

PushDualGen: Enabling LLMs to Generate Semantic IDs with Interpretable Copy for Industrial Push Recommendation

Manjia Lin

Da Li

Yan Wang

Kuaishou Technology, Beijing, China
linmanjia@kuaishou.com

Yong Jin

Zheming Ding

Wei Yuan

Kuaishou Technology, Beijing, China
jinyong@kuaishou.com

Lei Yan

Yanan Xia

Lu Zhang

Kuaishou Technology, Beijing, China
yanlei08@kuaishou.com

Fan Yang

Kuaishou Technology, Beijing, China
yangfan@kuaishou.com

Xuanping Li

Kuaishou Technology, Beijing, China
lixuanping@kuaishou.com

Yanan Niu

Kuaishou Technology, Beijing, China
niuyan@kuaishou.com

Abstract

Push recommendation in **KuaiShou** proactively delivers personalized content to nearly one billion users to facilitate their engagement. Recently, generative recommendation has achieved end-to-end user personalization through semantic ID. However, their black-box characteristics make recommendation logics difficult to trace, hindering their deployment. OneRec-Thinking addresses this by incorporating CoT before generating SIDs, but this significantly increases inference cost. To support large-scale industrial applications, we propose **PushDualGen**, a lightweight generator, which first generates the SID and then produces a copy as a skippable explanation. PushDualGen has been deployed in Kuaishou's push recommendation system. Online A/B tests demonstrate the effectiveness of PushDualGen, delivering significant improvements in both user attraction and satisfaction. The effective play rate for videos recommended to users has relatively increased by 8.50%, while the dissatisfaction rate has relatively fallen by 37.70%. In the long term, PushDualGen optimises the content ecosystem, providing more exposure for long-tail videos.

CCS Concepts

• **Information systems** → *Recommender systems*.

Keywords

Push Notification, Generative Recommendation, Semantic ID

ACM Reference Format:

Manjia Lin, Da Li, Yan Wang, Yong Jin, Zheming Ding, Wei Yuan, Lei Yan, Yanan Xia, Lu Zhang, Fan Yang, Xuanping Li, and Yanan Niu. 2018. PushDualGen: Enabling LLMs to Generate Semantic IDs with Interpretable Copy for Industrial Push Recommendation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 6 pages. <https://doi.org/XXXXXX.XXXXXXX>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, Woodstock, NY

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/2018/06

<https://doi.org/XXXXXX.XXXXXXX>

1 Introduction

Push recommendations utilise notifications to recommend content of interest to users without requiring app entry [3, 5, 11, 19]. Push serves as a primary re-engagement channel, reaching nearly one billion users on Kuaishou. Unlike passive feed recommendations, push notifications reach users proactively. Any misaligned recommendation thus risks directly triggering user dissatisfaction, imposing stricter requirements on item selection accuracy and traceability.

Recently, the generative recommendation paradigm based on the semantic ID (SID) [12], such as OneRec [2, 20, 21], has demonstrated competitive performance across a range of applications. Unlike traditional methods, they learn user preferences in an end-to-end manner and generate the corresponding semantic IDs (SIDs) for content of interest to users. OneRec-Thinking [9] incorporates a chain-of-thought before generating the SID, enhancing performance while providing the reasons for the recommendation. This provides a promising solution for the push recommendation, enabling the platform to deliver precise and reliable recommended content for user engagement. However, considering the efficiency and the throughput requirements, it is difficult to deploy in the push recommendation.

In this work, we propose **PushDualGen**, a generative recommendation framework tailored for push notifications. It includes three stages: (1) *Semantic ID-enabled Context Compression* to encode videos into the SIDs and train LLMs to use these SIDs; (2) *Personalized SID and Copy Generation* enables LLMs to generate the SID first, and then selectively produce copy as an explanation; (3) *Representation Fusion for Online Service*. Currently, PushDualGen is deployed in the push recommendation service of Kuaishou, serving nearly a billion users with approximately 100K QPS. Online A/B testing across 150 million users demonstrated that PushDualGen delivers direct performance gains for the platform. Furthermore, PushDualGen optimises the content ecosystem, providing exposure for long-tail videos and enhancing the attraction of the Kuaishou.

