Taobao Display Advertising

Some recent tech advances

Jian Xu
Alibaba Group

阿里妈妈

AGENDA

1. Taobao Display Advertising
2. Matching
3. Prediction
4. Strategies

Taobao Display Advertising

阿里妈妈

Taobao

600M+ Mobile MAU*

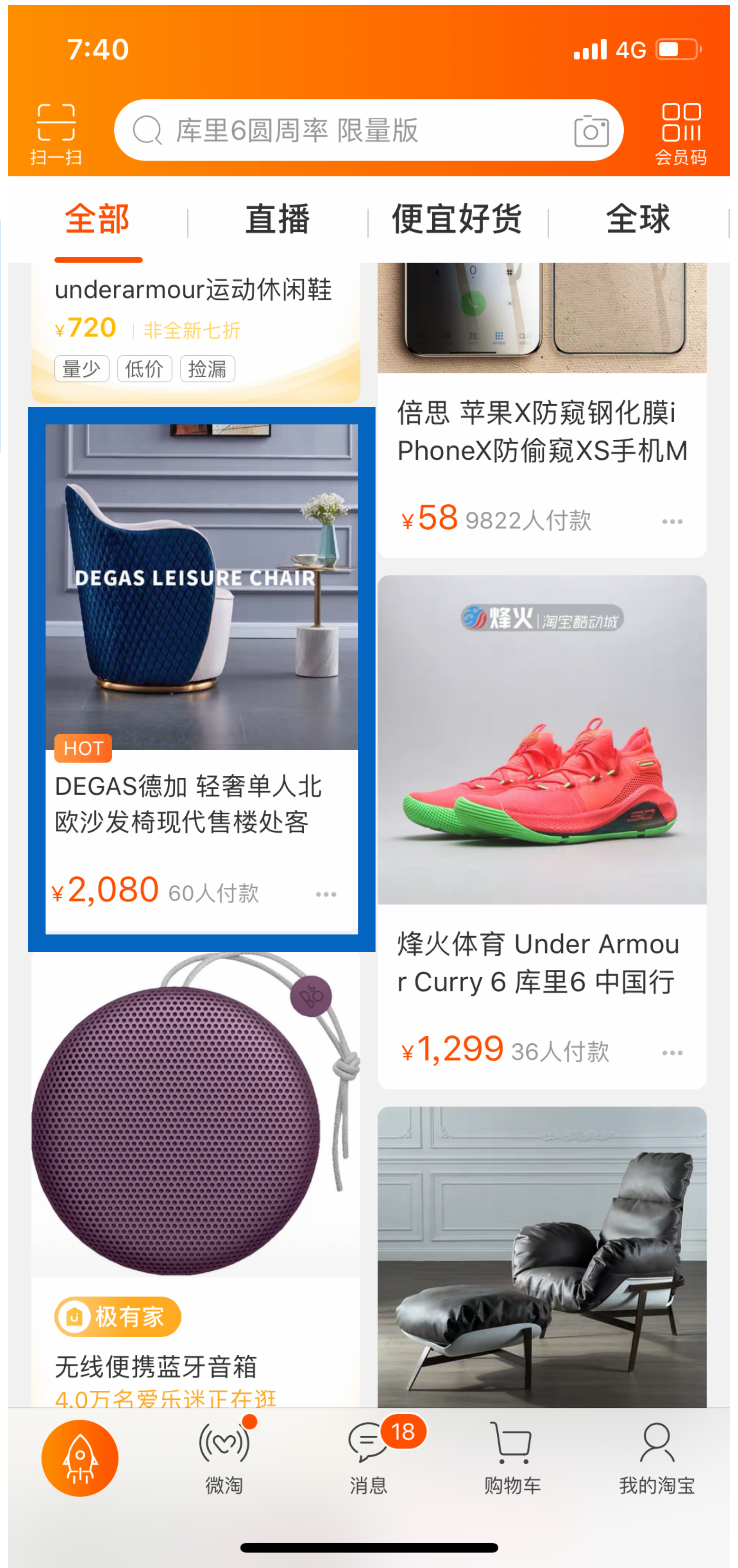
Alimama

Commercial Advertising Solutions

* Data source: https://www.alibabagroup.com/en/ir/presentations/Investor_Day_2018_Taobao.pdf

阿里妈妈

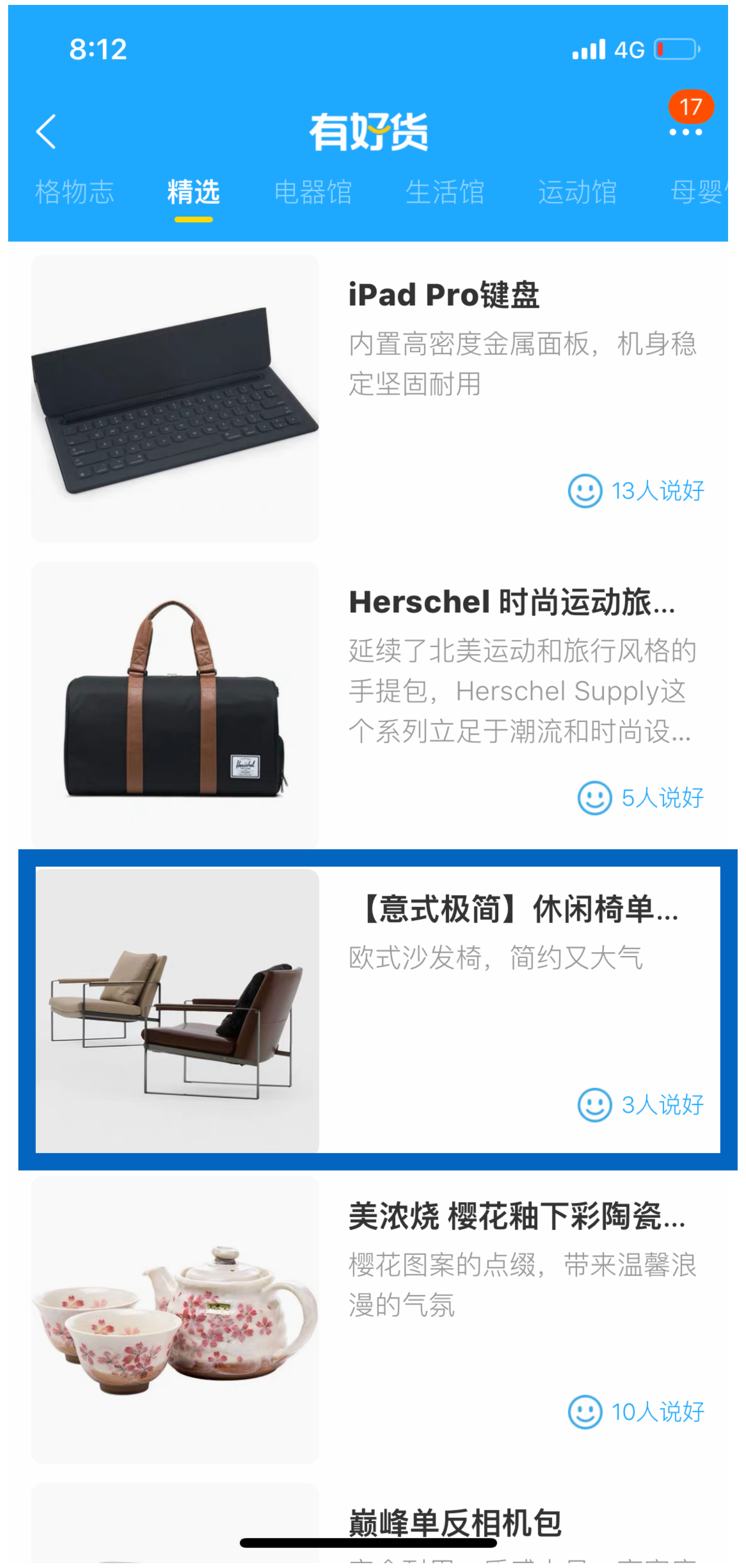
Banner Ads



Feed Ads
Guess What You Like

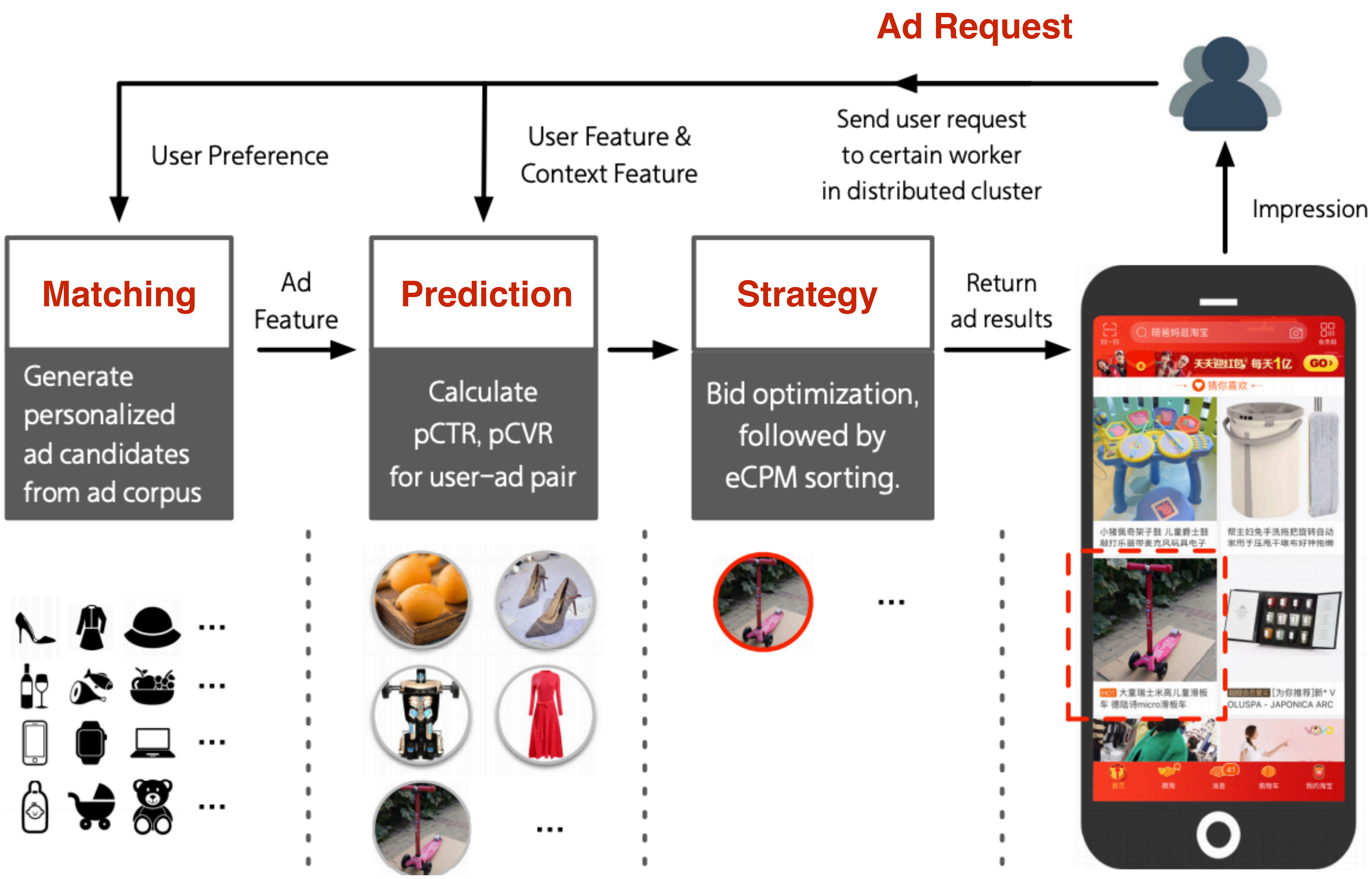
Feed Ads
Weitao

Item Ads

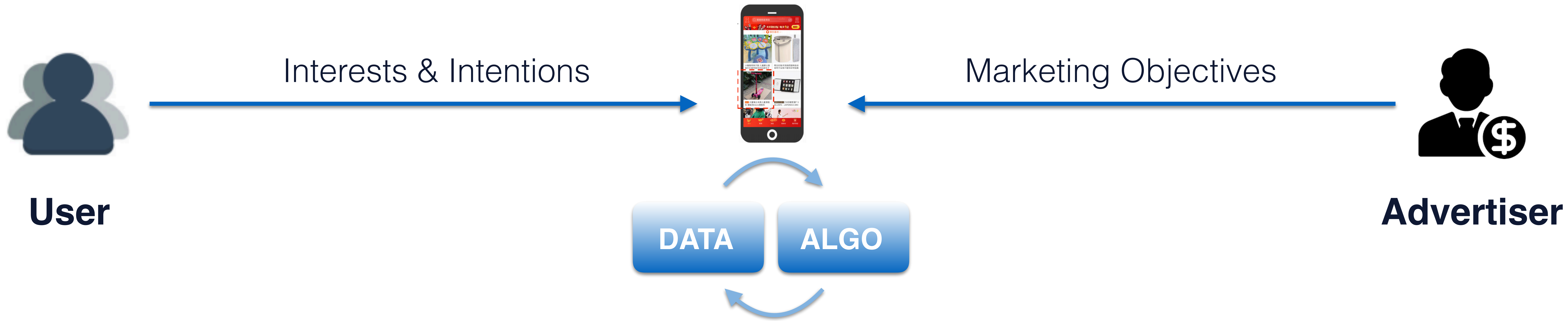


Feed Ads
Top Selected

阿里妈妈


Ad Request
User Preference
User Feature &
Context Feature
Send user request
to certain worker
in distributed cluster
Impression
Matching
Generate personalized ad candidates from ad corpus
Ad Feature
Prediction
Calculate pCTR, pCVR for user-ad pair
Strategy
Bid optimization, followed by eCPM sorting.
Return ad results
...

