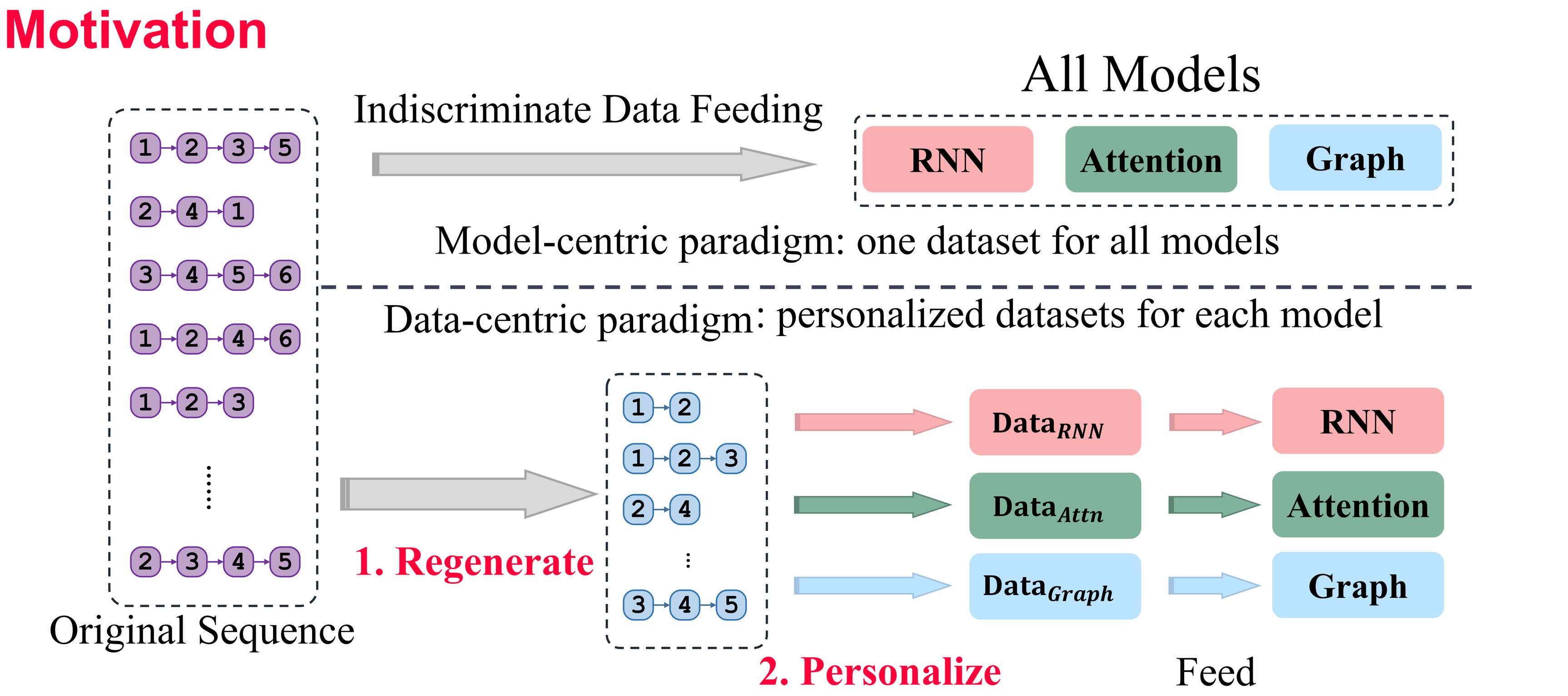
Dataset Regeneration for Sequential Recommendation

Mingjia Yin[1], Hao Wang[1]*, Wei Guo[2], Yong Liu[2],
Suojuan Zhang[1], Sirui Zhao[1], Defu Lian[1], Enhong Chen[1]
1.University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
2.Huawei Singapore Research Center

