August 3-7, 2025

KDD 2025

TORONTO

Breaking the Top-K Barrier:
Advancing Top-K Ranking Metrics Optimization in Recommender Systems

Weiqin Yang[1], Jiawei Chen*[1], Shengjia Zhang[1], Peng Wu[2], Yuegang Sun[3], Yan Feng[1], Chun Chen[1], Can Wang[1]

[1]Zhejiang University

[2]Beijing Technology and Business University

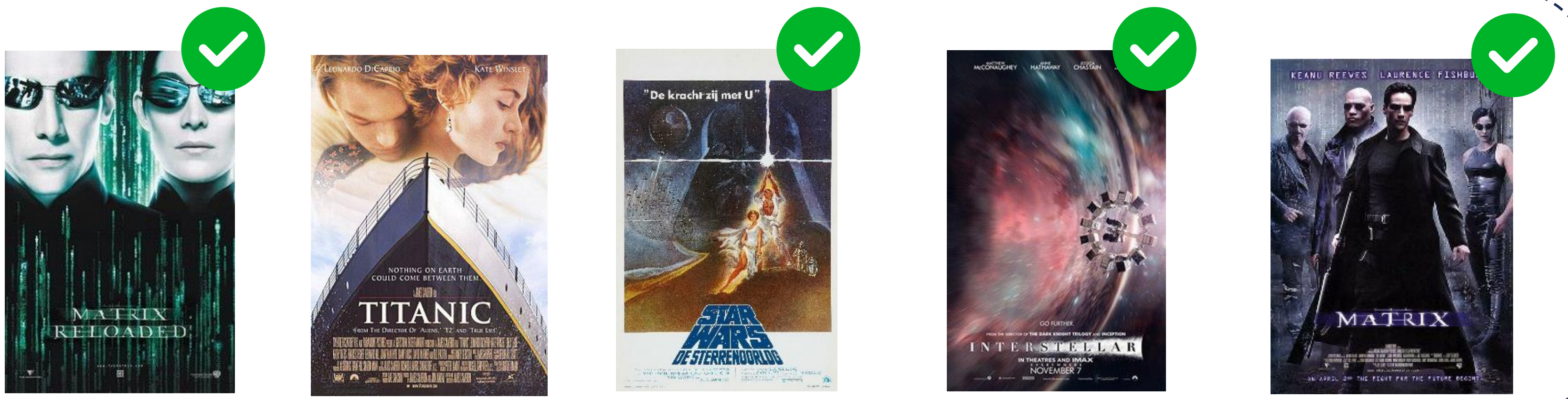
[3]Intelligence Indeed

Top-K Recommendation

Ranking 1 @5

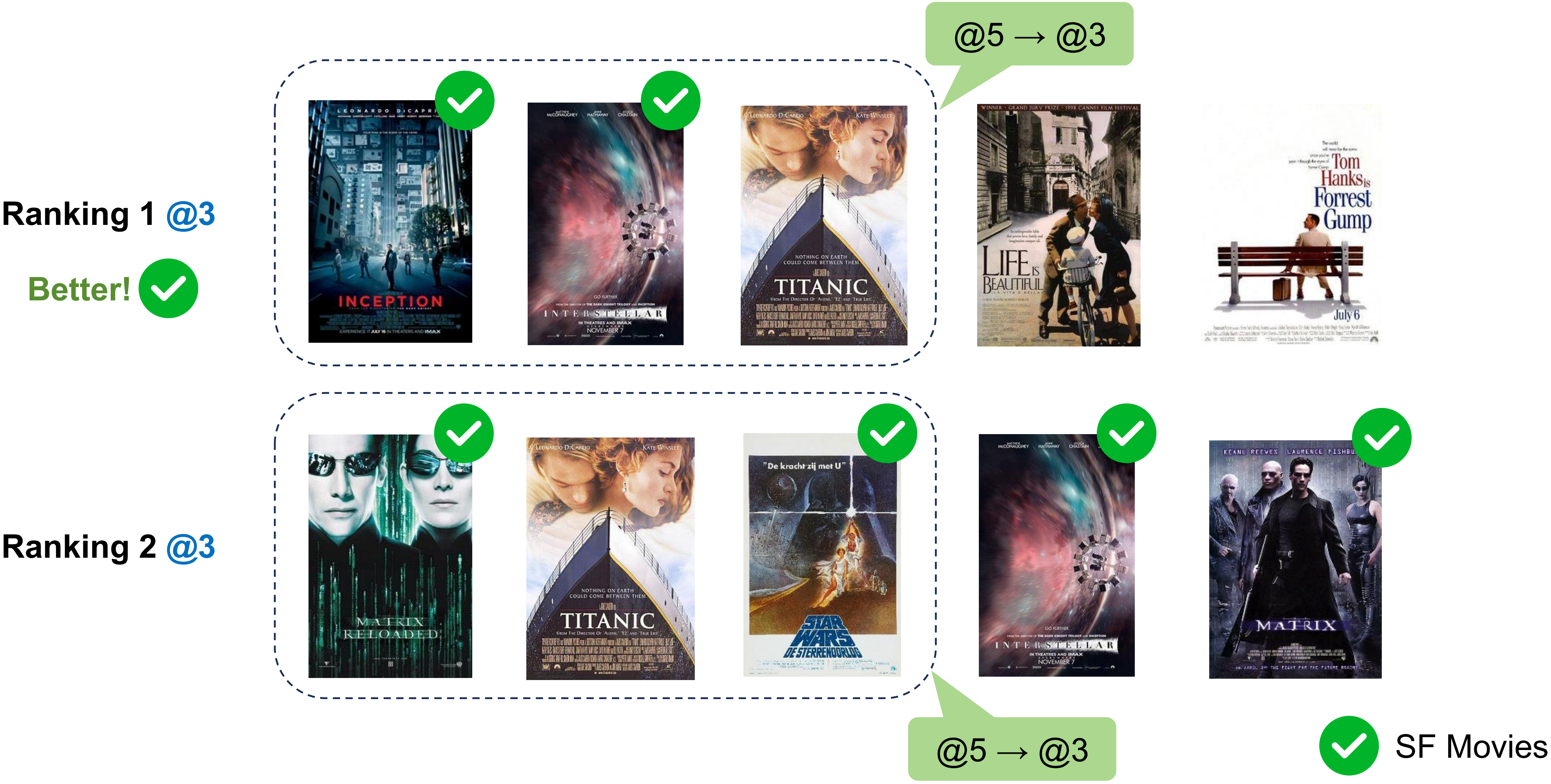
Ranking 2 @5

Better!



SF Movies

Top-K Recommendation



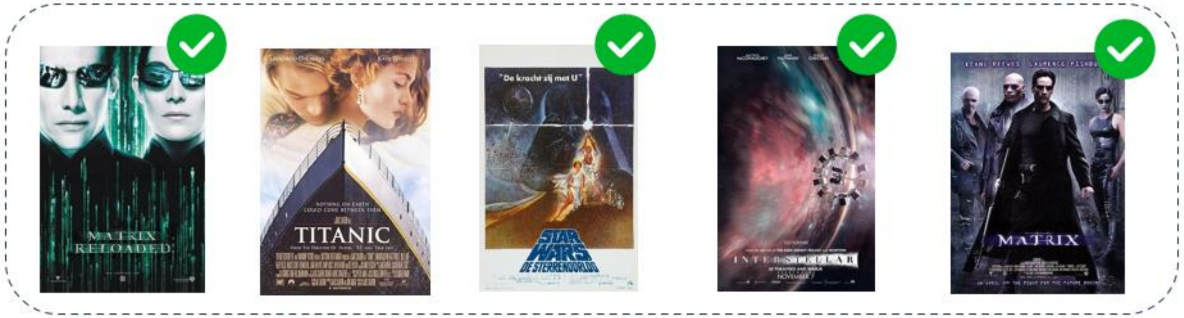
Even for the same ranking list,
***different** values of K may lead to*
***different** recommendation performance!*

NDCG Metric

NDCG (Normalized Discounted Cumulative Gain):

- One of the most representative **ranking metrics** in recommender systems (RS).
- Emphasizes accurate predictions at the **top** of the ranking list.
- Evaluates performance over the **entire** ranked list.

$$\mathrm{NDCG}(u) = \frac{\mathrm{DCG}(u)}{\mathrm{IDCG}(u)}, \quad \mathrm{DCG}(u) = \sum_{i \in \mathcal{P}_u} \frac{1}{\log_2(\pi_{ui} + 1)}$$



π_{ui} is the ranking position of item i in user u's recommendation list.

