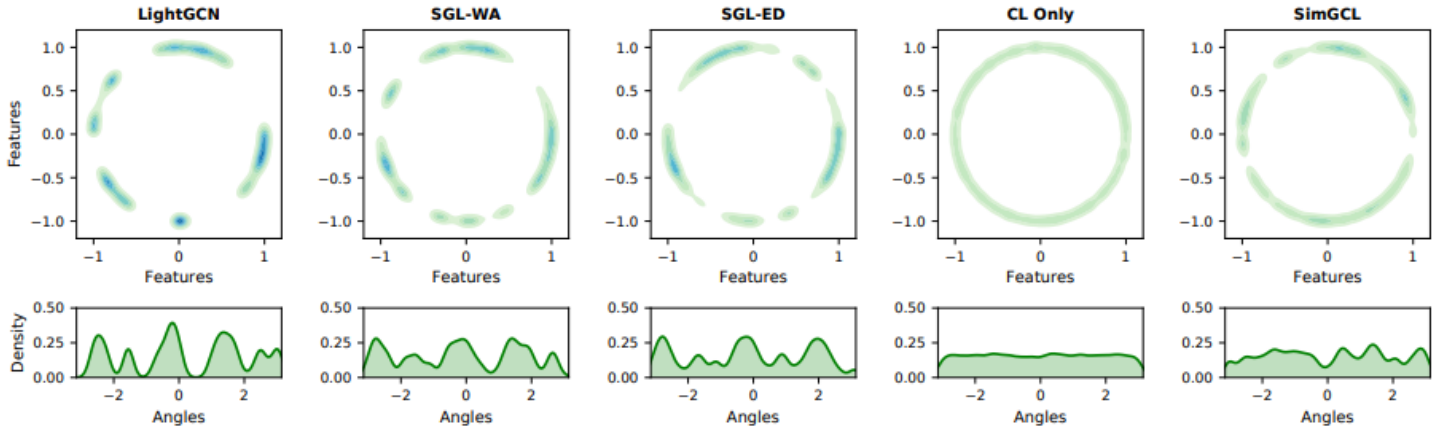
$$\mathcal{L}_{cl} = \sum_{i \in \mathcal{B}} -\log \frac{\exp(1/\tau)}{\sum_{j \in \mathcal{B}} \exp(\mathbf{z}_i^\top \mathbf{z}_j / \tau)}$$

When the graph augmentation is detached, the performance gains are still so remarkable!

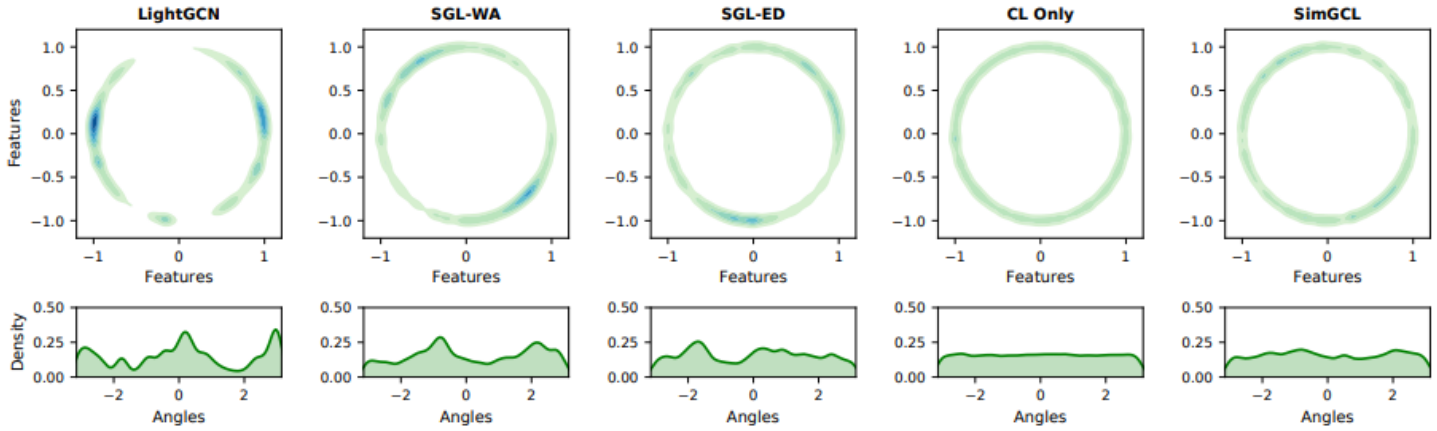
Revisit Graph CL in Recommendation

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Visualization of Representation



(a) Distribution of item representations learned from the dataset of Yelp2018.