paper code

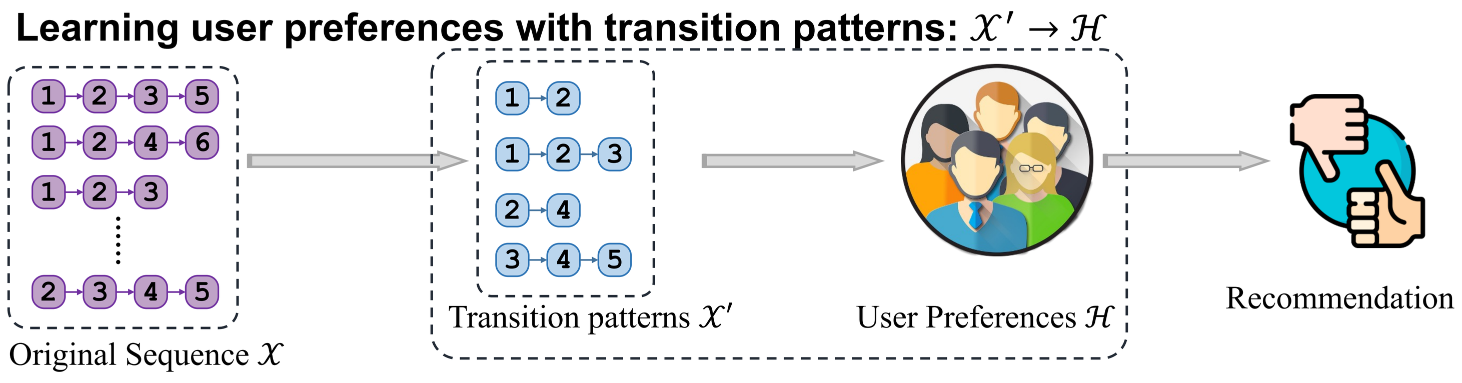
Motivation



Learning **Datasets** rather than **Models**

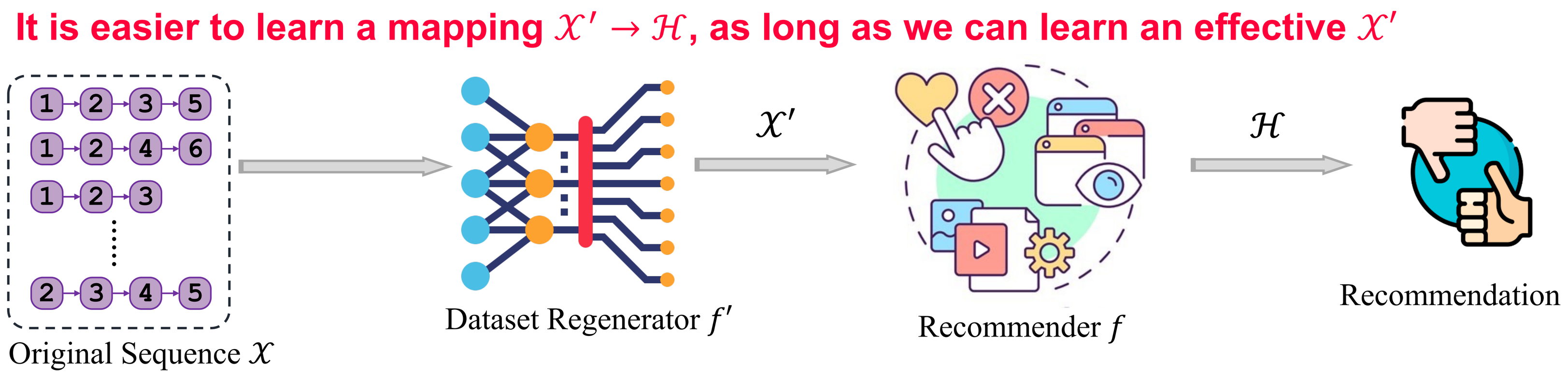
Key Idea

- We decompose sequential user modeling into two stages
  - Extracting the item transition patterns: $\mathcal{X} \rightarrow \mathcal{X}'$
  - Learning user preferences with transition patterns: $\mathcal{X}' \rightarrow \mathcal{H}$