2 Related Work

2.1 Push Recommendation

Push recommendation determines what content to deliver to users. Traditional methods retrieve candidates according to the atomic-level scoring. TIGER [12] proposes generative retrieval paradigm by

reframing candidate generation as an autoregressive sequence generation problem over SIDs. OneRec [2, 20, 21] scales this paradigm for large-scale industrial deployment using a unified generative model and further incorporates iterative preference alignment to improve generation quality. OneRec-Think [9] further augments SID generation with chain-of-thought; the added decoding overhead is prohibitive for latency-sensitive push scenarios [7, 16]. And RecGPT [13, 14] improves semantic coverage by augmenting online retrieval with LLM-predicted item tags as an additional retrieval signal.

2.2 Semantic ID for Recommendation System

Generative recommendation formulates the recommendation task as a sequence generation problem, where each item is represented by a SID, a structured and discrete codeword sequence obtained from continuous item embeddings via quantization for generation. Recent studies, such as the OneRec series [2, 20, 21], have demonstrated the effectiveness of SIDs in generative recommendation. Most existing methods construct SIDs with Residual Quantization (RQ), which hierarchically quantizes item embeddings to capture fine-grained semantics. However, RQ suffers from error accumulation: each level quantizes only the residual from the previous stage, making deeper SID tokens increasingly noisy and unreliable [4]. To alleviate this issue, several works have explored parallel SID generation [4, 10, 15]. Nevertheless, these methods still derive SIDs from a single embedding.

3 Method

3.1 Semantic ID-enabled Context Compression

The core of push recommendation is to infer user preferences from interaction history. However, user behavioral sequences often exceed the context window of LLMs, while heuristic strategies such as truncation or random sampling may discard signals. To make full use of the input context, we replaced the original input with SIDs inspired by the parallel embedding learned from CREM [8].

Parallel Semantic ID. Unlike residual SID methods, which suffer from hierarchical dependency and error accumulation, parallel SID approaches derive all SID tokens from a set of parallel embedding slots rather than from a single dense item representation. We construct SIDs from multiple parallel embedding slots. The details of the parallel representation model are described in Appendix A.

Specifically, each video v is represented by M embeddings $\{\mathbf{e}^{(m)}\}_{m=0}^{M-1}$. We quantize each embedding independently using K-means clustering. This yields M codebooks $\{C^{(m)}\}_{m=0}^{M-1}$, each containing K centroids. For slot m , the SID token is obtained by nearest-centroid assignment:

$$s^{(m)}(v) = \arg \min_{k \in \{0, \dots, K-1\}} \|\mathbf{e}^{(m)}(v) - \mathbf{c}_k^{(m)}\|^2, \quad (1)$$

where $\mathbf{c}_k^{(m)} \in C^{(m)}$ is the k -th centroid of the m -th codebook. The final SID obtained through quantization is defined as

$$\text{SID}(v) = [s^{(0)}(v), s^{(1)}(v), \dots, s^{(M-1)}(v)]. \quad (2)$$

By replacing each video with its SID, we compress long behavioral histories into compact, discrete semantic sequences, allowing the LLM to retain more interaction history within a limited context. In practice, we set $M = 8$ and $K = 512$. We adopt K-means due to

its simplicity and efficiency. Exploring more advanced quantization strategies is left for future work.

Scenario-Aware SID Adaptation. Since SIDs are discrete codebook indices absent from the LLM vocabulary, LLMs cannot understand or generate them without alignment. We address this in two steps. First, we register all $L \times K$ codebook indices as special tokens with random initialization. Second, we train LLMs via two complementary tasks over video v_i with push copy c_i and description d_i . We begin by utilising the text c_i and d_i to predict the corresponding SID of video v (denoted as $\langle \text{T2S} \rangle$), compressing scenario-related information into the discrete SID. Simultaneously, we designed the symmetrical task, utilising the SID of the video to generate c_i and d_i (denoted as $\langle \text{S2T} \rangle$), enabling LLMs to learn to utilise the semantic information contained in SIDs:

$$\begin{aligned} s_i^0 s_i^1 \dots s_i^{L-1} &= \mathcal{G}(c_i, d_i, \langle \text{T2S} \rangle; \theta), \\ c_i d_i &= \mathcal{G}(s_i^0, s_i^1, \dots, s_i^{L-1}, \langle \text{S2T} \rangle; \theta), \end{aligned}$$

where s_i^l denotes the SID of v_i in the l -th codebook, with $l \in \{0, \dots, L-1\}$. Both tasks are jointly optimized on Qwen3-0.6B [18] via next-token prediction:

$$\mathcal{L}_{\text{adapt}} = - \sum_{m=0}^{M-1} \log P(y^m | x, y^{<m}), \quad (3)$$

where x denotes the input and y^m is the m -th target token. To ensure training efficiency and stability, we freeze the original vocabulary embeddings and optimize only the SID token embeddings.