(b) Distribution of item representations learned from the dataset of Amazon-Book.

LightGCN shows highly clustered features that mainly reside on some narrow arcs.

For SGL variants, the distributions become more evenly distributed, and the density estimation curves are less sharp, no matter if the graph augmentations are applied.

Revisit Graph CL in Recommendation

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Findings

The uniformity of the distribution is the underlying factor that has a decisive impact on the recommendation performance in SGL, rather than the dropout-based graph augmentations. **The InfoNCE Loss Influences More!**

Optimizing the CL loss is actually minimizing the cosine similarity between different nodes embedding, which can be seen as an implicit way to debias.

SimGCL: Simple Graph Contrastive Learning for Recommendation

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

By adjusting the uniformity of the learned representation in a certain scope, the optimal recommendation performance can be reached.

Manipulating the graph structure for a more evenly-distributed representation space is intractable and time-consuming. We directly add random noises to the representation for an efficient and effective augmentation.

SimGCL: Simple Graph Contrastive Learning for Recommendation

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

Formulation

Given a node i and its representation $\mathbf{e}_i$ in the d-dimensional embedding space, we can implement the following representation-level augmentation:

$$\mathbf{e}_i' = \mathbf{e}_i + \Delta_i', \quad \mathbf{e}_i'' = \mathbf{e}_i + \Delta_i''$$

Two constraints

The added noise vectors Δ_i' and Δ_i'' are subject to $\|\Delta\|_2 = \epsilon$

$$\Delta = \bar{\Delta} \odot \text{sign}(\mathbf{e}_i), \ \bar{\Delta} \in \mathbb{R}^d \sim U(0,1)$$

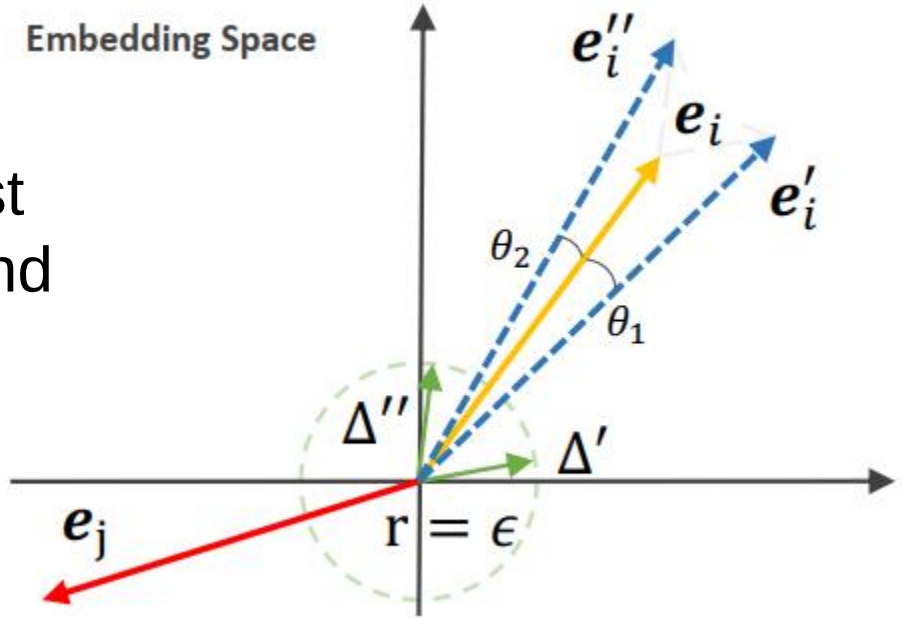
The first constraint controls the magnitude. The second constraint requires that $\mathbf{e}_i$, Δ' and Δ'' should be in the same hyperoctant.

SimGCL: Simple Graph Contrastive Learning for Recommendation

Illustration

By adding the scaled noise vectors to the original representation, we rotate $\mathbf{e}_i$ by two small angles (θ_1 and θ_2). Each rotation leads to an augmented representation ($\mathbf{e}_i'$ and $\mathbf{e}_i''$).

The augmented representation retains most information of the original representation and meanwhile also keeps some variance.