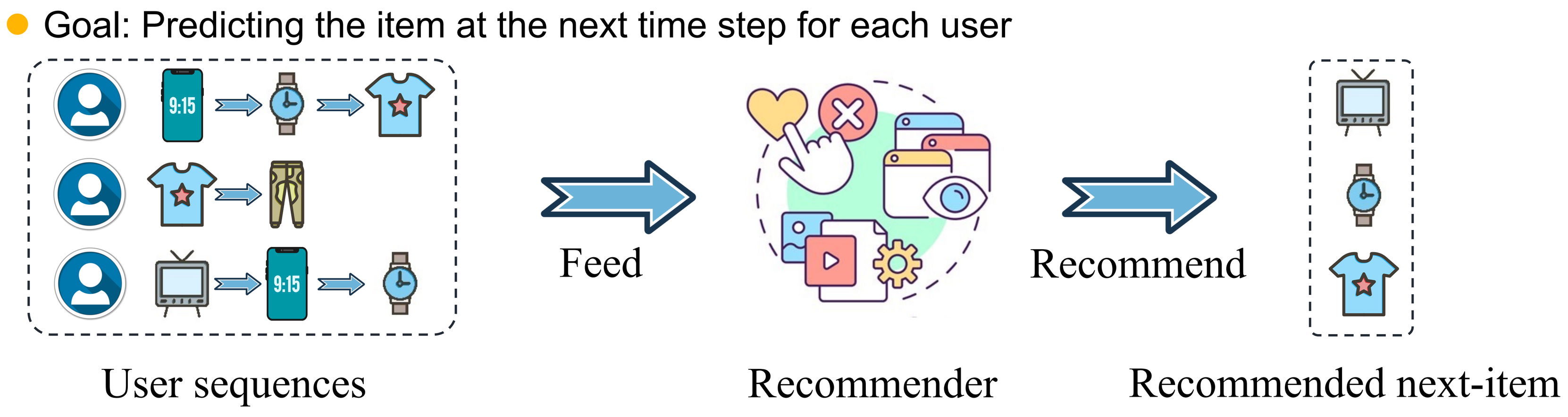
- Traditional model-centric paradigm directly learns the mapping $\mathcal{X} \rightarrow \mathcal{H}$
  - It is hard to learn a satisfying f since it contains two mappings $\mathcal{X} \rightarrow \mathcal{X}'$ and $\mathcal{X}' \rightarrow \mathcal{H}$

- Our paradigm develops a new dataset that explicitly represents the transition patterns $\mathcal{X}'$
  - It is easier to learn a mapping $\mathcal{X}' \rightarrow \mathcal{H}$, as long as we can learn an effective $\mathcal{X}'$

