RecGPT-Mobile: On-Device Large Language Models for User Intent Understanding in Taobao Feed Recommendation

Bin Zhang*
Weipeng Huang*
Dimin Wang
Jialin Zhu
tianji.zb@taobao.com
weipeng.hwp@taobao.com
dimin.wdm@taobao.com
xiafei.zjl@taobao.com
Taobao & Tmall Group of Alibaba
HangZhou, China

Yuning Jiang
Zhaode Wang
Chengfei Lv
Jian Wang
mengzhu.jyn@taobao.com
zhaode.wzd@taobao.com
chengfei.lcf@taobao.com
krod.wj@taobao.com
Taobao & Tmall Group of Alibaba
HangZhou, China

Qichao Ma
Li Chen
Junqing Wu
Yipeng Yu
maqichao.mqc@taobao.com
cl121469@taobao.com
junqing.wjq@taobao.com
linxin.yyp@taobao.com
Taobao & Tmall Group of Alibaba
HangZhou, China

arXiv:2605.04726v1 [cs.IR] 6 May 2026

Abstract

Predicting a user's next search query from recent interaction behaviors is a critical problem in modern e-commerce systems, particularly in scenarios where user intent evolves rapidly. Large Language Models (LLMs) offer strong semantic reasoning capabilities and have recently been adopted to enhance training data construction for next-query prediction. However, due to resource constraints on mobile devices, existing applications are deployed on cloud servers, resulting in high inference costs. In this paper, we propose **RecGPT-Mobile**, a framework that designs a lightweight LLM-based intent understanding agent to improve recommendation quality in mobile e-commerce scenarios. By deploying LLM directly on mobile devices, our approach can capture the evolving interests of users more quickly and adjust the recommendation results in real time. Extensive offline analyzes and online experiments demonstrate that our method significantly improves the accuracy of recommendation results, laying a practical path for LLM deployment in production-scale recommendation systems on mobile devices, as well as a scalable solution for integrating LLMs into real-world next-query prediction systems.

CCS Concepts

• **Information systems** → **Language models**.

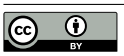
Keywords

On-device LLMs, User Intent Understanding, Feed Recommendation

ACM Reference Format:

Bin Zhang, Weipeng Huang, Dimin Wang, Jialin Zhu, Yuning Jiang, Zhaode Wang, Chengfei Lv, Jian Wang, Qichao Ma, Li Chen, Junqing Wu, and Yipeng Yu. 2026. RecGPT-Mobile: On-Device Large Language Models for User Intent Understanding in Taobao Feed Recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3805712.3808410

*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution 4.0 International License.

SIGIR '26, Melbourne, VIC, Australia

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3808410

1 Introduction

The exponential growth of digital content have made personalized recommendation systems indispensable in modern applications. Although traditional cloud-based recommendation systems [3, 10, 24] can utilize user behaviors effectively, it still struggles to meet the requirements for low-latency real-time inference in the mobile environment. Besides, centralized architectures introduce unavoidable communication delays between edge devices and remote servers and impede the perception of rapidly changing user intents.

By building feature centers and collecting user behaviors locally, device-level recommendation systems [7–9, 21] make the results more real-time. However, deploying sophisticated model directly on resource-constrained devices presents significant challenges: LLMs [1, 2, 14, 22] are typically too massive in size and computationally intensive for direct mobile deployment. Recent advances in model compression techniques such as quantization [6], pruning [11], knowledge distillation [12] have emerged to bridge this gap, making it possible to deploy complex models on mobile devices.

To address these limitations, we propose RecGPT-Mobile, a novel device-level framework that deploys a lightweight LLM as an intent agent directly on the user's mobile device. In summary, our contributions are as follows:

- To our knowledge, this work presents the first implementation of an LLM-based recommendation system on mobile devices. We apply preference-based optimization and lightweight model compression to enable efficient deployment of LLM-based retrieval systems.
- We design an intent agent that translates implicit user behaviors into explicit intent queries, as well as a frequency control mechanism to reduce redundant inference on the client device and improve resource utilization.
- We conducted extensive experiments on Mobile Taobao, involving millions of real users and diverse shopping sessions. The results show that RecGPT-Mobile significantly improves the relevance and interpretability of recommendations.