SimGCL: Simple Graph Contrastive Learning for Recommendation

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

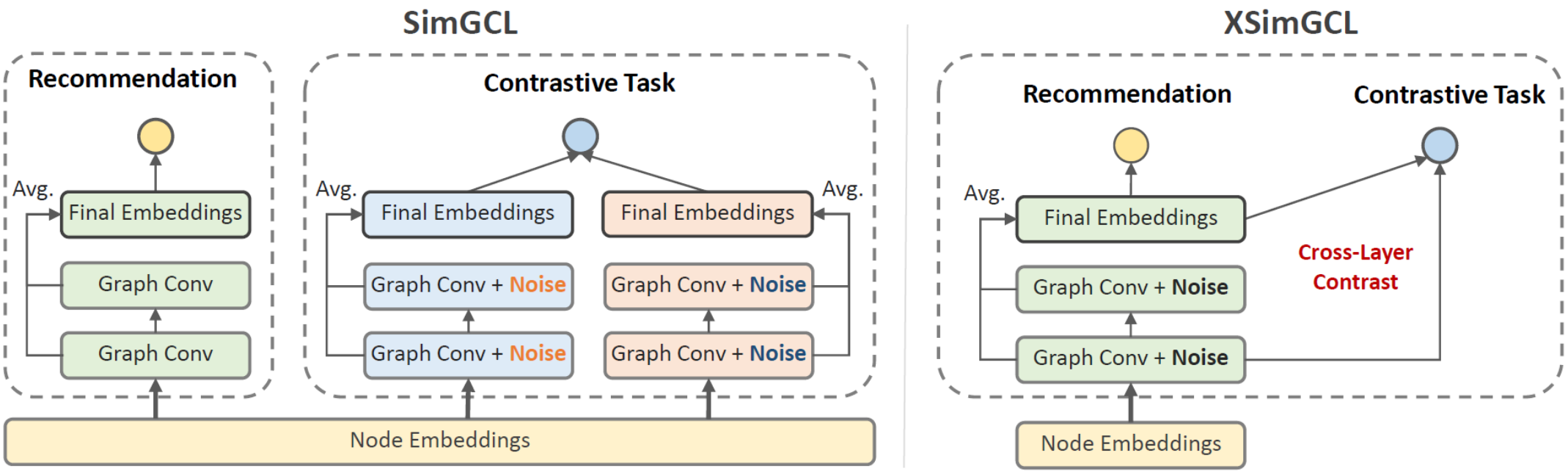
Representation Learning

At each layer, different scaled random noises are imposed on the current node embeddings. The final perturbed node representations are learned by

$$\mathbf{E}' = \frac{1}{L}\big((\tilde{\mathbf{A}}\mathbf{E}^{(0)} + \mathbf{\Delta}^{(1)}) + (\tilde{\mathbf{A}}(\tilde{\mathbf{A}}\mathbf{E}^{(0)} + \mathbf{\Delta}^{(1)}) + \Delta^{(2)})) + \ldots$$
$$+(\tilde{\mathbf{A}}^{L}\mathbf{E}^{(0)} + \tilde{\mathbf{A}}^{L-1}\mathbf{\Delta}^{(1)} + \ldots + \tilde{\mathbf{A}}\mathbf{\Delta}^{(L-1)} + \mathbf{\Delta}^{(L)})\big)$$

LightGCN is adopted as the graph encoder to propagate node information (the initial embedding is skipped).

XSimGCL: Extremely Graph Contrastive Learning for Recommendation



There is a sweet pot when utilizing CL where the mutual information between correlated views is neither too high nor too low.

What if we contrast different layer embeddings?