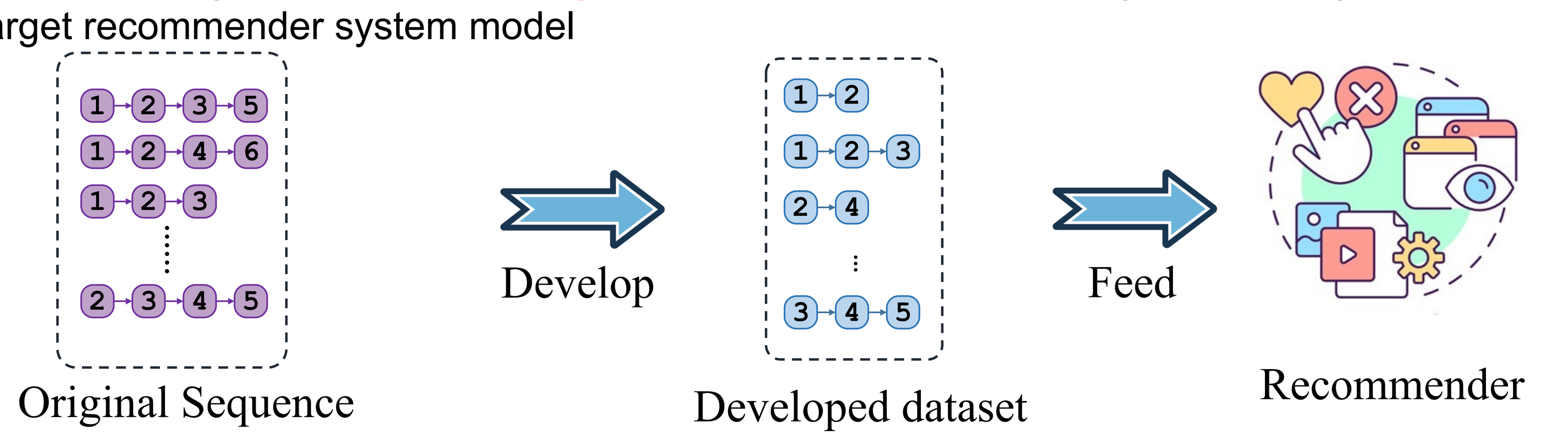


Problem Statement

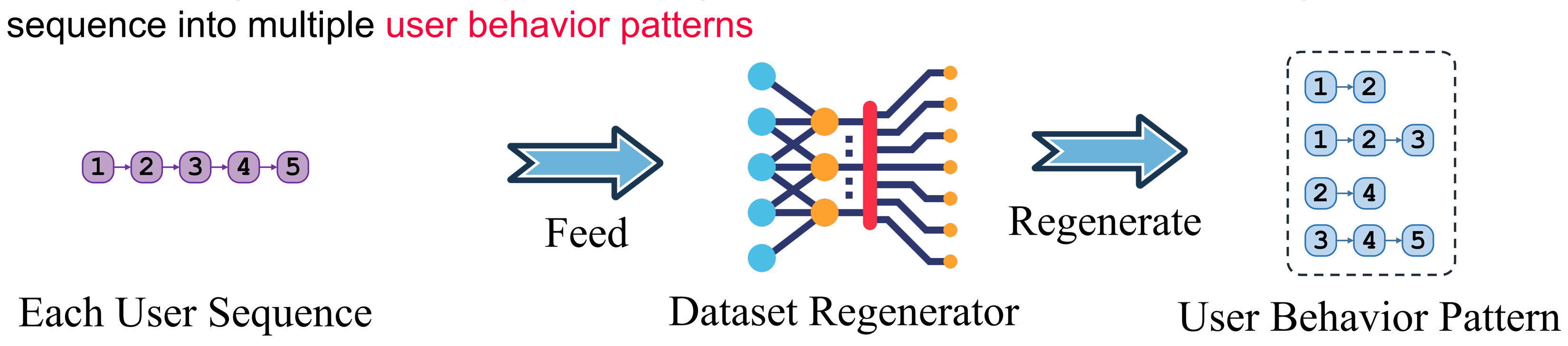
- **Sequential Recommendation**
  - Goal: Predicting the item at the next time step for each user



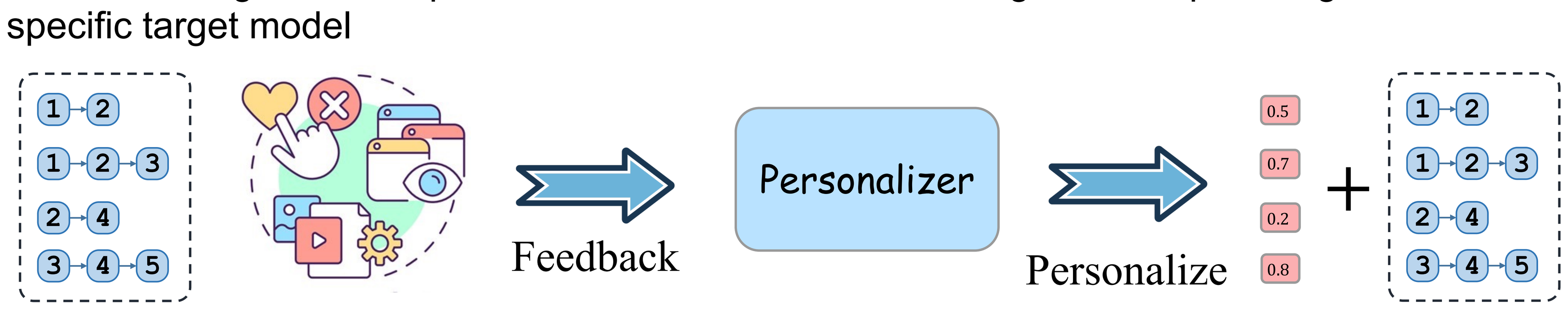
- **Training Data Development**
  - Goal: Learning an informative and generalizable dataset, facilitating the learning process of a target recommender system model



- **Dataset Regeneraiton**
  - Goal: Learning an one-to-many dataset regenerator, which transforms each original user sequence into multiple user behavior patterns

