ICML
International Conference
On Machine Learning

Learning Cascade Ranking as One Network

Yunli Wang, Zhen Zhang, Zhiqiang Wang, Zixuan Yang, Yu Li, Jiang Yang, Shiyang Wen, Peng Jiang, Kun Gai

2025.07

What is cascade ranking?

Notations:

$\mathscr{M}_i$: The i-th stage in cascade ranking

$\mathscr{Q}_i$: The sample space of the i-th stage

q_i : The size of $\mathscr{Q}_i$

$\mathscr{I}$: The impression space

$\mathscr{F}^{\downarrow}_{\mathscr{M}}(S)$: The Ordered list sorted by the score of model $\mathscr{M}$ (on space S) in descending order

$\mathscr{F}^{\downarrow}_{\mathscr{M}}(S)[:K]$: The top K set of $\mathscr{F}^{\downarrow}_{\mathscr{M}}(S)$

$\mathscr{F}^{\downarrow}_{\mathscr{M}_T}((\cdots\mathscr{F}^{\downarrow}_{\mathscr{M}_1}(\mathscr{Q}_0)[:q_1]\cdots))[:q_T]$: The outputs of the system, named CS_{out}

T: total stage number of the cascade ranking

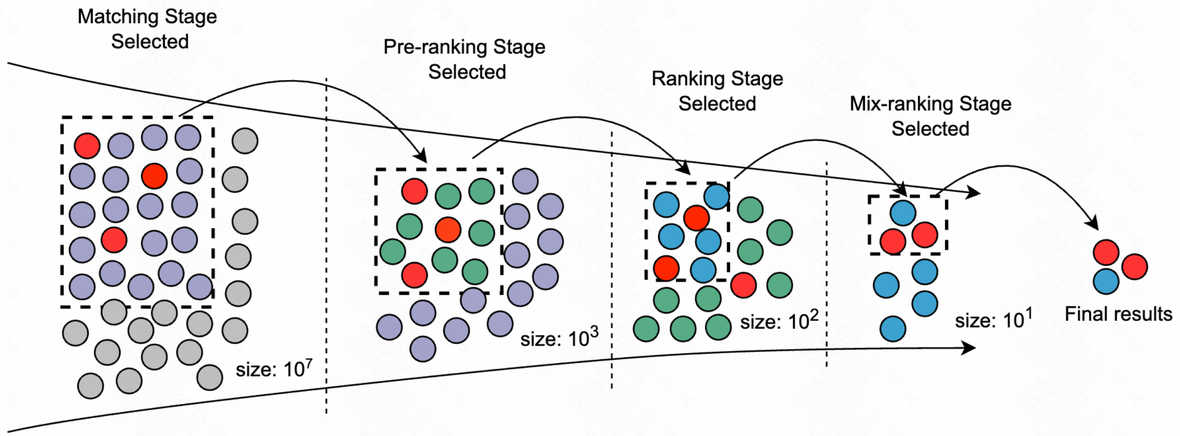


Figure 1. A typical cascade ranking architecture, including four stages: Matching, Pre-ranking, Ranking, and Mix-ranking. The red points represent the ground truth for the selection.