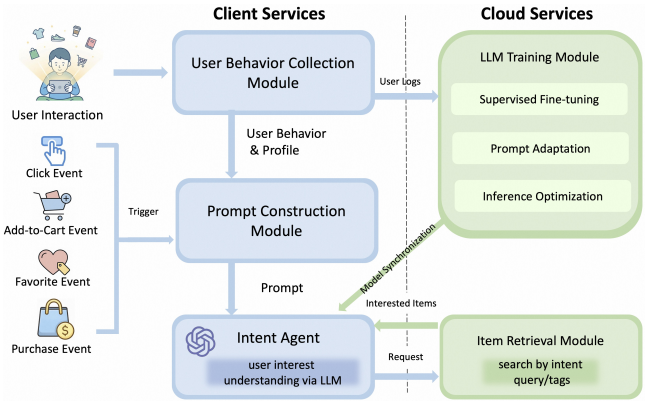


Figure 1: Framework of RecGPT-Mobile.

2 Related Work

On-device recommendation Systems. EdgeRec [8] was the first to implement ranking models directly on mobile devices to reduce signal latency. Gong et al. [7] implemented a real-time recommendation framework in the Kuaishou app, which processes user feedback locally on the device. To tackle the limitations of delayed processing in cloud-based re-ranking, DIR [18] directly integrates re-ranking framework on mobile devices.

Cloud-based LLMs for recommendation Systems. HSTU [23] reformulated recommendation as a sequential transduction task within a generative framework. OneRec [4] is a unified end-to-end generative framework that replaces the traditional cascaded strategy. RecGPT [20] is a LLM-based production-scale recommendation framework that replaces log-fitting methods with intent-centric reasoning. It integrates user interest mining, item tagging, retrieval, and explanation generation into a closed loop.

Mobile LLMs. Deploying LLMs on resource-constrained edge devices [15–17, 19] is crucial for low latency and data locality, necessitating advances in architecture and compression to reduce memory use. Collaborative edge-cloud deployment and hardware-aware optimizations, such as leveraging mobile GPUs and NPUs, help balance resource usage and inference speed.

3 Methodology

3.1 System Overview

Fig 1 shows the overall architecture of RecGPT-Mobile framework. **User Behavior Collection Module** functions as a local repository to collect and cache user behavior data. Triggered by specific high-intent actions, **Prompt Construction Module** synthesizes raw behavior and profile data into a structured prompt. By processing the synthesized prompt, the **Intent Agent** transforms complex behavioral signals into a clear user intent query. Upon receiving a request from the intent agent, **Item Retrieval Module** executes a search based on the generated intent query, and returns a set of interested items back to the client side. The **LLM Training Module** employs supervised fine-tuning, prompt adaptation and inference optimization to ensure the model remains both adaptive and accurate. In the following sections, we focus on the design and implementation of the LLM Training Module.

Table 1: Training Sample Construction and Composition

Sample Type	Data Source	Ratio
Behavior-driven	Purchase & search logs	60%
Co-purchase	Co-purchase item matrix	20%
LLM-based	GPT-based rewriting	15%
Human-annotated	Manually annotated data	5%

3.2 Preliminaries

We focus on the next-query prediction task given heterogeneous user behavior sequences. Formally, let a user behavior sequence be:

$$\mathcal{B} = \{(i_1, a_1, t_1), (i_2, a_2, t_2), \ldots, (i_T, a_T, t_T)\}, \tag{1}$$

where i_T denotes the item interacted with at timestamp T, $a_T \in \{\text{click}, \text{cart}, \text{favorite}, \text{purchase}\}$ denotes the action type, and t_T denotes the action time. Given the observed behavior sequence $\mathcal{B}$, the task is to predict the next potential search query $q \in \mathcal{Q}$, where $\mathcal{Q}$ is the space of search queries. The objective can be expressed as:

$$q^* = \arg\max_{q \in \mathcal{Q}} P(q \mid \mathcal{B}) \text{ s.t. Resource Constrains}, \tag{2}$$

where $P(q \mid \mathcal{B})$ captures the conditional probability of a search query given the user's recent behaviors.