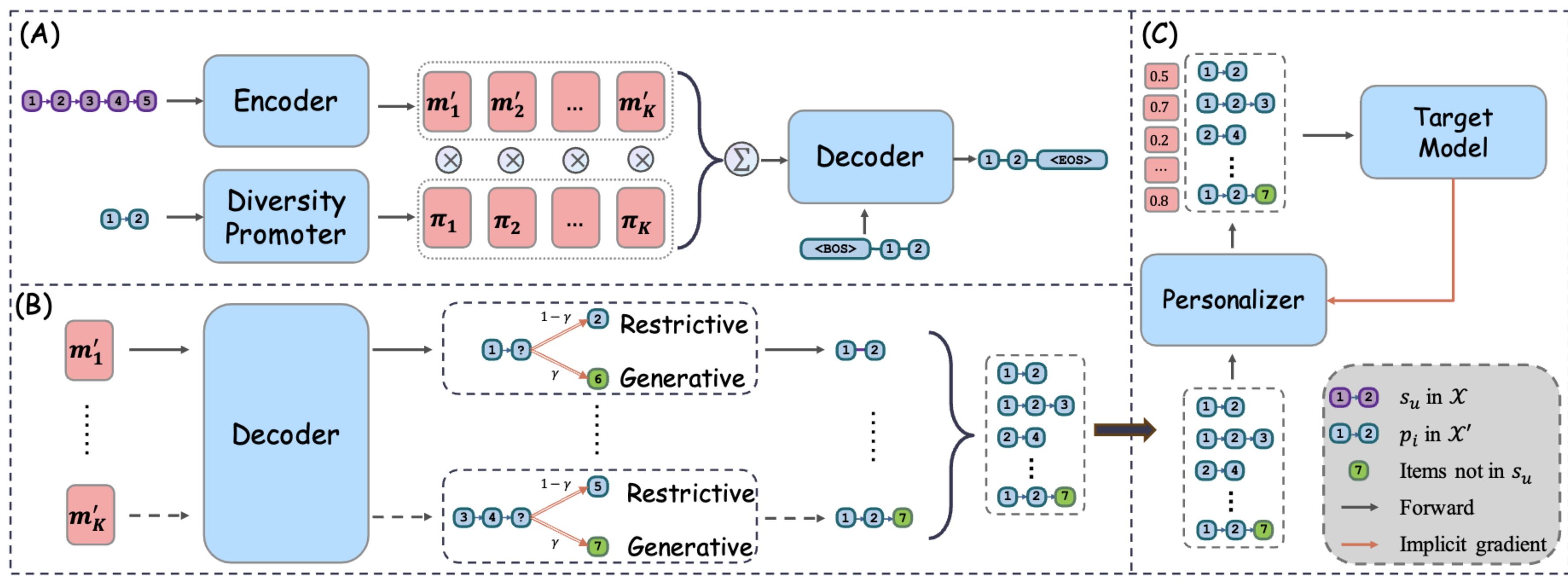


- **Model-aware Dataset Regeneration**
  - Goal: Learning a dataset personalizer, which scores each regenerated pattern given a specific target model

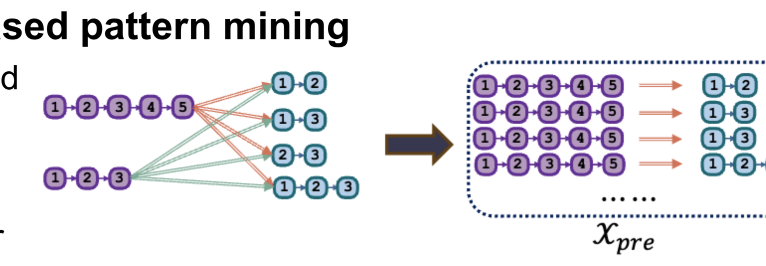


Methodology

- Overview



- **(A) Model-agnostic dataset regeneration (DR4SR)**
  - **Pre-training dataset construction with rule-based pattern mining**
    - Extracting patterns that appear more than a specified number of times within a given sliding window size



  - **Diversity-promoted regenerator**
    - **Architecture**: Encoder-Decoder-based Transformer
    - **Input**: original user sequences $\mathcal{X}$; **Output**: sequential behavior patterns $\mathcal{X}'$
    - **Challenge**: It is hard for vanilla transformer to model the one-to-many relationship between the source sequences and target patterns
    - **Solution**: We introduce a diversity promoter which transforms the memory generated by the encoder into a target-aware memory
    - **Formulation**:
      - Projecting the encoded memory into k different latent spaces: $m_k' = MLP_k(m)$
      - Generating a probability vector with target information: $\pi = Softmax(MLP(h_{pattern}^{(l)}))$
      - Generating a target-aware memory: $m_{final} = \sum_{k=1}^{K} \pi_k m_k'$
    - **Learning**:
      - Reconstruct each target pattern with the source sequence as input

$$L_{recon} = -\sum_{(s_u, p_i)}^{|\mathcal{X}_{pre}|} \sum_{t=1}^{T} \log\left(P(p_{it} | h_u^{(l)}, \hat{p}_{<t})\right)$$

