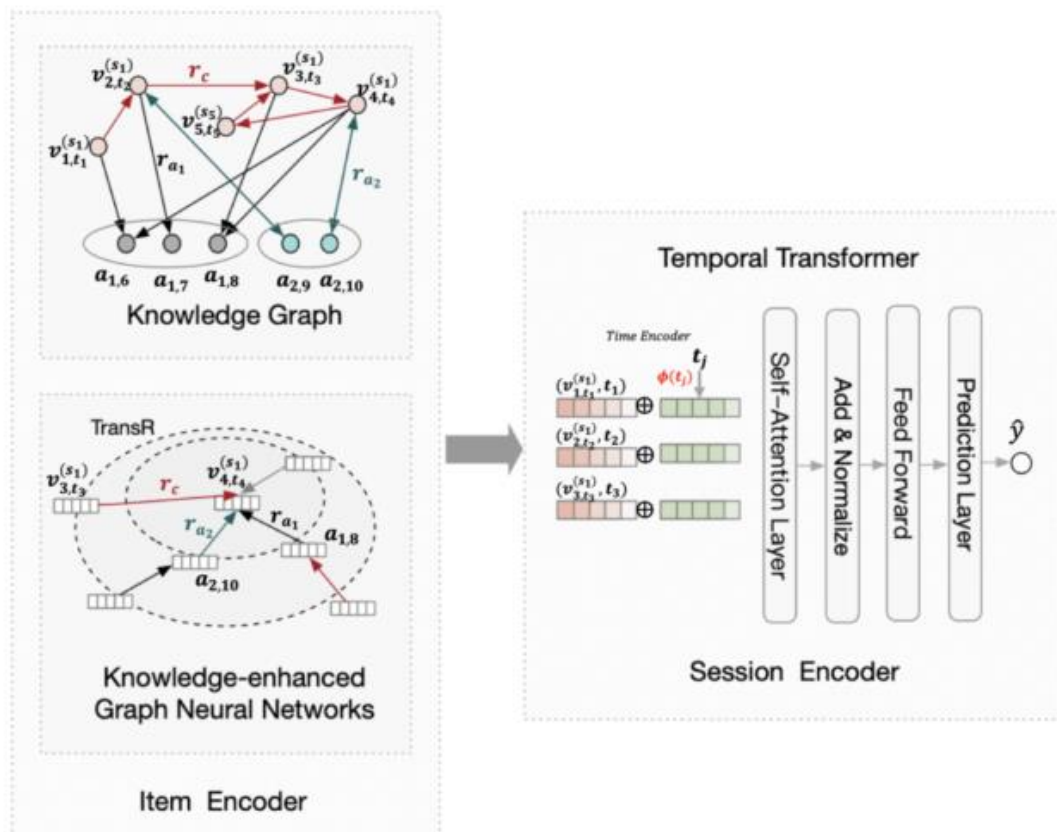


## Knowledge-enhanced Session-based Recommendation with Temporal Transformer (KSTT)

Morivation:

- SBR 中缺少对物品的知识信息的利用，因此提出在 SBR 中引入知识图谱
- SBR 中缺少对物品之间点击的时间间隔的利用——使用时间编码器，将时间编码，并替代 transformer 中的位置编码（Temporal Transformer）

Framework:



知识图谱用 TransR (head+relation=tail) 学习，得到的 embedding 和时间编码相加，然后放入 Transformer 的 encoder。

时间编码器:

- TBE

$$\Phi_{bucket}(t) = \text{One-hot}(\lfloor \log_2(\hat{t} - t) \rfloor)$$

- T2v

$$\Phi_{T2V}^i(t) = \begin{cases} w_i t + b_i, & \text{if } i = 1 \\ \sin(w_i t + b_i), & \text{if } 1 \leq i < d \end{cases}$$

- MTE

$$\Phi_{MTE}(t) = [\Phi_{w_1,d}(t), \Phi_{w_2,d}(t), ..., \Phi_{w_k,d}(t)]^T$$

$$\Phi_{w,d}(t) = [\sqrt{c_1}, \sqrt{c_{2j}}\cos(\frac{j\pi t}{\omega}), \sqrt{c_{2j+1}}\sin(\frac{j\pi t}{\omega}), ...]$$

