

全民K歌直播推荐算法实践

—— timmili, kevinshuang, graywang



腾讯音乐娱乐集团
TENCENT MUSIC ENTERTAINMENT

目录

CONTENT

- 01 业务背景及团队介绍
- 02 推荐系统架构及挑战
- 03 召回模型算法设计
- 04 精排模型算法设计



01 业务背景及团队介绍

背景介绍

腾讯音乐娱乐集团（TME），2018年从腾讯拆分独立上市，涵盖四大移动音乐产品，月活用户量超过8亿。其中全民K歌以唱为核心，不断丰富音乐功能和娱乐玩法，当前月活1.5亿+。



QQ音乐



酷狗音乐



酷我音乐



全民K歌



中央曲库



原创音乐



节目制作



合资厂牌

关于全民K歌

在线K歌用户的首选APP

81.5%

用户行业平均
渗透

77.7%

月活占在线K歌
行业总数量

81.5%

日活跃设备
数占比达

数据来源：艾瑞咨询《2020年中国在线K歌社交娱乐行业研究报告》



团队介绍

全民K歌算法团队，负责全民K歌各场景的推荐功能算法开发与工程实现，目前团队**20+**人，分布在深圳、北京两地。



期待加入，base地点深圳&北京

简历扫描二维码



业务背景

全民K歌涉及多样化的推荐场景，涵盖内容、直播、歌房、点歌等多种形态。

优质UGC推荐

关注流推荐

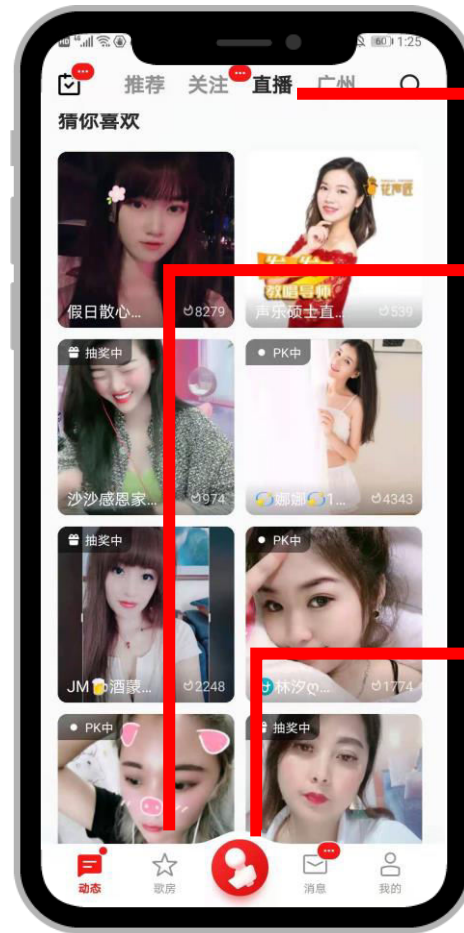
同城社交推荐



直播推荐

歌房推荐

点歌推荐





推荐Feed直播间



UGC直播间



直播Tab



上下滑、左滑



关注Feed

直播

关注

直播TAB点击率
老用户场景
时长

平台伴奏偏好用户
↓
推荐FEED点击率
新用户场景
UGC广播点击率
↑
社交偏好用户

完成度 平稳度
流量控制
细粒度

送礼

送花

社区指标