阿里妈妈



MATCHING

Retrieve the most promising Top K candidates from LARGE corpus

In this talk

Tree-based Deep Match (TDM)

PREDICTION

Predict user response probabilities (e.g., CTR) with accurately

In this talk

From **Deep Interest Network** to **Multi-channel Interest Memory Network**

STRATEGY

Publisher: Impression allocation
Advertiser: Ad delivery strategies

In this talk

Single-agent Bidding
Multi-agent Bidding

Matching

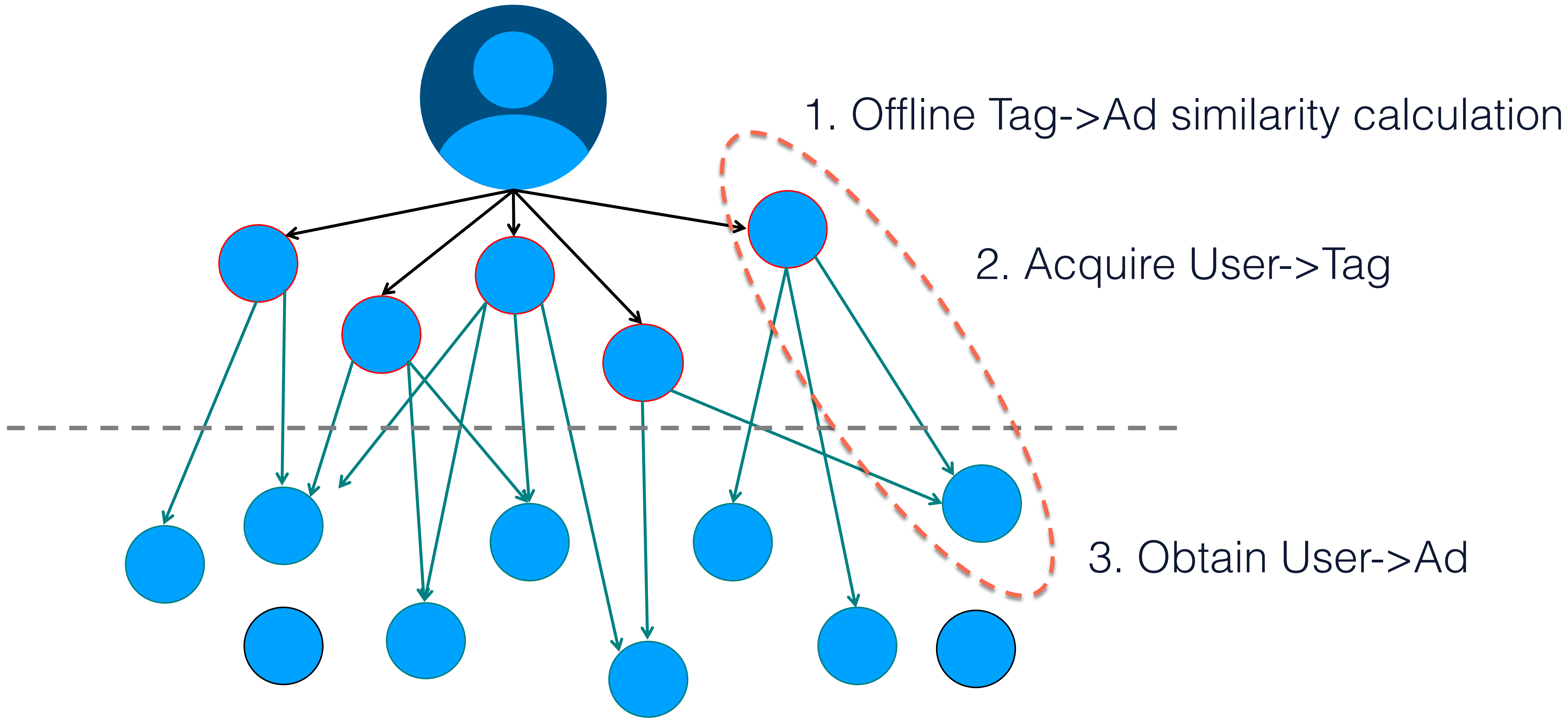
阿里妈妈

阿里妈妈

Classical Matching Methods

User->Tag, Tag->Ad matching

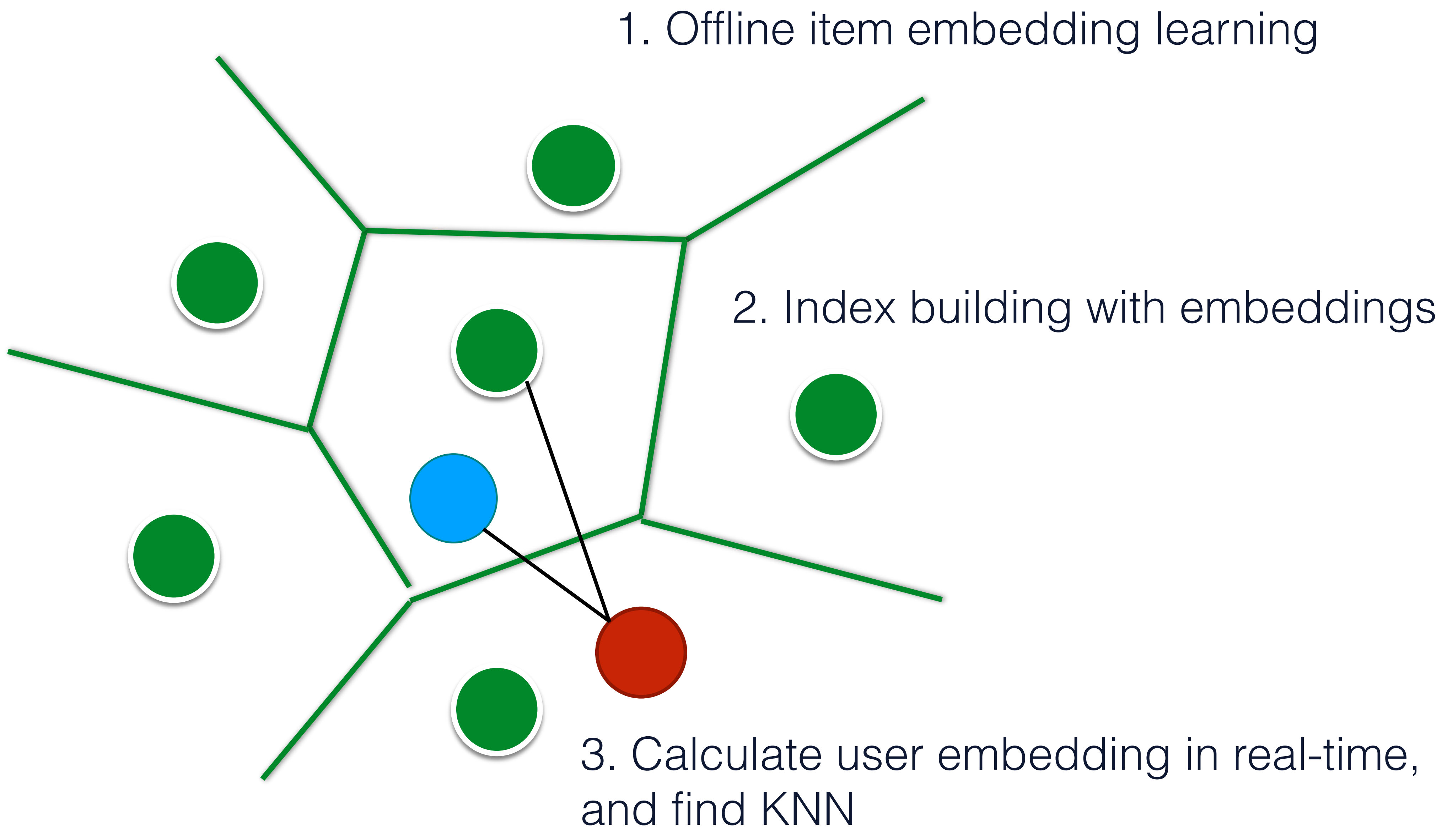
Typical model: item-based collaborative filtering [1]



Difficult to do full corpus matching

Vector-space matching

Typical model: Inner-product based candidate generation [2]



Difficult to accommodate advanced models

[1] G Linden, B Smith, J York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering, IEEE Internet Computing, 2003
[2] P Covington, J Adams, E Sargin. Deep Neural Networks for YouTube Recommendation. RecSys 2016

阿里妈妈

Tree-based Deep Match - Motivation

Challenges Revisited

$$\boxed{T} \times \boxed{N} \leq \boxed{Bound}$$

computational cost on a single item | Number of computations | System capacity

Dilemma

Advanced models —> large T

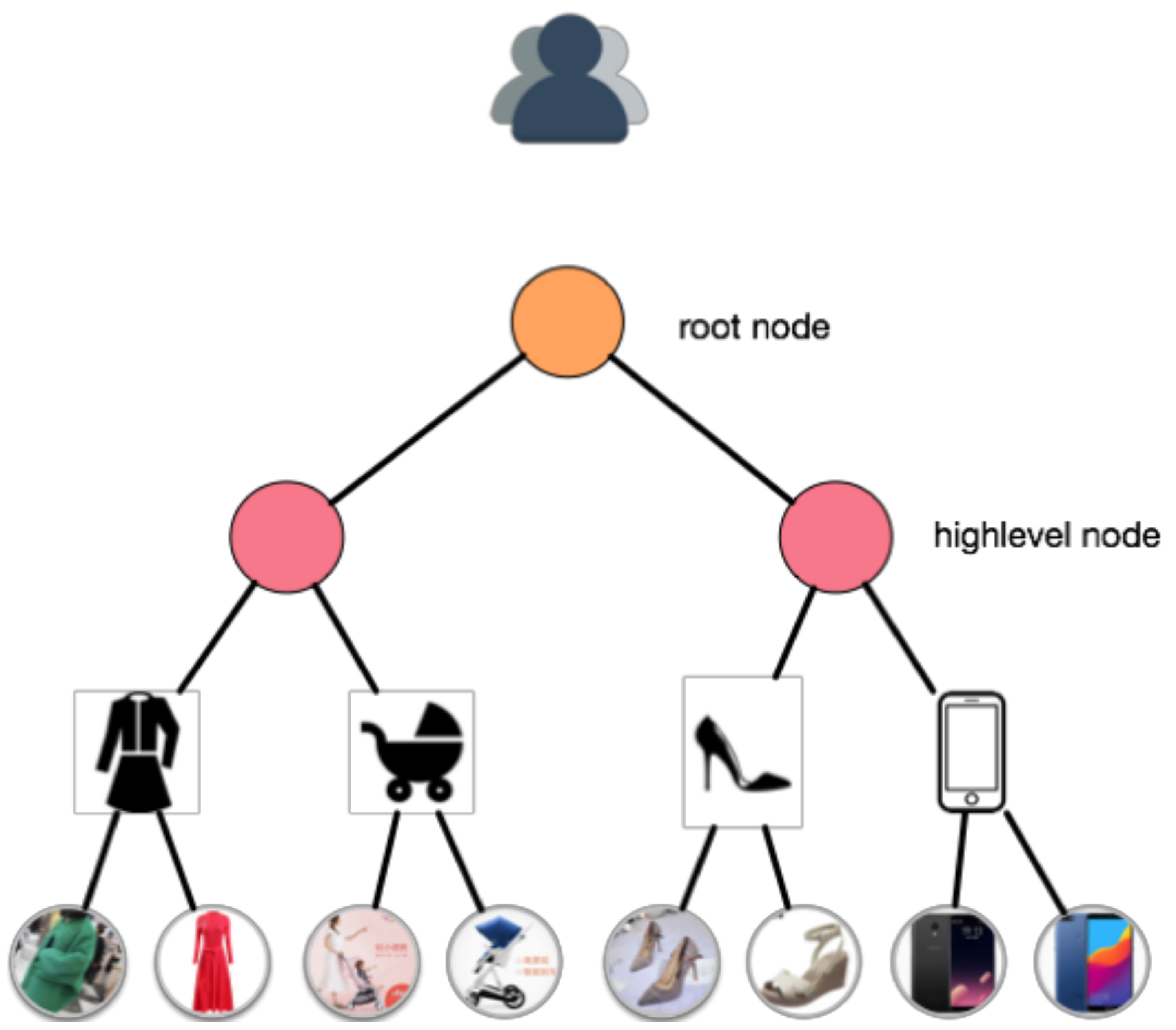
Full corpus computation —> large N

Idea

Full corpus matching != Full corpus computation

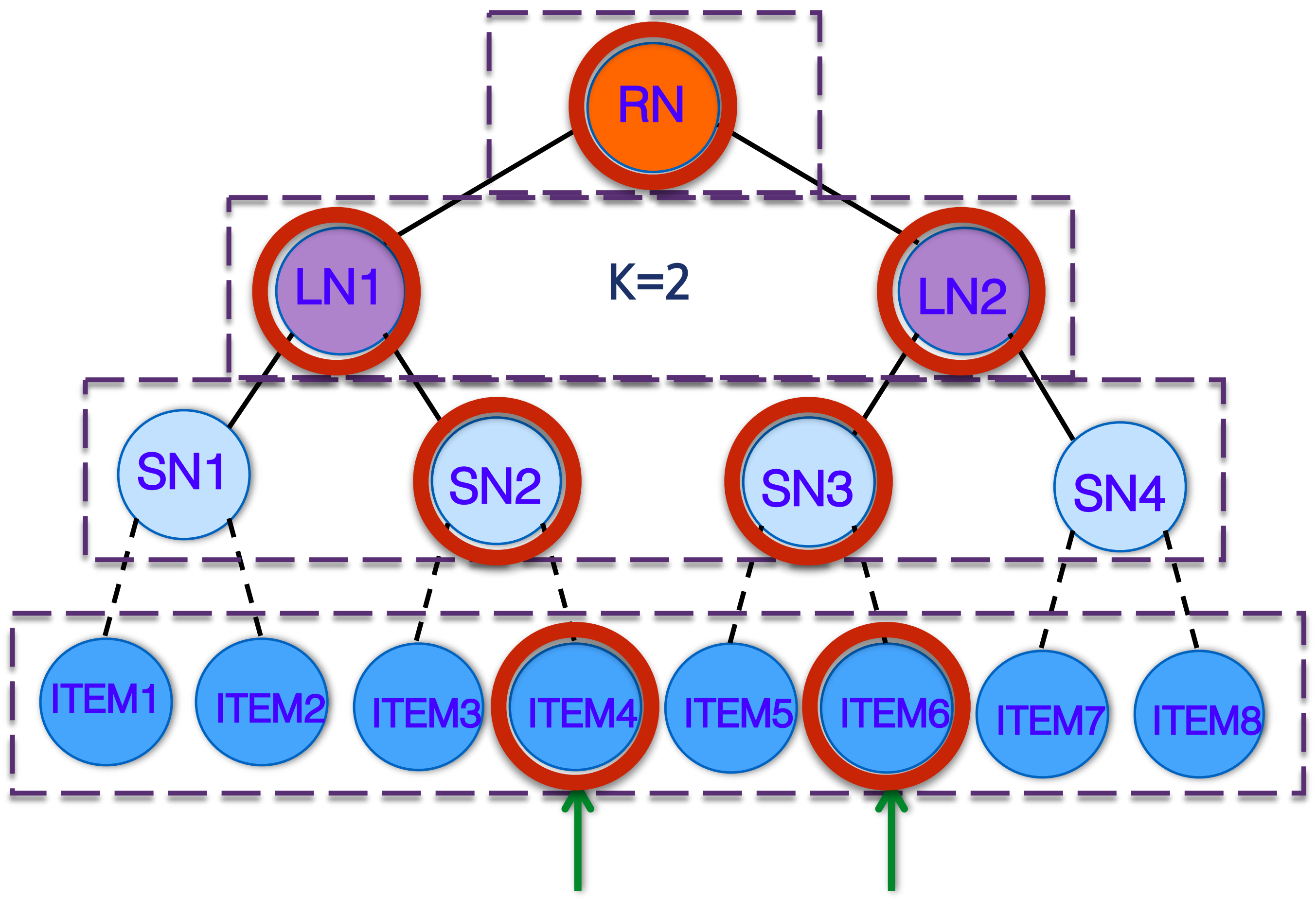
A tree index can facilitate efficient matching

Like the human intuition: interest is hierarchically organized
e.g., baby creams -> baby bath&skin care -> baby