The Goal of Cascade Ranking

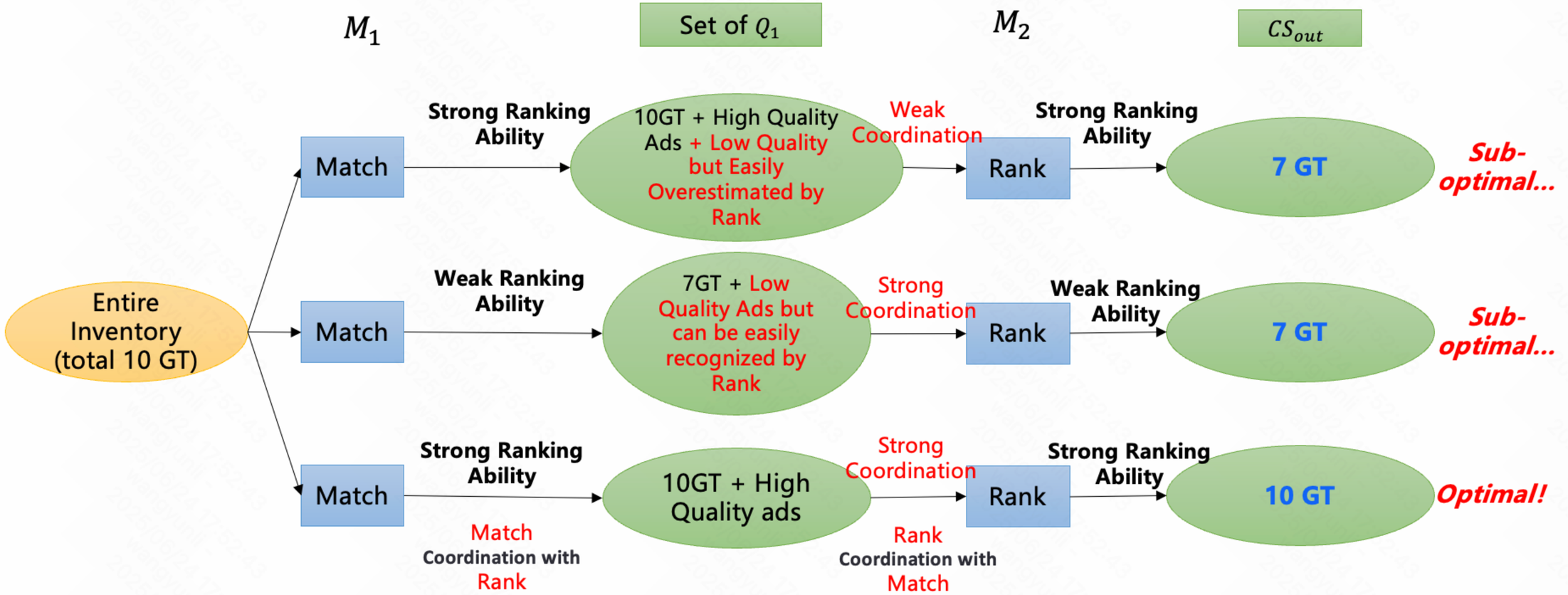
The goal of training paradigms for cascade ranking is to optimize the end-to-end Recall of CS_{gt} using training data collected from the system:

$$Recall@\mathscr{K}@q_T = \frac{\sum_{i=1}^{q_T} 1(item_i \in CS_{out}) 1(item_i \in CS_{gt})}{\sum_{j=1}^{\mathscr{K}} 1(item_j \in CS_{gt})}$$

Notations:

- CS_{gt} : the ground truth set of items considered relevant or optimal based on user feedback or expert annotations.
- $\mathscr{K}$: the size of CS_{gt}, and $\mathscr{K} \leq q_T$.
- $1(\cdot)$: the indicator function.

Key Requirements for Efficient Cascade Ranking



Efficient Cascade Ranking = Strong Ranking Ability of Per Stage + Strong Coordination and Complementarity Between Stages

- **Match Coordination with Rank**: Preemptively avoid easily overestimated ads of “Rank” stage.
- **Rank Coordination with Match**: Accurately identify ground-truth items in output set of the “Match” stage.