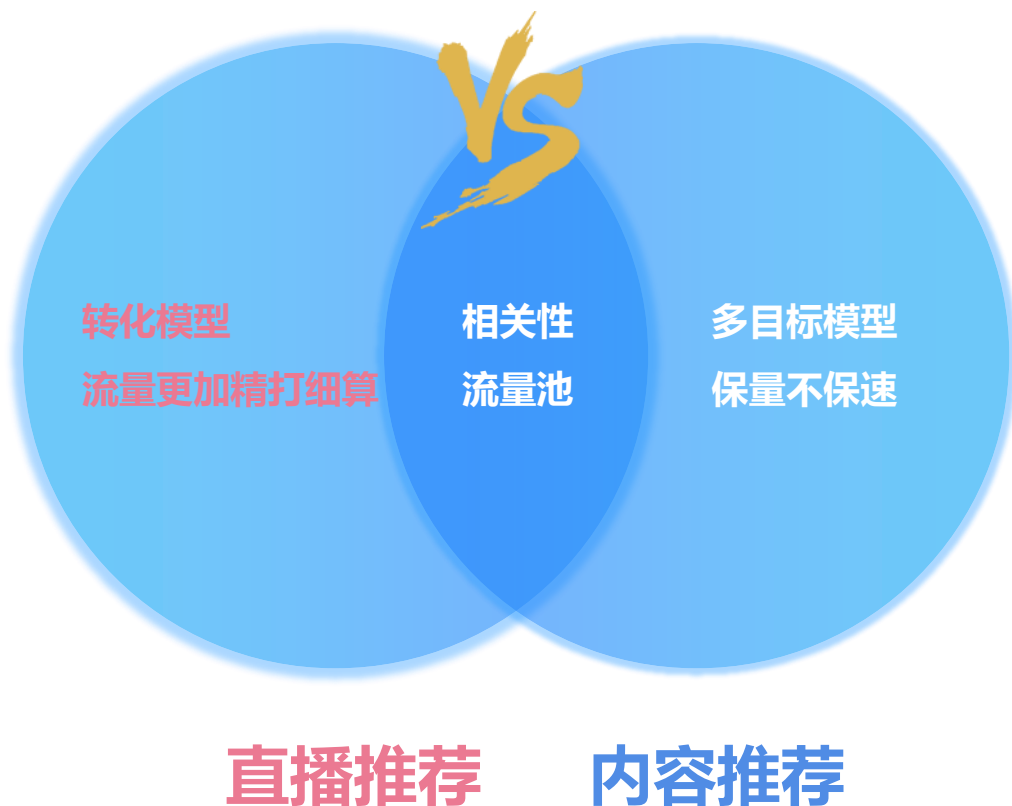
增长指标

运营指标

商业指标



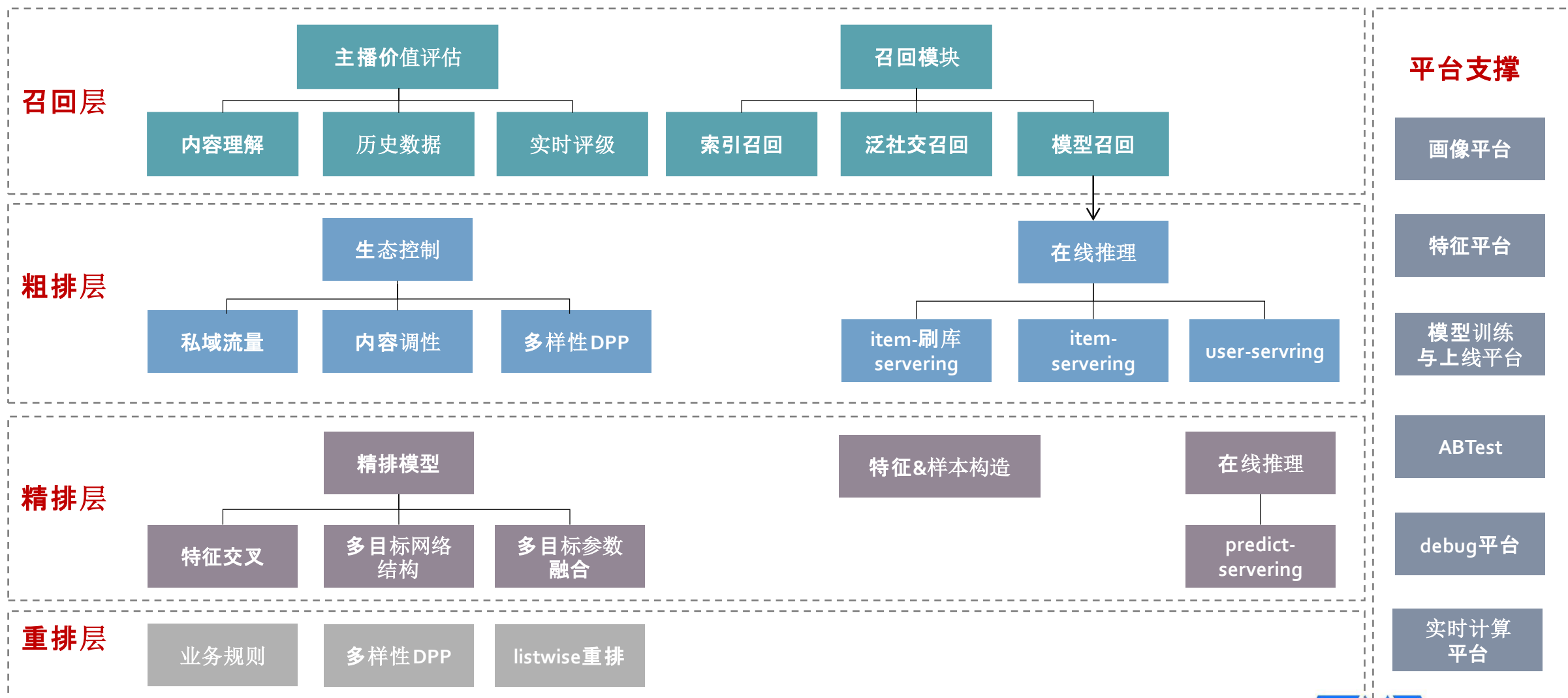
业务背景





02 推荐系统架构及挑战

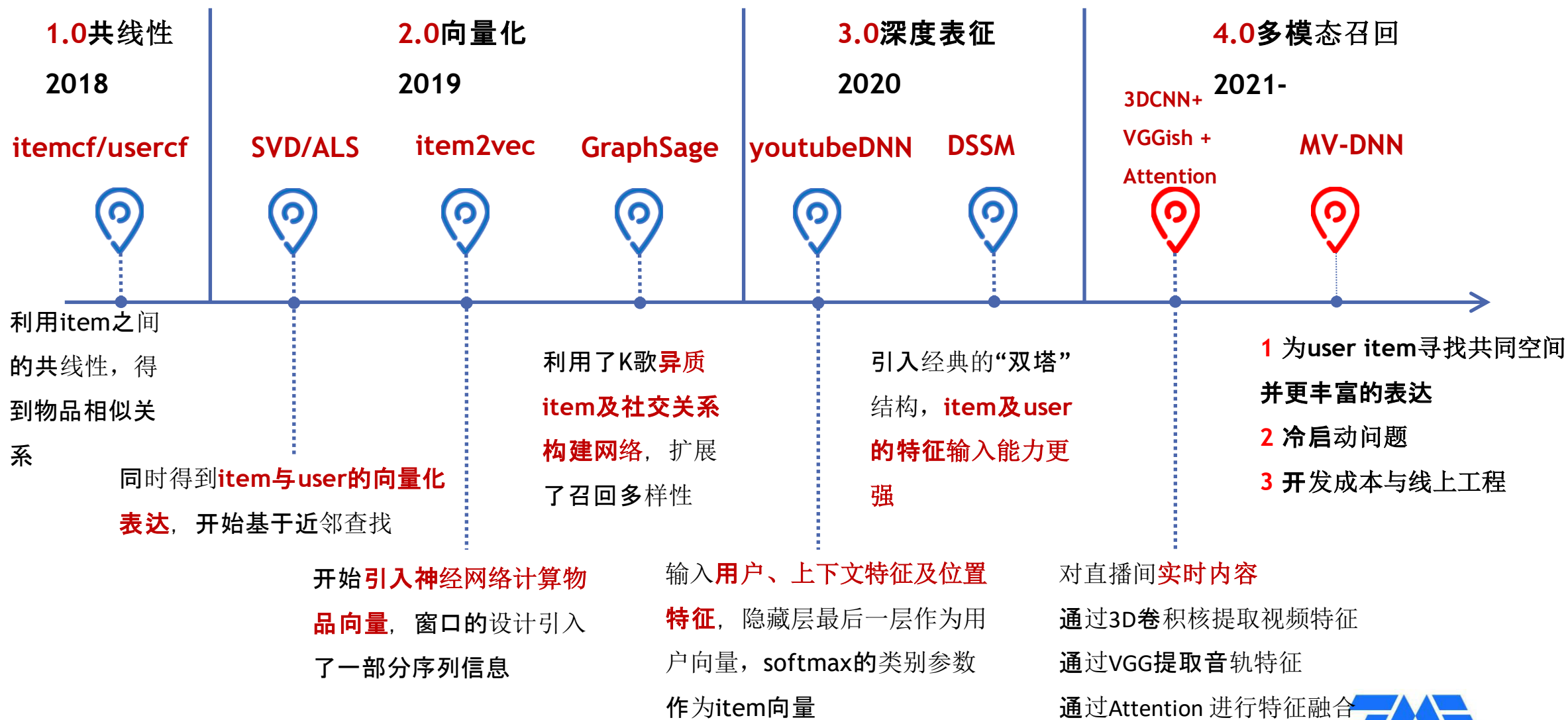
推荐系统架构



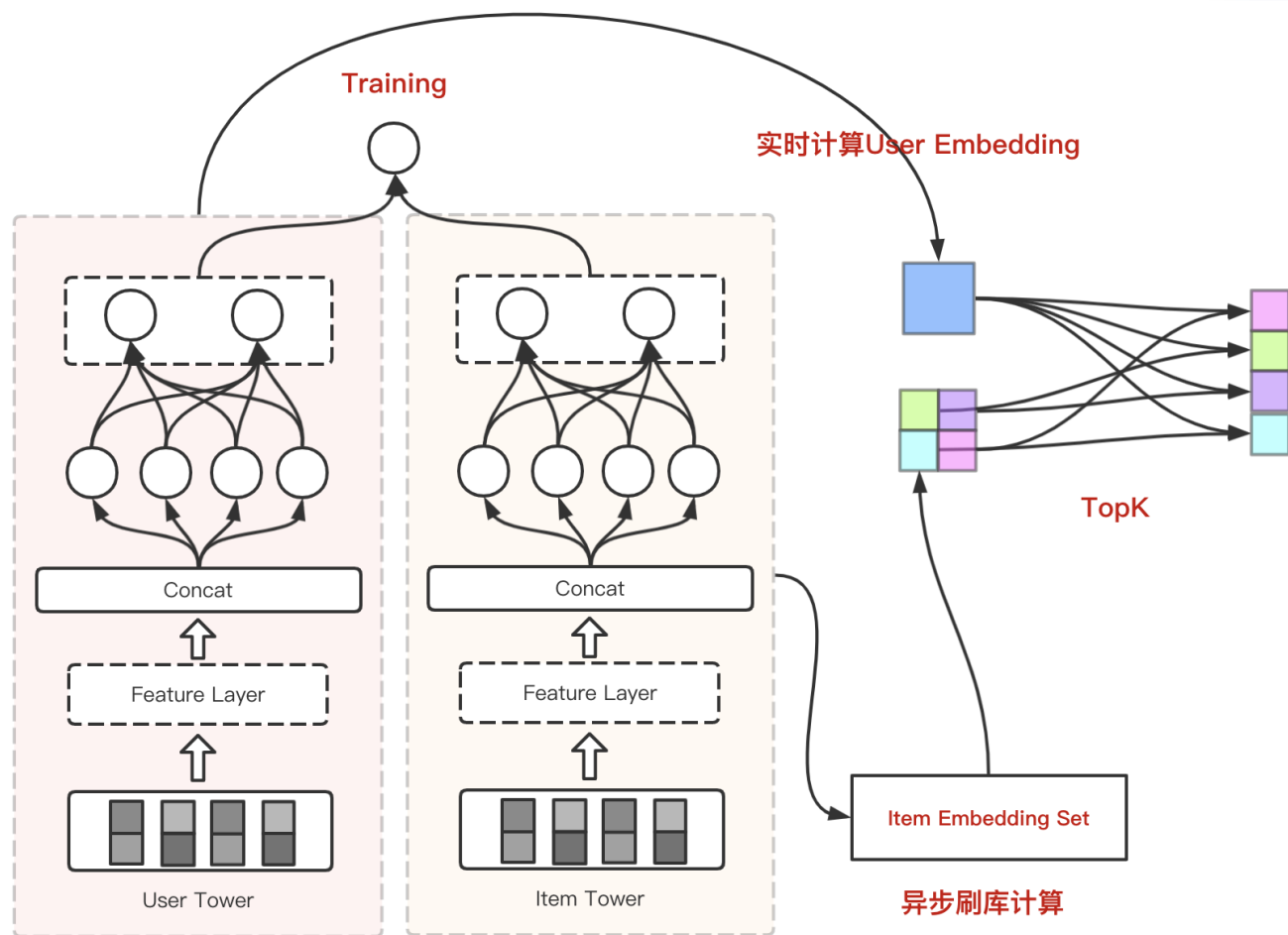


03 召回模型算法设计

召回模型迭代



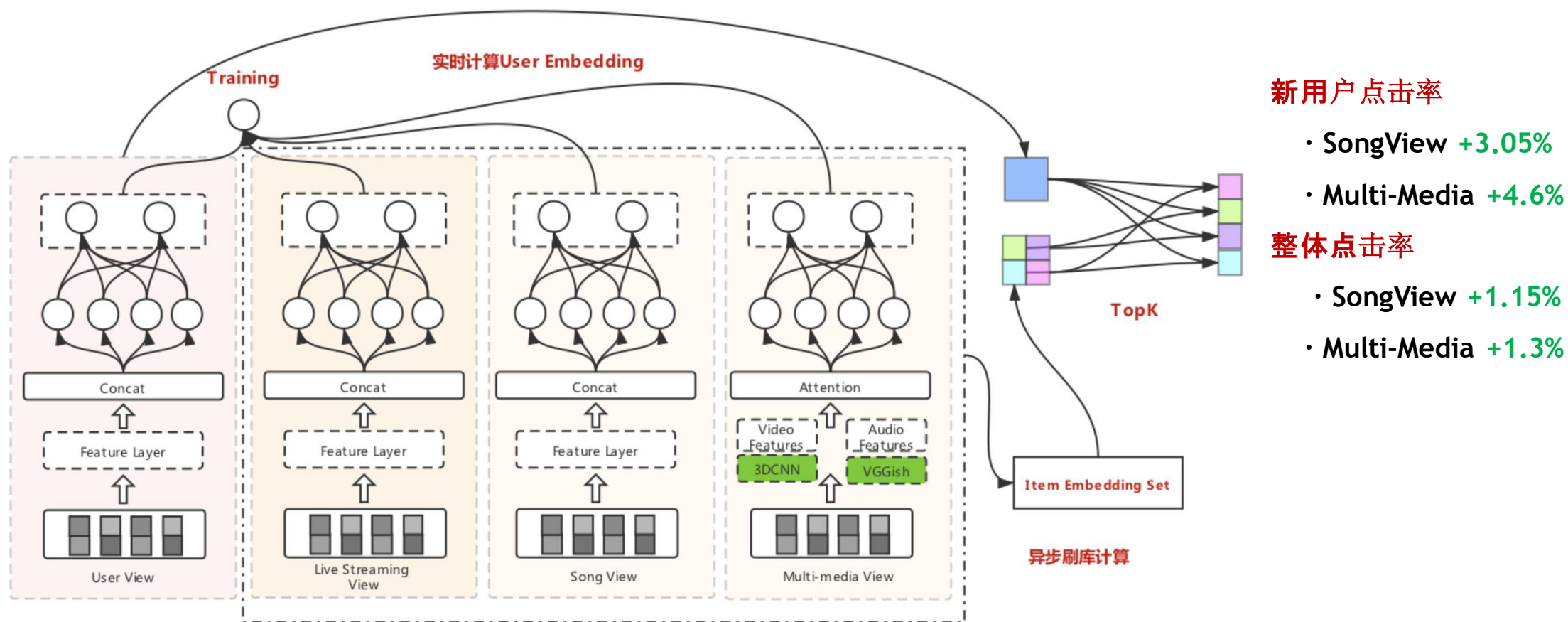
召回双塔模型



方法	模型能力	索引效能	存在的问题
Content-based(cb)	启发式统计	倒排	粗略
item-based(i2i)	启发式统计	Trigger	非学习, 非面向全库
向量检索(i2i)	内积模型	全库&分段请求	缺乏U&I联合优化
内积模型向量检索(u2i)	内积模型	全库&一次请求	