- **(B) Dataset regeneration with hybrid inference strategy**
  - **Basic process**: Re-feeding the original sequences into the regenerator and conduct inference
  - **Restrictive mode (Exploitation)**: Decoding is limited to selecting items from the input sequence
  - **Generative mode (Exploration)**: No restrictions, exploring patterns that not exist in the original data
  - **Hybrid mode (Balanced)**: A probability γ to adopt generative mode and $1-\gamma$ for restrictive mode
  - **Note**: No target patterns input for the diversity promoter. We just respectively input each projected memory into the decoder to generate K patterns
- **(C) Model-aware dataset regeneration (DR4SR+)**
  - **Dataset personalizer (MLP)**
    - **Input**: sequential behavior patterns $\mathcal{X}'$; **Output**: sample weight for each training instances W
    - **Learning**:

$$w_{i,t} = \frac{\exp((g_\phi(h_t^i)_0 + Gumbel\_Noise)/\tau)}{\sum_{k=0}^{1} \exp((g_\phi(h_t^i)_k + Gumbel\_Noise)/\tau)}, where\ g_\phi(h_t^i) \in \mathbb{R}^2$$

$$L_{next-item}(i,t) = -\log\left(\sigma(h_{t-1}^i \cdot v_t^i)\right) - \sum_{v_j \notin p_i} \log\left(1 - \sigma(h_{t-1}^i \cdot v_j)\right)$$

$$L_{rec} = \sum_{i=1}^{|\mathcal{X}'|} \sum_{t=2}^{|p_i|} \boxed{w_{i,t}} L_{next-item}(i,t)$$

    - **Challenge**: Model collapse, $w_{i,t} \approx 0$ for all training instances
    - **Solution**: We formalize the problem as a bi-level optimization problem

$$\phi^* = \arg\min_\phi L_{rec-ori}(\theta^*(\phi)),$$
$$s.t.\quad \theta^*(\phi) = \arg\min_\theta L_{rec}(\theta, \phi)$$

      - **Efficiently optimized with implicit gradient**

$$\nabla_\phi \mathrm{L}_{\mathrm{rec-ori}} = -\nabla_\theta L_{rec-ori} \cdot \sum_{n=0}^{K} \left(I - \nabla_\theta^2 L_{rec}\right)^n \cdot \nabla_\phi \nabla_\theta L_{rec}$$