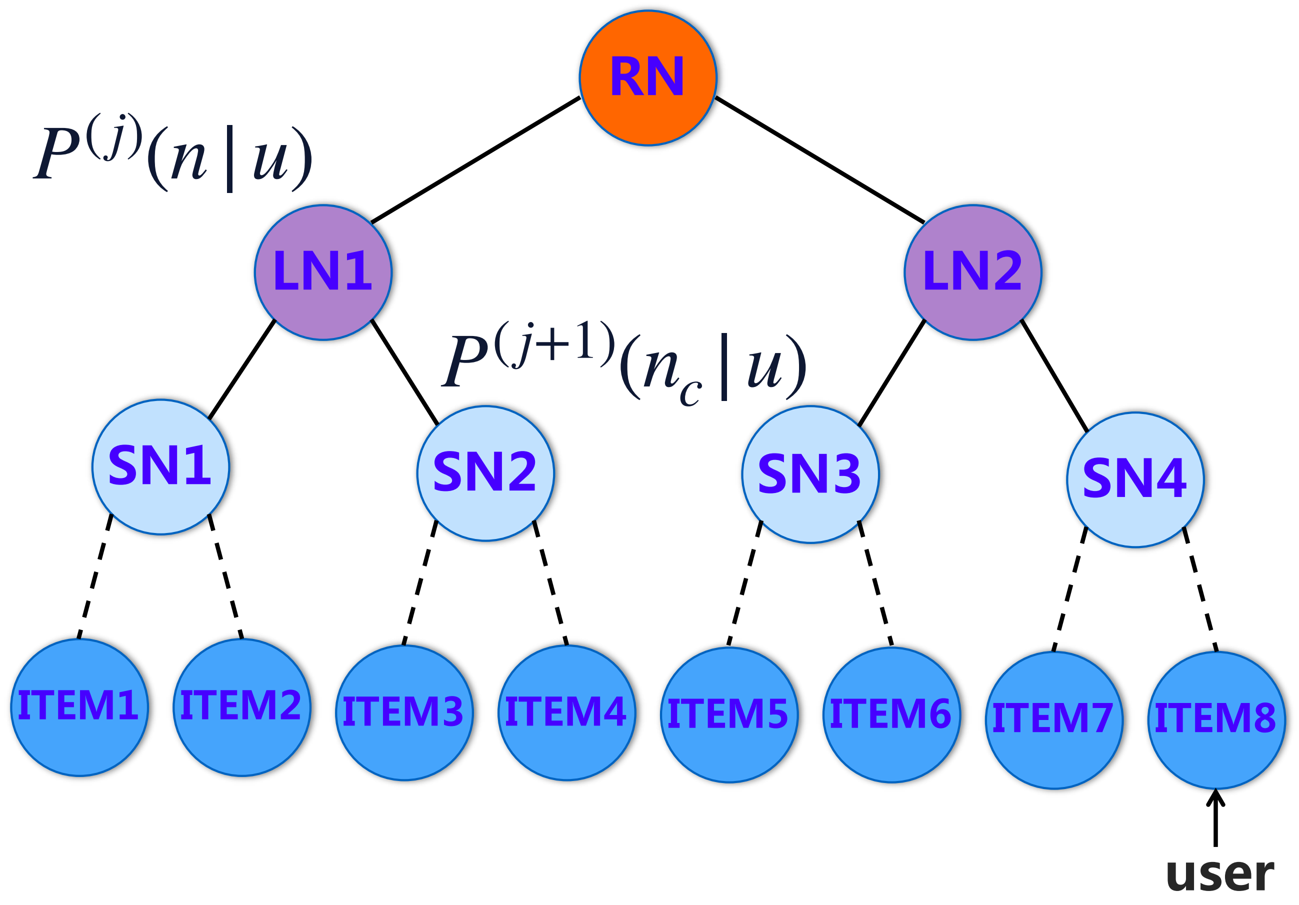
Tree-based Deep Match - Core Ingredients

Ideally: Beam search



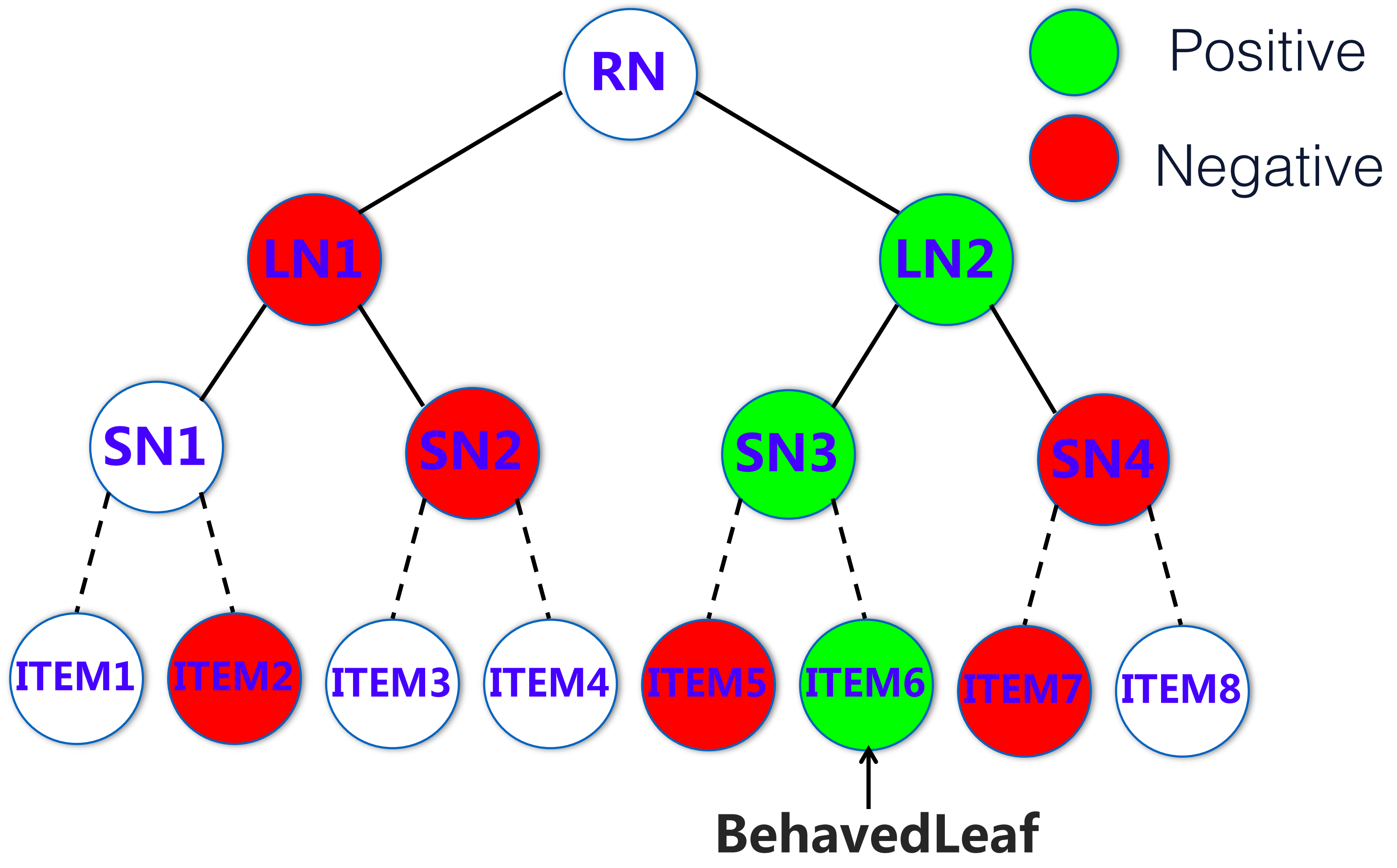
Complexity: $O(logN \cdot K)$

Max-heap like tree structure



$$P^{(j)}(n\,|\,u) = \frac{\max_{n_c \in \{n's\ children\}} P^{(j+1)}(n_c\,|\,u)}{\alpha^{(j)}}$$

User-node preference model learning



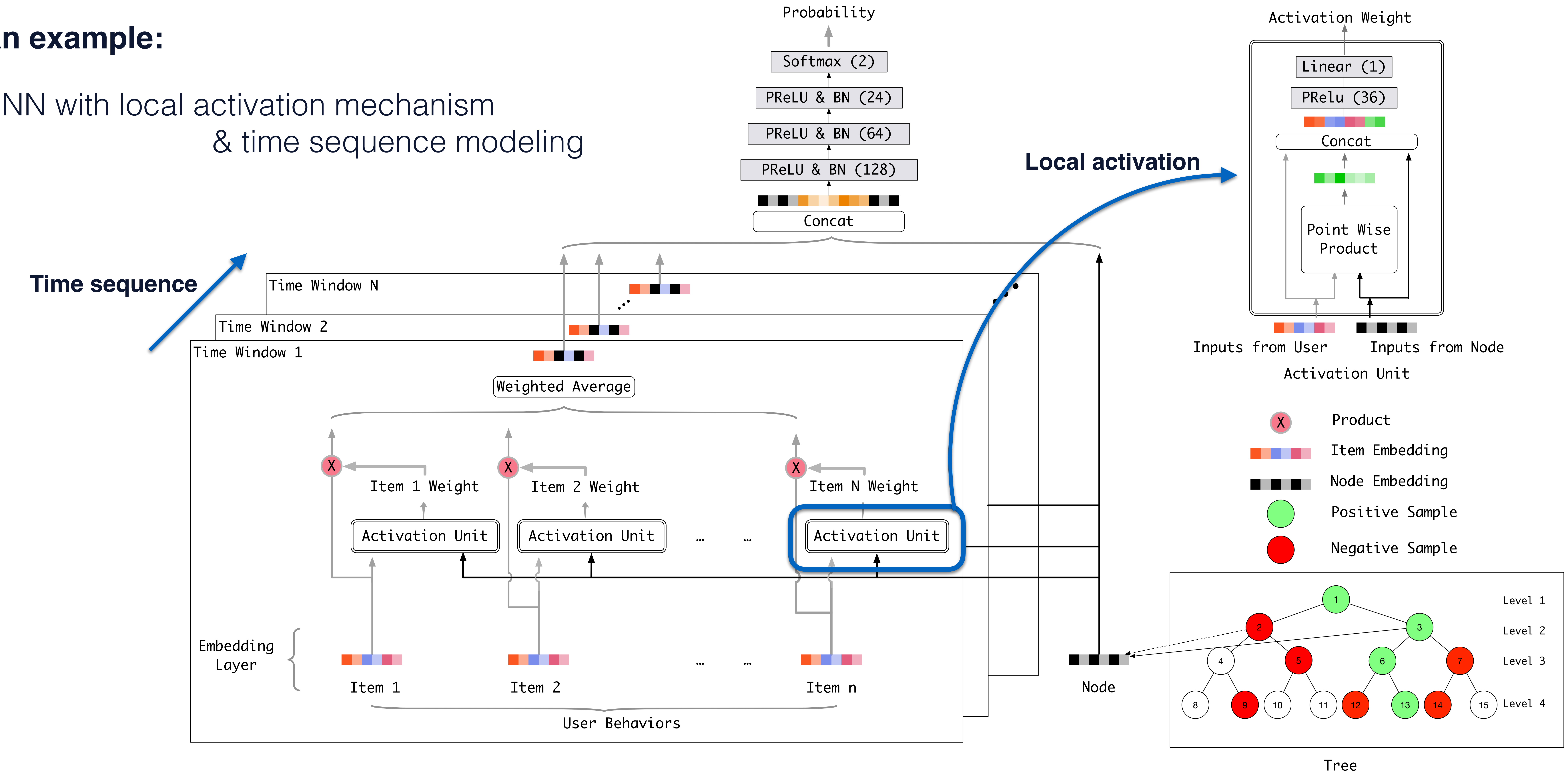
Train *user-node preference model* to fit the max-heap probability distribution

Sampling + Layer-wise modeling

Tree-based Deep Match - Accommodating Arbitrary Model

An example:

DNN with local activation mechanism
& time sequence modeling



Tree-based Deep Match - Joint Optimization

Tree structure is crucial to the performance

- It is the index for matching
- Decides the non-leaf node sample distribution
- Decides the model performance upper-bound

Learn tree structure and user-node preference model jointly

$$\mathcal{L}(\theta, \pi) = -\sum_{i=1}^{n} \sum_{j=0}^{l_{max}} \log \hat{p}\left(b_j(\pi(c^{(i)}))|u^{(i)}; \theta, \pi\right)$$

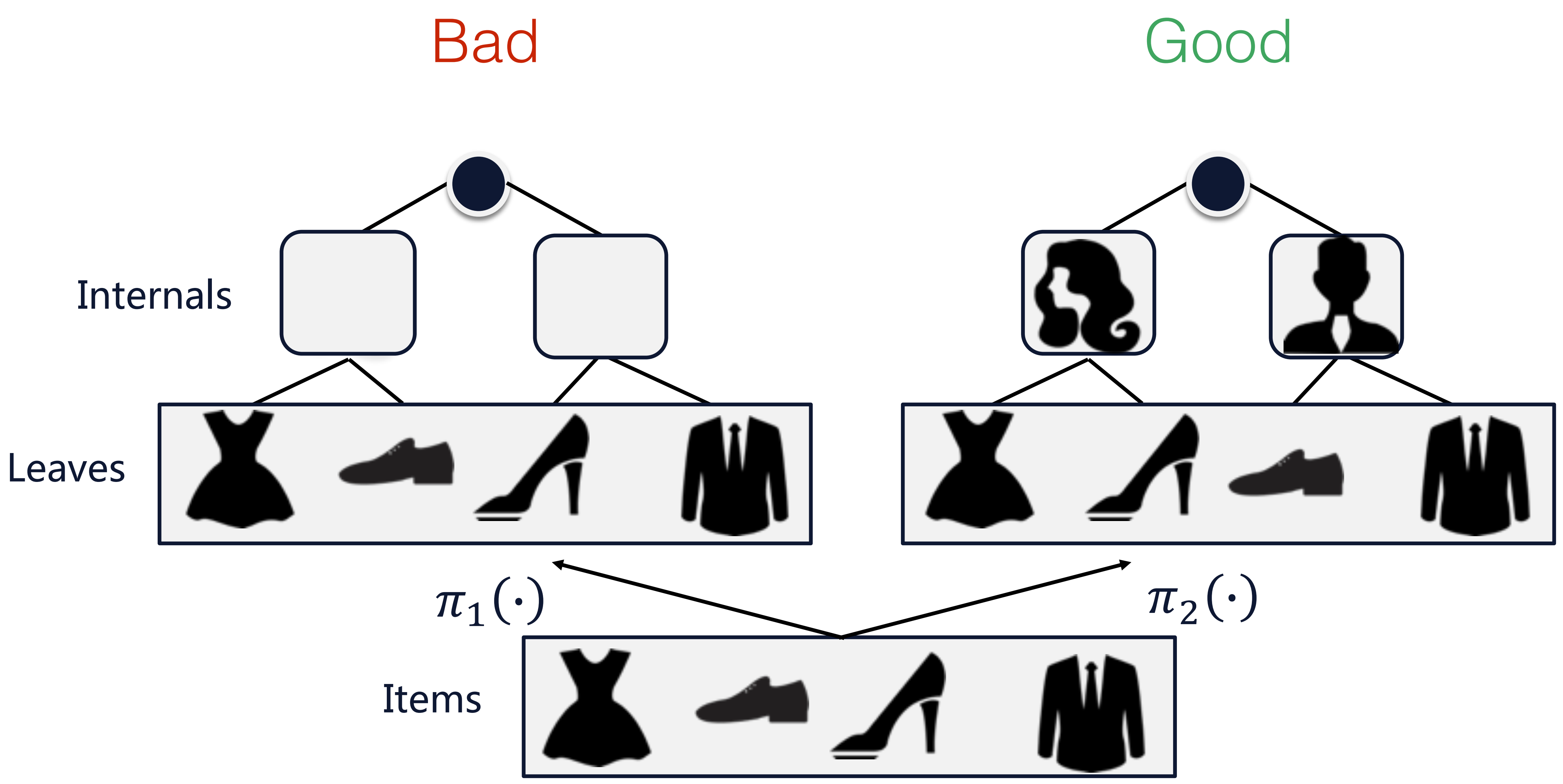
Model parameters

Item->leaf assignment

Ancestor at level j

Training set $\{(u^{(i)}, c^{(i)})\}_{i=1}^{n}$

Solve $\min_{\pi} \mathcal{L}(\theta, \pi)$ and $\min_{\theta} \mathcal{L}(\theta, \pi)$ alternately