$\mathcal{P}_u$ is the set of positive items.

IDCG is the ideal DCG used for normalization.

NDCG@K Metric

NDCG@K (Normalized Discounted Cumulative Gain with Top-K Truncation):

- **Truncated** version of NDCG, which is used to evaluate the real-world RS.
- Only the **top K ranked items** are taken into account.
- NDCG is equivalent to NDCG@∞.

Top-K truncated

$$\mathrm{NDCG@}K(u) = \frac{\mathrm{DCG@}K(u)}{\mathrm{IDCG@}K(u)}, \quad \mathrm{DCG@}K(u) = \sum_{i \in \mathcal{P}_u} \frac{\mathbb{I}(\pi_{ui} \le K)}{\log_2(\pi_{ui} + 1)}$$

π_{ui} is the ranking position of item i in user u's recommendation list.

$\mathcal{P}_u$ is the set of positive items.

IDCG is the ideal DCG used for normalization.

NDCG Surrogate Loss: Softmax Loss

Softmax Loss (SL, a.k.a. Cross Entropy Loss in RS):

- **SOTA** recommendation loss.
- Initially derived from the MLE objective [1].
- Later proven to be **an upper bound of –log NDCG** [2].
- Minimizing SL leads to improved NDCG (a **theoretical guarantee**).

$$\mathcal{L}_{\mathrm{SL}}(u) = -\sum_{i \in \mathcal{P}_u} \log \left(\frac{\exp\left(s_{ui}/\tau\right)}{\sum_{j \in \mathcal{I}} \exp\left(s_{uj}/\tau\right)} \right)$$

τ is a temperature parameter controlling the sharpness of softmax operator.

s_{ui} is the predicted score of user u on item i.

Backbone	Loss	Health		Electronic		Gowalla		Book	
		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
MF	BPR	0.1627	0.1234	0.0816	0.0527	0.1355	0.1111	0.0665	0.0453
	SL	0.1719	0.1261	0.0821	0.0529	0.2064	0.1624	0.1559	0.1210

[1] Wu, Jiancan, et al. On the Effectiveness of Sampled Softmax Loss for Item Recommendation. TOIS 2024.
[2] Yang, Weiqin, et al. PSL: Rethinking and Improving Softmax Loss from Pairwise Perspective for Recommendation. NIPS 2024

*SL optimizes for **full ranking metric NDCG**.*

How can we specifically optimize for
***Top-K ranking metrics**, e.g., **NDCG@K**?*

Challenge 1: Top-K Truncation

$$\mathrm{NDCG@}K(u) = \frac{\mathrm{DCG@}K(u)}{\mathrm{IDCG@}K(u)}, \quad \mathrm{DCG@}K(u) = \sum_{i \in \mathcal{P}_u} \frac{\mathbb{I}(\pi_{ui} \le K)}{\log_2(\pi_{ui} + 1)}$$

How to handle the Top-K truncation term: $\mathbb{I}(\pi_{ui} \le K)$?

This term involves **estimating the ranking position** for each interaction,

which is **intractable** for large scale RS (complexity: $O(|UI \log I|)$, U/I is the size of user/item set).

Quantile Technique

How to handle the Top-K truncation term: $\mathbb{I}(\pi_{ui} \le K)$?

Given the **Top-K quantile** defined as

$$\beta_u^K := \inf\{s_{ui} : \pi_{ui} \le K\}$$

for each user u, then the Top-K truncation term can be rewritten as

$$\mathbb{I}(\pi_{ui} \le K) = \mathbb{I}(s_{ui} \ge \beta_u^K)$$

This term does not require the global ranking results;

it only needs to **estimate** the Top-K quantile β_u^K.

Quantile Estimation

How to to estimate the Top-K quantile β_u^K?

Method 1: Sample Quantile Estimation (complexity: $O(UN \log N)$)

- Sample N scores and sort them.
- Estimate the original Top-K quantile by the Top-$\left(\frac{K}{I}\right) * N$ quantile of the sampled scores.
- The estimation error decreases **exponentially** w.r.t. the sample size N.