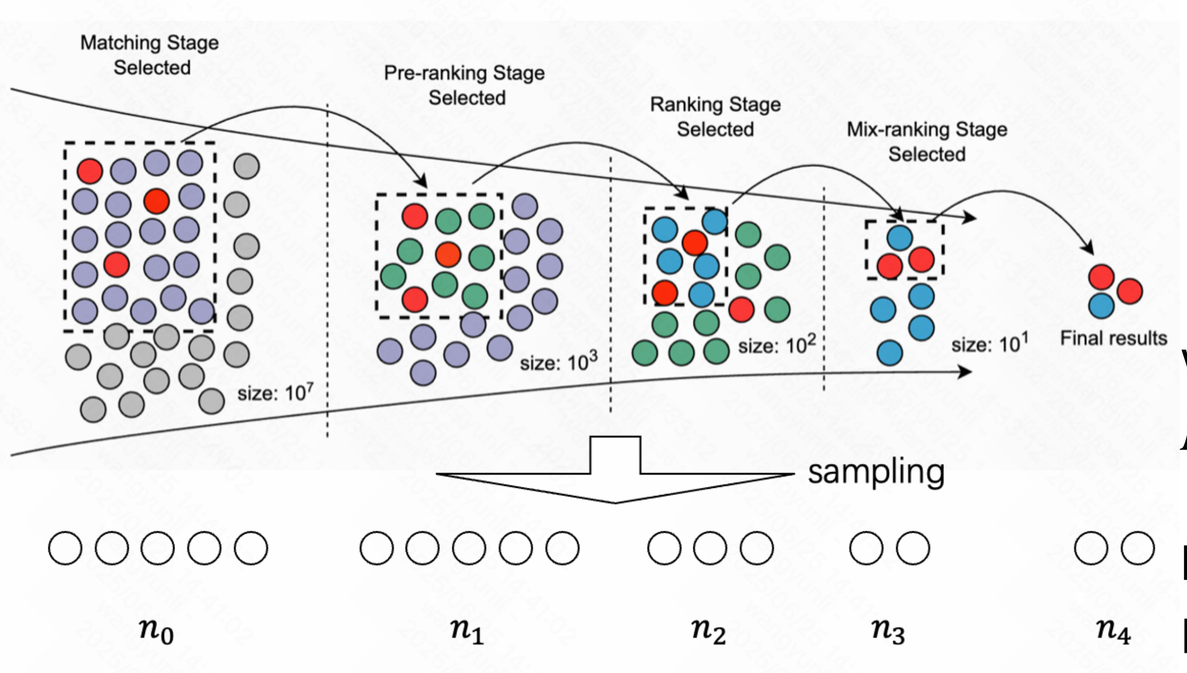
Key Challenges for Traditional Methods

- **Misalignment of Training Objectives**: Most of existing methods do not directly optimize the end-to-end recall of cascade ranking.
- **Lack of Learning to Collaborate**: Some existing methods lack the ability to learn these interactions and collaborations, especially for methods that separately train different stages.

To the best of our knowledge, no existing approach simultaneously addresses both challenges, highlighting the need for a more comprehensive solution.

To address these issues, we propose **LCRON**, which is the abbreviation of "**L**earning **C**ascade **R**anking as **O**ne **N**etwork".

LCRON—Sample Organization



Full-stage training samples:

The training set is:

$$D = \left(u, \{(item_j, y_j) \mid 0 \le j < N\}\right)$$

$$= \left(u, \bigcup_{i=0}^{3} \{(item_j, y_j) \mid 0 \le j < n_i \textbf{ and } item_j \in \mathcal{Q}_i\}\right)$$

where n_i is the number of samples drawn from $\mathcal{Q}_i$, and $N = n_0 + n_1 + n_2 + n_3$.

List-wise Supervision:

For any two pairs $(u, item_i)$ and $(u, item_j)$, the label y_i and y_j satisfy the following condition:

$$1(y_i > y_j) = 1(\mathcal{S}_i > \mathcal{S}_j) \vee \left(1(\mathcal{S}_i = \mathcal{S}_j) \wedge 1(R_i > R_j)\right)$$

where $\mathcal{S}_i > \mathcal{S}_j$ indicates that $item_i$ belongs to a later stage than $item_j$, and $R_i > R_j$ indicates that $item_i$ has a higher rank than $item_j$ within the same stage.

LCRON—End-to-end Recall Optimization

Notations:

Let $P_{\mathscr{M}_i}^{q_i}$ represent the probability vector of each ad in D being selected by the cascade ranking for top-q_i selection.

Let $\pi \in \{0,1\}^N$ denote the sampling result from $P_{\mathscr{M}_1}^{q_1}$, and let P_π represent the probability of sampling π.

$\langle \cdot, \cdot \rangle$ denotes dot product, and $\odot$ denotes element-wise product.

$\mathbf{1}$ represents a vector where all elements are 1.