实验部分：

	Method	<i>Diginetica</i>		<i>Yoochoose1/64</i>		<i>Yoochoose1/4</i>		<i>Last-fm</i>	
		R@20	MRR@20	R@20	MRR@20	R@20	MRR@20	R@20	MRR@20
Classical	POP	0.89	0.20	6.71	1.65	1.33	0.30	5.26	1.26
	S-POP	21.06	13.68	30.44	18.35	27.08	17.75	22.59	8.71
	Item-KNN	35.75	11.57	51.60	21.81	52.31	21.70	14.84	4.85
	BPR-MF	5.24	1.98	31.31	12.08	3.40	1.57	14.05	5.01
	FPMC	26.53	6.95	45.62	15.01	51.86	17.50	17.68	4.99
DNN-based	GRU4REC	29.45	8.33	60.64	22.89	59.53	22.60	17.90	5.39
	NARM	49.70	16.17	68.32	28.63	69.73	29.23	29.94	10.85
	STAMP	45.64	14.32	68.74	29.67	70.44	30.00	29.24	11.33
	RepeatNet	47.79	17.66	69.13	30.24	70.71	31.03	32.28	12.03
GNN-based	SR-GNN	50.73	17.59	70.57	30.94	71.36	31.89	30.75	12.53
	GC-SAN	52.48	18.20	71.20	30.48	71.46	31.47	32.46	12.22
	KSTT	<b>53.25</b>	<b>18.29</b>	<b>72.25</b>	<b>31.05</b>	<b>72.35</b>	<b>32.30</b>	<b>32.63</b>	<b>12.90</b>

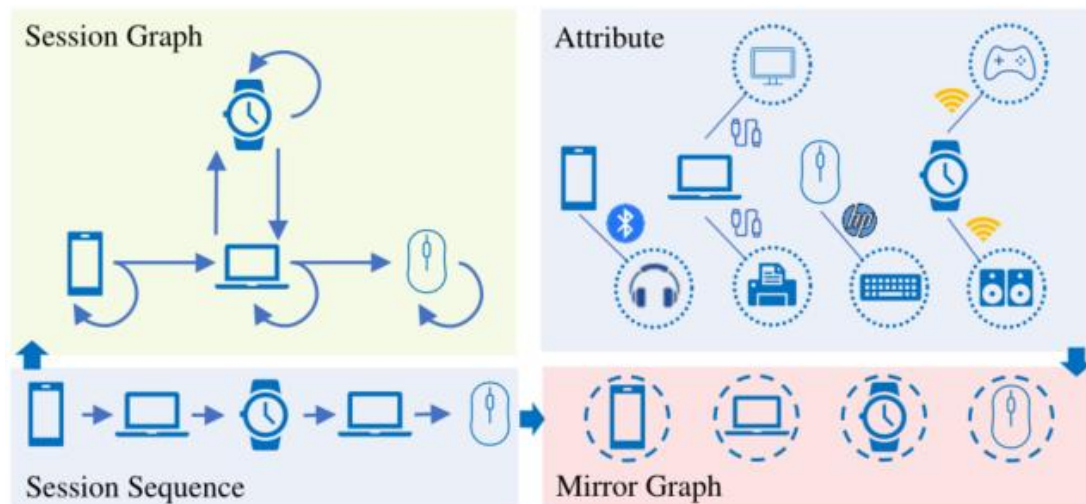
消融实验：baseline 为 SRGNN，在 SRGNN 上加上 KG、SAN 和时间编码（Temporal Transformer 的 encoder 部分由时间编码和 SAN 组成），来验证 KG 和 TT 的有效性。

Models	<i>Yoochoose1/64</i>		<i>Diginetica</i>	
	R@20	MRR@20	R@20	MRR@20
Baseline	70.48	30.81	50.73	17.59
Baseline (+KGNN)	70.74	30.98	52.17	17.88
Baseline (+SAN)	71.20	30.48	52.48	18.20
Baseline (+KGNN, +SAN)	71.53	30.50	53.10	18.25
Baseline (+KGNN, +SAN, +MTE)	72.09	30.63	53.12	18.24
Baseline (+KGNN, +SAN, +T2v)	72.10	30.70	53.15	18.27
Baseline (+KGNN, +SAN, +TBE)	<b>72.25</b>	<b>31.05</b>	<b>53.25</b>	<b>18.29</b>