Examples of good and bad item -> leaf assignments

阿里妈妈

Tree-based Deep Match - Experiments

Dataset*

	UserBehavior
# of users	969,529
# of items	4,162,024
# of categories	9,439
# of records	100,020,395

* UserBehavior is a subset of Taobao user behavior data

** F1 score is calculated based on Recall@200 and Precision@200

Performance evaluation against existing methods**

Method	F1 Score	F1 Score Lift
Item-CF	0.0230	0.00%
Youtube product-DNN	0.0336	46.09%
TDM-All	0.0634	175.65%

Performance evaluation with TDM variations

Method	F1 Score	F1 Score Lift
TDM-Base(DNN)	0.0340	0.00%
TDM-Base+Attention	0.0395	16.18%
TDM-Base+JointLearning	0.0373	9.71%
TDM-All	0.0634	86.47%

1 + 1 > 2

TDM-ALL: TDM-Base + Attention + JointLearning

Tree-based Deep Match - Summary

Where we were …

To effectively and efficiently find Top-K candidates from **LARGE** corpus

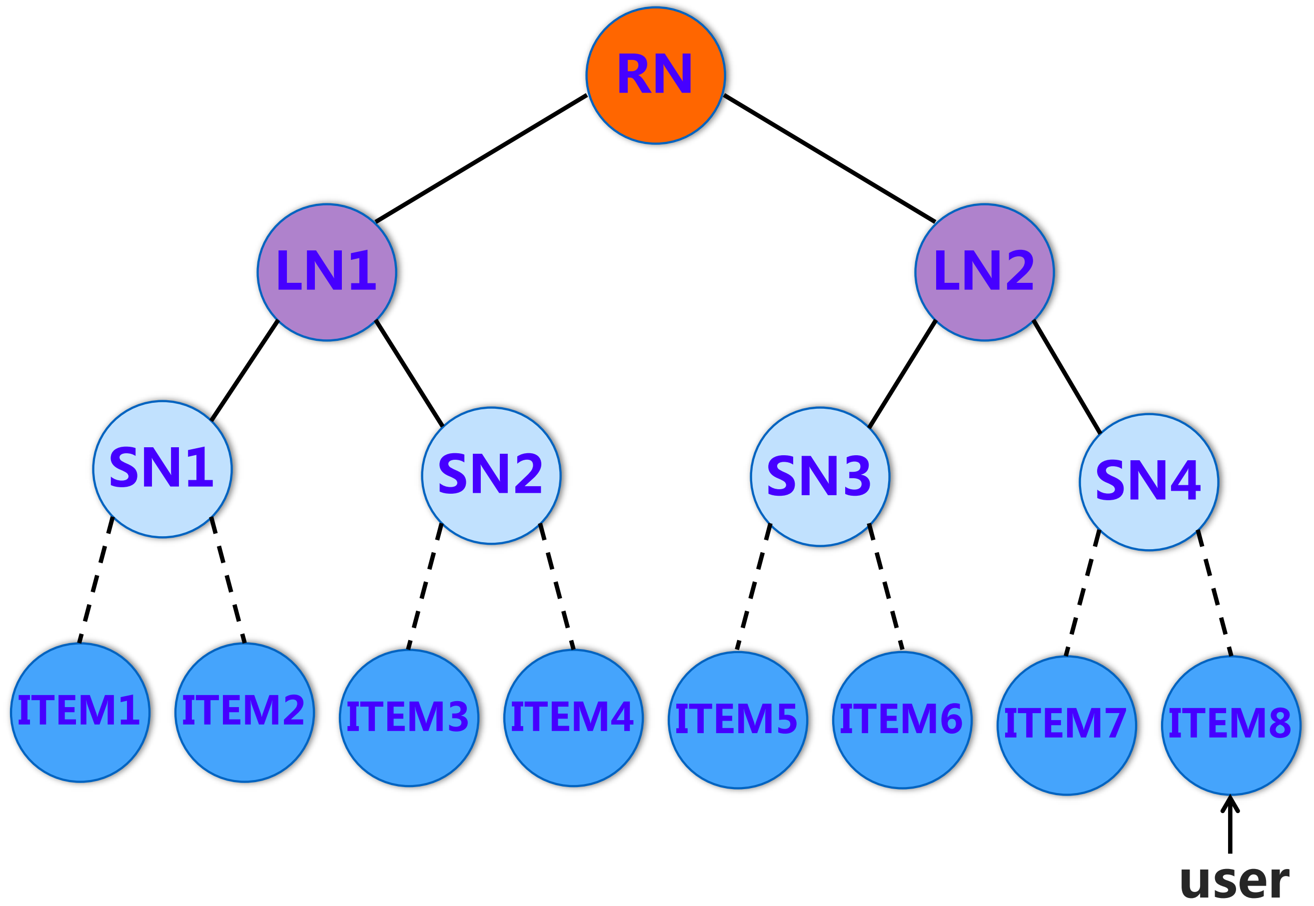
$$\boxed{T} \times \boxed{N} \le \boxed{Bound}$$

Unit computational cost

Number of computations

System capacity

Our thoughts …

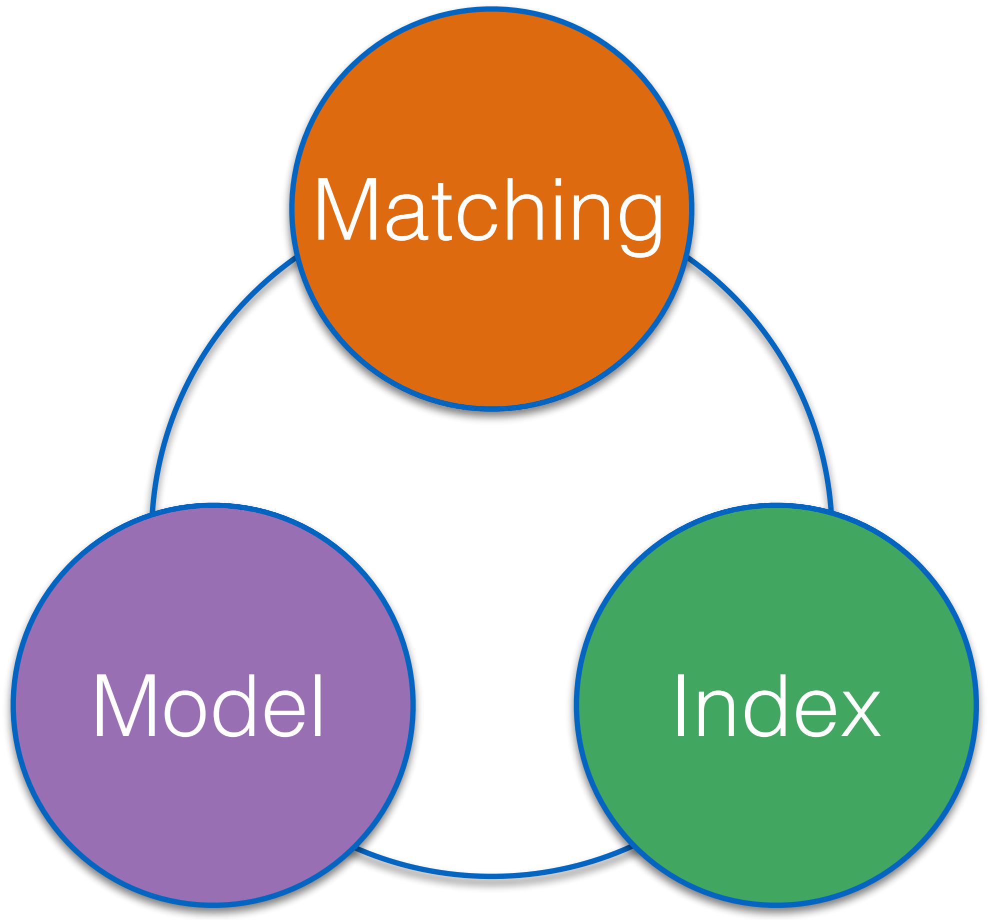


Advanced Models + Full corpus matching

depend on

Max-heap like tree index

And the adventure …



Beam search based matching
+
Arbitrary advanced models
+
Tree structure joint optimization

TDM is open-sourced with X-Deep Learning. (https://github.com/alibaba/x-deeplearning)

Prediction

阿里妈妈

Deep CTR Prediction - Overview

Click probability?

Group-wise Embedding Network

A Deep Learning Practice

Deep Interest Network (KDD18)

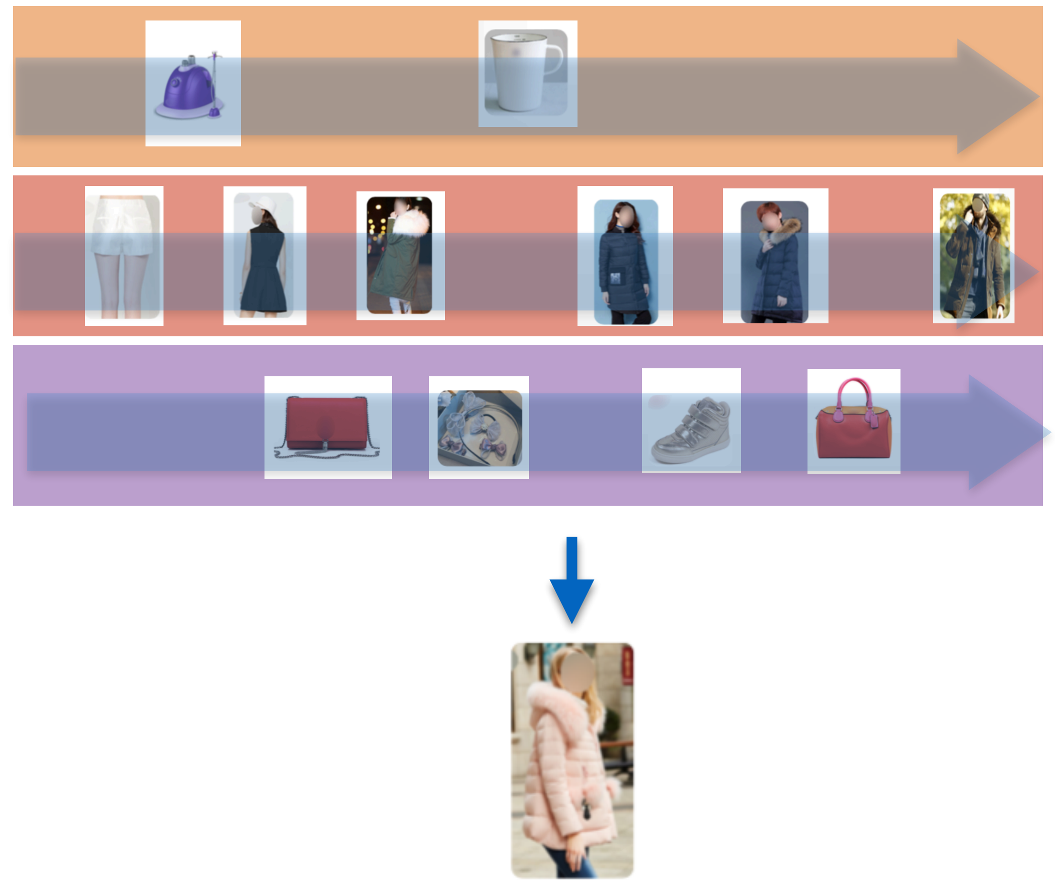
Local Activation

Deep Interest Evolution Network (AAAI19)

Interest Evolution + Local Activation

Multi-channel Interest Memory Network (KDD19)

Long-sequence Interests Modeling

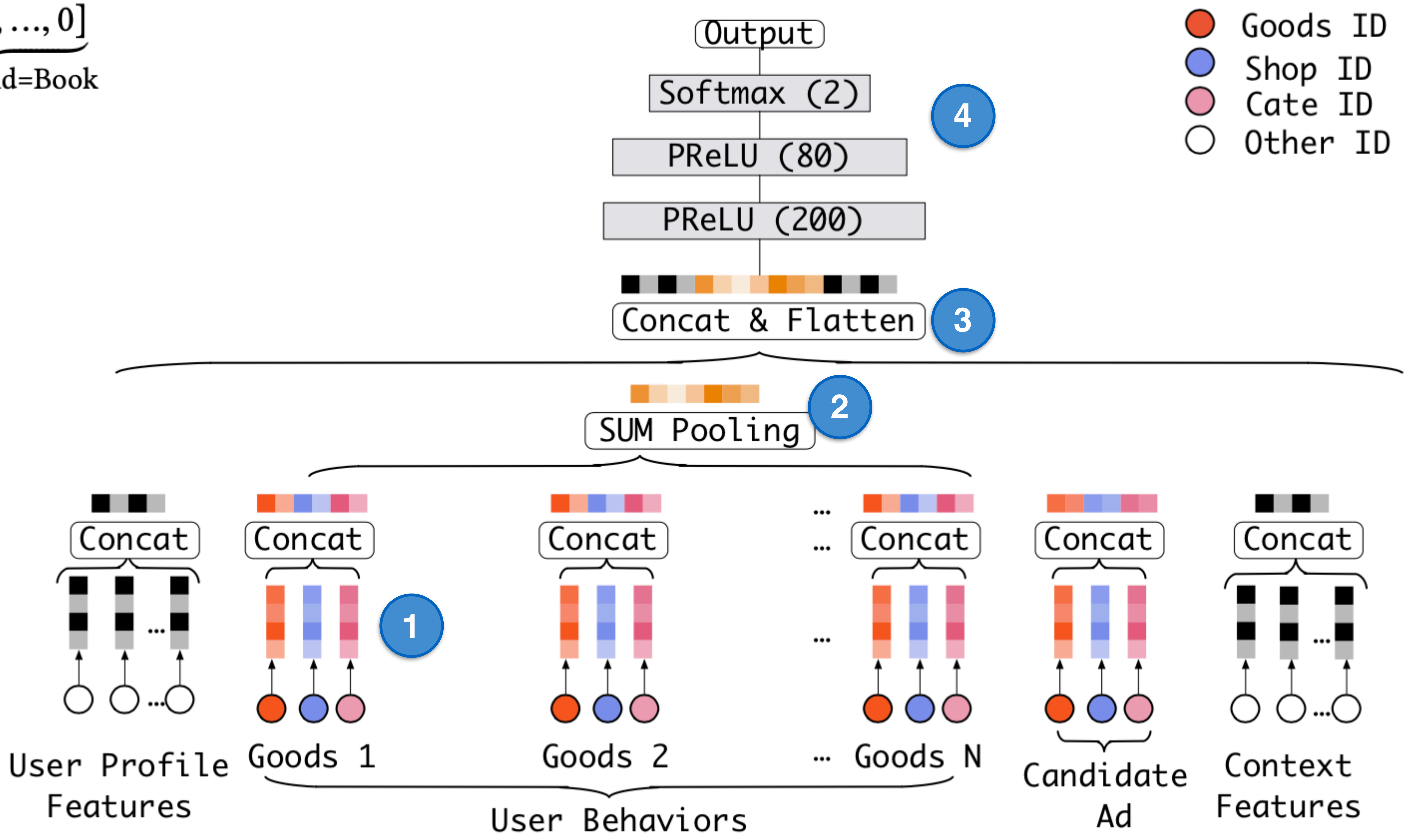


Deep CTR Prediction - First Attempt

Group-wise Embedding Network (GwEN)