LCRON—End-to-end Recall Optimization

Take two-stage cascade ranking as an example, the **survival probability of ground-truth items** can be formulated as:

$$
\begin{aligned}
P_{CS}^{q_2} &= \mathbb{E}_{\pi \sim P_\pi} \frac{P_{\mathscr{M}_2}^{q_2} \odot \pi}{\langle \pi, P_{\mathscr{M}_2}^{q_2} \rangle / \langle 1, P_{\mathscr{M}_2}^{q_2} \rangle} && \text{Intractable} \\
&\geq \mathbb{E}_{\pi \sim P_\pi} P_{\mathscr{M}_2}^{q_2} \odot \pi && \text{Lower Bound} \\
&= \prod_i^2 P_{\mathscr{M}_i}^{q_i} && \text{The denominator is always less than or equal to 1.} \\
&= \widehat{P_{CS}^{q_2}} .
\end{aligned}
$$

Thus, we can optimize the lower bound of $P_{CS}^{q_2}$ to improve the probability that ground-truth items appears in CS_{out}, namely to directly optimize end-to-end recall.

LCRON—End-to-end Recall Optimization

Then, the end-to-end loss is formulated as:

$$
\begin{aligned}
L_{e2e} &= -\sum_i (y_i \ln((\widehat{P^{q_2}_{CS}})_i) + (1-y_i) ln(1-(\widehat{P^{q_2}_{CS}})_i) \\
&= -\sum_i (y_i \ln((\prod_i^2 P^{q_i}_{\mathscr{M}_i})_i) + (1-y_i) ln(1-(\prod_i^2 P^{q_i}_{\mathscr{M}_i})_i) \\
&= -\sum_j y_j ln(\prod_i^2 \frac{\sum_{j=1}^{q_i} (\hat{\mathscr{P}}^{\downarrow}_{\mathscr{M}_i})_{j,:}}{\oslash\ sp(\sum_{t=1} (\hat{\mathscr{P}}^{\downarrow}_{\mathscr{M}_i})_{t,:})} - \sum_j (1-y_j) ln(1-\prod_i^2 \frac{\sum_{j=1}^{q_i} (\hat{\mathscr{P}}^{\downarrow}_{\mathscr{M}_i})_{j,:}}{\oslash\ sp(\sum_{t=1} (\hat{\mathscr{P}}^{\downarrow}_{\mathscr{M}_i})_{t,:})})
\end{aligned}
$$

where $\hat{\mathscr{P}}^{\downarrow}_{\mathscr{M}_i} \in [0,1]^{N \times N}$ is the soft permutation matrix for model $\mathscr{M}_i$, where $(\hat{\mathscr{P}}^{\downarrow}_{\mathscr{M}_i})_{j,k}$ represents the soft probability that item k is ranked at position j.

Soft permutation matrix can be obtained by differentiable sorting techniques, such as NeuralSort.

LCRON—Auxiliary Loss for Tightening the Bound

The gap between $P_{CS}^{q_2}$ and $\widehat{P_{CS}^{q_2}}$ can be formulated as:

$$
\begin{aligned}
\Delta &= P_{CS}^{q_2} - \widehat{P_{CS}^{q_2}} \\
&= \mathbb{E}_{\pi \sim P_\pi} \left[\frac{P_{\mathscr{M}_2}^{q_2} \odot \pi}{\langle \pi, P_{\mathscr{M}_2}^{q_2} \rangle / q_2} - P_{\mathscr{M}_2}^{q_2} \odot \pi \right] \\
&= \mathbb{E}_{\pi \sim P_\pi} \left[P_{\mathscr{M}_2}^{q_2} \odot \pi \left(\frac{q_2}{\langle \pi, P_{\mathscr{M}_2}^{q_2} \rangle} - 1 \right) \right] \\
&\leq \mathbb{E}_{\pi \sim P_\pi} \left[P_{\mathscr{M}_2}^{q_2} \left(\frac{q_2}{\langle \pi, P_{\mathscr{M}_2}^{q_2} \rangle} - 1 \right) \right] \\
&= \left[P_{\mathscr{M}_2}^{q_2} \left(\frac{q_2}{\mathbb{E}_{\pi \sim P_\pi} \langle \pi, P_{\mathscr{M}_2}^{q_2} \rangle} - 1 \right) \right] \\
&= \left[P_{\mathscr{M}_2}^{q_2} \left(\frac{q_2}{\langle P_{\mathscr{M}_1}^{q_1}, P_{\mathscr{M}_2}^{q_2} \rangle} - 1 \right) \right] \\
&= \Delta'
\end{aligned}
$$

If the top q_2 sets of the two models are consistent, it helps to reduce Δ'.