3.2 Personalized SID and Copy Generation

After compression, each video can be encoded as a parallel SID, and the generator gained the ability to understand and generate SIDs. Based on these foundations, we fine-tune the generator using the interaction history of user u , enabling it to generate the SID and the copy corresponding to the video of interest to the user. In addition to the interaction history, the input $\mathcal{X}(u)$ also includes a task instruction, the user profile, and temporal recency markers.

Training Objective. We train the generator to generate SIDs first, and then produce the corresponding copy. To separate them, we insert a special token $\langle \text{id2text_sep} \rangle$ into the middle. Consequently, the training objective is divided into two parts. The first is the prediction of the SID, formalised as follows:

$$\mathcal{L}_{\text{SID}} = - \sum_{l=1}^L \log P(s_{T+1}^l | \mathcal{X}(u), s_{T+1}^{<l}). \quad (4)$$

The second is the generation of copy based on the predicted SID:

$$\mathcal{L}_{\text{Copy}} = - \sum_{j=1}^{\lfloor \text{CT}_{T+1} \rfloor} \log P(t_j | \mathcal{X}(u), \text{SID}(v_{T+1}), \langle \text{id2text_sep} \rangle, t_{<j}). \quad (5)$$

The overall objective is the weighted sum of them:

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{SID}} + \lambda_{\text{Copy}} \mathcal{L}_{\text{Copy}}, \quad (6)$$

where λ_{Copy} is a hyperparameter used to balance these different training objectives.

Multi-Token Binding. Although SIDs compactly encode video, the input also contains a large amount of text, which results in a high computational cost. We address this by merging frequent

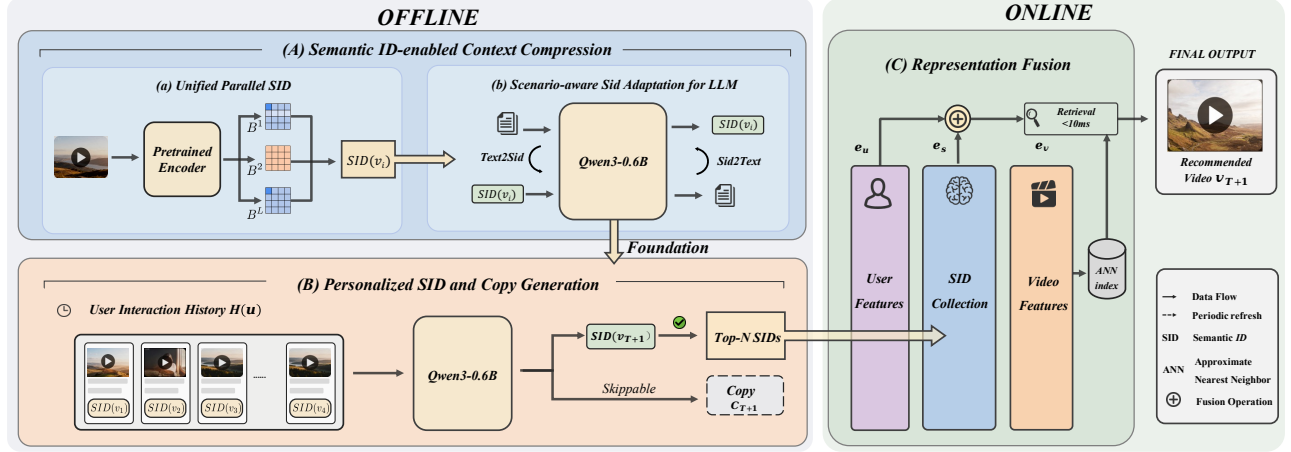


Figure 1: Overview of PushDualGen. Module (A) compresses videos into Parallel SID and aligns them with the token of LLMs via Text2SID/SID2Text fine-tuning. Module (B) generates SIDs of videos that users may be interested in and skippable copy from the interaction history. Module (C) fuses LLM-derived preference signals with user features for ANN search.

token sequences into single tokens, which are sourced from high-frequency n -grams ($n \in \{2, 3, 4\}$) and semantically atomic entities such as SID. And each new token embedding is initialized by mean pooling over its constituents.