Method 2: Quantile Regression (complexity: $O(UNT)$, T is interaction steps)

- Minimizing the following quantile regression loss:

$$\beta_u^K = \arg\min_{\beta} \mathcal{L}_{\mathrm{QR}}(\beta; u) := \sum_{i \in \mathcal{I}} \left(1 - \frac{K}{I}\right)(s_{ui} - \beta)_+ + \frac{K}{I}(\beta - s_{ui})_+$$

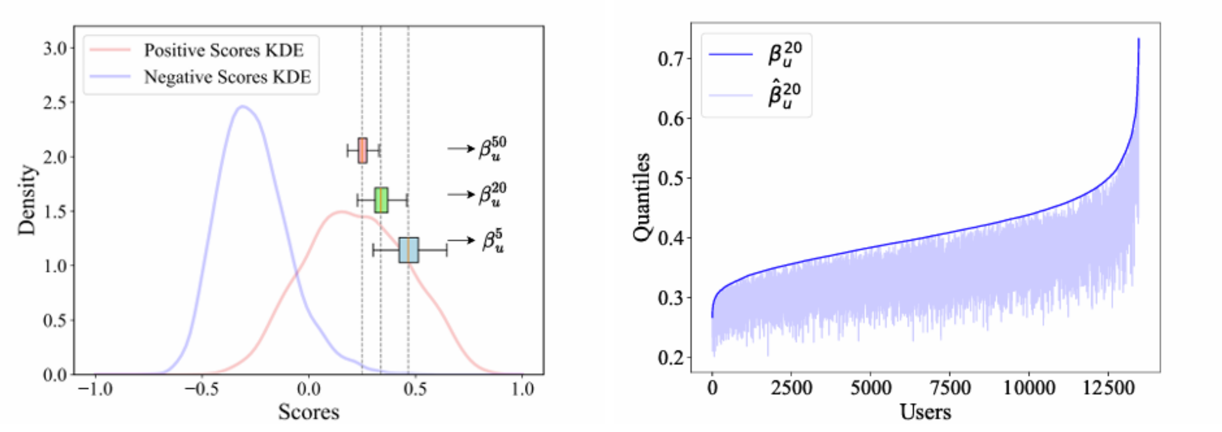
- The optimal solution is precisely the Top-K quantile.

Quantile Estimation

How to to estimate the Top-K quantile β_u^K?

Negative Sampling Trick for RS

- Instead of sampling from the entire item set, we **sample only from the negative items**.
- Because there is a significant **distributional difference** between positive and negative scores.
- **Accurate quantile estimation** can be achieved with a **small sample size** N.



(a) Quantile distribution. (b) Quantile estimation.

Figure 2: (a) Quantile distribution. The distributions of ideal quantiles β_u^{20} and the positive/negative scores are illustrated using Kernel Density Estimation (KDE) [54]. (b) Quantile estimation. The estimated quantile $\hat{\beta}_u^{20}$ and ideal quantile β_u^{20} are illustrated. The estimation error is 0.06 ± 0.03.

Challenge 2: Discontinuity

We derive an upper bound of –log DCG@K:

$$-\log \mathrm{DCG}@K(u) \leq \sum_{i \in \mathcal{P}_u} \delta(s_{ui} - \beta_u^K) \cdot \log \left(\sum_{j \in \mathcal{I}} \delta(d_{uij}) \right)$$

where $d_{uij} = s_{uj} - s_{ui}$, and $\delta(\cdot)$ is the Heaviside step function.

However, this bound is still **non-smooth**! We can smooth the two Heaviside functions with sigmoid and exp functions, respectively.

$$\mathcal{L}_{\mathrm{SL}@K}(u) = \sum_{i \in \mathcal{P}_u} \underbrace{\sigma_w(s_{ui} - \beta_u^K)}_{\text{weight: } w_{ui}} \cdot \underbrace{\log \left(\sum_{j \in \mathcal{I}} \sigma_d(d_{uij}) \right)}_{\text{SL term: } \mathcal{L}_{\mathrm{SL}}(u,i)}$$

$d_{uij} = s_{uj} - s_{ui}$

$\sigma_w(x) = 1/(1 + e^{-x/\tau_w})$ is the sigmoid function.

$\sigma_d(x) = e^{x/\tau_d}$ is the exponential function.

NDCG@K Surrogate Loss: SL@K

The resulted **SL@K loss** offers several desirable properties:

Property 1: Theoretical Guarantees

SL@K serves as an upper bound of –log NDCG@K --- minimizing SL@K leads to improved NDCG@K.