3.3 Supervised Fine-tuning

To effectively model next-query prediction from heterogeneous user behavior sequences, we construct training data from multiple complementary sources, each capturing different aspects of user intent. As shown in Table 1, our training set consists of four types of samples: **behavior-driven samples**, extracted from purchase logs and search logs by linking purchased items to post-purchase queries and inferring complementary relations; **co-purchase relation samples**, automatically generated from the item-level co-purchase matrix to capture complementary signals from purchase co-occurrence; **LLM-based augmentation**, which rewrites rule-based samples with LLM to increase linguistic diversity while preserving semantics; and **human-annotated samples**, a small manually reviewed set used for quality calibration and reliable evaluation. **Prompt 1** below is used for supervised fine-tuning of the intent agent.

> **Prompt 1: Next-query Prediction Prompt.**
>
> **Input:** Given a timestamp t and a user behavior sequence $\mathcal{B} = \{(i_1, a_1, t_1), (i_2, a_2, t_2), \ldots, (i_n, a_n, t_n)\}$, please infer the user's next search intent, and generate the most likely search query that reflects the user's latent requirement.
>
> **Output:** <Predicted search query>.

3.4 Adaptive Prompt Construction

Algorithm 1 describes the adaptive prompt construction procedure for next-query prediction under on-device constraints. Given a user behavior sequence $\mathcal{B}$ and the scenario context s, the algorithm first summarizes heterogeneous user interactions into a compact

Algorithm 1 Adaptive Prompt Construction with Template & Structural Adaptation

Input: Behavior sequence $\mathcal{B} = \{(i_t, a_t, t_t)\}_{t=1}^{T}$; scenario s; template pool $\mathcal{T}$; component set C; scorer M_{score}; on-device budget $C_{\max}$ (latency/memory/token).
Output: Adaptive prompt P^*.

Stage 1: Feature Extraction
Compute behavior features $\Phi \leftarrow \Phi(\mathcal{B}) = [\Phi_{\text{act}}, \Phi_{\text{rec}}, \Phi_{\text{div}}, \Phi_{\text{freq}}]$, where Φ_{act} represents action type, Φ_{rec} encodes recency, Φ_{div} denotes diversity, and Φ_{freq} is the action frequency.
Stage 2: Template-level Adaptation
for all $T_k \in \mathcal{T}_s$ **do**
    $\alpha_k \leftarrow M_{\text{score}}(T_k, \Phi, s)$
end for
$p_k \leftarrow \exp(\beta\alpha_k) \,/\, \sum_j \exp(\beta\alpha_j)$
$T^* \leftarrow \arg\max_{T_k \in \mathcal{T}_s} p_k$
Initialize prompt $P \leftarrow T^*$
Stage 3: Structural-level Adaptation
for all $c \in C$ **do**
    $\Delta(c) \leftarrow M_{\text{score}}(P \oplus c, \Phi, s) - M_{\text{score}}(P, \Phi, s)$
    if $\Delta(c) > \tau$ **and** $\text{Cost}(P \oplus c) \le C_{\max}$ **then**
        $P \leftarrow P \oplus c$
    end if
end for
Stage 4: Budget Enforcement & Finalization
$P^* \leftarrow \arg\max_{P' \subseteq P} M_{\text{score}}(P', \Phi, s)$ **s.t.** $\text{Cost}(P') \le C_{\max}$
Instantiate behavior tokens from $\mathcal{B}$ into P^*
return P^*

behavior feature vector $\Phi(\mathcal{B})$, which captures action distribution, temporal recency, semantic diversity, and interaction frequency.