3.3 Representation Fusion for Online Serving

We train PushDualGen based on click history. Although the click contains direct signals of interest, it can only partially reflect users' preferences; user interests are widely distributed across different situations, such as browsing and commenting. We incorporate the SIDs generated by PushDualGen into the original user feature and fuse them with heterogeneous online features to yield a more comprehensive preference representation of the user. Specifically, we use three independent encoders to project (1) user features into $\mathbf{e}_u \in \mathbb{R}^d$, (2) candidate video features into $\mathbf{e}_v \in \mathbb{R}^d$, and (3) the Top- N SIDs into $\mathbf{e}_s \in \mathbb{R}^d$, which summarizes the preference signal captured in PushDualGen. In practice, we set $N=20$. To incorporate the generative signal into the user representation, we fuse $\mathbf{e}_u$ and $\mathbf{e}_s$ via a weighted combination, formalised as follows:

$$\mathbf{e}'_u = \alpha \mathbf{e}_u + \beta \mathbf{e}_s, \quad (7)$$

where α and β are hyperparameters, which are set to 1 in applications. $\mathbf{e}'_u$ is used to search for videos of interest to users from a collection consisting of $\mathbf{e}_v$, using ANN search.

4 Experiments

4.1 Setup

Training Data. Our training data is derived from the push-click logs of Kuaishou, which are collected from nearly a billion users and span millions of candidate videos. For *SID-enabled Context Compression*, we use over 4 million videos sampled from the content pool. We filtered various types of videos to ensure content coverage and diversity. Finally, the corpus contains approximately 0.72B tokens. For *Personalized SID and Copy Generation*, we conduct incremental

training based on user click logs with approximately 3.6B tokens every day.

Implementation Details. We instantiate the three components of PushDualGen as follows. For *Semantic ID-enabled Context Compression*, we adopt Qwen2.5-Omni-3B [17] with 8 learnable compression tokens to map the representation of videos into SIDs. For *Scenario-aware SID Adaptation* and *SIDs and Copy Generation*, we employ Qwen3-0.6B [18] as the backbone generator. The two parts are trained with learning rates of 1×10^{-5} and 1×10^{-7} for 3 epochs and 1 epoch, respectively.

Evaluation. We conduct a 14-day A/B test by allocating 15% of platform traffic, covering approximately 150M users, with an equal split between PushDualGen and the online service based on a cascading pipeline. To mitigate potential pre-experiment imbalance introduced by hash-based traffic partitioning, we estimate treatment effects using CUPED [1]. We evaluate performance using four metrics. *Click PV* and *DAU* are used to measure notification engagement, and *Eff. Play Rate* (Effective Play Rate) and *Dis. Rate* (Dislike Rate) are used to evaluate post-click satisfaction in terms of content relevance and explicit negative feedback.

Table 1: Online A/B Test Results. M denotes million. Improvements significant at $p < 0.05$ are marked with *.

		Online Service	PushDualGen
Notification	Click PV $\uparrow$	82.04M	82.39M (+0.43%*)
	DAU $\uparrow$	414.89M	415.08M (+0.05%*)
Post-Click	Dis. Rate $\downarrow$	0.053%	0.033% (-37.70%*)
	Eff. Play Rate $\uparrow$	64.46%	69.94% (+8.50%*)

4.2 Online A/B Test Results

We evaluate PushDualGen in the online push recommendation system through a controlled A/B test. And the results are shown in Table 1. We can find that PushDualGen achieves statistically significant improvements across all metrics. For notification, the

gains in Click PV and DAU indicate that PushDualGen effectively attracts users back into Kuaishou and the user behaviors after clicking further confirm that the pushed content aligns with user interests. In post-click, Dis. Rate is relatively reduced by 37.70%, and Eff. Play Rate is relatively improved by 8.50%. These suggest that PushDualGen improves users' satisfaction, and reduces negative feedback.

4.3 Ablation Study

To analyse the contributions of components in PushDualGen, we use the task of SID prediction to evaluate the performance of PushDualGen and its variants. And Pass@ k is introduced to measure the probability that at least one of k independently sampled SIDs matches the ground-truth SID of a user-interested item. We construct these variants by replacing or disabling modules one by one and retraining the model. For the variant w/o Parallel SID, we replace the parallel SID with the SID used in OneRec[2]. And for the variant w/o SID Adaptation, we retrained Qwen3-0.6B to generate SIDs and text corresponding to videos directly. We also analysed the training strategies of Token Freeze and Multi-token Binding. Results are shown in Table 2. Furthermore, we show the performance of Qwen3-0.6B after training on the SID used by OneRec [2].