$\underbrace{[0, 0, 0, 0, 1, 0, 0]}_{\text{weekday=Friday}}$ $\underbrace{[0, 1]}_{\text{gender=Female}}$ $\boxed{\underbrace{[0, .., 1, ..., 1, ...0]}_{\text{visited\_cate\_ids=\{Bag,Book\}}}}$ $\underbrace{[0, .., 1, ..., 0]}_{\text{ad\_cate\_id=Book}}$

Feature group (field)



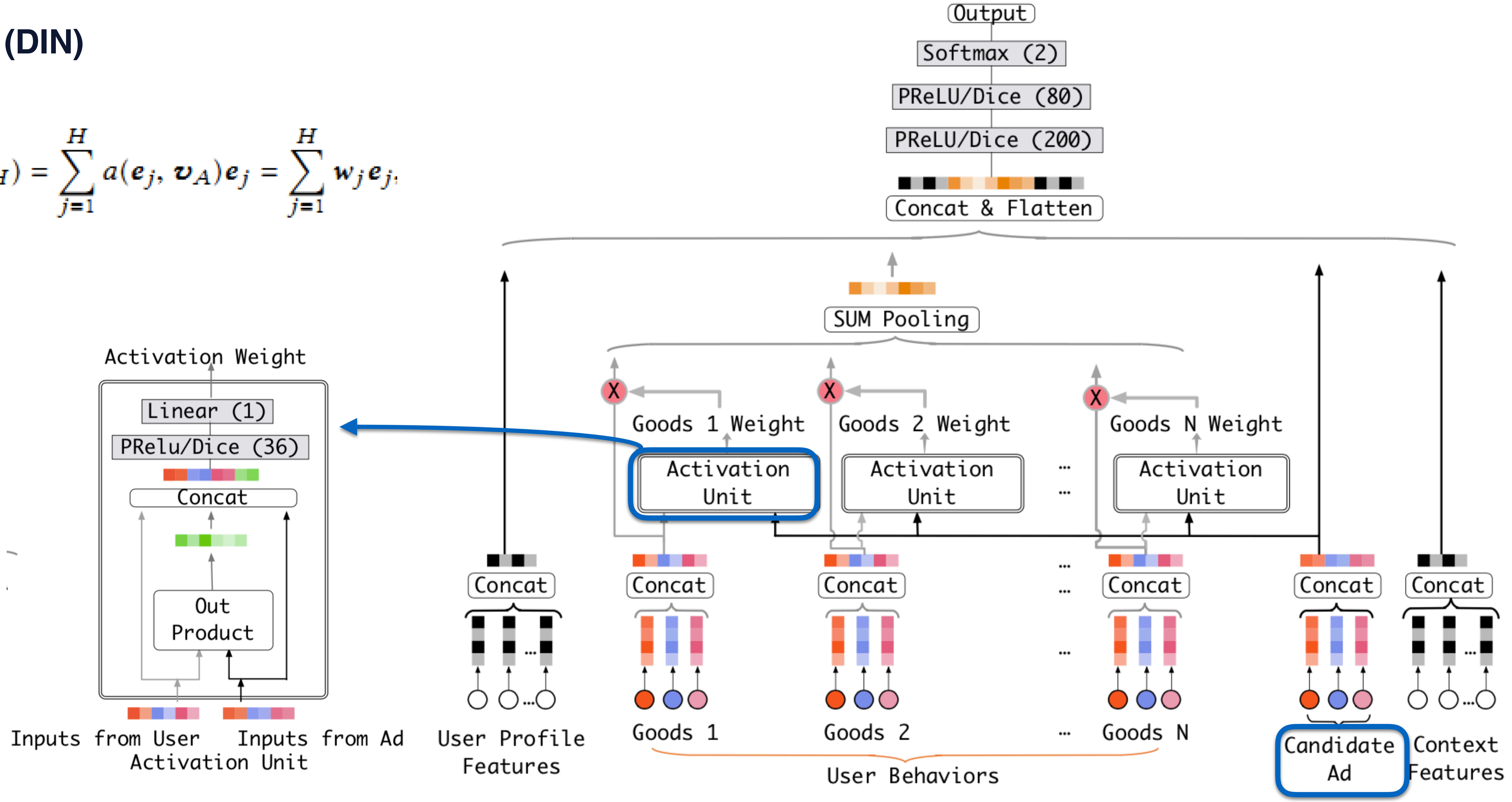
Deep CTR Prediction - Deep Interest Network

Deep Interest Network (DIN)

$$\boldsymbol{v}_U(A) = f(\boldsymbol{v}_A, \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_H) = \sum_{j=1}^{H} a(\boldsymbol{e}_j, \boldsymbol{v}_A)\boldsymbol{e}_j = \sum_{j=1}^{H} \boldsymbol{w}_j \boldsymbol{e}_j,$$

Local Activation Mechanism

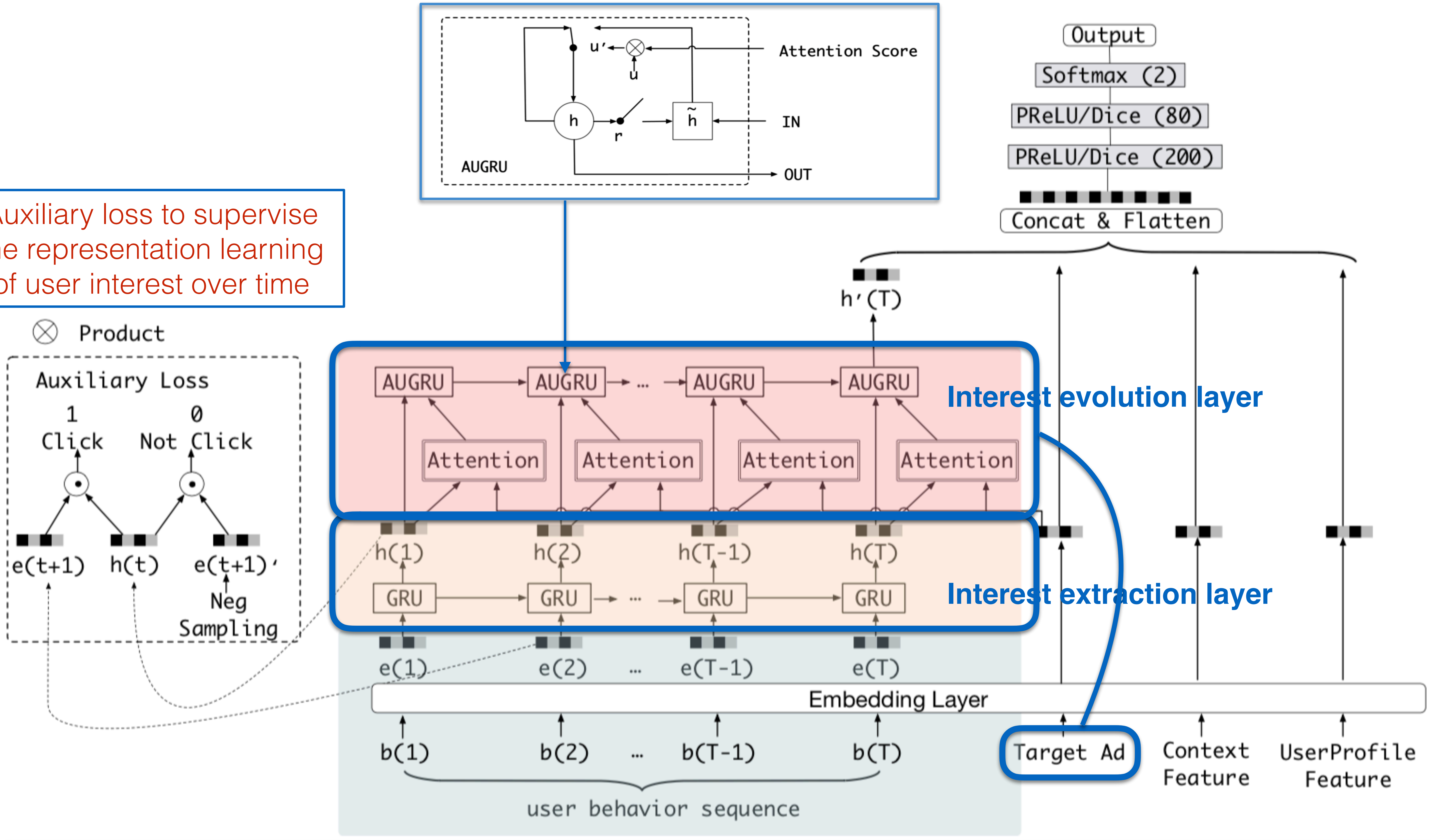
Share similar thoughts
with attention mechanism



Deep CTR Prediction - Deep Interest Evolution Network

Deep Interest Evolution Network (DIEN)

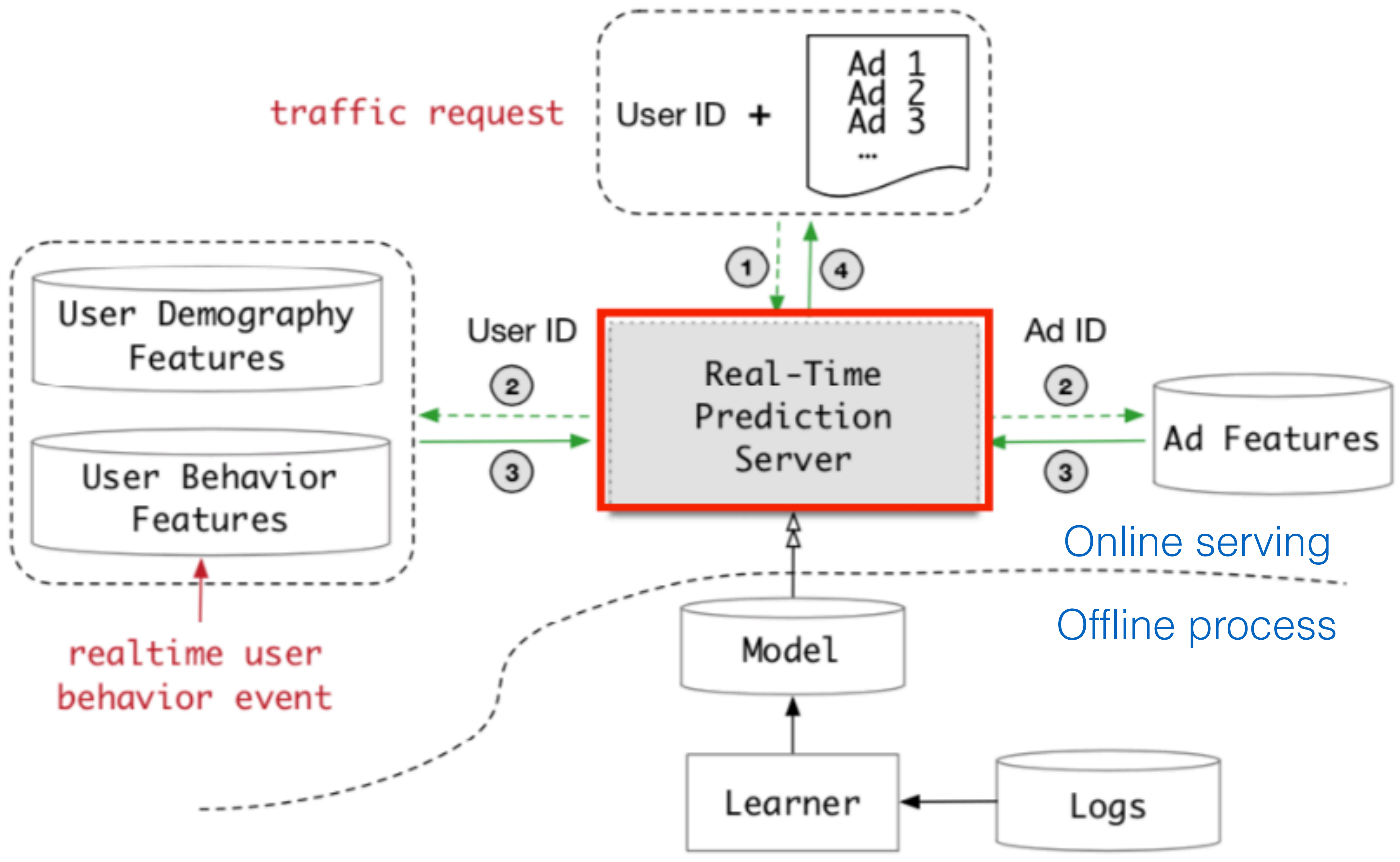
Auxiliary loss to supervise the representation learning of user interest over time



Deep CTR Prediction - Architecture Revisited

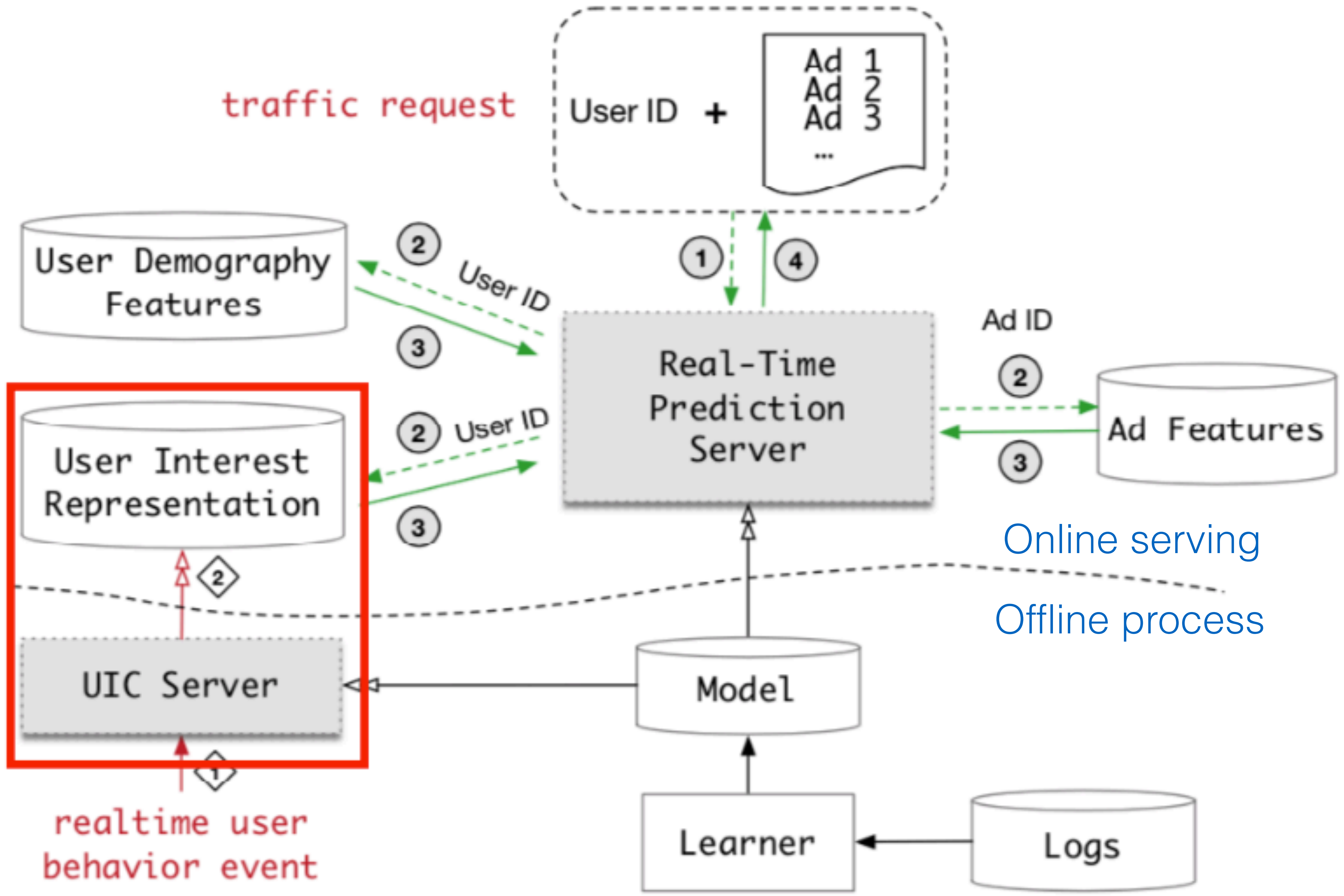
Local activation —Challenges→ online serving efficiency