Based on the extracted features, the algorithm performs template-level adaptation by scoring candidate prompt templates using a lightweight scoring model and selecting the most suitable template conditioned on both the behavior characteristics and scenario context. After the template is selected, structural-level adaptation is applied to incrementally refine the prompt structure. Specifically, candidate structural components are evaluated according to their marginal utility gain, and only those that contribute positively while satisfying the on-device budget constraints are incorporated into the prompt. Finally, a budget-aware pruning step is conducted to ensure that the constructed prompt maximizes utility under strict latency, memory, and token constraints. The resulting prompt is instantiated with concrete behavior tokens and used as the input to the on-device language model for next-query prediction.

3.5 Mobile Device Inference Optimization

The core objective of the Intent Agent is to update semantic representation when the user's intent changes, while avoiding frequent computation when behavior is stable, thus achieving a balance between on-device running performance and recommendation accuracy. To this end, we designed the intent agent trigger pipeline, as shown in Fig 2.

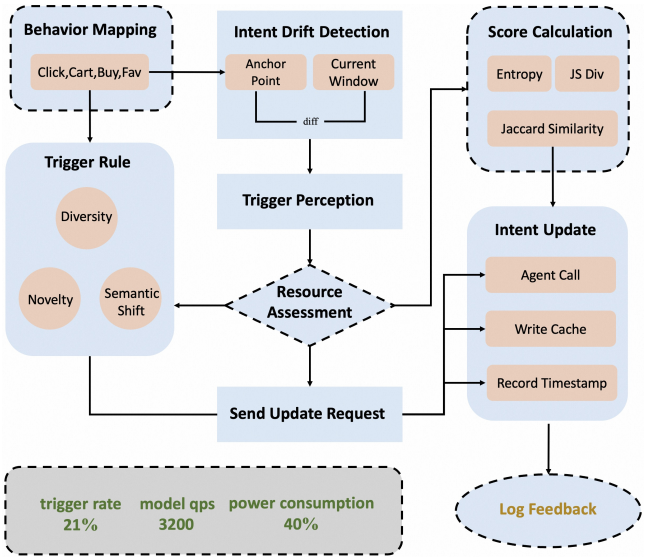


Figure 2: Mobile Intent Agent Trigger Pipeline.

Given a user behavior sequence $\mathcal{B}$ within a sliding window, each interaction is mapped to a discrete semantic tag (e.g., category, brand, or intent type). This yields a normalized tag distribution $P_{\mathcal{B}}$ over the current window. Let $P_{\mathcal{B}}^{(t)}$ and $P_{\mathcal{B}}^{(t-1)}$ denote the tag distributions at the current step and the previous trigger point, respectively. We quantify intent drift from three complementary perspectives. First, we measure the change in uncertainty of user intent using entropy:

$$H(P_{\mathcal{B}}) = -\sum_{k} P_{\mathcal{B}}(k) \log P_{\mathcal{B}}(k). \tag{3}$$

The absolute entropy difference

$$\Delta H = \left| H(P_{\mathcal{B}}^{(t)}) - H(P_{\mathcal{B}}^{(t-1)}) \right| \tag{4}$$

captures whether the user intent becomes more focused or more exploratory. Second, we assess semantic overlap between consecutive behavior windows using Jaccard similarity:

$$JA(\mathcal{Z}^{(t)}, \mathcal{Z}^{(t-1)}) = \frac{|\mathcal{Z}^{(t)} \cap \mathcal{Z}^{(t-1)}|}{|\mathcal{Z}^{(t)} \cup \mathcal{Z}^{(t-1)}|}, \tag{5}$$