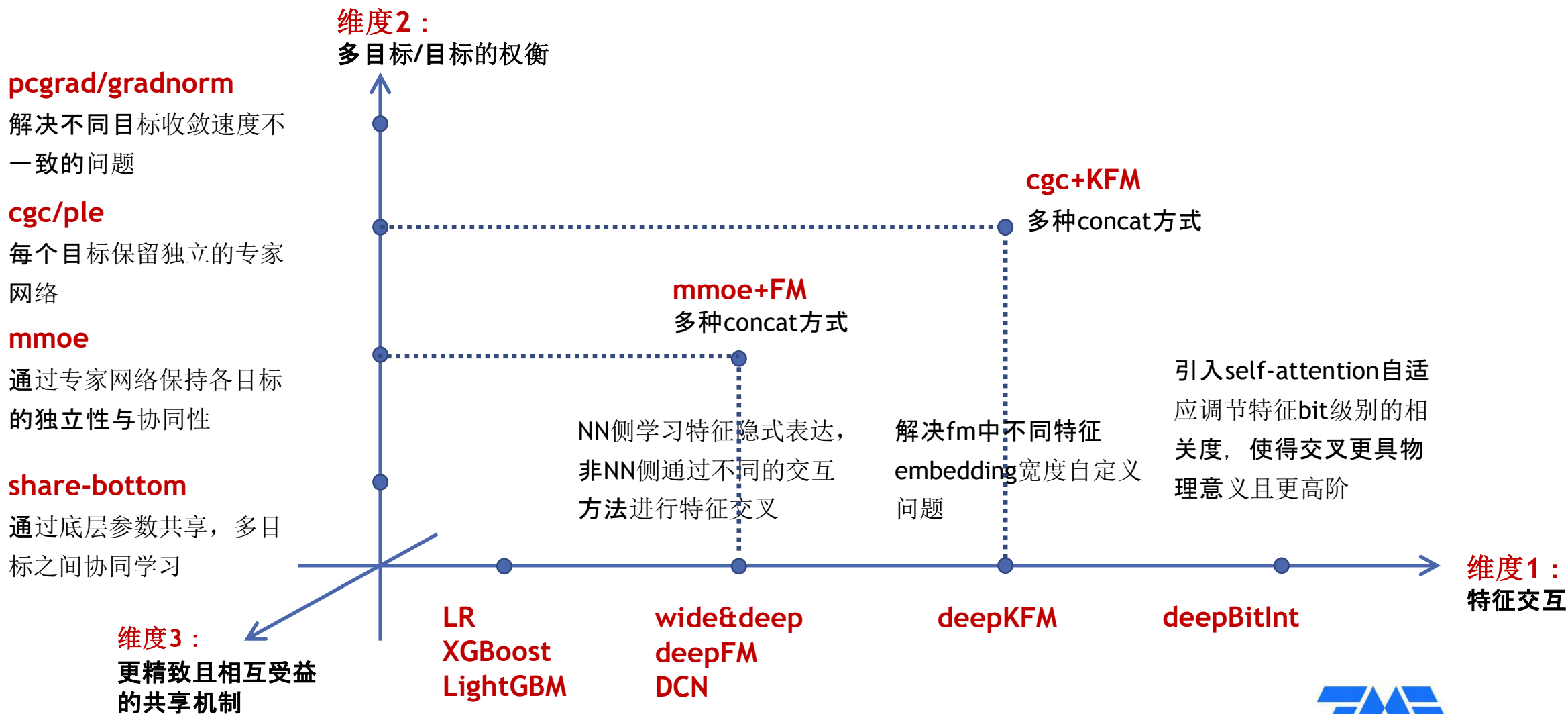
召回多模态模型实践



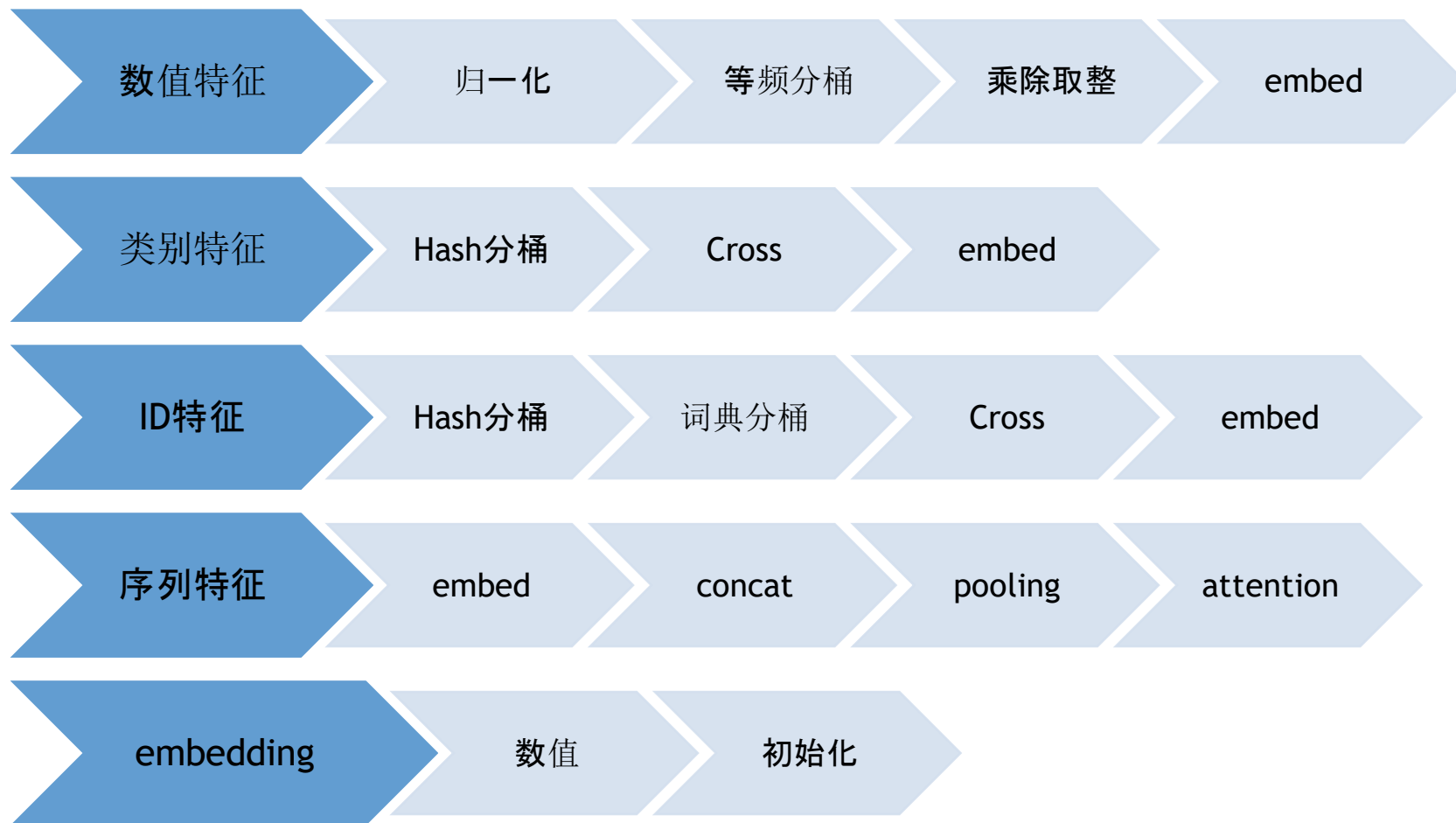


04 精排模型算法设计

精排模型多维度迭代



第一节：特征处理



第一节：采样及加权

➤ Skip-Above

使用场景：CVR/CTR

➤ Click-Around

使用场景：CTR/CVR

样本加权：

$$\sum_i w_i f_i(x)$$

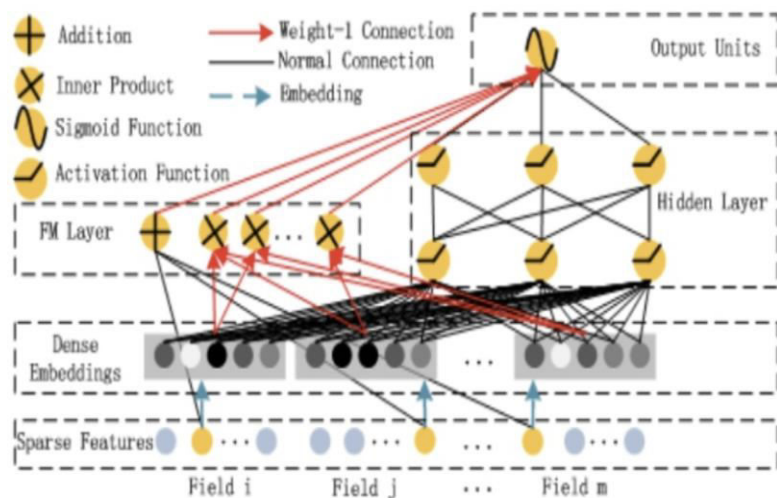
i代表用户行为，包括：送花/送礼/观看直播/观看异步作品等

直播间1	
直播间2	
直播间3	
直播间4	
直播间5	→ last click
直播间...	

直播间1	
直播间2	
直播间3	
直播间4	→ click
直播间5	
直播间...	



第二节：特征交叉-KFM



$$\hat{y} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n f(x_i, x_j) + \widehat{y_{dnn}} + b$$

AUC比对: DeepFM->DeepNFM->DeepKFM
+0.3% +0.4%

$$\text{FM} \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

$$\text{NFM} \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^n x_i \mathbf{v}_i \odot \mathbf{v}_j x_j$$

KFM：更普适的核函数，将低阶特征更自由的映射到共同空间进行交叉

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n x_i^T \phi_{i,j} x_j \quad (1)$$