Long sequence behaviors —Improve→ prediction performance



User Interest Computation at **EVERY** Request

Redundant Synchronous



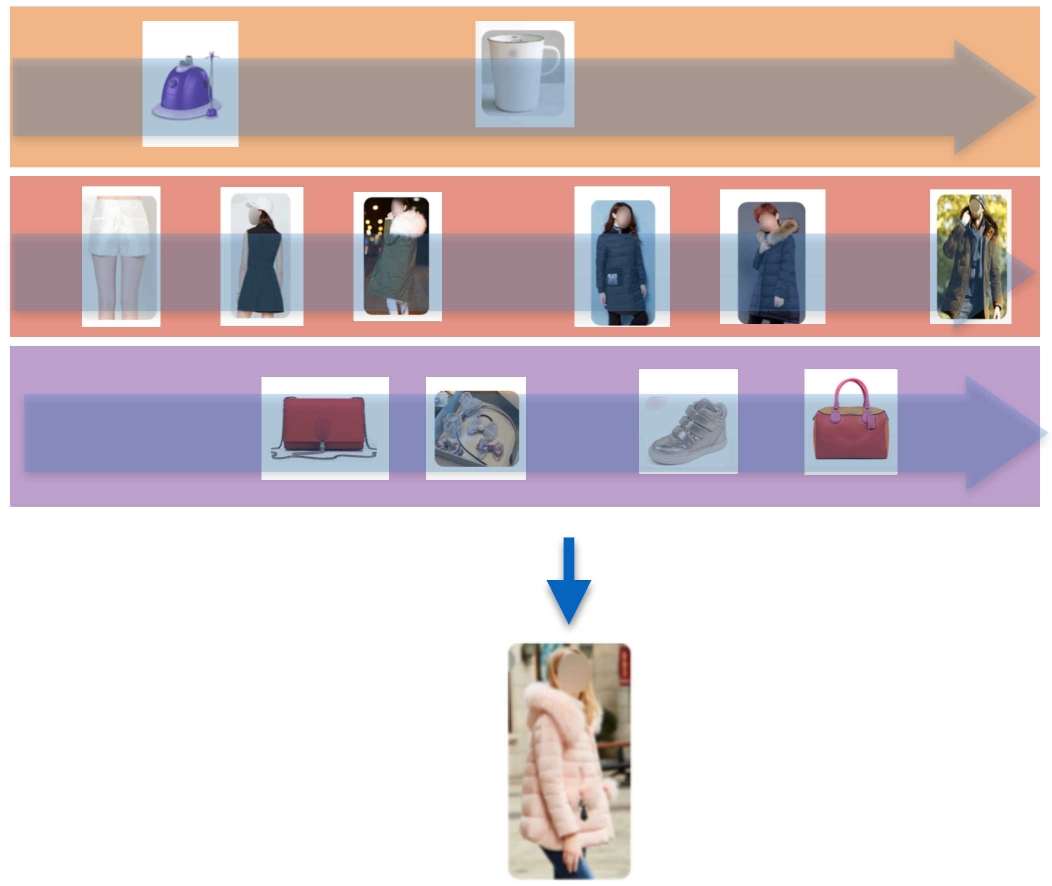
Independent User Interest Computation

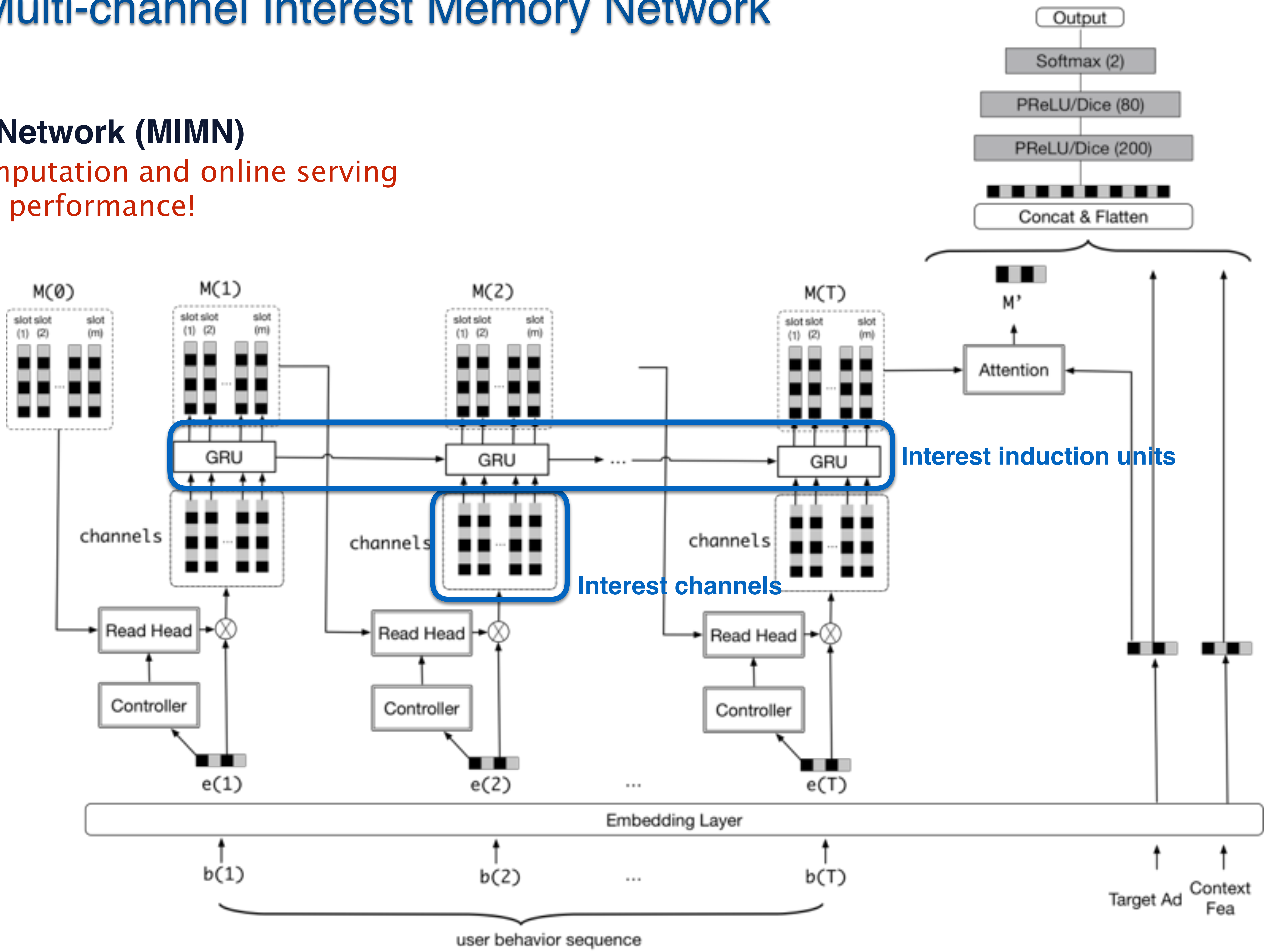
Incremental **Asynchronous**

Deep CTR Prediction - Multi-channel Interest Memory Network

Multi-channel Interest Memory Network (MIMN)

The separation of user interest computation and online serving should not degrade the prediction performance!





Right Here!

Section 3 of IDLUGHET (EKLUTNA) EXHIBIT HALL
7:00pm to 9:30pm on August 6

阿里妈妈

Deep CTR Prediction - Experiments

Compared with some latest work

GRU4REC [3] - bases on RNN and is the first work using the recurrent cell to model sequential user behaviors.

ARNN - a variation of GRU4Rec which uses attention mechanism to weighted sum over all the hidden states along time for better user sequence representation

RUM [4] - uses an external memory to store user's behavior features. It also utilizes soft-writing and attention reading mechanism to interact with the memory. We use the feature-level RUM to store sequence information

Datasets

Dataset	Users	Items[a]	Categories	Instances
Amazon(Book).	75053	358367	1583	150016
Taobao.	987994	4162024	9439	987994

AUC Performance

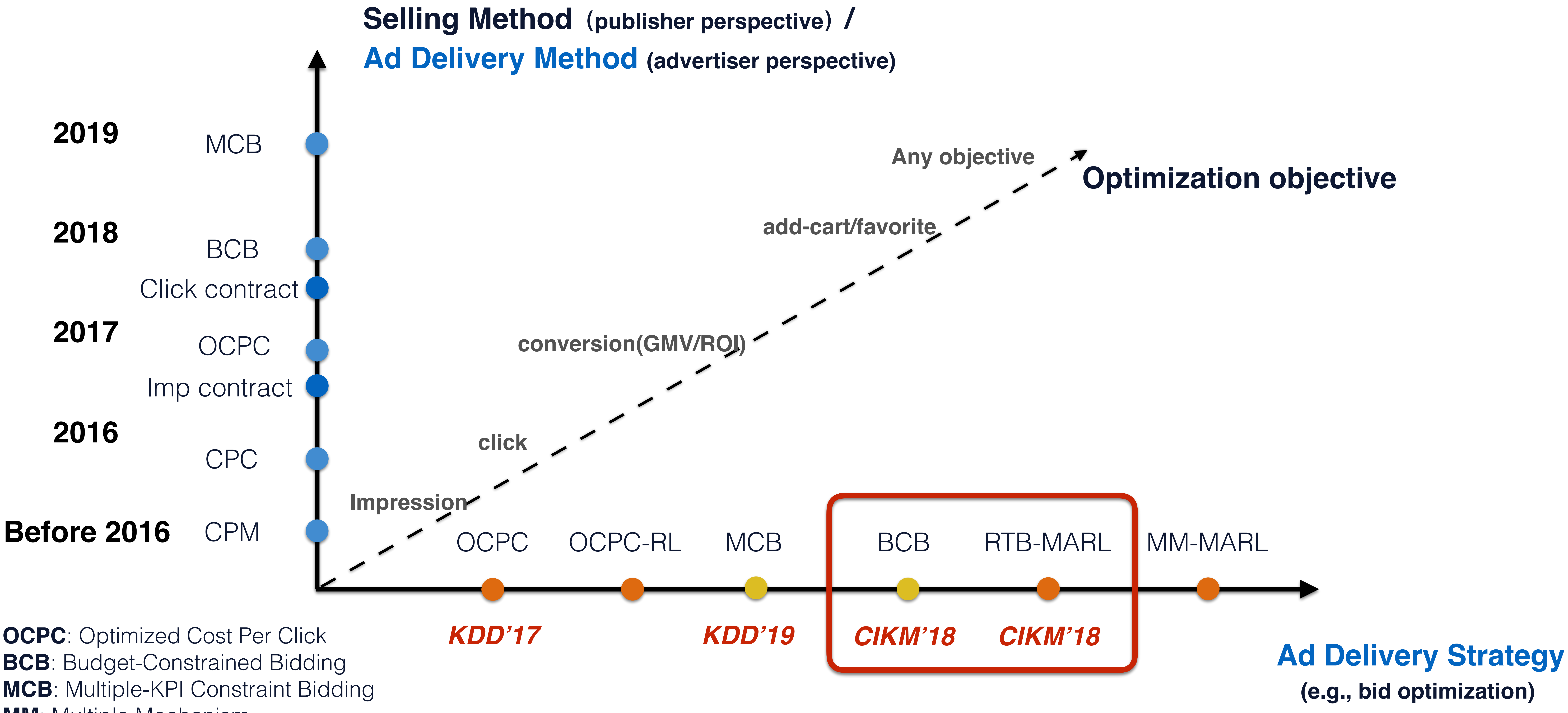
Model	Taobao (mean ± std)	Amazon (mean ± std)
Embedding&MLP	0.8709 ± 0.00184	0.7367 ± 0.00043
DIN	0.8833 ± 0.00220	0.7419 ± 0.00049
GRU4REC	0.9006 ± 0.00094	0.7411 ± 0.00185
ARNN	0.9066 ± 0.00420	0.7420 ± 0.00029
RUM	0.9018 ± 0.00253	0.7428 ± 0.00041
DIEN	0.9081 ± 0.00221	0.7481 ± 0.00102
MIMN	**0.9179 ± 0.00325**	**0.7593 ± 0.00150**

[3] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. ICLR 2016.
[4] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. WSDM. 2018.