Theorem 3.2 (NDCG@K surrogate). *For any user u, if the Top-K hits $H_u^K > 1$*[4]*, SL@K serves as an upper bound of* $-\log \mathrm{DCG@}K$*, i.e.,*

$$-\log \mathrm{DCG@}K(u) \leq \mathcal{L}_{\mathrm{SL@}K}(u). \tag{3.7}$$

When the Top-K hits $H_u^K = 1$, a marginally looser yet effective bound holds, i.e., $-\frac{1}{2}\log \mathrm{DCG@}K(u) \leq \mathcal{L}_{\mathrm{SL@}K}(u)$.

NDCG@K Surrogate Loss: SL@K

Property 1: Theoretical Guarantees

SL@K achieves **superior performance** in NDCG@K optimization, with avg. Imp.% of **6.03%**!

Table 2: Top-20 recommendation performance comparison of SL@K with existing losses. The best results are highlighted in bold, and the best baselines are underlined. "Imp." denotes the improvement of SL@K over the best baseline.

Backbone	Loss	Health		Electronic		Gowalla		Book	
		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
MF	BPR	0.1627	0.1234	0.0816	0.0527	0.1355	0.1111	0.0665	0.0453
	GuidedRec	0.1568	0.1093	0.0644	0.0385	0.1135	0.0863	0.0518	0.0361
	SONG@20	0.0874	0.0650	0.0708	0.0444	0.1237	0.0970	0.0747	0.0542
	LLPAUC	0.1644	0.1209	0.0821	0.0499	0.1610	0.1189	0.1150	0.0811
	SL	0.1719	0.1261	0.0821	0.0529	0.2064	0.1624	0.1559	0.1210
	AdvInfoNCE	0.1659	0.1237	0.0829	0.0527	0.2067	0.1627	0.1557	0.1172
	BSL	0.1719	0.1261	0.0834	0.0530	0.2071	0.1630	0.1563	0.1212
	PSL	0.1718	0.1268	0.0838	0.0541	0.2089	0.1647	0.1569	0.1227
	SL@20 (Ours)	**0.1823**	**0.1390**	**0.0901**	**0.0590**	**0.2121**	**0.1709**	**0.1612**	**0.1269**
	Imp. %	+6.05%	+9.62%	+7.52%	+9.06%	+1.53%	+3.76%	+2.74%	+3.42%
LightGCN	BPR	0.1618	0.1203	0.0813	0.0524	0.1745	0.1402	0.0984	0.0678
	GuidedRec	0.1550	0.1073	0.0657	0.0393	0.0921	0.0686	0.0468	0.0310
	SONG@20	0.1353	0.0960	0.0816	0.0511	0.1261	0.0968	0.0820	0.0573
	LLPAUC	0.1685	0.1207	0.0831	0.0507	0.1616	0.1192	0.1147	0.0810
	SL	0.1691	0.1235	0.0823	0.0526	0.2068	0.1628	0.1567	0.1220
	AdvInfoNCE	0.1706	0.1264	0.0823	0.0528	0.2066	0.1625	0.1568	0.1177
	BSL	0.1691	0.1236	0.0823	0.0526	0.2069	0.1628	0.1568	0.1220
	PSL	0.1701	0.1270	0.0830	0.0536	0.2086	0.1648	0.1575	0.1233
	SL@20 (Ours)	**0.1783**	**0.1371**	**0.0903**	**0.0591**	**0.2128**	**0.1729**	**0.1625**	**0.1280**
	Imp. %	+4.51%	+7.95%	+8.66%	+10.26%	+2.01%	+4.92%	+3.17%	+3.81%
XSimGCL	BPR	0.1496	0.1108	0.0777	0.0508	0.1966	0.1570	0.1269	0.0905
	GuidedRec	0.1539	0.1088	0.0760	0.0473	0.1685	0.1277	0.1275	0.0951
	SONG@20	0.1378	0.0948	0.0525	0.0320	0.1367	0.0985	0.1281	0.0964
	LLPAUC	0.1519	0.1083	0.0781	0.0481	0.1632	0.1200	0.1363	0.1008
	SL	0.1534	0.1113	0.0772	0.0490	0.2005	0.1570	0.1549	0.1207
	AdvInfoNCE	0.1499	0.1072	0.0776	0.0489	0.2010	0.1564	0.1568	0.1179
	BSL	0.1649	0.1201	0.0800	0.0507	0.2037	0.1597	0.1550	0.1207
	PSL	0.1579	0.1143	0.0801	0.0507	0.2037	0.1593	0.1571	0.1228
	SL@20 (Ours)	**0.1753**	**0.1332**	**0.0869**	**0.0571**	**0.2095**	**0.1717**	**0.1624**	**0.1277**
	Imp. %	+6.31%	+10.91%	+8.49%	+12.40%	+2.85%	+7.51%	+3.37%	+3.99%