Experiments

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

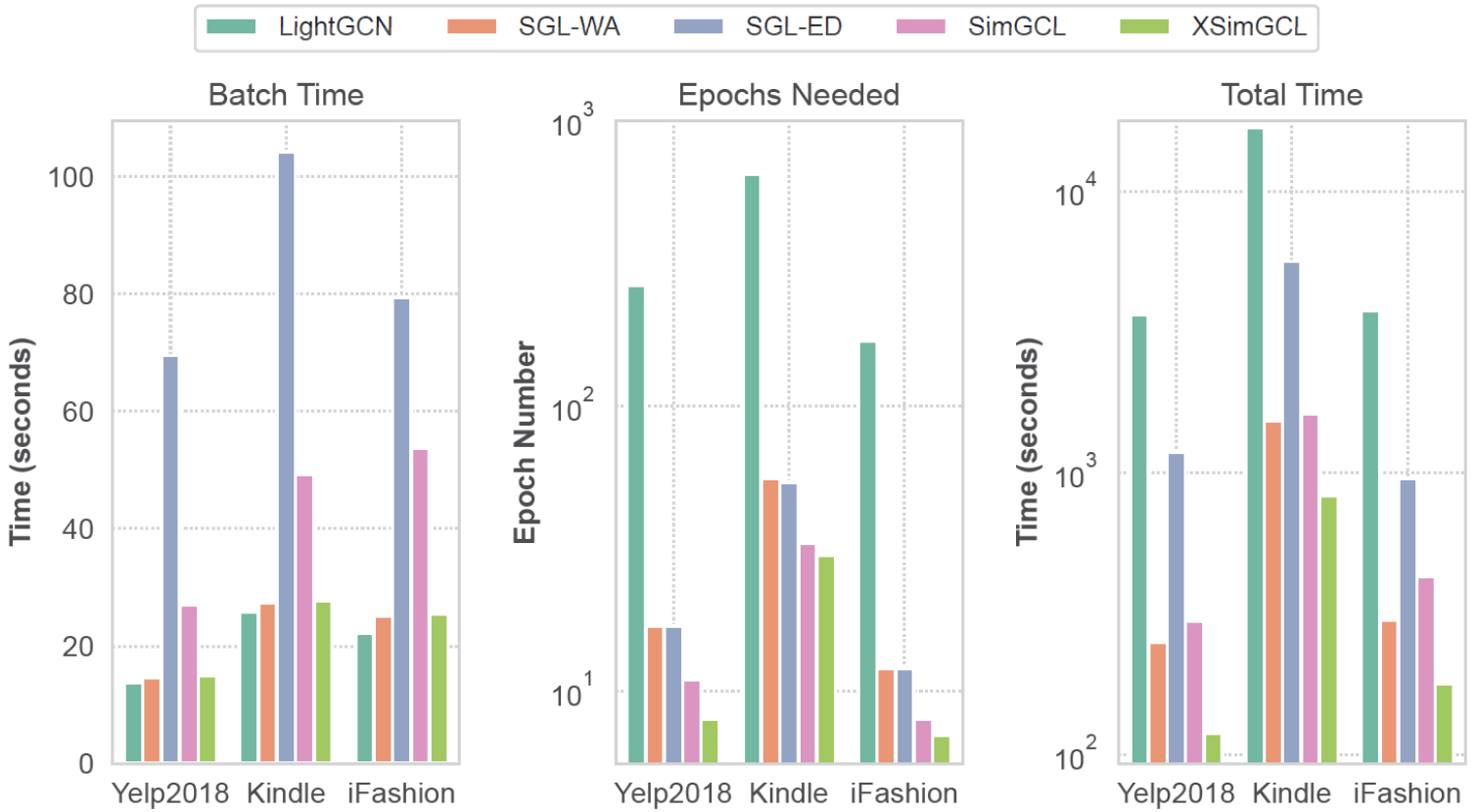
Performance Comparison

Method		Yelp2018		Amazon-Kindle		Alibaba-iFashion	
		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
1-Layer	LightGCN	0.0631	0.0515	0.1871	0.1186	0.0845	0.0390
	SGL-ND	0.0643 (+1.9%)	0.0529 (+2.7%)	0.1880 (+0.5%)	0.1192 (+0.5%)	0.0896 (+6.0%)	0.0432(+10.8%)
	SGL-ED	0.0637 (+1.0%)	0.0526 (+2.1%)	0.1936 (+3.5%)	0.1231 (+3.8%)	0.0932 (+10.3%)	0.0447 (+14.6%)
	SGL-RW	0.0637 (+1.0%)	0.0526 (+2.1%)	0.1936 (+3.5%)	0.1231 (+3.8%)	0.0932 (+10.3%)	0.0447 (+14.6%)
	SGL-WA	0.0628 (-0.4%)	0.0525 (+1.9%)	0.1918 (+2.5%)	0.1221 (+2.9%)	0.0913 (+8.0%)	0.0440 (12.8%)
	SimGCL	0.0689 (+9.2%)	0.0572 (+11.1%)	**0.2087 (+11.5%)**	**0.1361 (+14.8%)**	0.1036 (+22.6%)	0.0505 (+29.5%)
	XSimGCL	**0.0692 (+9.7%)**	**0.0582 (+13.0%)**	0.2071 (+10.7%)	0.1339 (+12.9%)	**0.1069 (+26.5%)**	**0.0527 (+35.1%)**
2-Layer	LightGCN	0.0622	0.0504	0.2033	0.1284	0.1053	0.0505
	SGL-ND	0.0658 (+5.8%)	0.0538 (+6.7%)	0.2020 (-0.6%)	0.1307 (+1.8%)	0.0993 (-5.7%)	0.0484 (-4.2%)
	SGL-ED	0.0668 (+7.4%)	0.0549 (+8.9%)	0.2084 (+2.5%)	0.1341 (+4.4%)	0.1062 (+0.8%)	0.0514 (+1.8%)