$$\sum_{i=1}^{n-1} x_i^T \sum_{j=i+1}^n \phi_{i,j} x_j \quad (2)$$

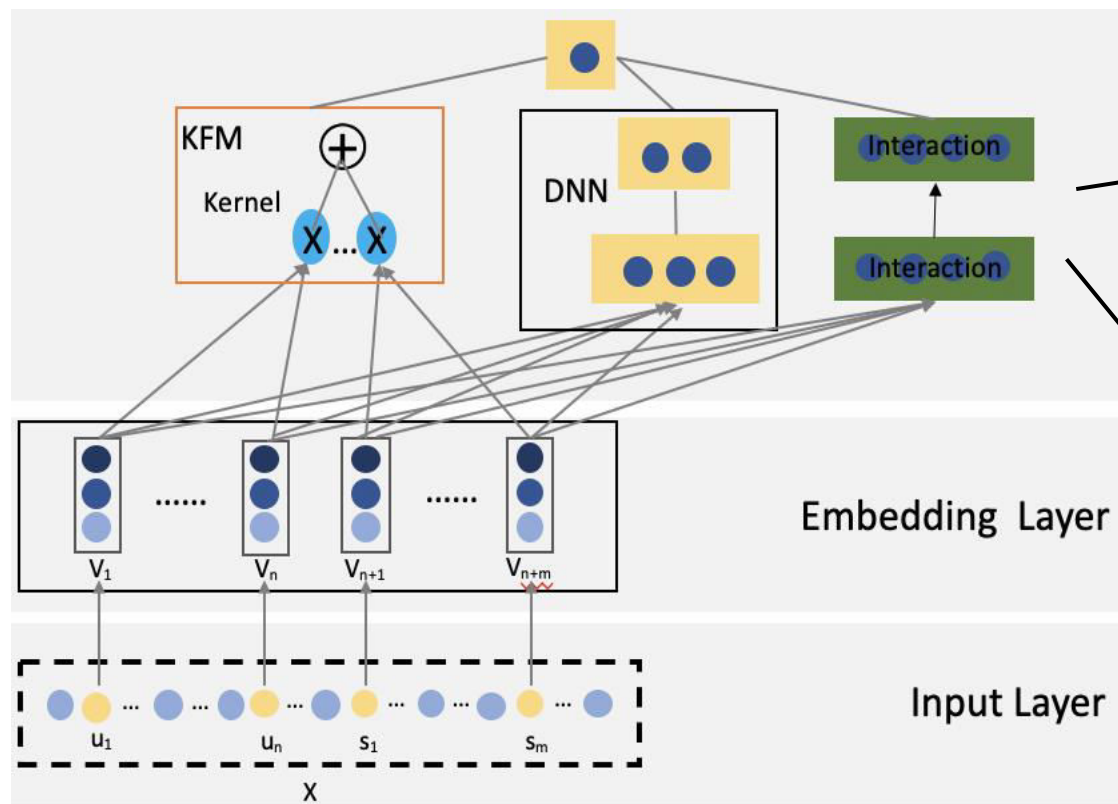
$$\sum_{i=1}^{n-1} x_i^T [\phi_{i,1}, \phi_{i,2}, \dots, \phi_{i,n}] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (4)$$

(3) (4)的计算在Tensorflow中的算法复杂度是O(n+T)



第二节：特征交叉—DeepKFM&Cross | AutoInt



通过引入Cross层和AutoInt层加强特征的高阶交叉能力

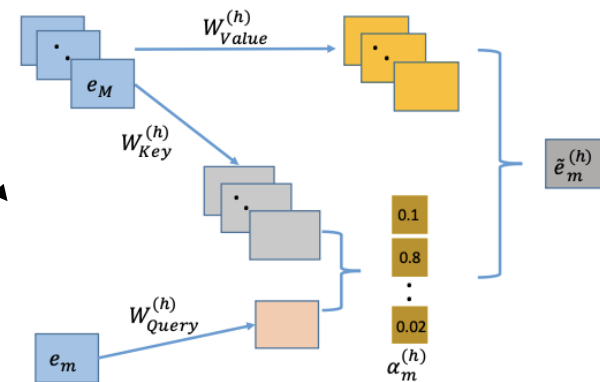
- +Cross AUC+0.004
- +AutoInt AUC+0.005

Diagram illustrating the **Cross** layer operation:

$$y = x_0 * x' * w + b + x$$

The diagram shows the **Output** (a vector of three orange circles) is calculated as the product of **Feature Crossing** (a vector of three brown circles) and **Input** (a vector of three yellow circles), followed by a summation with **Bias** (a vector of three grey circles) and the **Input** vector.