Strategy

阿里妈妈

Strategies - Overview



OCPC: Optimized Cost Per Click
BCB: Budget-Constrained Bidding
MCB: Multiple-KPI Constraint Bidding
MM: Multiple Mechanism

阿里妈妈

Budget Constrained Bidding

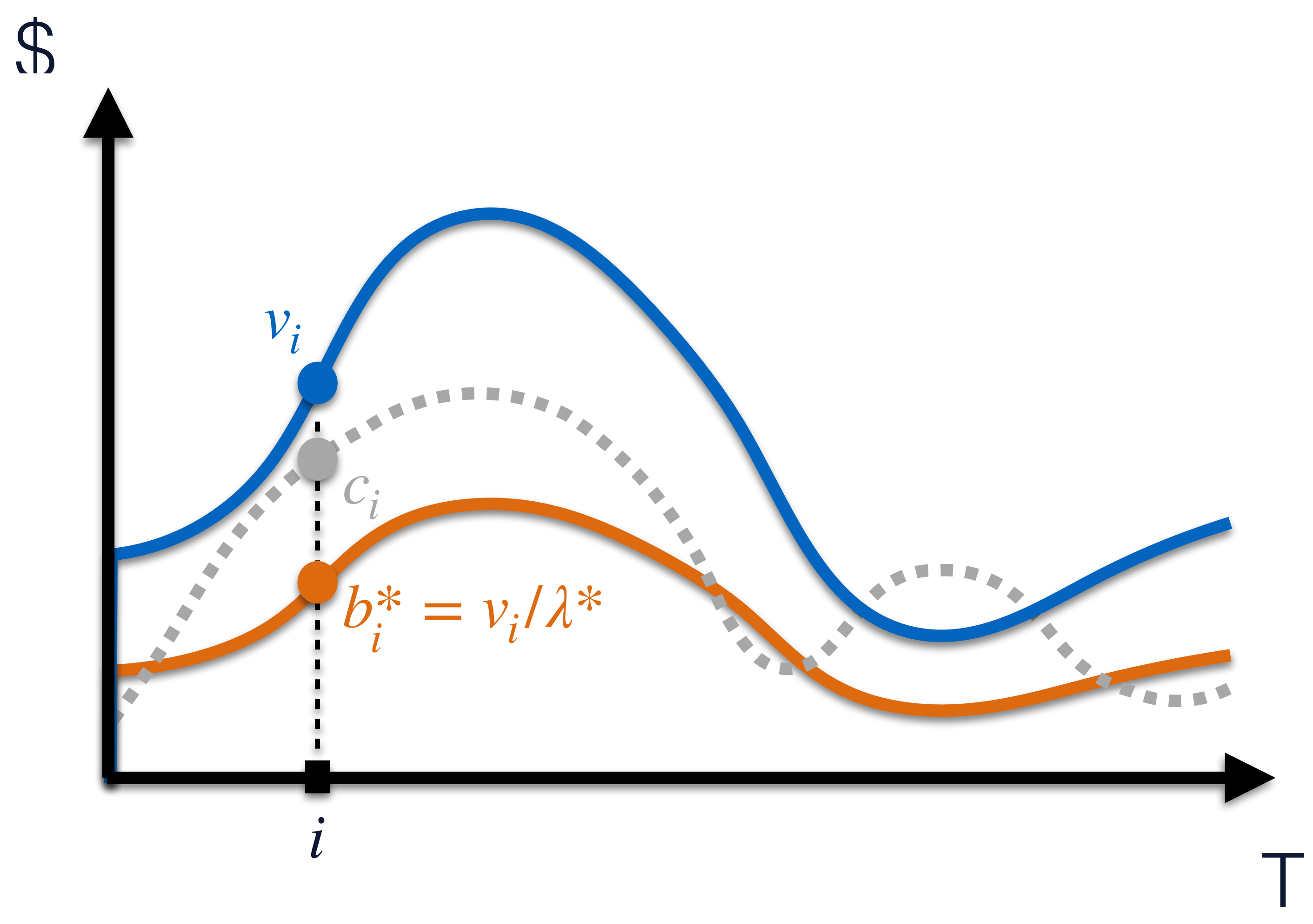
Problem formulation

Given budget B, acquire as much value as possible. In RTB, it is to decide the bid b_i for each impression i to maximize the winning value.

$$\begin{aligned} \underset{b_i}{\text{maximize}} \quad & \sum_i \mathbf{1}[b_i > c_i] \cdot \boxed{v_i} \\ \text{subject to} \quad & \sum_i \mathbf{1}[b_i > c_i] \cdot \boxed{c_i} \leq B \end{aligned}$$

impression value

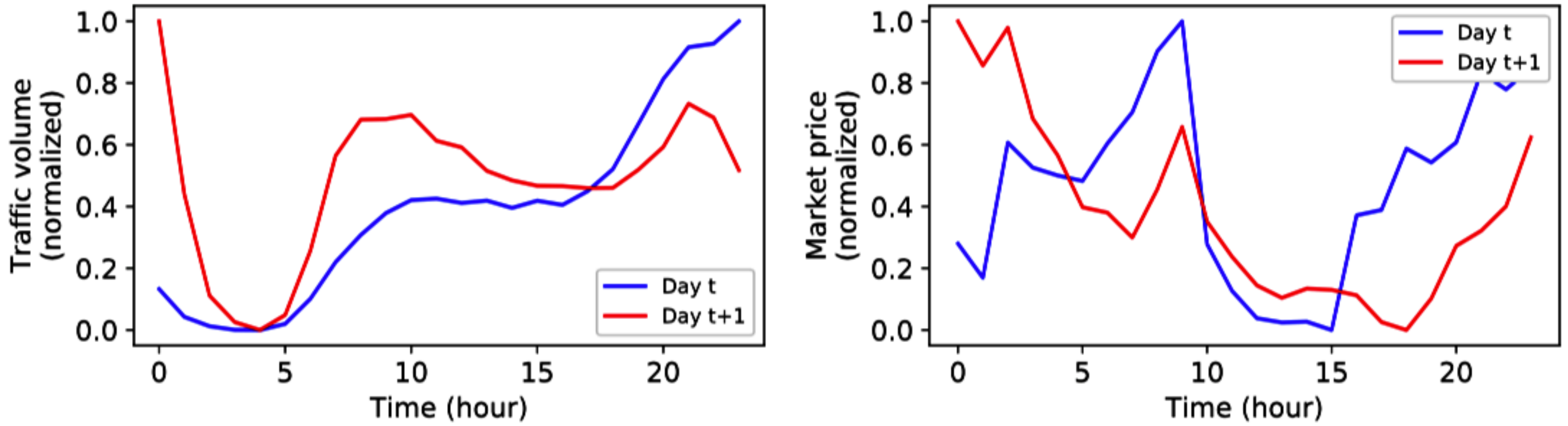
market price



Optimal Solution [5]

$$b_i^* = v_i/\lambda^*$$

Challenge is the dynamic and unstable environment!



Example of unstable traffic and market price patterns

[5] Zhang W, Ren K, Wang J. Optimal real-time bidding frameworks discussion[J]. arXiv preprint arXiv:1602.01007, 2016.

Budget Constrained Bidding

Model-free Reinforcement Learning

$\mathcal{S}$: s_t campaign stats at timestep t

Remaining budget, win-rate, spending rate, rpm, etc

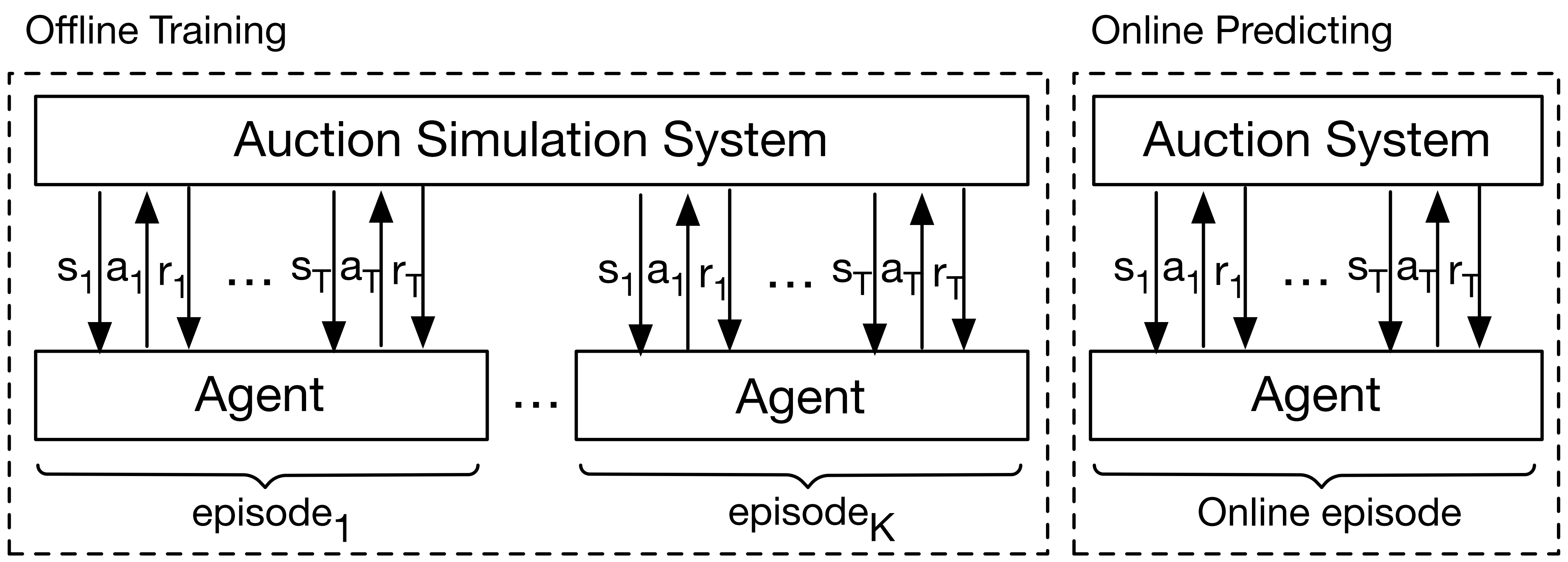
$\mathcal{A}$: a_t adjusts λ by $\lambda_t = \lambda_{t-1}(1 + a_t)$

$\mathcal{T}$: ignored with model-free RL

r_t : cummulative value between step $t-1$ and t

c_t : cummulative cost between step $t-1$ and t

γ : $\gamma = 1$ in this scenario

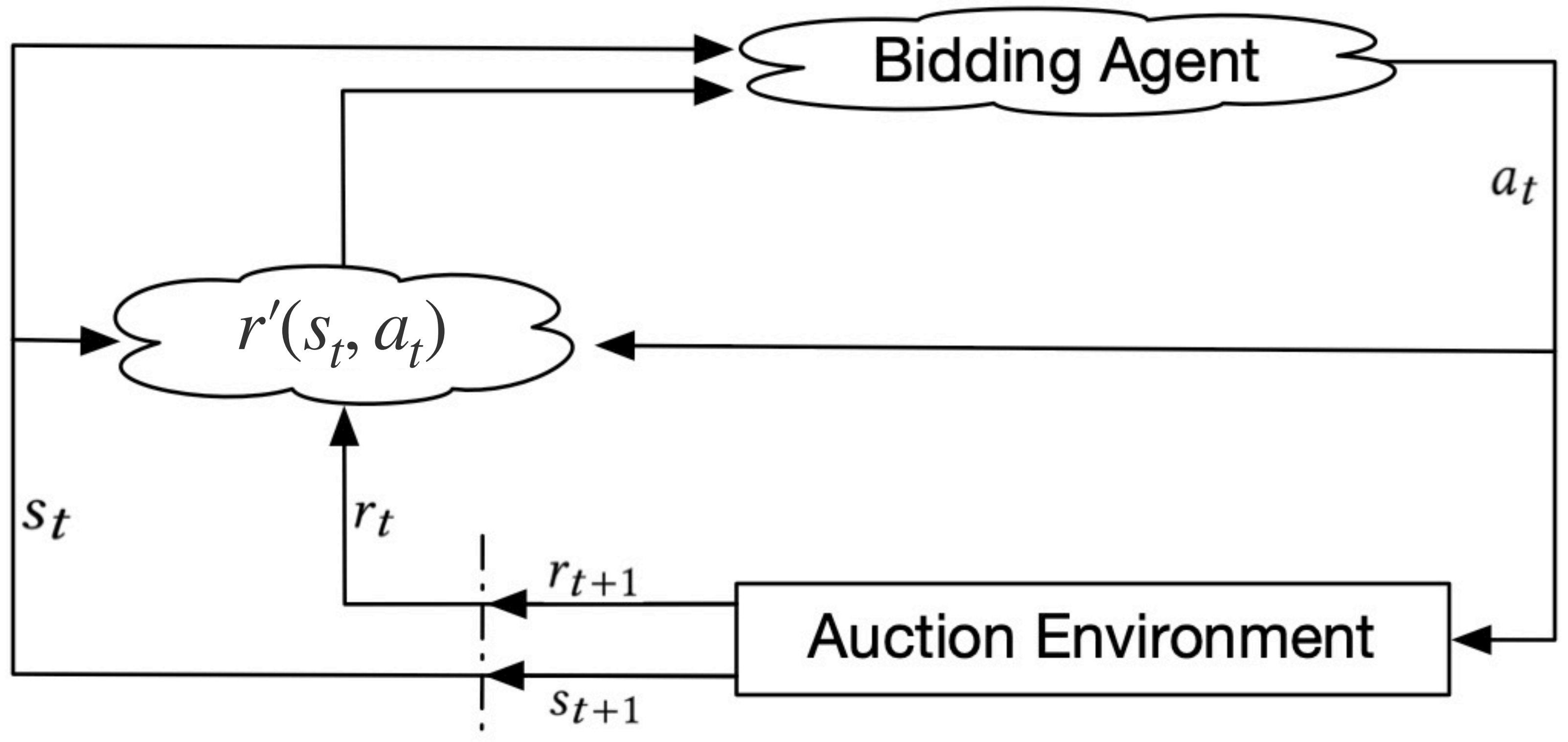


Avoiding the Immediate Reward Pitfall

$$r'(s,a) = \max_{e \in E(s,a)} \boxed{\sum_{t=1}^{T} r_t^{(e)}}$$

Episode-level reward observed with (s,a) pair

THEOREM Let $\pi^*_{r'}$ be an optimal policy with the shaped reward r'. If the MDP has a unique initial state, $\pi^*_{r'}$ is also an optimal policy in the original MDP formulation with immediate reward r.