	SGL-RW	0.0644 (+3.5%)	0.0530 (+5.2%)	0.2088 (+2.7%)	0.1345 (+4.8%)	0.1053 (+0.0%)	0.0512 (+1.4%)
	SGL-WA	0.0653 (+5.0%)	0.0544 (+7.9%)	0.2068 (+1.7%)	0.1330 (+3.6%)	0.1028 (-2.4%)	0.0501 (-0.8%)
	SimGCL	0.0719 (+15.6%)	0.0601 (+19.2%)	0.2071 (+1.9%)	0.1341 (+4.4%)	0.1119 (+6.3%)	0.0548 (+8.5%)
	XSimGCL	**0.0722 (+16.1%)**	**0.0604 (+19.8%)**	**0.2114 (+4.0%)**	**0.1382 (+7.6%)**	**0.1143 (+8.5%)**	**0.0559 (+10.7%)**
3-Layer	LightGCN	0.0639	0.0525	0.2057	0.1315	0.0955	0.0461
	SGL-ND	0.0644 (+0.8%)	0.0528 (+0.6%)	0.2069 (+0.6%)	0.1328 (+1.0%)	0.1032 (+8.1%)	0.0498 (+8.0%)
	SGL-ED	0.0675 (+5.6%)	0.0555 (+5.7%)	0.2090 (+1.6%)	0.1352 (+2.8%)	0.1093 (+14.5%)	0.0531 (+15.2%)
	SGL-RW	0.0667 (+4.4%)	0.0547 (+4.5%)	0.2105 (+2.3%)	0.1351 (+2.7%)	0.1095 (+14.7%)	0.0531 (+15.2%)
	SGL-WA	0.0671 (+5.0%)	0.0550 (+4.8%)	0.2084 (+1.3%)	0.1347 (+2.4%)	0.1065 (+11.5%)	0.0519 (+12.6%)
	SimGCL	0.0721 (+12.8%)	0.0601 (+14.5%)	0.2104 (+2.3%)	0.1374 (+4.5%)	0.1151 +(20.5%)	0.0567 (+23.0%)
	XSimGCL	**0.0723 (+13.1%)**	**0.0604 (+15.0%)**	**0.2147 (+4.4%)**	**0.1415 (+7.6%)**	**0.1196 (+25.2%)**	**0.0586 (27.1%)**