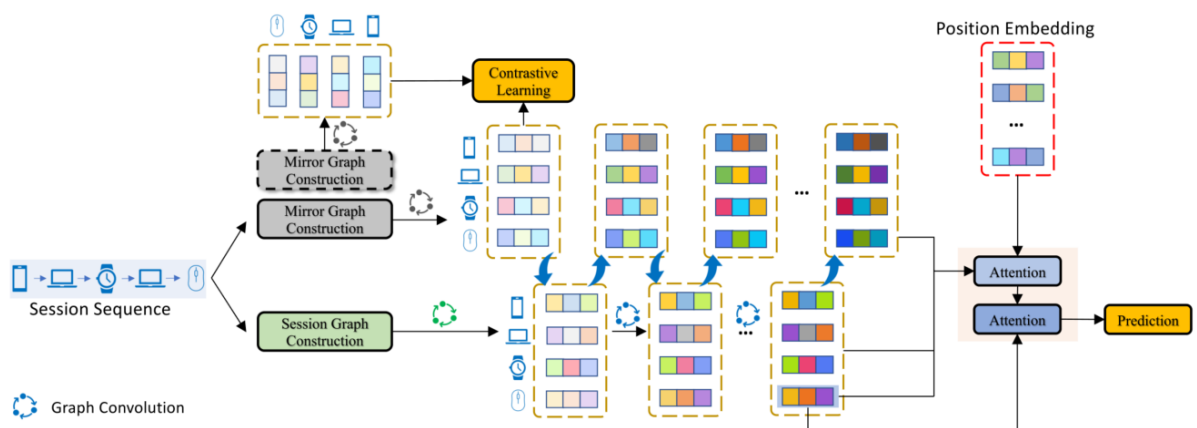
## An Attribute-Driven Mirror Graph Network for Session-based Recommendation (MGS)

Motivation: SBR 只挖掘了物品之间有限的短期序列信息，没有考虑物品的知识信息，这会造成数据稀疏性问题。物品与物品之间可以通过属性相连，所以引入物品的属性来构成一个额外的图，以此丰富物品额外的知识信息。对于每个会话，都构建两个图：

- Session graph: 学习会话间的序列转换模式
- Mirror graph: 学习物品额外的知识信息



Framework:



对于任意一个会话，先构造 session graph 和 mirror graph，两个 graph 先学习到两组不同的 embedding，两个图上都是使用的 GAT。得到两组不同的 embedding 后，通过迭代双重细化（iterative dual refinement）将两组 embedding 迭代训练，来融合会话语义和属性语义。

以下是两个 graph 上向量的学习方式：

- Session graph

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{e}_{ij}^T (\mathbf{x}_i^{(l-1)} \odot \mathbf{x}_j^{(l-1)})))}{\sum_{\mathbf{x}_k \in N_{\mathbf{x}_i}} \exp(\text{LeakyReLU}(\mathbf{e}_{ik}^T (\mathbf{x}_i^{(l-1)} \odot \mathbf{x}_k^{(l-1)})))}$$

$$\mathbf{x}_i^{(l)} = \sum_{x_j \in \mathcal{N}_{x_i}} \alpha_{ij} \mathbf{x}_j^{(l-1)}$$

其中  $e_{ij}$  表示  $i$  和  $j$  连接的类型（包括出边、入边、自环、双向），每种类型维护一个向量。

● Mirror graph

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{q}_{r_p}^\top [\mathbf{x}_i^{(0)} \parallel \mathbf{x}_j^{(0)}]))}{\sum_{x_k \in \mathcal{N}_{m_i}^{r_p}} \exp(\text{LeakyReLU}(\mathbf{q}_{r_p}^\top [\mathbf{x}_i^{(0)} \parallel \mathbf{x}_k^{(0)}]))}$$

$$\mathbf{m}_i^{r_p} = \sum_{x_j \in \mathcal{N}_{m_i}^{r_p}} \alpha_{ij} \mathbf{x}_j^{(0)}$$

$$\mathbf{m}_i = \mathbf{W}_1[\mathbf{m}_i^{r_1} \parallel \dots \parallel \mathbf{m}_i^{r_M}],$$

Mirror graph 中的边可以表示为  $\langle \text{item}, \text{attribute type}, \text{attribute value} \rangle$ ，约等于一个知识图谱， $ij$  通过属性间接相连， $q_{rp}$  代表 attribute relation 的向量， $\mathcal{N}$  代表与  $i$  通过属性  $rp$  间接相连的邻居，然后经过一个 softmax，这类似一个 RGAT，然后将  $i$  的每个属性都这样计算，再拼接起来，得到  $\mathbf{m}_i$ ，代表  $i$  在 mirror graph 上得到的 embedding。