AutoInt



第二节：特征交叉—DeepBitInt

$$\alpha_{i,k}^{(h)} = \frac{\exp(\phi^{(h)}(e_i, e_k))}{\sum_{l=1}^N \exp(\phi^{(h)}(e_i, e_l))}$$

$$\alpha_{i,k}^{(h)} = \frac{\exp(\phi^{(h)}(T_i, T_k))}{\sum_{l=1}^N \exp(\phi^{(h)}(T_i, T_l))}$$

Field级 $\phi^{(h)}(e_i, e_k) = \left\langle W_{Query}^{(h)} e_k, W_{Key}^{(h)} e_i \right\rangle \Rightarrow$ Unit级

$$\phi^{(h)}(T_i, T_l) = \left\langle W_{Query}^{(h)}, T \right\rangle_i \times \left\langle W_{Key}^{(h)}, T \right\rangle_l$$

$$\tilde{e}_i^{(h)} = \sum_{k=1}^N \alpha_{i,k}^{(h)} (W_{Value}^{(h)} e_k)$$

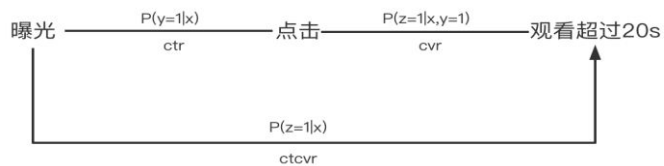
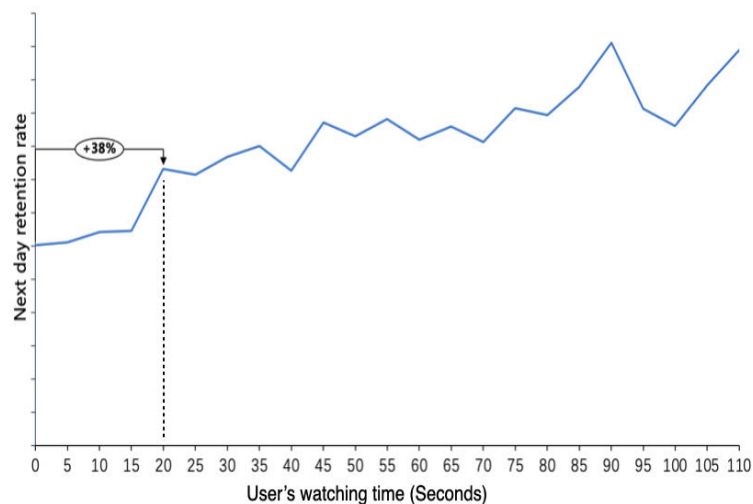
$$\tilde{v}_i^{(h)} = \sum_{k=1}^N \alpha_{i,k}^{(h)} \left\langle W_{Value}^{(h)}, T \right\rangle_k$$