where $\mathcal{Z}^{(t)}$ denotes the set of observed tags. A lower Jaccard score indicates a larger shift in semantic focus. Finally, we explicitly model distributional drift using Jensen–Shannon (JS) divergence:

$$\text{JS}\left(P_{\mathcal{B}}^{(t)}, P_{\mathcal{B}}^{(t-1)}\right) = \frac{1}{2}\text{KL}\left(P_{\mathcal{B}}^{(t)} \,\middle\|\, M\right) + \frac{1}{2}\text{KL}\left(P_{\mathcal{B}}^{(t-1)} \,\middle\|\, M\right), \tag{6}$$

where $M = \frac{1}{2}\left(P_{\mathcal{B}}^{(t)} + P_{\mathcal{B}}^{(t-1)}\right)$. We fuse the above signals into a single intent drift score:

$$\Delta_{\text{intent}} = \lambda_1 \cdot \Delta H + \lambda_2 \cdot \left(1 - JA(\mathcal{Z}^{(t)}, \mathcal{Z}^{(t-1)})\right) + \lambda_3 \cdot \text{JS}\left(P_{\mathcal{B}}^{(t)}, P_{\mathcal{B}}^{(t-1)}\right), \tag{7}$$

where $\lambda_1, \lambda_2, \lambda_3 \ge 0$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$ control the relative importance of uncertainty change, semantic overlap, and distributional drift. The LLM is triggered to update the prompt and generate a new prediction only if

$$\Delta_{\text{intent}} > \tau, \tag{8}$$

with τ being a predefined threshold.

Table 2: LLM-based Automatic Evaluation Results

Eval. Model	Target	S_{sem}	S_{logic}	S_{style}	Total
Qwen3-4B	Base	0.752	0.621	0.657	0.677
	LoRA	**0.885**	**0.792**	**0.811**	**0.829**
	LoRA+Quant	0.844	0.754	0.785	0.794
Qwen3-8B	Base	0.657	0.554	0.627	0.613
	LoRA	**0.807**	**0.755**	**0.786**	**0.783**
	LoRA+Quant	0.780	0.746	0.754	0.760
Qwen3-30B	Base	0.671	0.654	0.663	0.654
	LoRA	**0.839**	**0.797**	**0.787**	**0.808**
	LoRA+Quant	0.812	0.774	0.762	0.780

4 Experiments

4.1 Offline Evaluation

Prompt 2 is designed to evaluate and decompose generation quality into three complementary dimensions: semantic relevance (S_{sem}), logical consistency (S_{logic}), and linguistic style (S_{style}). The final score is computed as a weighted aggregation of the three sub-scores, providing a holistic assessment of generation quality.

Prompt 2: LLM-based Evaluation Prompt.

Input: Given a user behavior sequence $\mathcal{B} = \{(i_1, a_1, t_1), (i_2, a_2, t_2), \ldots, (i_n, a_n, t_n)\}$, and a candidate search query q generated for next-query prediction, please evaluate the quality of q from the following three aspects:

- **Semantic Consistency**: whether q is semantically aligned with the latent intent implied by $\mathcal{B}$.
- **Logical Coherence**: whether q reflects a reasonable intent transition given the behavior sequence.
- **Expression Quality**: whether q avoids trivial template reuse and is expressed in a natural manner.

Output: Three normalized scores $S_{\text{sem}}, S_{\text{logic}}, S_{\text{style}} \in [0, 1]$

Table 2 reports the automatic evaluation results across different evaluation model sizes and deployment settings. We use Qwen3-0.6B as the base model and compare to LoRA-adapted models[13], and quantized LoRA models [5] under the same evaluation protocol. Overall, LoRA models achieve the highest scores across most dimensions, indicating the effectiveness of lightweight fine-tuning. Notably, quantized LoRA models consistently preserve a large portion of the semantic, logical, and stylistic quality of their full-precision counterparts, with only marginal degradation in total scores. This shows that our approach remains robust under resource-constrained deployment scenarios, validating the practicality of on-device inference with minimal quality loss.