Key Results

- **Overall Performance: Integrating DR4SR and DR4SR+ with various backbones**

Dataset	Beauty				Sports				Toys				Yelp			
Metric	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
∞-AE	0.0478	0.0661	0.0262	0.0308	0.0256	0.0373	0.0144	0.0173	0.0450	0.0593	0.0268	0.0304	0.0252	0.0424	0.0121	0.0162
MELT	0.0577	0.0879	0.0303	0.0379	0.0311	0.0488	0.0163	0.0208	0.0709	0.0987	0.0401	0.0473	0.0293	0.0497	0.0143	0.0195
GRU4Rec	0.0204	0.0382	0.0107	0.0150	0.0160	0.0279	0.0085	0.0115	0.0212	0.0357	0.0099	0.0136	0.0215	0.0364	0.0105	0.0143
DR4SR	0.0252**	0.0448**	0.0128**	0.0177**	0.0208**	0.0341**	0.0102**	0.0135**	0.0252**	0.0418**	0.0124**	0.0165**	0.0235**	0.0403**	0.0114**	0.0156**
Improv	23.5%	17.3%	19.6%	18.0%	30.0%	22.2%	20.0%	17.4%	18.9%	22.4%	25.3%	21.3%	9.30%	10.7%	8.57%	9.09%
DR4SR+	**0.0292****	**0.0473****	**0.0149****	**0.0194****	**0.0223****	**0.0360****	**0.0116****	**0.0151****	**0.0274****	**0.0456****	**0.0134****	**0.0179****	**0.0243****	**0.0415****	**0.0120****	**0.0164****
Improv	43.1%	23.8%	39.3%	29.3%	39.4%	29.0%	36.5%	31.3%	29.2%	27.7%	35.4%	31.6%	13.0%	14.0%	14.3%	14.7%
SASRec	0.0553	0.0847	0.0291	0.0368	0.0297	0.0449	0.0156	0.0194	0.0682	0.0951	0.0381	0.0448	0.0289	0.0488	0.0143	0.0193
DR4SR	0.0595**	0.0906**	0.0317**	0.0395**	0.0330**	0.0512**	0.0174**	0.0220**	0.0762**	0.1049**	0.0432**	0.0504**	0.0304*	0.0512*	0.0151*	0.0202*
Improv	7.59%	6.97%	8.93%	7.34%	11.1%	14.0%	11.5%	13.4%	11.7%	10.3%	13.4%	12.5%	5.19%	4.92%	5.59%	4.66%
DR4SR+	**0.0619****	**0.0919****	**0.0337****	**0.0412****	**0.0349****	**0.0525****	**0.0191****	**0.0235****	**0.0773****	**0.1068****	**0.0453****	**0.0527****	**0.0317****	**0.0523****	**0.0159****	**0.0211****
Improv	11.9%	8.50%	15.8%	12.0%	17.5%	16.9%	22.4%	21.1%	13.3%	12.3%	18.9%	17.6%	9.69%	7.17%	11.2%	9.33%
FMLP	0.0602	0.0934	0.0311	0.0394	0.0323	0.0524	0.0166	0.0217	0.0676	0.0982	0.0377	0.0447	0.0297	0.0495	0.0143	0.0197
DR4SR	0.0635**	0.0993**	0.0332**	0.0421	0.0345	0.0559	0.0177**	0.0230**	0.0717**	0.1061**	0.0400**	0.0486**	0.0316**	0.0524**	0.0158**	0.0210**
Improv	5.48%	6.32%	6.75%	6.85%	6.81%	6.68	6.63%	5.99%	6.07%	8.04%	6.10%	8.72%	6.40%	5.86%	10.5%	6.60%
DR4SR+	**0.0687****	**0.1056****	**0.0357****	**0.0449****	**0.0384****	**0.0597****	**0.0198****	**0.0253****	**0.0788****	**0.1136****	**0.0437****	**0.0524****	**0.0353****	**0.0582****	**0.0171****	**0.0231****
Improv	14.1%	13.1%	14.8%	14.0%	18.9%	13.9%	19.3%	16.6%	16.6%	15.7%	15.9%	17.2%	18.9%	17.6%	19.6%	17.3%
GNN	0.0570	0.0859	0.0311	0.0384	0.0311	0.0476	0.0167	0.0211	0.0697	0.0958	0.0403	0.0469	0.0242	0.0430	0.0118	0.0166
DR4SR	0.0611**	0.0926**	0.0324*	0.0406*	0.0336**	0.0525**	0.0182**	0.0230**	0.0736**	0.1031**	0.0424**	0.0498**	0.0268**	0.0451*	0.0129**	0.0175*
Improv	7.19%	7.80%	4.18%	5.73%	8.04%	10.3%	8.98%	9.00%	5.60%	7.62%	5.21%	6.18%	10.7%	4.88%	9.32%	5.42%
DR4SR+	**0.0637****	**0.0953****	**0.0334****	**0.0414****	**0.0351****	**0.0545****	**0.0189****	**0.0238****	**0.0771****	**0.1082****	**0.0442****	**0.0521****	**0.0272****	**0.0471****	**0.0134****	**0.0184****
Improv	11.8%	10.9%	7.40%	7.81%	12.9%	14.5%	13.2%	12.8%	10.6%	12.9%	9.68%	11.1%	12.4%	9.53%	13.6%	10.8%
CL4SRec	0.0653	0.0947	0.0370	0.0441	0.0381	0.0559	0.0215	0.0259	0.0781	0.1075	0.0456	0.0530	0.0322	0.0535	0.0159	0.0212
DR4SR	0.0732**	0.1016**	0.0423**	0.0495**	0.0401**	0.0600**	0.0227**	0.0274**	0.0821**	0.1113*	0.0481**	0.0551*	0.0344**	0.0561**	0.0174**	0.0229**
Improv	12.1%	7.29%	14.3%	12.2%	5.25%	7.33%	5.58%	5.79%	5.12%	3.53%	5.48%	3.96%	6.83%	4.86%	9.43%	8.02%
DR4SR+	**00756****	**0.1062****	**0.0440****	**0.0517****	**0.0448****	**0.0655****	**0.0247****	**0.0299****	**0.0829****	**0.1140****	**0.0489****	**0.0567****	**0.0363****	**0.0598****	**0.0183****	**0.0241****
Improv	15.8%	1.12%	18.9%	17.2%	17.6%	17.2%	14.8%	15.4%	6.15%	6.05%	7.24%	6.98%	12.7%	11.8%	15.1%	13.7%