To tighten the bound of L_{e2e}, we design **single-stage loss** $L_{single}^{\mathscr{M}_i}$ directly optimizes the ranking consistency of each model with the same supervision, thereby implicitly aligning $\mathscr{M}_1(D)$ and $\mathscr{M}_2(D)$. This helps to reduce Δ', indirectly optimizing the bound Δ.

The auxiliary loss can be formulated as:

$$
L_{single}^{\mathscr{M}_i} = -\sum_j y_j ln\left(\frac{\sum_{j=1}^{\mathscr{K}} (\hat{\mathscr{P}}^{\downarrow}_{\mathscr{M}_i})_{j,:}}{\oslash\ sp(\sum_{t=1} (\hat{\mathscr{P}}^{\downarrow}_{\mathscr{M}_i})_{t,:})}\right) - \sum_j (1 - y_j) ln\left(1 - \frac{\sum_{j=1}^{\mathscr{K}} (\hat{\mathscr{P}}^{\downarrow}_{\mathscr{M}_i})_{j,:}}{\oslash\ sp(\sum_{t=1} (\hat{\mathscr{P}}^{\downarrow}_{\mathscr{M}_i})_{t,:})}\right)
$$

Experiments—Overview

- Public Experiments
  - Experimental Setup
  - Results on two-stage cascade ranking
  - Results on three-stage cascade ranking
  - Runtime and Space Complexity Analysis
- Online Experiments
  - Experiment Setup
  - Online A/B Results

Public Experiments—Experimental Setup

- **Benchmarks**: RecFlow, the only public benchmark that collects data from all stages of real-world cascade ranking systems.

- **Evaluation Metrics**:
  - Golden Metric: End-to-end Recall
  - Auxiliary Observation Metrics: Recall & NDCG for each single stage

- **Baselines**: BCE, ICC, RankFlow, FS-RankNet, FS-LambdaLoss, ARF, ARF-v2

Results on two-stage cascade ranking

- Architecture: DSSM (Retrieval Stage) + DIN (Ranking Stage)

Table 2. Main results of public experiments on RecFlow. Each method was run 5 times, and the results are reported as mean±std. $*$ indicates the best results. Bold numbers indicate that LCRON shows statistically significant improvements over the baselines, as determined by a t-test at the 5% significance level. Note that the *Recall*@10@20 of *Joint* is the golden metric for the whole cascade ranking system. The test set is the last day, with the remaining data used for training.

Method/Metric	Joint	Ranking		Retrieval	
	Recall@10@20 ↑	Recall@10@20 ↑	NDCG@10 ↑	Recall@10@30 ↑	NDCG@10 ↑
BCE	0.8539±0.0006	0.8410±0.0007	0.7043±0.0008	0.9706±0.0004*	0.7150±0.0019
ICC	0.8132±0.0003	0.8100±0.0003	0.6980±0.0003	0.9288±0.0003	0.6155±0.0003
RankFlow	0.8647±0.0007	0.8629±0.0006	0.7274±0.0010	0.9656±0.0006	0.7087±0.0003
FS-RankNet	0.7881±0.0007	0.7908±0.0008	0.6864±0.0004	0.9321±0.0004	0.6710±0.0005
FS-LambdaLoss	0.8666±0.0016	0.8660±0.0018	0.7306±0.0027	0.9691±0.0004	0.7190±0.0027*
ARF	0.8608±0.0006	0.8616±0.0007	0.6655±0.0027	0.9631±0.0008	0.5437±0.0110
ARF-v2	0.8678±0.0009	0.8679±0.0009	0.7269±0.0005	0.9684±0.0006	0.7152±0.0028
LCRON (ours)	**0.8732±0.0005***	**0.8729±0.0004***	0.7291±0.0008	0.9700±0.0004	0.7151±0.0009