4.2 Online A/B Testing

We conducted online experiments in four scenarios of mobile Taobao during a one-month testing, covering tens of millions of users. Considering storage limitations, RecGPT-Mobile adopts Qwen3-0.6B-Quant as mobile deployment model, with hyperparameter settings

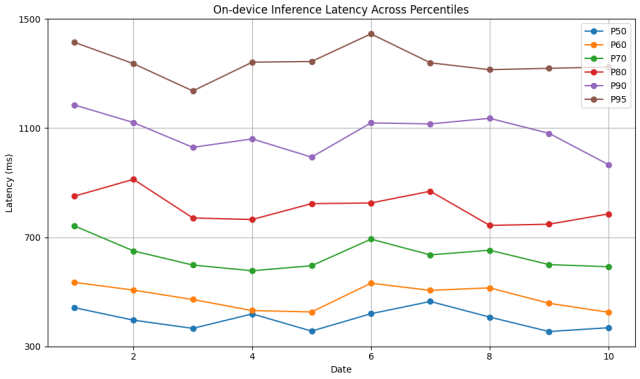


Figure 3: Running latency on real-world mobile devices under different percentile

of $\lambda_1 = 0.4, \lambda_2 = 0.3, \lambda_3 = 0.3, \tau = 0.8$, which were determined via experimental heuristic search. As shown in Table 3, RecGPT-Mobile achieves a definitely significant improvement across all four feed scenarios, contributing 1.8% CLICK, 2.7% PAY and 2.5% GMV promotions on average. Fig 3 presents the on-device inference latency across multiple percentiles on different dates. While higher percentiles exhibit increased latency as expected, the overall temporal trends remain consistent across P50 to P95. This indicates that the model maintains stable execution behavior even under amplified tail-latency conditions. Moreover, the moderate divergence among percentile curves reflects realistic device-level variance rather than systematic performance degradation, suggesting that the proposed deployment is robust to runtime fluctuations commonly encountered in real-world client environments.

Table 3: Results of online experiment.

Scenario	CLICK	PAY	GMV
Payment Success Page	+1.3%	+2.3%	+2.5%
Shipment Tracking Page	+2.4%	+2.9%	+3.0%
Shopping Cart Page	+2.5%	+2.7%	+2.9%
Order List Page	+0.8%	+1.8%	+1.8%
Average	+1.8%	+2.7%	+2.5%

5 Conclusion

This paper presents an on-device framework for next-query intent prediction from user behavior sequences, leveraging large language models with adaptive prompt construction under strict resource constraints. By dynamically adjusting prompt templates and structures, and incorporating a mobile device trigger optimization scheme, the proposed approach enables effective training and robust inference without extensive human supervision. Experimental results show that the model maintains strong intent understanding ability while preserving stable runtime performance across various tail-latency conditions, demonstrating its practicality for deploying recommendation systems on mobile devices.

Presenter Biography

Bin Zhang is an algorithm expert at Taobao, focusing on researching and applying large language models in recommendation systems that enhance user experience through a deeper understanding of user intent. During his work at Taobao, he has applied large language models to develop innovative methods for user intent understanding in recommendations.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).

[3] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[4] Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965* (2025).

[5] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems* 35 (2022), 30318–30332.

[6] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. arXiv:2210.17323 [cs.LG] https://arxiv.org/abs/2210.17323

[7] Xudong Gong, Qinlin Feng, Yuan Zhang, Jiangling Qin, Weijie Ding, Biao Li, Peng Jiang, and Kun Gai. 2022. Real-time short video recommendation on mobile devices. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 3103–3112.

[8] Yu Gong, Ziwen Jiang, Yufei Feng, Binbin Hu, Kaiqi Zhao, Qingwen Liu, and Wenwu Ou. 2020. EdgeRec: recommender system on edge in Mobile Taobao. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2477–2484.