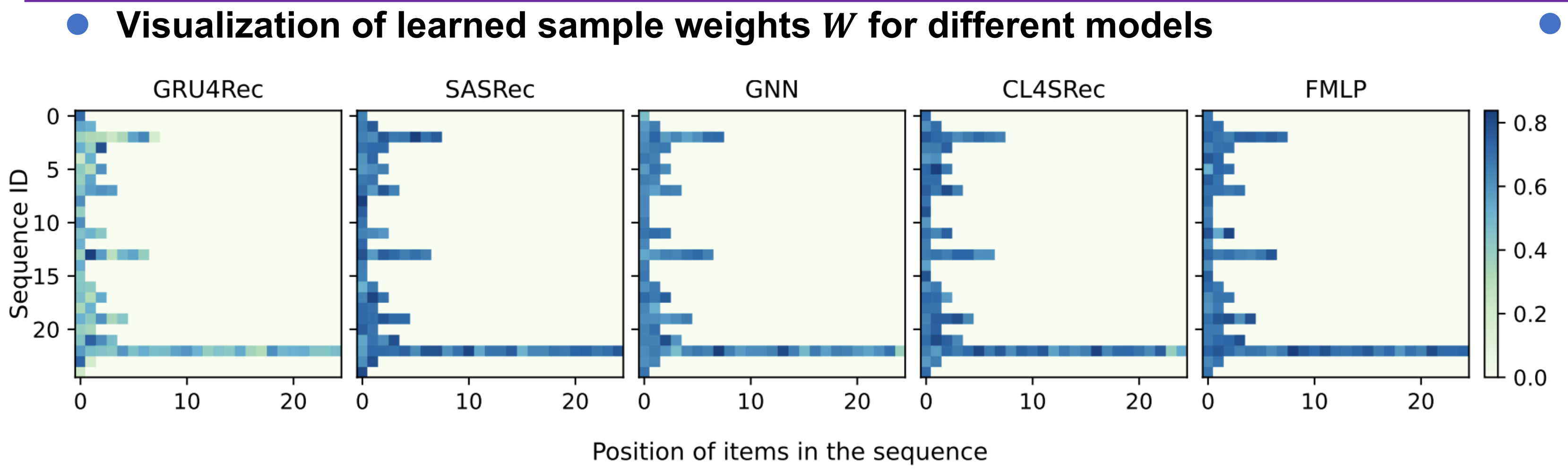
- **Ablation**

Dataset	Beauty	Sport	Toys	Yelp
SASRec	0.0368	0.0194	0.0448	0.0193
DR4SR+	**0.0412**	**0.0235**	**0.0527**	**0.0211**
(A) -diversity	0.0365	0.0211	0.0470	0.0196
(B) pattern	0.0181	0.0184	0.0407	0.0141
(C) end-to-end	0.0026	0.0029	0.0067	0.0035

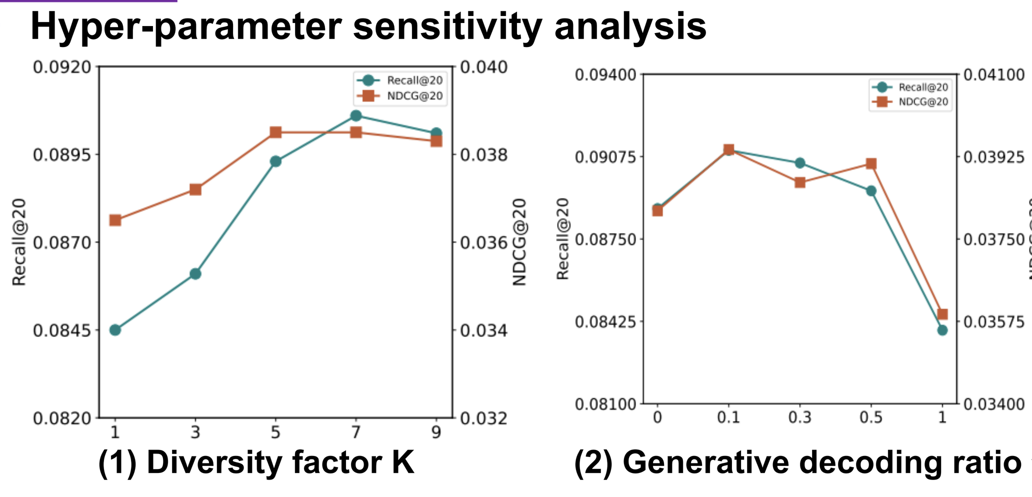
- **Time and space complexity**

Dataset	Metric	Beauty	Sport	Toys	Yelp
BASE	Runtime(s/epoch)	7.618	15.345	13.370	17.738
	GPU memory (MB)	1930	2194	1968	2254
w/ bi-level optimization	Runtime(s/epoch)	9.476	18.952	14.213	22.41
	GPU memory (MB)	2342	2626	2382	2688

- **Visualization of learned sample weights W for different models**



- **Hyper-parameter sensitivity analysis**



- **Comparison of using (Original / Regenerated) data to construct graphs or augmentation data**