Table 2: Results on SID Prediction. The best and optimal results are shown by Bold and underlined, respectively.

	Pass@1	Pass@4	Pass@20
PushDualGen	0.335	0.391	0.422
w/o Token Freeze	0.333	0.386	0.412
w/o Multi-token Binding	0.332	0.384	<u>0.416</u>
w/o SID Adaptation	0.319	0.352	0.366
w/o Parallel SID	0.326	0.379	0.401
Qwen3-0.6B	0.312	0.347	0.358

We found that without the training phase of SID Adaptation, PushDualGen exhibited a significant drop in performance. This is because LLMs have never encountered SID-related data during their previous training, so an additional training process is necessary to enable them to understand and use SIDs. In contrast, the token freeze and multi-token binding primarily optimize the training process, so the performance gains are relatively slight. Compared with the results of the vanilla Qwen3-0.6B, each module we have introduced has had a positive impact on PushDualGen, and they can be combined for further performance improvements.

5 Impact on Content Ecosystem

For content platforms, sustained core competitiveness relies on the long-term health of the entire content ecosystem. In this part, we analyse how PushDualGen reshapes the content ecosystem. Based on their content, videos shared on the Kuaishou platform are automatically categorised into 39 different categories. We grouped the 39 categories into three groups: **Head** (>5%), **Torso** (1%–5%), and **Tail** (<1%) according to their share of total exposure. As shown in Figure 2, PushDualGen adjusts the exposure distribution across different groups on the platform. This allows some long-tail videos to be recommended to users on Kuaishou. From a long-term perspective, this helps the platform attract more users to share videos.

This also demonstrates that PushDualGen provides additional results from different preferences for the online service. Unlike online services that tend to favour historically popular content, the SIDs generated by PushDualGen offer better generalisation capabilities and perform better in cold-start scenarios.

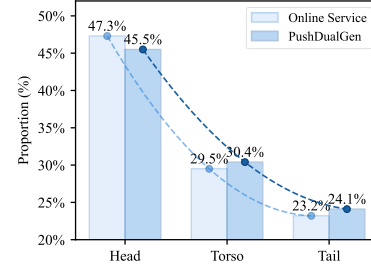


Figure 2: The proportion of exposure among different groups. PushDualGen offers fairer content exposure than the online service and is friendly to long-tail content.

6 Case Study

PushDualGen retains the ability to generate copy then SIDs. This helps us to understand what is contained within SIDs, as well as the recommendation logic behind PushDualGen. To gain a clear understanding, we conducted qualitative studies on representative examples. In addition to the recommended videos, we allow PushDualGen to generate subsequent copy. We present these results in Figure 3. It is clear that PushDualGen captures the users' preferences effectively, which is capable of adaptively taking into account user preferences manifested through a single video click, as well as across the entire click history.

History		Recommended video		Interpretable Copy	
		...			Handmade Liangpi Tutorial: No Additives, Healthier
		...			Cute Dogs Deliver Mooncakes: Adorable vibes

Figure 3: Qualitative analysis. Each row represents a user. Red indicates the link between the recommended video and the interaction history.

7 Conclusion

In this paper, we introduced PushDualGen, a lightweight generative framework for push recommendation. Unlike OneRec-Thinking, PushDualGen first generates SIDs and then produces subsequent copies as skippable interpretations. This design enhances performance while ensuring that the reasoning logic remains interpretable. Moreover, its flexible inference enables PushDualGen to be broadly deployed. In the future, we plan to experiment with increasing the size of the PushDualGen, while expanding the application of PushDualGen to scenarios beyond push recommendations.