Experiments

Running Speed Comparison

Running time for per epoch

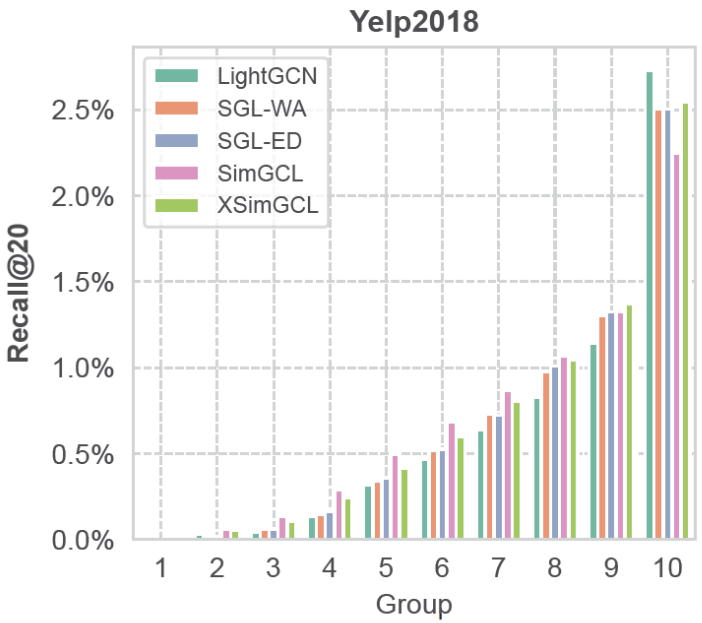


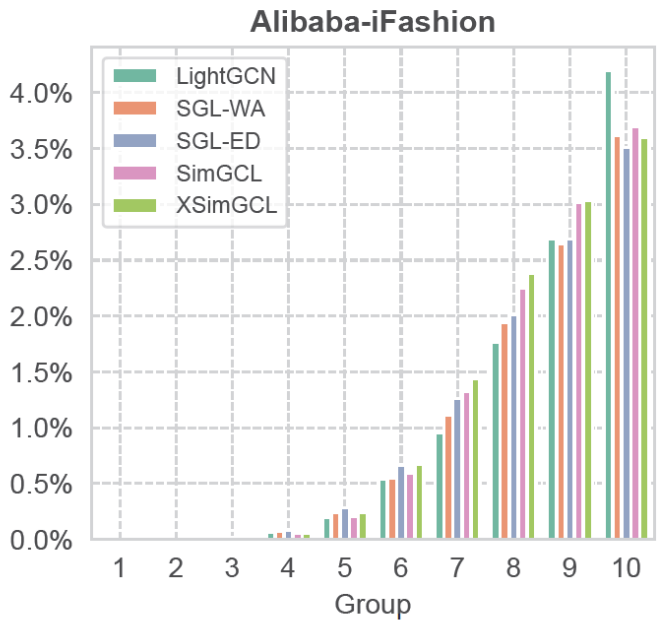
Running Faster!

Converge Faster!

Experiments

Ability to Debias





- XSimGCL/SimGCL's improvements all come from the items with lower popularity.
- LightGCN is inclined to recommend popular items.
- SGL variants fall between LightGCN and XSimGCL on exploring long-tail items.

THE WEB CONFERENCE ACM

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

06

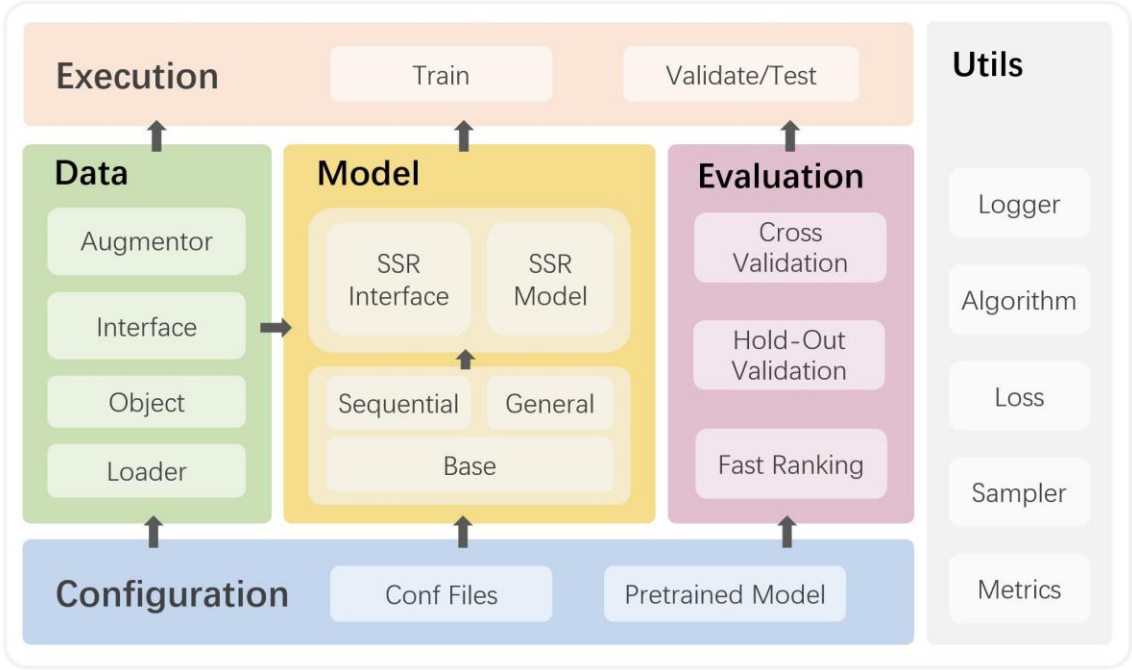
Library and Findings

- A Library for self-supervised recommendation

Tutorial

SELFRec

- SELFRec incorporates multiple high-quality datasets and metrics.
- There are more than 20+ state-of-the-art SSR methods implemented in SELFRec for fair empirical comparison.
- SELFRec is developed with Python 3.7 and PyTorch 1.7+.
- SELFRec provides a set of simple and high-level interfaces, by which new SSR models can be easily added in a plug-and-play fashion.
- SELFRec decouples the model design from other procedures.