效果：AutoInt的交互粒度从Field到Unit。 离线AUC+0.007

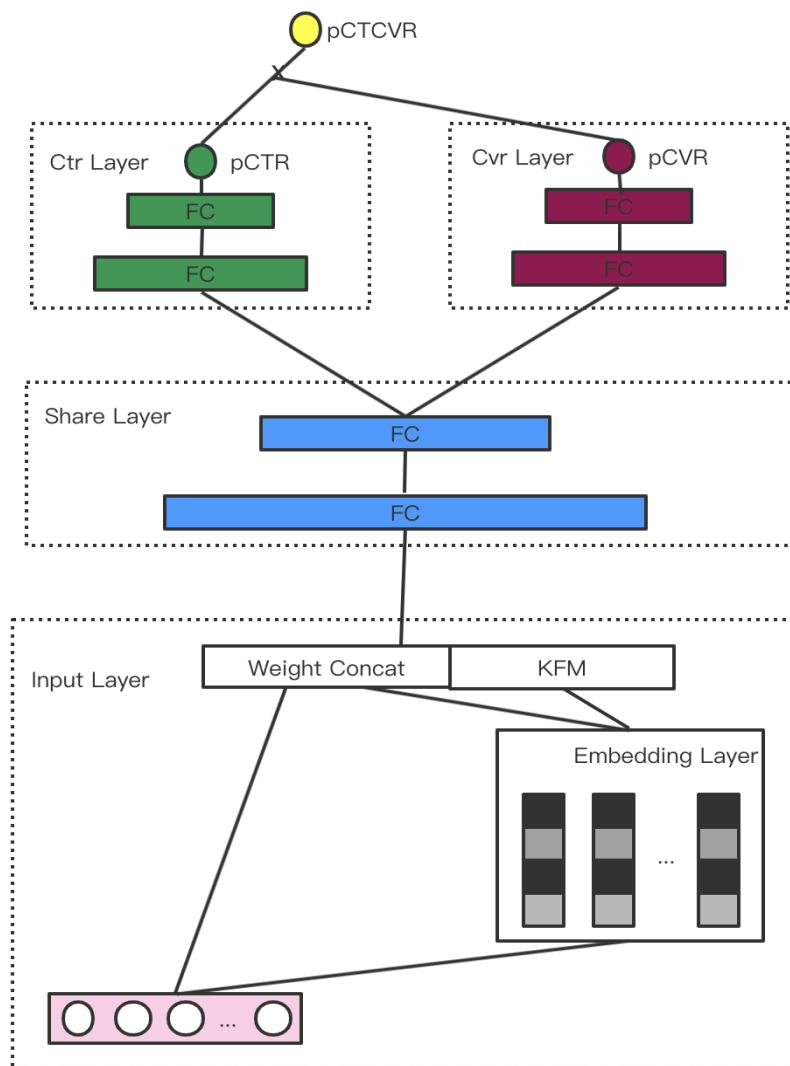


第三节：CVR预估-ESMM

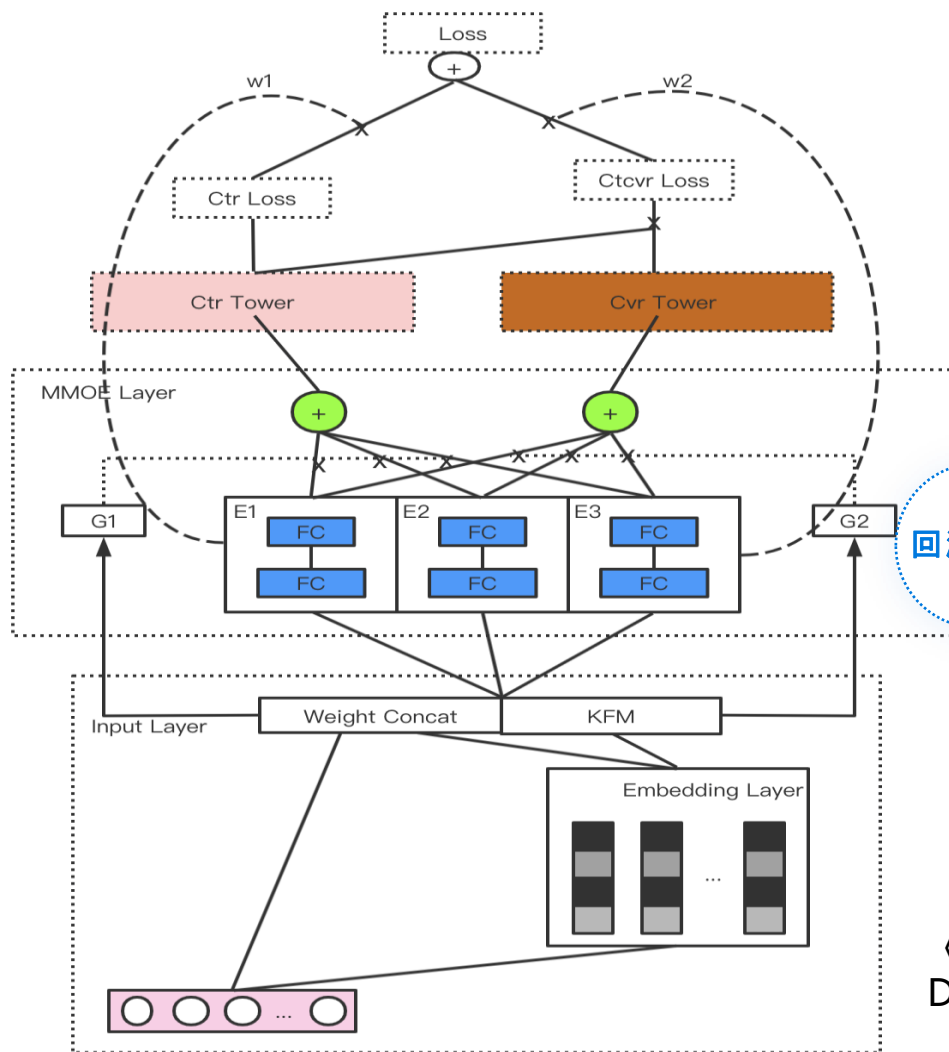
- 直播推荐的本质不是ctr而是cvr
- cvr和ctr从结果和过程都有一定的相关性



三者间关系: $ctcvr = ctr * cvr$



第三节：CVR预估-GradNorm



回溯目标

$$L_{\text{grad}}(t; w_i(t)) = \sum_i \left| G_W^{(i)}(t) - \bar{G}_W(t) \times [r_i(t)]^\alpha \right|_1$$

回溯对象

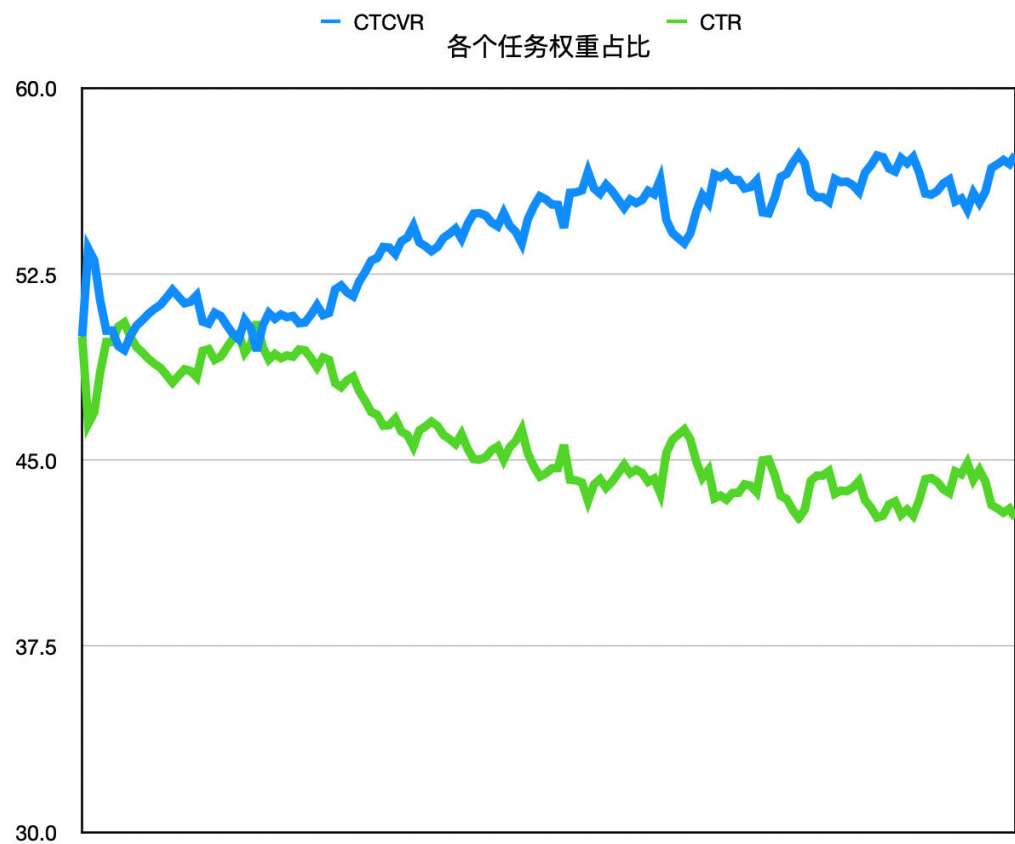
回溯方法

- 1 根据每个任务的**加权**loss和共享特征的weights, 求每个任务的梯度g。
- 2 求每个任务的梯度的2范数 G, 并设定**理想梯度**为G的平均。
- 3 考虑每个任务的进步的速度, 即求训练速度的逆系数, 直观来讲, 速度快的任务权重会减小。
- 4 根据理想梯度求loss-grad, 更新每个任务的权重。

《GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks》



第三节：CVR预估-GradNorm



Hard Task



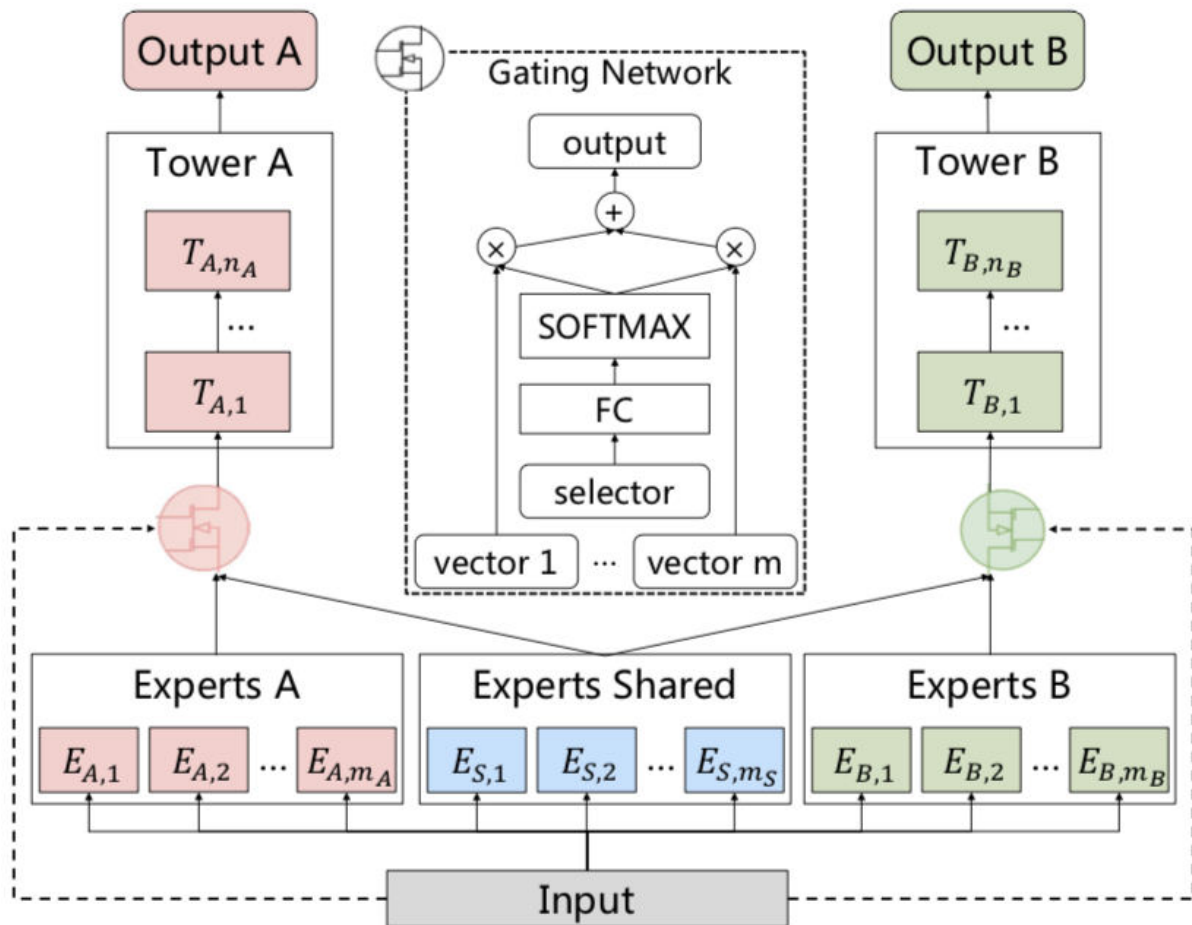
Easy Task

Table 3. Online experimental results.

Compared Method	$\alpha = 0.00$	$\alpha = 0.25$	$\alpha = 0.50$	$\alpha = 0.75$	$\alpha = 1.00$
ESMM	1.90%	1.98%	2.02%	2.05%	2.07%
ESMM+GradNorm	1.91%	2.02%	2.04%	2.06%	2.08%
ESMM+FM	1.89%	1.98%	2.03%	2.05%	2.08%
ESMM+FM+GradNorm	1.98%	2.08%	2.10%	2.14%	2.16%
ESMM+FM+MMoE	1.92%	2.01%	2.06%	2.09%	2.11%
ESMM+FM+MMoE+GradNorm	2.05%	2.21%	2.24%	2.27%	2.29%



第三节：CVR预估-CGC



Customized Gate Control (CGC) Model

进入指标

- 点击率 +4.01%
- 有效点击 +4.5%

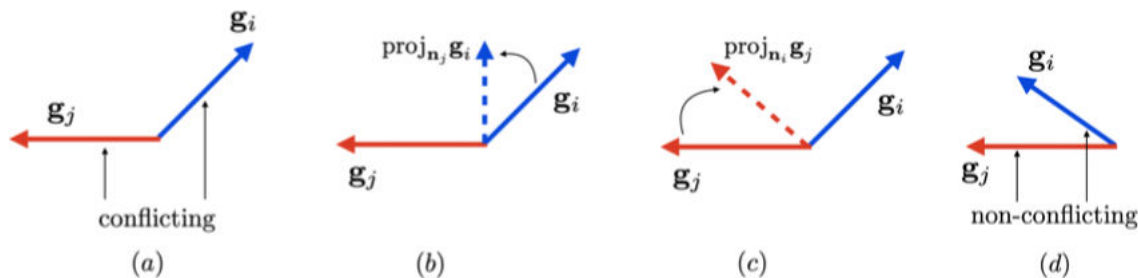
沉浸指标

- 人均时长 +4.3%
- 总时长 +9.11%

《Progressive Layered Extraction (PLE):
A Novel Multi-Task Learning (MTL) Model for Personalized
Recommendations》



第三节：CVR预估-PCGrad



在 (a) 中，任务i和j具有相互冲突的梯度方向，这可能导致相消干扰。在 (b) 和 (c) 中，我们说明了在梯度冲突的情况下的PCGrad算法。PCGrad将任务i的梯度投影到任务j的梯度的法线向量上，反之亦然。在PCGrad下，不冲突的任务梯度 (d) 不会更改，从而可以进行**有益的交互**。

Algorithm 1 PCGrad Update Rule

Require: Model parameters θ , task minibatch $\mathcal{B} = \{\mathcal{T}_k\}$

- 1: $\mathbf{g}_k \leftarrow \nabla_{\theta} \mathcal{L}_k(\theta) \quad \forall k$
- 2: $\mathbf{g}_k^{\text{PC}} \leftarrow \mathbf{g}_k \quad \forall k$
- 3: **for** $\mathcal{T}_i \in \mathcal{B}$ **do**
- 4: **for** $\mathcal{T}_j \stackrel{\text{uniformly}}{\sim} \mathcal{B} \setminus \mathcal{T}_i$ **in random order do**
- 5: **if** $\mathbf{g}_i^{\text{PC}} \cdot \mathbf{g}_j < 0$ **then**
- 6: *// Subtract the projection of \mathbf{g}_i^{PC} onto \mathbf{g}_j*
- 7: Set $\mathbf{g}_i^{\text{PC}} = \mathbf{g}_i^{\text{PC}} - \frac{\mathbf{g}_i^{\text{PC}} \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2} \mathbf{g}_j$
- 8: **return** update $\Delta\theta = \mathbf{g}^{\text{PC}} = \sum_i \mathbf{g}_i^{\text{PC}}$

ctcvr auc提升**0.8**个千分点， ctr auc保持不变

欢迎加入

期待加入，base地点深圳&北京

可扫描以下二维码



创 造 音 乐 无 限 可 能

CREATING ENDLESS
OPPORTUNITIES WITH MUSIC