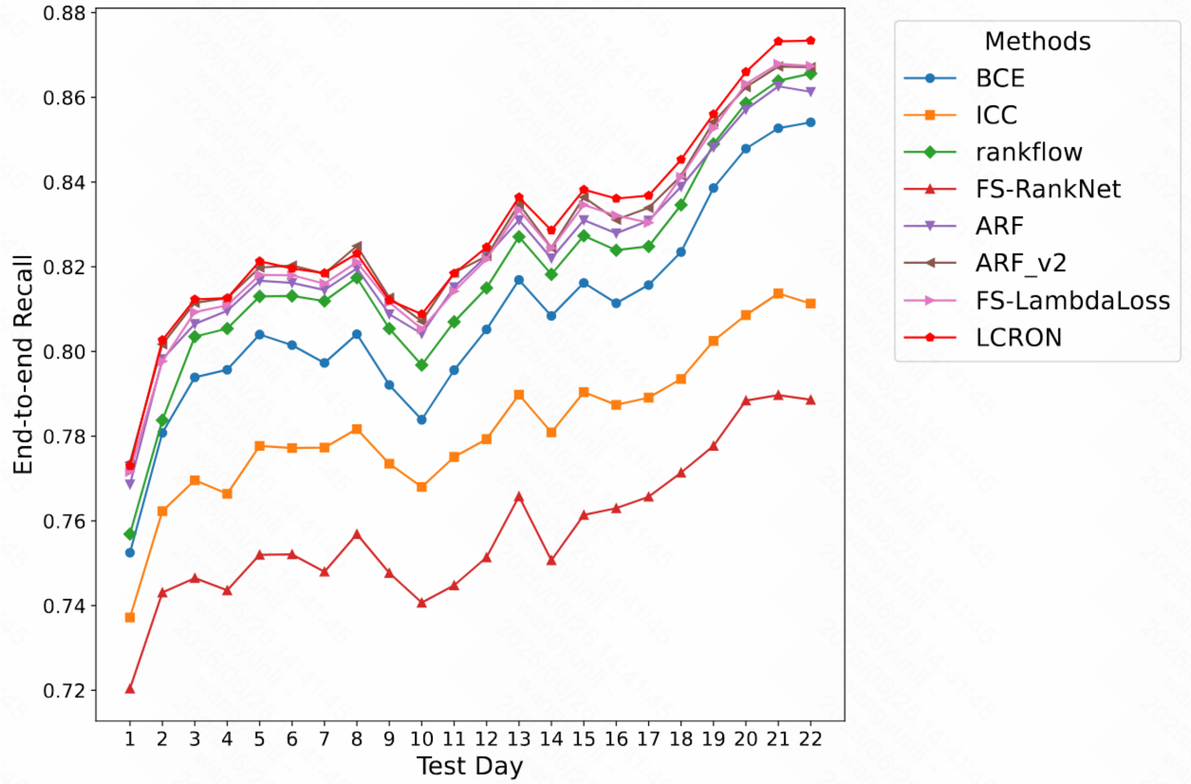


Figure 2. The evaluation results of different methods on RecFlow, in a streaming manner.

Results on three-stage cascade ranking

- Architecture: DSSM (Retrieval Stage) + MLP (Pre-ranking Stage) + DIN (Ranking Stage)

Table 6. Experimental results for three-stage cascade ranking. Each method was run 5 times, and the results are reported as mean±std. $*$ indicates the best results. Bold numbers indicate that LCRON shows statistically significant improvements over the baselines, as determined by a t-test at the 5% significance level. The test set is the last day, with the remaining data used for training.