NDCG@K Surrogate Loss: SL@K

Property 1: Theoretical Guarantees

SL@K shows **closely alignment** with NDCG@K when K is varying.

Table 3: NDCG@K (D@K) comparisons with varying K on Health and Electronic datasets and MF backbone. The best results are highlighted in bold, and the best baselines are underlined. "Imp." denotes the improvement of SL@K over the best baseline.

Method	Health						Electronic					
	D@5	D@10	D@20	D@50	D@75	D@100	D@5	D@10	D@20	D@50	D@75	D@100
BPR	<u>0.0940</u>	0.1037	0.1234	<u>0.1621</u>	<u>0.1804</u>	<u>0.1925</u>	0.0345	0.0419	0.0527	0.0690	0.0777	0.0845
GuidedRec	0.0769	0.0881	0.1093	0.1484	0.1671	0.1811	0.0228	0.0294	0.0385	0.0551	0.0635	0.0703
SONG	0.0353	0.0392	0.0488	0.0709	0.0834	0.0930	0.0316	0.0393	0.0493	0.0661	0.0744	0.0803
SONG@K	0.0503	0.0535	0.0650	0.0896	0.1037	0.1135	0.0276	0.0349	0.0444	0.0581	0.0651	0.0706
LLPAUC	0.0887	0.0996	0.1209	0.1592	0.1765	0.1892	0.0305	0.0388	0.0499	0.0686	0.0778	<u>0.0848</u>
SL	0.0922	0.1037	0.1261	0.1620	0.1791	0.1924	0.0353	0.0430	0.0529	0.0696	0.0783	0.0845
AdvInfoNCE	0.0926	0.1038	0.1237	0.1608	0.1789	0.1920	0.0341	0.0423	0.0527	0.0697	0.0782	0.0843
BSL	0.0922	0.1037	0.1261	0.1620	0.1791	0.1924	0.0344	0.0425	0.0530	0.0691	0.0776	0.0843
PSL	<u>0.0940</u>	<u>0.1048</u>	<u>0.1268</u>	0.1613	0.1789	0.1912	<u>0.0356</u>	<u>0.0434</u>	<u>0.0541</u>	<u>0.0700</u>	<u>0.0784</u>	0.0845
SL@K (Ours)	**0.1080**	**0.1190**	**0.1390**	**0.1736**	**0.1916**	**0.2035**	**0.0402**	**0.0484**	**0.0590**	**0.0760**	**0.0844**	**0.0908**
Imp. %	+14.89%	+13.55%	+9.62%	+7.09%	+6.21%	+5.71%	+12.92%	+11.52%	+9.06%	+8.57%	+7.65%	+7.08%

NDCG@K Surrogate Loss: SL@K

Property 1: Theoretical Guarantees

The best NDCG@K' results are always achieved when $K = K'$ in SL@K.

Table 4: Performance exploration of SL@K on NDCG@K' with varying K and K'. The best results are highlighted in bold.

SL@K	Health						Electronic					
	D@5	D@10	D@20	D@50	D@75	D@100	D@5	D@10	D@20	D@50	D@75	D@100
SL@5	**0.1080**	0.1180	0.1379	0.1724	0.1906	0.2032	**0.0402**	0.0480	0.0583	0.0753	0.0839	0.0900
SL@10	0.1077	**0.1190**	0.1377	0.1734	0.1909	0.2028	0.0400	**0.0484**	0.0583	0.0755	0.0839	0.0901
SL@20	0.1076	0.1188	**0.1390**	0.1733	0.1909	0.2029	0.0400	0.0483	**0.0590**	0.0759	0.0837	0.0900
SL@50	0.1062	0.1167	0.1364	**0.1736**	0.1901	0.2020	0.0398	0.0481	0.0587	**0.0760**	0.0842	0.0907
SL@75	0.1073	0.1179	0.1387	0.1734	**0.1916**	0.2031	0.0397	0.0481	0.0587	0.0759	**0.0844**	0.0907
SL@100	0.1071	0.1177	0.1375	0.1727	0.1904	**0.2035**	0.0399	0.0481	0.0587	0.0759	0.0843	**0.0908**
SL (@∞)	0.0922	0.1037	0.1261	0.1620	0.1791	0.1924	0.0353	0.0430	0.0529	0.0696	0.0783	0.0845