得到两个图上物品的表示之后，将两组表示进行双重迭代细化，将物品的序列信息和知识信息融合在一起。

● Session item: 通过  $\mathbf{x}$  和  $\mathbf{m}$  之间的加权元素方向偏移来细化 session item embedding.

$$\beta_i = \frac{(\mathbf{W}_1^s \mathbf{x}_i^{(l)})^\top \mathbf{W}_2^s \mathbf{m}_i^{(l-1)}}{\sqrt{d}}$$

$$\mathbf{x}_i^{(l)} = \mathbf{x}_i^{(l)} + \beta_i (\mathbf{m}_i^{(l-1)} - \mathbf{x}_i^{(l)})$$

● Mirror item: 通过 attention 来评估每个 mirror 节点对每个 item 节点的重要性来细化 mirror item embedding（评估每个 mirror 节点与其邻居的重要性听起来更合理，但是这样做效果很差，后面会做相关实验）。

$$a_{ij} = \frac{\exp((\mathbf{W}_1^m \mathbf{m}_i^{(l-1)})^\top \mathbf{W}_2^m \mathbf{x}_j^{(l)})}{\sum_{k=1}^n \exp((\mathbf{W}_1^m \mathbf{m}_i^{(l-1)})^\top \mathbf{W}_2^m \mathbf{x}_k^{(l)})}$$

$$\mathbf{m}_i^{(l)} = \sum_{j=1}^n a_{ij} \mathbf{m}_j^{(l-1)}$$

● Final session item embedding: 经过  $L$  次迭代后，通过一个 highway network 来导出每个会话 item 的最终表示。

$$\mathbf{x}_i^{(L)} = \pi_i \mathbf{x}_i^{(0)} + (1 - \pi_i) \mathbf{x}_i^{(L)}$$

$$\pi_i = \text{sigmoid}(\mathbf{W}_h [\mathbf{x}_i^{(0)} \parallel \mathbf{x}_i^{(L)}])$$

得到细化的物品表示后，开始学习会话表示。

首先给物品的 embedding 加上位置编码：

$$\mathbf{h}_i = \mathbf{x}_i^{(L)} + \mathbf{p}_i.$$

然后利用 attention 计算每个 item 和最后一个 item 的相关性，作为权重，再加

权相加得到 session representation:

$$\beta_i = \mathbf{g}^\top \text{sigmoid}(\mathbf{W}_2 \mathbf{h}_i + \mathbf{W}_3 \mathbf{m}_i^{(L)} + \mathbf{W}_4 \mathbf{x}_n^{(L)} + \mathbf{b})$$

$$\mathbf{z}_s = \sum_{i=1}^n \beta_i \mathbf{h}_i,$$

然后使用门控机制来融合最后一个 item 和 session representation, 得到最终的 session representation:

$$\boldsymbol{\theta} = \text{sigmoid}(\mathbf{W}_5[\mathbf{z}_s \parallel \mathbf{x}_n^{(L)}])$$

$$\mathbf{z}_s = (1 - \mu \boldsymbol{\theta}) \odot \mathbf{z}_s + \mu \boldsymbol{\theta} \odot \mathbf{x}_n^{(L)}$$

得到会话表示后, 将每个 candidate items 的初始 embedding 与会话表示点积, 再经过一个 softmax, 得到 candidates item 下一次被点击的概率:

$$\hat{y}_c = \frac{\exp(\mathbf{z}_s^\top \mathbf{x}_c^{(0)})}{\sum_{i=1}^N \exp(\mathbf{z}_s^\top \mathbf{x}_i^{(0)})},$$

Loss Function:

$$\mathcal{L}_{main}(\hat{y}) = - \sum_{c=1}^N y_c \log(\hat{y}_c) + (1 - y_c) \log(1 - \hat{y}_c),$$

由于每个会话项有大量的属性相同的邻居, 因此每个相关属性的采样邻居物品肯定会包含一些噪声, 所以在卷积开始的时候表现可能会很差。所以这里引入一个对比学习的方法来监督 mirror graph 的生成。随机抽样属性相同的邻居, 为同一会话生成两个 mirror graph,  $\mathbf{m}'$  和  $\mathbf{m}''$  代表这两个 mirror graph 学习到的物品的 embedding,  $t$  代表温度系数, 在原来的 loss 上再加上对比学习的 loss:

$$\mathcal{L}_{ssl} = - \sum_{i=1}^n \log \frac{\exp(\cos(\mathbf{m}'_i, \mathbf{m}''_i)/\tau)}{\sum_{j=1}^n \exp(\cos(\mathbf{m}'_i, \mathbf{m}''_j)/\tau)}$$

$$\mathcal{L} = \mathcal{L}_{main} + \phi \mathcal{L}_{ssl} + \lambda \|\Phi\|_2^2,$$