Method/Metric	Joint	Ranking		Pre-ranking		Retrieval	
	Recall@10@20 ↑	Recall@10@20 ↑	NDCG@10 ↑	Recall@10@30 ↑	NDCG@10 ↑	Recall@10@40 ↑	NDCG@10 ↑
BCE	0.7191±0.0005	0.6574±0.0011	0.5714±0.0009	0.8814±0.0009	0.6382±0.0007	0.9709±0.0006*	0.6350±0.0019*
ICC	0.6386±0.0071	0.6196±0.0120	0.5794±0.0038	0.7682±0.0408	0.4925±0.0463	0.8526±0.0467	0.4754±0.0679
RankFlow	0.7308±0.0005	0.7230±0.0008	0.6400±0.0008	0.8729±0.0014	0.6396±0.0008	0.9611±0.0008	0.6265±0.0013
FS-RankNet	0.6200±0.0010	0.6224±0.0008	0.5756±0.0006	0.8038±0.0006	0.5733±0.0008	0.9373±0.0008	0.5678±0.0008
FS-LambdaLoss	0.7319±0.0038	0.7292±0.0042	0.6431±0.0014*	0.8803±0.0029	0.6443±0.0024*	0.9662±0.0012	0.6297±0.0022
ARF	0.7256±0.0004	0.7251±0.0005	0.5675±0.0036	0.8712±0.0008	0.5099±0.0074	0.9612±0.0004	0.4268±0.0031
ARF-v2	0.7332±0.0020	0.7285±0.0051	0.6430±0.0015	0.8777±0.0064	0.6438±0.0031	0.9649±0.0029	0.6284±0.0039
LCRON	**0.7390±0.0008***	0.7338±0.0008*	0.6017±0.0009	0.8859±0.0007*	0.6010±0.0031	0.9678±0.0012	0.5758±0.0055

- Conclusions:
  - LCRON consistently outperforms all the baselines on end-to-end recall under different settings, showing the effectiveness and robustness of our method.
  - LCRON does not dominate all individual stage metrics (e.g., the Recall of the Ranking model), it achieves substantial improvements in joint metrics, highlighting the importance of stage collaboration.

Runtime and Space Complexity Analysis

	Time Complexity	Space Complexity	Runtime of two-stage experiments	GPU Memory Usage of two-stage experiments
BCE	O(DN)	O(DN)	5358s	37.7GB
ICC	O(DN^2)	O(DN^2)	5376s	38.2GB
RankFlow	O(DN)	O(DN)	5057s	38.2GB
FS-RankNet	O(DN^2)	O(DN^2)	5104s	37.7GB
FS-LambdaLoss	O(DN^2)	O(DN^2)	5076s	38.2GB
ARF	O(DN^2)	O(DN^2)	5362s	37.3GB
ARF-v2	O(DN^2)	O(DN^2)	5573s	37.3GB
LCRON	O(DN^2)	O(DN^2)	5418s	38.2GB

D is impression number, N is the number of items per impression

Conclusions: In real-world applications, N is usually small, LCRON does not introduce significant computational or memory overhead compared to existing methods, making it well-suited for industrial deployment.

Online Experiments

- Architecture: DSSM (Retrieval Stage) + MLP (Pre-ranking Stage)

Table 5. Industrial experimental results for 15 days on a real-world advertising system. Each method was allocated 10% of the online traffic. For online metrics, we calculate the relative improvement of other methods compared to FS-LambdaLoss as the baseline.

Method/Metric	Offline Metrics	Online Metrics	
	Joint Recall	Revenue	Ad Conversions
FS-LambdaLoss	0.8210	–	–
ARF-v2	0.8237	+1.66%	+0.65%
LCRON (ours)	0.8289	+4.1%	+1.6%

Conclusions:

- LCRON model achieves significant improvements in both revenue and ad conversions compared to the baselines.
- Based on the results of rigorous A/B testing, LCRON has been fully rolled out on the Kuaishou advertising platform since January 2025. The two models trained under LCRON have successfully replaced the primary pathways (i.e., those with the highest weight) in the Matching and Pre-ranking stages, marking a major milestone in its industrial deployment.

Further Information & Contact

For more details, please refer to our paper and GitHub repository. We have included QR codes below for easy access.

Paper

GitHub

If you have any questions need further information, please feel free to contact us.

We are also hiring! We’re actively looking for talented researchers and developers to join our team. If you're interested in AI and advertising systems, and want to work on impactful projects, don't hesitate to reach out!

Contact E-mail: wangyunli@kuaishou.com

Thanks for listening!