References

- [1] Alex Deng, Ya Xu, Ron Kohavi, and Toby Walker. 2013. Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining (WSDM '13)*. Association for Computing Machinery, 123–132. <http://doi.org/10.1145/2433396.2433413>
- [2] Jiaxin Deng, Shuo Wang, Kang Cai, Long Ren, Qigen Hu, Qiang Luo, Qian Wang, Wei Ding, and Guorui Zhou. 2025. OneRec: Unifying Retrieve and Rank with Generative Recommender and Iterative Preference Alignment. *arXiv preprint arXiv:2502.18965* (2025). <http://doi.org/10.48550/arXiv.2502.18965>
- [3] Yan Gao, Viral Gupta, Jinyun Yan, Changji Shi, Zhongen Tao, PJ Xiao, Curtis Wang, Shipeng Yu, Römer Rosales, Ajith Muralidharan, and Shaunaq Chatterjee. 2018. Near Real-Time Optimization of Activity-Based Notifications. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. Association for Computing Machinery, London, United Kingdom, 283–292. <http://doi.org/10.1145/3219819.3219880>
- [4] Yupeng Hou, Jiacheng Li, Ashley Shin, Jinsung Jeon, Abhishek Santhanam, Wei Shao, Kaveh Hassani, Ning Yao, and Julian McAuley. 2025. Generating Long Semantic IDs in Parallel for Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25)*. Association for Computing Machinery, Toronto, ON, Canada, 956–966. <http://doi.org/10.1145/3711896.3736979>
- [5] Huxiao Ji, Haitao Yang, Linchuan Li, Shunyu Zhang, Cunyi Zhang, Xuanping Li, and Wenwu Ou. 2024. TIM: Temporal Interaction Model in Notification System. In *Proceedings of the 2024 International Conference on Multimedia Retrieval (ICMR '24)*. Association for Computing Machinery, Phuket, Thailand, 1120–1124. <http://doi.org/10.1145/3652583.3657614>
- [6] Da Li, Yuxiao Luo, Keping Bi, Jiafeng Guo, Wei Yuan, Biao Yang, Yan Wang, Fan Yang, Tingting Gao, and Guorui Zhou. 2026. Compressing then Matching: An Efficient Pre-training Paradigm for Multimodal Embedding. *arXiv preprint arXiv:2511.08480* (2026). <http://doi.org/10.48550/arXiv.2511.08480>
- [7] Xinyu Lin, Chaoqun Yang, Wenjie Wang, Yongqi Wang, Fuli Feng, and Tat-Seng Chua. 2025. Efficient Inference for Large Language Model-based Generative Recommendation. In *The Thirteenth International Conference on Learning Representations (ICLR 2025)*. <https://openreview.net/forum?id=VdrACDjN40>
- [8] Lihao Liu, Yan Wang, Biao Yang, Da Li, Jiangxia Cao, Yuxiao Luo, Xiang Chen, Xiangyu Wu, Wei Yuan, Fan Yang, Guiguang Ding, Tingting Gao, and Guorui Zhou. 2026. CREM: Compression-Driven Representation Enhancement for Multimodal Retrieval and Comprehension. *arXiv preprint arXiv:2602.19091* (2026). <http://doi.org/10.48550/arXiv.2602.19091>
- [9] Zhanyu Liu, Jiaxin Deng, Junda She, Kuo Cai, Lejian Ren, Qigen Hu, Qiang Luo, Shiyao Wang, Xinchun Luo, Xingmei Wang, Yi Su, Yunfan Wu, Zixing Zhang, and Guorui Zhou. 2025. OneRec-Think: In-Text Reasoning for Generative Recommendation. *arXiv preprint arXiv:2510.11639* (2025). <http://doi.org/10.48550/arXiv.2510.11639>
- [10] Zhao Liu, Yichen Zhu, Yiqing Yang, Guoping Tang, Rui Huang, Qiang Luo, Xiao Lv, Ruiming Tang, Kun Gai, and Guorui Zhou. 2025. DiffGRM: Diffusion-based Generative Recommendation Model. *arXiv preprint arXiv:2510.21805* (2025). <http://doi.org/10.48550/arXiv.2510.21805>
- [11] Conor O'Brien, Huasen Wu, Shaodan Zhai, Dalin Guo, Wenzhe Shi, and Jonathan J. Hunt. 2022. Should I Send This Notification? Optimizing Push Notifications Decision Making by Modeling the Future. *arXiv preprint arXiv:2202.08812* (2022). <http://doi.org/10.48550/arXiv.2202.08812>