Datasets:

Dataset	Tmall	Diginetica	30music
# click	818,479	982,961	1,429,251
# train	351,268	719,470	1,153,622
# test	25,898	60,858	122,517
# items	40,728	43,097	132,648
# attribute num	2	1	1
avg. AS num.	848.72	243.58	41.89
avg. len.	6.69	5.12	9.33

Performance:

Models	Tmall				Diginetica				30music			
	P@10	MRR@10	P@20	MRR@20	P@10	MRR@10	P@20	MRR@20	P@10	MRR@10	P@20	MRR@20
FPMC	13.10	7.12	16.06	7.32	15.43	6.20	26.53	6.95	1.51	0.55	2.40	0.61
GRU4REC	9.47	5.78	10.93	5.89	17.93	7.33	29.45	8.33	15.91	10.46	18.28	10.95
NARM	19.17	10.42	23.30	10.70	35.44	15.13	49.70	16.17	37.81	25.95	39.40	26.55
STAMP	22.63	13.12	26.47	13.36	33.98	14.26	45.64	14.32	36.13	25.97	42.57	26.27
SR-GNN	23.41	13.45	27.57	13.72	36.86	15.52	50.73	17.59	36.49	<u>26.71</u>	39.93	<u>26.94</u>
GCE-GNN	29.19	15.55	34.35	15.91	41.54	<b>18.29</b>	<u>54.64</u>	<b>19.20</b>	39.93	21.21	44.71	21.55
S <sup>2</sup> -DHCN	26.22	14.60	31.42	15.05	41.16	18.15	53.18	18.44	<u>40.05</u>	17.58	<u>45.49</u>	17.97
COTREC	<u>30.62</u>	<u>17.65</u>	<u>36.35</u>	<u>18.04</u>	<b>41.88</b>	18.16	54.18	19.07	39.88	17.42	45.15	17.79
MGS	<b>35.39*</b>	<b>18.15*</b>	<b>42.12*</b>	<b>18.62*</b>	<u>41.80</u>	<u>18.20</u>	<b>55.05*</b>	<u>19.13</u>	<b>41.51*</b>	<b>27.67*</b>	<b>46.46*</b>	<b>28.01*</b>

Ablation Study:

● Impact of Attribute information:

首先定义两个 MRR， $\alpha$  表示包含了与 ground-truth 具有相同属性值的 items 的 session 的性能是否得到了改善。S<sub>asg</sub> 表示包含了与 ground-truth 具有相同属性的 items 的会话

$\beta$  表示是否更好的考虑了与 ground-truth 有相同属性的 items，V<sub>ass</sub> 表示与 ground-truth 具有相同属性的 items。

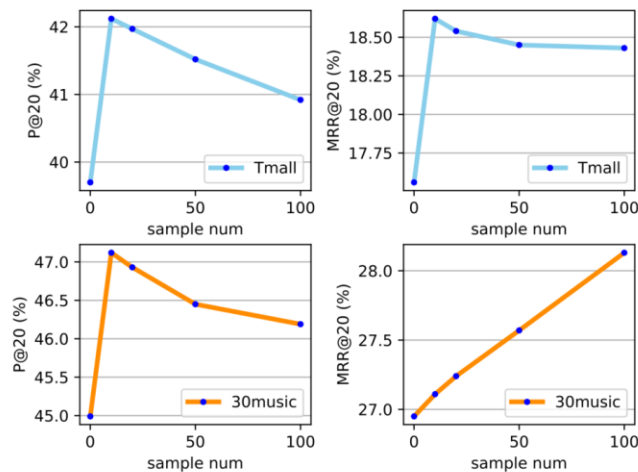
$$MRR_{\alpha} = \frac{1}{|S_{asg}|} \sum_{x_{gt} \in S_{asg}} \frac{1}{Rank(x_{gt})},$$

$$MRR_{\beta} = \frac{1}{|V_{ass}|} \sum_{x_{as} \in V_{ass}} \frac{1}{Rank(x_{as})},$$

Method	Tmall		30music	
	$MRR_{\alpha}$	$MRR_{\beta}$	$MRR_{\alpha}$	$MRR_{\beta}$
MGS (No-attri)	26.72	0.37	50.88	5.04
MGS	<b>28.16</b>	<b>0.43</b>	<b>51.46</b>	<b>5.85</b>