NDCG@K Surrogate Loss: SL@K

Property 2: Ease of Implementation

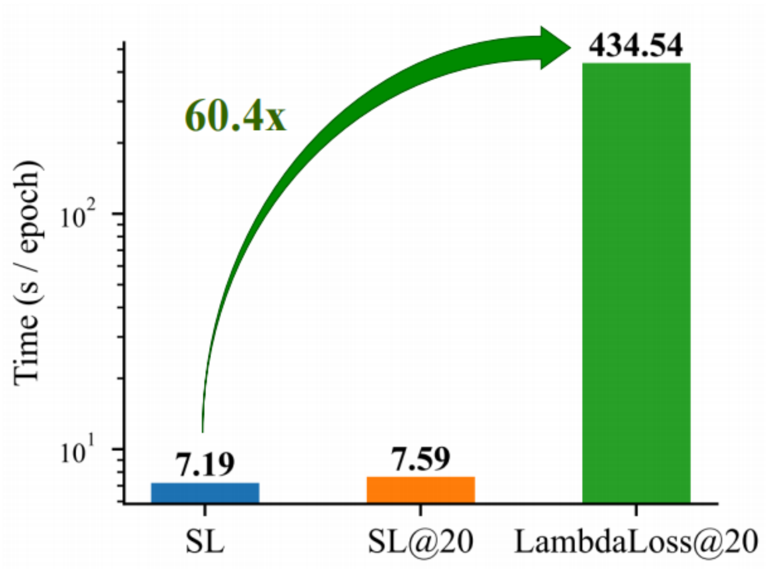
SL@K is a **weighted SL**, which introduces only a quantile-based weight w_{ui}.
Thus, SL@K can be seamlessly integrated into existing RS with **minimal modifications**.

$$\mathcal{L}_{\mathrm{SL@}K}(u) = \sum_{i \in \mathcal{P}_u} \underbrace{\sigma_w(s_{ui} - \beta_u^K)}_{\text{weight: } w_{ui}} \cdot \underbrace{\log\left(\sum_{j \in \mathcal{I}} \sigma_d(d_{uij})\right)}_{\text{SL term: } \mathcal{L}_{\mathrm{SL}}(u,i)}$$

NDCG@K Surrogate Loss: SL@K

Property 3: Computational Efficiency

- **SL complexity**: $O(UPN)$
  - where U is the size of user set, P is the number of positive items, N is the negative sampling size.
- **SL@K complexity**: $O(UPN + UN\log N)$
  - The additional computational overhead is **typically negligible** in practice.