SELFREC

Framework for Self-Supervised Recommendation

https://github.com/Coder-Yu/SELFRec

THE WEB CONFERENCE ACM

THE UNIVERSITY OF QUEENSLAND AUSTRALIA

07

Limitations and Future Research

Tutorial

Limitations and Future Research

- **Theory for Augmentation Selection**
  - Cannot seamlessly transplant augmentation approaches designed for other fields to recommendation
  - Most augmentation approaches are based on heuristics
  - Cumbersome trial-and-error work is needed to search for useful augmentations

  A solid recommendation-specific theoretical foundation for the augmentation selection is urgently needed.

Example

In vision tasks, views should not share too much information (left) or too little information (right), but should find an optimal mix (the "sweet spot", middle) that maximizes the downstream performance.

What is the criterion for augmentation selection in recommendation?

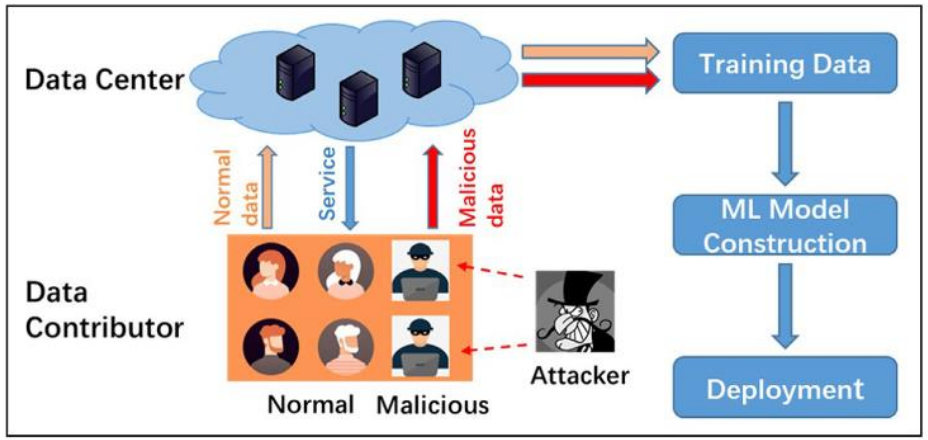
Y, Tian et al. Understanding View Selection for Contrastive Learning. Google AI Blog

Limitations and Future Research

- **Attacking and Defending in Pre-trained Recommendation Models**
  - Recommender systems are vulnerable to the data poisoning attack
  - It remains unknown if the recommendation models pretrained in a self-supervised way are robust to such attacks

Developing new-type attacks and defending self-supervised pretrained recommender systems against these attacks will be an interesting future research direction.

Example



Are pre-trained recommendation models robust to data poisoning attacks?

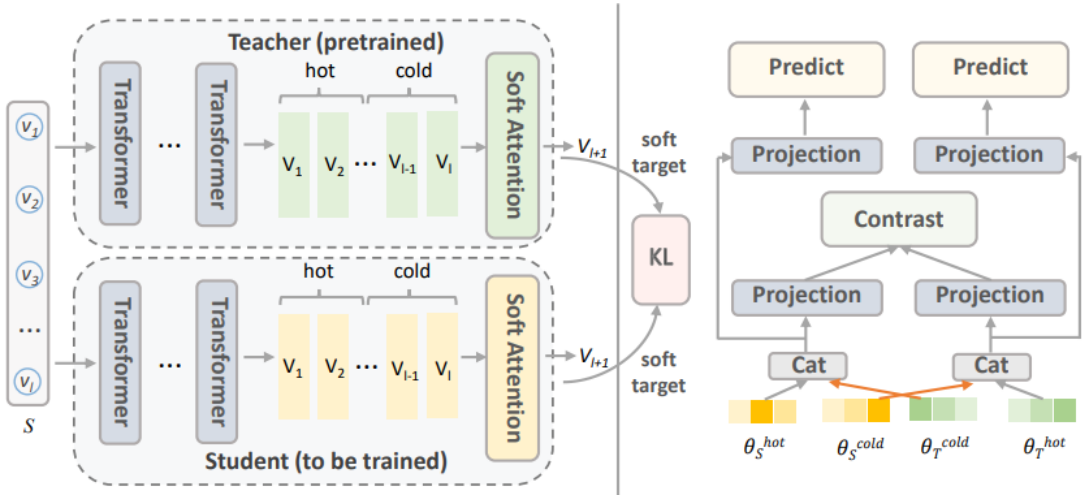
Pic from Poisoning attacks and countermeasures in intelligent networks: Status quo and prospects. Digital Communications and Networks

Limitations and Future Research

- **On-Device Self-Supervised Recommendation**
  - On-device recommender systems are compromised by the highly compressed model size and limited labeled data.
  - Combined with the technique of knowledge distillation, SSL may largely compensate for the accuracy degradation of on-device recommendation models.