MGS (No) 表示去掉了 SSL 和双向迭代细化，这两项指标确实都提高了，证明了对于具有相同属性的物品，在会话层面和物品层面都得到了更多的考虑。

● Impact of Number of Neighbors:



横坐标是指对于 mirror graph 的 attribute same 的邻居的取样数值, 对于 Tmall, 在大概 10 的时候 P 和 MRR 都达到峰值, 而 music 的 MRR 呈一个不断上升的趋势, 这是因为 music 的平均 attribute-same 邻居数量较少。

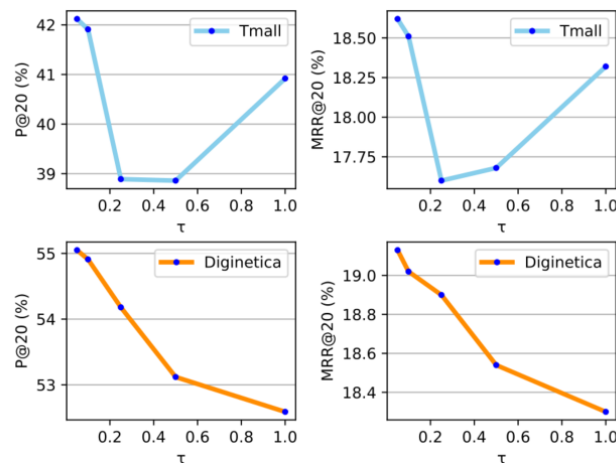
● Impact of SSL:

$$Sim = - \sum_{x_i \in \mathcal{X}} \sum_{x_j \in N_{x_i}^{as}} \log(\text{sigmoid}(\cos(\mathbf{x}_i, \mathbf{x}_j)))$$

Sim 代表  $x_i, x_j$  的距离, 越小代表越相似。

Method	Tmall			Diginetica			30music		
	P@20	MRR@20	Sim	P@20	MRR@20	Sim	P@20	MRR@20	Sim
MGS-NS	39.38	17.60	227.36	53.96	18.68	80.65	45.60	23.66	697.79
MGS	<b>42.12</b>	<b>18.62</b>	<b>178.86</b>	<b>55.05</b>	<b>19.13</b>	<b>42.18</b>	<b>46.46</b>	<b>28.01</b>	<b>393.94</b>

NS 代表去掉对比学习 loss, 下图代表对 temperature 的实验, 当  $t$  越大时, 会引入更少的辨别信息, 当  $t$  比较小时, 在应用 softmax 函数后, 对比对象之间的归一化差异会缩小, 从而导致模型的高梯度。

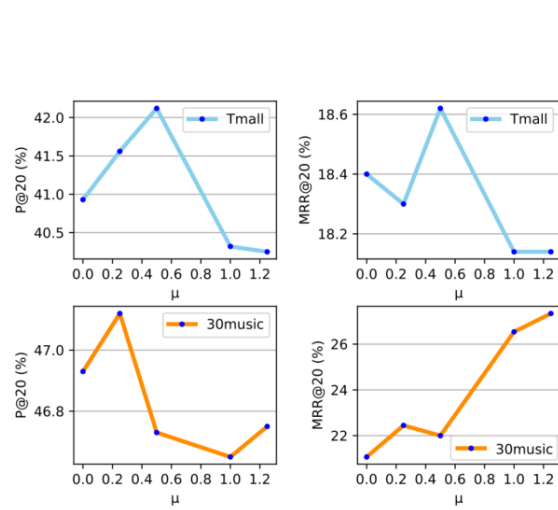


● Impact of Iterative Dual Refinement:

Method	Tmall		Diginetica	
	P@20	MRR@20	P@20	MRR@20
MGS (Adj)	40.46	18.11	54.94	19.01
MGS	<b>42.12</b>	<b>18.62</b>	<b>55.05</b>	<b>19.13</b>

Adj 表示只将每个 mirror 节点连接到其相应会话节点的相邻项。

● Impact of the Last Clicked Item:



对  $u$  系数做实验,  $u=0$  代表不加 last clicked item, Tmall 对 last 依赖不大, 表明模型需要更全面的信息来更好地理解用户的偏好。而 music 更依赖 last, 这是因为音乐软件用户更喜欢在短时间内连续收听类似的专辑。这些结果表明, 能手动调系数的门控机制对于不同的真实场景是可行的。