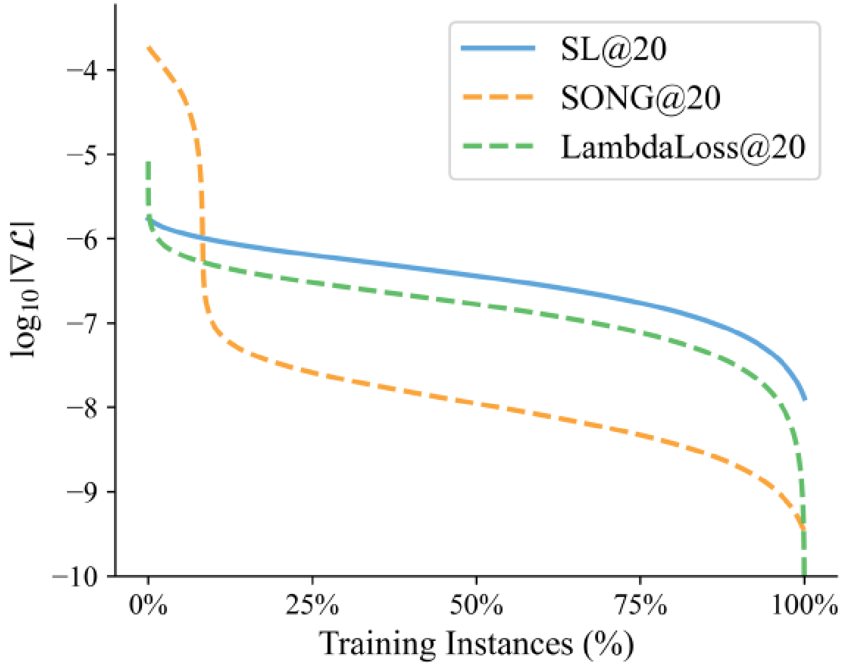


(b) Execution time.

NDCG@K Surrogate Loss: SL@K

Property 4: Gradient Stability

- The weight w_{ui} is bounded ($\in (0.1, 1)$ when $\tau_w \geq 1$)
- SL@K will **not** significantly amplify the gradient variance.
- Training other NDCG@K surrogate losses is **instable** in RS.



(c) Gradient distribution.

NDCG@K Surrogate Loss: SL@K

Property 5: False Positive Noise Robustness

- **False positive noise** is prevalent in RS, e.g., clickbait, item position bias, or accidental interactions.
- The weight w_{ui} in SL@K helps mitigate this issue, since false positives tends to have **lower score** s_{ui}, resulting in **more insignificant contributions** in SL@K.

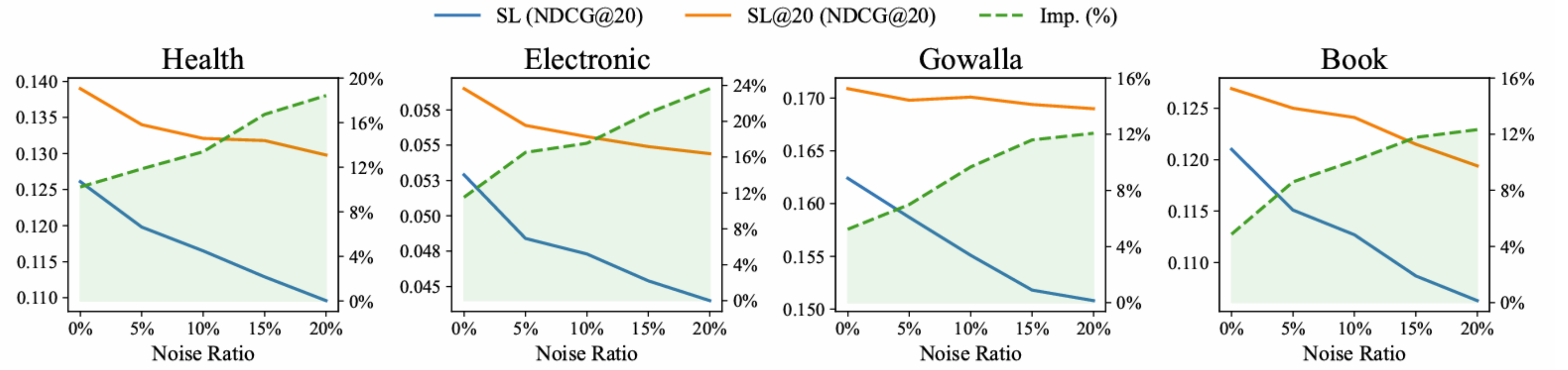


Figure 8: NDCG@20 performance of SL@K compared with SL under varying ratios of imposed false positive instances. "Noise Ratio" denotes the ratio of false positive instances. "Imp." indicates the improvement of SL@K over SL.

Conclusion & Future Work

Contributions:

- We propose a **novel recommendation loss, SL@K**, tailored for **NDCG@K optimization** with theoretical guarantees and consistent & significant performance improvement in practice.
- SL@K is a **weighted SL**, which is elegant in form and introduces minimal additional complexity.

Future Work:

- More effective and accurate **quantile estimation**.
- **Variance analysis** of SL@K.
- Application in **other domains** with Top-K metrics.

Thank You!

浙江大學
ZHEJIANG UNIVERSITY

1st auth. Email: futurelover10032@gmail.com

GitHub

Weiqin Yang
(1st auth.)

Jiawei Chen
(corr. auth.)