[9] Renjie Gu, Chaoyue Niu, Yikai Yan, Fan Wu, Shaojie Tang, Rongfeng Jia, Chengfei Lyu, and Guihai Chen. 2022. On-device learning with cloud-coordinated data augmentation for extreme model personalization in recommender systems. *arXiv preprint arXiv:2201.10382* (2022).

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[11] Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both Weights and Connections for Efficient Neural Networks. arXiv:1506.02626 [cs.NE] https://arxiv.org/abs/1506.02626

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [stat.ML] https://arxiv.org/abs/1503.02531

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr* 1, 2 (2022), 3.

[14] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).

[15] Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Wei Liu, Jian Luan, Xiwen Zhang, Nicholas D Lane, and Mengwei Xu. 2025. Demystifying small language models for edge deployment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 14747–14764.

[16] Xubin Wang, Zhiqing Tang, Jianxiong Guo, Tianhui Meng, Chenhao Wang, Tian Wang, and Weijia Jia. 2025. Empowering edge intelligence: A comprehensive survey on on-device ai models. *Comput. Surveys* 57, 9 (2025), 1–39.

[17] Zhaode Wang, Jingbang Yang, Xinyu Qian, Shiwen Xing, Xiaotang Jiang, Chengfei Lv, and Shengyu Zhang. 2024. MNN-LLM: A Generic Inference Engine for Fast Large Language Model Deployment on Mobile Devices. In *MMAsia '24 Workshops*.

[18] Yunjia Xi, Weiwen Liu, Yang Wang, Ruiming Tang, Weinan Zhang, Yue Zhu, Rui Zhang, and Yong Yu. 2023. On-device integrated re-ranking with heterogeneous behavior modeling. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5225–5236.

[19] Jiajun Xu, Zhiyuan Li, Wei Chen, Qun Wang, Xin Gao, Qi Cai, and Ziyuan Ling. 2024. On-Device Language Models: A Comprehensive Review. arXiv:2409.00088 [cs.CL] https://arxiv.org/abs/2409.00088

[20] Chao Yi, Dian Chen, Gaoyang Guo, Jiakai Tang, Jian Wu, Jing Yu, Mao Zhang, Sunhao Dai, Wen Chen, Wenjun Yang, Yuning Jiang, Zhujin Gao, Bo Zheng, Chi Li, Dimin Wang, Dixuan Wang, Fan Li, Fan Zhang, Haibin Chen, Haozhuang Liu, Jialin Zhu, Jiamang Wang, Jiawei Wu, Jin Cui, Ju Huang, Kai Zhang, Kan Liu, Lang Tian, Liang Rao, Longbin Li, Lulu Zhao, Na He, Peiyang Wang, Qiqi Huang, Tao Luo, Wenbo Su, Xiaoxiao He, Xin Tong, Xu Chen, Xunke Xi, Yang Li, Yaxuan Wu, Yeqiu Yang, Yi Hu, Yinnan Song, Yuchen Li, Yujie Luo, Yujin Yuan, Yuliang Yan, Zhengyang Wang, Zhibo Xiao, Zhixin Ma, Zile Zhou, and Ziqi Zhang. 2025. RecGPT Technical Report. arXiv:2507.22879 [cs.IR] https://arxiv.org/abs/2507.22879

[21] Hongzhi Yin, Liang Qu, Tong Chen, Wei Yuan, Ruiqi Zheng, Jing Long, Xin Xia, Yuhui Shi, and Chengqi Zhang. 2025. On-device recommender systems: A comprehensive survey. *Data Science and Engineering* (2025), 1–30.

[22] Yipeng Yu. 2026. Deep Research of Deep Research: From Transformer to Agent, From AI to AI for Science. arXiv:2603.28361 [cs.AI] https://arxiv.org/abs/2603.28361

[23] Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. 2024. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152* (2024).

[24] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.