On-device self-supervised recommendation is still under-explored, and it deserves further study.

Example



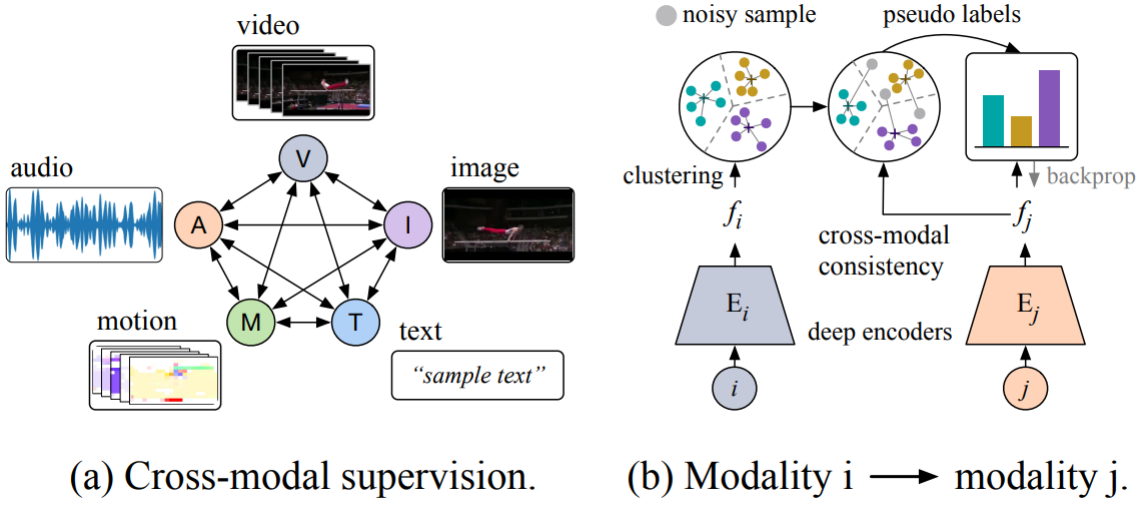
How to empower on-device recommendation with SSL?

Pic from On-Device Next-Item Recommendation with Self-Supervised Knowledge Distillation. SIGIR'22

Limitations and Future Research

- **Towards General-Purpose Pre-Training**
  - Data in recommender systems is multi-modal and scenarios are rather diverse
  - Explore general-purpose recommendation models which are pre-trained with the multi-modal SSL on large-scale data.
  - Enable models to adapt to multiple downstream recommendation tasks with the cheap finetuning.

Example



(a) Cross-modal supervision. (b) Modality i ⟶ modality j.

How to develop a once and for all pretraining technique for diverse modalities and downstream tasks?

Pic from Once and for All: Self-supervised Multi-modal Co-training on One-billion Videos at Alibaba. MM'21

Get More Information

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING

Self-Supervised Learning for Recommender Systems: A Survey

Junliang Yu, Hongzhi Yin*, Xin Xia, Tong Chen, Jundong Li, and Zi Huang

Abstract—Neural architecture-based recommender systems have achieved tremendous success in recent years. However, when dealing with highly sparse data, they still fall short of expectation. Self-supervised learning (SSL), as an emerging technique to learn with unlabeled data, recently has drawn considerable attention in many fields. There is also a growing body of research proceeding towards applying SSL to recommendation for mitigating the data sparsity issue. In this survey, a timely and systematical review of the research efforts on self-supervised recommendation (SSR) is presented. Specifically, we put forward an exclusive definition of SSR, on top of which we build a comprehensive taxonomy to divide existing SSR methods into four categories: contrastive, generative, predictive, and hybrid. For each category, the narrative unfolds along its concept and formulation, the involved methods, and its pros and cons. Meanwhile, to facilitate the development and evaluation of SSR models, we release an open-source library SELFRec, which incorporates multiple benchmark datasets and evaluation metrics, and has implemented a number of state-of-the-art SSR models for empirical comparison. Finally, we shed light on the limitations in the current research and outline the future research directions.

Index Terms—Recommendation, Self-Supervised Learning, Contrastive Learning, Pre-Training, Data Augmentation.

✦

https://arxiv.org/abs/2203.15876


THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA
Thanks
Q&A
ONLY ONLY
END