Budget Constrained Bidding

Experiments

Evaluation metric R/R^* The total value of the winning impressions VS theoretically optimal value (with λ^*)

Performance comparison against SOTAs

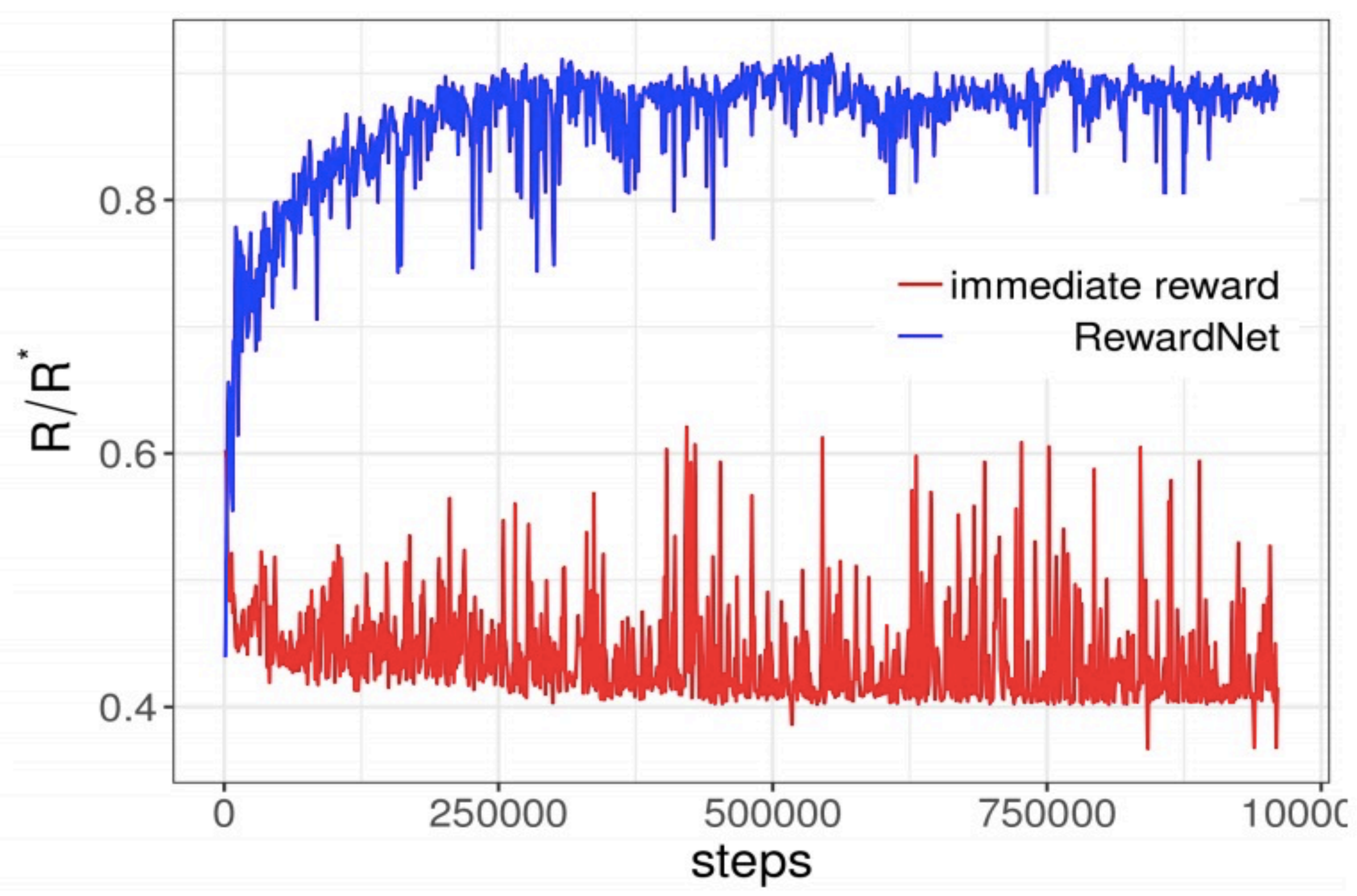
λ Deviation	Campaigns	R/R^*				Improvements of R/R^*		
		FLB	BSLB	RLB	DRLB	FLB	BSLB	RLB
$[-100\%, -80\%)$	43	0.436	0.525	0.430	**0.878**	101.38%	67.24%	104.19%
$[-80\%, -40\%)$	89	0.434	0.647	0.800	**0.884**	103.69%	36.63%	10.50%
$[-40\%, -20\%)$	66	0.697	0.901	0.927	**0.945**	35.58%	4.88%	1.94%
$[-20\%, 0\%)$	41	0.863	0.936	**0.965**	0.953	10.43%	1.82%	-1.24%
$[0\%, 20\%)$	39	0.825	0.925	0.944	**0.950**	15.15%	2.70%	0.64%
$[20\%, 40\%)$	48	0.491	0.947	0.895	**0.948**	93.08%	0.11%	5.92%
$[40\%, 80\%)$	85	0.391	0.904	0.832	**0.928**	137.34%	2.65%	11.54%
$[80\%, 160\%)$	57	0.307	0.813	0.709	**0.924**	200.98%	13.65%	30.32%
$[160\%, \infty)$	32	0.291	0.668	0.618	**0.904**	210.65%	35.33%	46.28%
Average		0.526	0.807	0.791	**0.924**	100.92%	18.33%	16.80%

FLB : fixed linear bidding $b_i = v_i/\lambda_0$

BSLB : budget smoothed linear bidding $b_i = v_i/(\lambda_0 \cdot \Delta)$

RLB : model-based RL approach to decide bid directly [6]

The effectiveness of the shaped reward



Right Here!

Multiple-KPI Constrained Bidding
Section 3 of IDLUGHET (EKLUTNA)
EXHIBIT HALL
7:00pm to 9:30pm on August 6

[6] Cai, Han, et al. Real-time bidding by reinforcement learning in display advertising. WSDM 2017.

阿里妈妈

Multi-Agent Optimal Bidding

Motivation

1. Bid optimization with budget constraint is a typical sequential decision problem
2. Lacking awareness to the bid environment can lead to suboptimal bidding strategies

⇨ Is MARL formulation helpful?

Problem formulation

N advertisers, let B_i be the budget of advertiser i, whose state space is S_i and action space is A_i (e.g., bid or bid adjustment)

$$\pi_i : S_1 \times S_2 \times \ldots \times S_N \rightarrow A_i$$

After each round of auctions, reward of advertiser i

$$r_i : S_1 \times S_2 \times \ldots \times S_N \times A_1 \times A_2 \times \ldots \times A_N \rightarrow R_i$$

and state transition happens (e.g., $B_i^{t+1} = B_i^t - cost_i^t$)

$$\tau : S_1 \times S_2 \times \ldots \times S_N \times A_1 \times A_2 \times \ldots \times A_N \rightarrow S_1 \times S_2 \times \ldots \times S_N$$

Goal: $$\pi_i^* = \arg\max_{\pi_i} \Sigma_{t=0}^{T} \gamma^t r_i^t$$

Challenges

1. Millions of bidders
2. Huge overhead in state/action synchronization

Multi-Agent Optimal Bidding

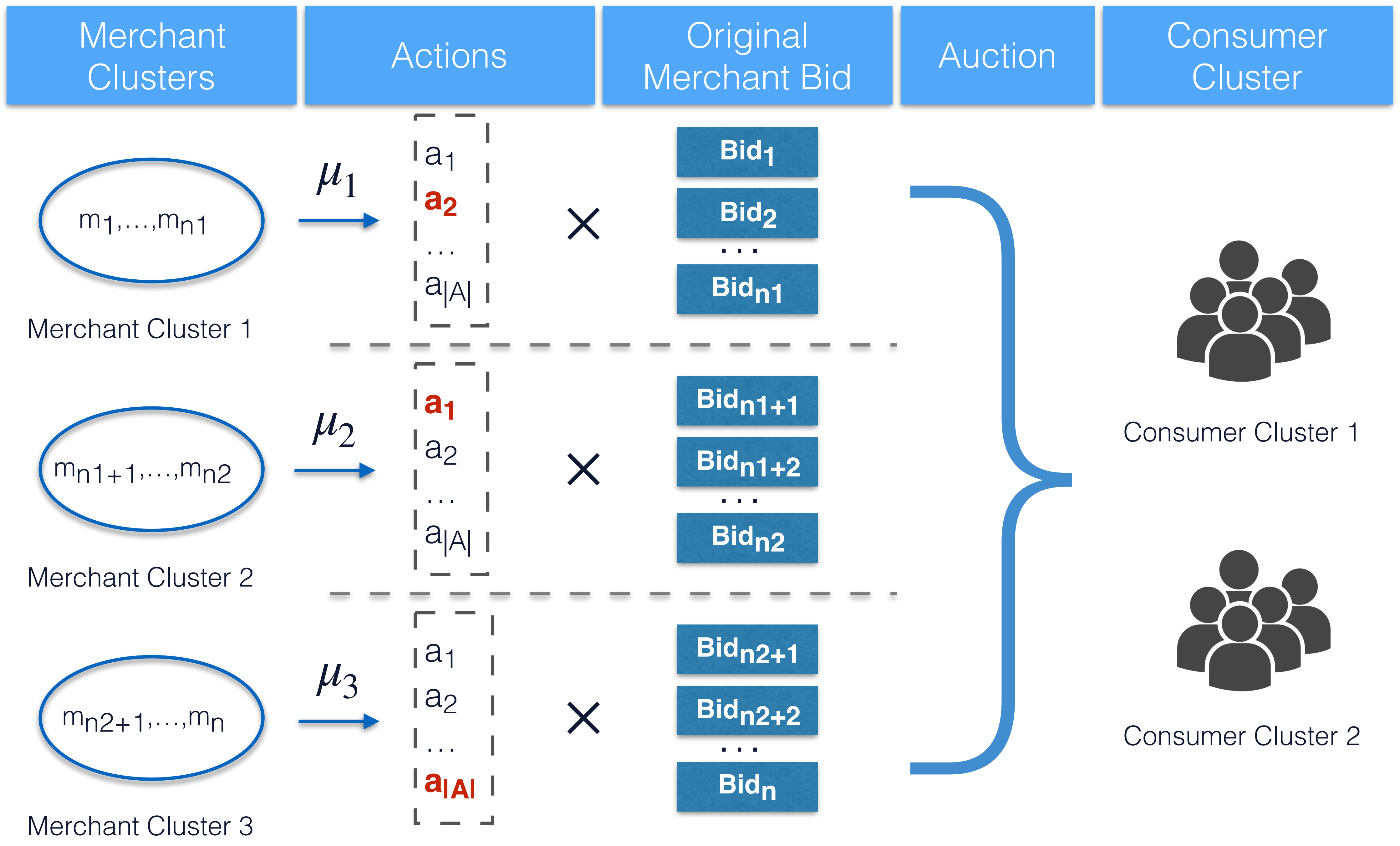
Solution - Distributed Coordinated Multi-Agent Bidding (DCMAB)

Resolve scalability issues

1. Group advertisers into clusters. Each cluster is an **agent**.
2. Group users into clusters.
3. State is summarized and synced in every T time window.
4. Algo: Distributed Coordinated Multi-Agent Bidding (DCMAB)

MDP formulation details

state: spending and GMV of ALL merchant clusters on ALL consumer clusters
action: selecting and applying a bid adjustment factor **a**
reward: reward is the GMV obtained by the agents



阿里妈妈

Multi-Agent Optimal Bidding

Experiments

GMV from different bidding agents

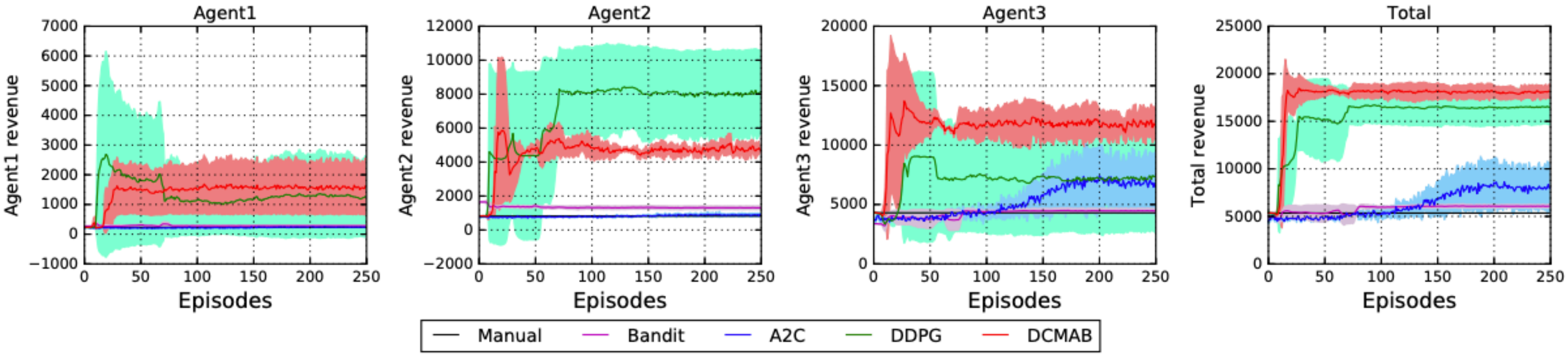
	AgentC1	AgentC2	AgentC3	Total
Manual	231	817	4299	5347
Bandit	281±21	1300±50	4422±171	6003±123
A2C	238±7	872±104	7365±2387	8477±2427
DDPG	1333±1471	7938±2538	7087±4311	16359±1818
DCMAB	1590±891	4763±721	11845±1291	18199±757

Manual: bid with original bids

Bandit: bid based only on impression feature (does not consider budget)

A2C: on-policy actor-critic w/o memory replay. Critic function doesn't take other agents' actions as input

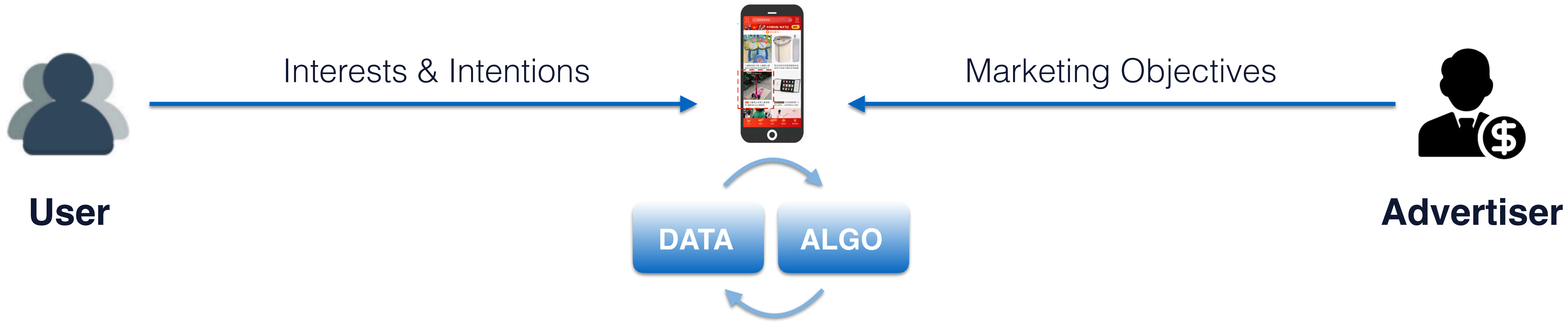
DDPG: off-policy actor-critic with memory replay. Critic function doesn't take other agents' actions as input



Summary

阿里妈妈

Summary



MATCHING	PREDICTION	STRATEGY
Retrieve the most promising Top K candidates from LARGE corpus	Predict user response probabilities (e.g., CTR) with accurately	Publisher: Impression allocation Advertiser: Ad delivery strategies
In this talk	In this talk	In this talk
Tree-based Deep Match (TDM)	From **Deep Interest Network** to **Multi-channel Interest Memory Network**	**Single-agent Bidding** **Multi-agent Bidding**
TDM -> FDM	**Thousands of activities -> Life-long model**	**Bid Optimization -> Mechanism Optimization**

References

List of publications discussed in this talk

Zhu, Han, et al. Learning Tree-based Deep Model for Recommender Systems. KDD 2018
Zhu, Han, et al. Joint Optimization of Tree-based Index and Deep Model for Recommender Systems. arXiv preprint 2019
Zhou, Guorui, et al. Deep interest network for click-through rate prediction. KDD 2018
Zhou, Guorui, et al. Deep interest evolution network for click-through rate prediction. AAAI 2019
Pi, Qi, et al. Practice on Long Sequential User Behavior Modeling for Click-Through Rate Prediction. KDD 2019
Wu, Di, et al. Budget constrained bidding by model-free reinforcement learning in display advertising.CIKM 2018
Jin, Junqi, et al. Real-time bidding with multi-agent reinforcement learning in display advertising. CIKM 2018

Related work discussed in this talk

[1] G Linden, B Smith, J York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering, IEEE Internet Computing, 2003
[2] P Covington, J Adams, E Sargin. Deep Neural Networks for YouTube Recommendation. RecSys 2016
[3] Balázs Hidasi, et al. Session-based recommendations with recurrent neural networks. ICLR 2016.
[4] Xu Chen, et al. Sequential recommendation with user memory networks. WSDM 2018.
[5] Zhang W, Ren K, Wang J. Optimal real-time bidding frameworks discussion[J]. arXiv preprint arXiv:1602.01007, 2016.
[6] Cai, Han, et al. Real-time bidding by reinforcement learning in display advertising. WSDM 2017.

THANK YOU

阿里妈妈