- [12] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. Curran Associates, Inc., 10299–10315. https://proceedings.neurips.cc/paper_files/paper/2023/hash/20dcab0f14046a5c6b02b61da9f13229-Abstract-Conference.html
- [13] RecGPT Team. 2025. RecGPT Technical Report. *arXiv preprint arXiv:2507.22879* (2025). <http://doi.org/10.48550/arXiv.2507.22879>
- [14] RecGPT Team. 2025. RecGPT-V2 Technical Report. *arXiv preprint arXiv:2512.14503* (2025). <http://doi.org/10.48550/arXiv.2512.14503>
- [15] Teng Shi, Chenglei Shen, Weijie Yu, Shen Nie, Chongxuan Li, Xiao Zhang, Ming He, Yan Han, and Jun Xu. 2025. LLaDA-Rec: Discrete Diffusion for Parallel Semantic ID Generation in Generative Recommendation. *arXiv preprint arXiv:2511.06254* (2025). <http://doi.org/10.48550/arXiv.2511.06254>
- [16] Chengbing Wang, Yang Zhang, Zhicheng Wang, Tianhao Shi, Keqin Bao, Fuli Feng, and Tat-Seng Chua. 2025. Decoding in Latent Spaces for Efficient Inference in LLM-based Recommendation. *arXiv preprint arXiv:2509.11524* (2025). <http://doi.org/10.48550/arXiv.2509.11524>
- [17] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. 2025. Qwen2.5-Omni Technical Report. *arXiv:2503.20215 [cs.CL]* <https://arxiv.org/abs/2503.20215>
- [18] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 Technical Report. *arXiv:2505.09388 [cs.CL]* <https://arxiv.org/abs/2505.09388>
- [19] Bo Zhao, Koichiro Narita, Burkay Orten, and John Egan. 2018. Notification Volume Control and Optimization System at Pinterest. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18)*. Association for Computing Machinery, London, United Kingdom, 1012–1020. <http://doi.org/10.1145/3219819.3219906>
- [20] Guorui Zhou, Honghui Bao, Jiaming Huang, Jiaxin Deng, Jinghao Zhang, Junda She, Kuo Cai, Lejian Ren, Lu Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rongzhou Zhang, Ruiming Tang, Shiyao Wang, Wuchao Li, Xiangyu Wu, Xinchun Luo, Xingmei Wang, Yifei Hu, Yunfan Wu, Zhanyu Liu, Zhiyang Zhang, Zixing Zhang, Bo Chen, Bin Wen, Chaoyi Ma, Chengru Song, Chenglong Chu, Defu Lian, Fan Yang, Feng Jiang, Hongtao Cheng, Huanjie Wang, Kun Gai, Pengfei Zheng, Qiang Wang, Rui Huang, Siyang Mao, Tingting Gao, Wei Yuan, Yan Wang, Yang Zhou, Yi Su, Zexuan Cheng, Zhixin Ling, and Ziming Li. 2026. OpenOneRec Technical Report. *arXiv preprint arXiv:2512.24762* (2026). <http://doi.org/10.48550/arXiv.2512.24762>
- [21] Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, Pengfei Zheng, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Ruiming Tang, Shiyao Wang, Shujie Yang, Tao Wu, Wuchao Li, Xinchun Luo, Xingmei Wang, Yi Su, Yunfan Wu, Zexuan Cheng, Zhanyu Liu, Zixing Zhang, Bin Zhang, Boxuan Wang, Chaoyi Ma, Chengru Song, Chenhui Wang, Chenglong Chu, Di Wang, Dongxue Meng, Dunju Zang, Fan Yang, Fangyu Zhang, Feng Jiang, Fuxing Zhang, Gang Wang, Guowang Zhang, Han Li, Honghui Bao, Hongyang Cao, Jiaming Huang, Jiapeng Chen, Jiaqiang Liu, Jinghui Jia, Kun Gai, Lantao Hu, Liang Zeng, Qiang Wang, Qidong Zhou, Rongzhou Zhang, Shengzhe Wang, Shihui He, Shuang Yang, Siyang Mao, Sui Huang, Tiantian He, Tingting Gao, Wei Yuan, Xiao Liang, Xiaoxiao Xu, Xugang Liu, Yan Wang, Yang Zhou, Yi Wang, Yiwu Liu, Yue Song, Yufei Zhang, Yunfeng Zhao, Zhixin Ling, and Ziming Li. 2025. OneRec-V2 Technical Report. *arXiv preprint arXiv:2508.20900* (2025). <http://doi.org/10.48550/arXiv.2508.20900>

A Details of the Parallel Embedding Model and Semantic ID

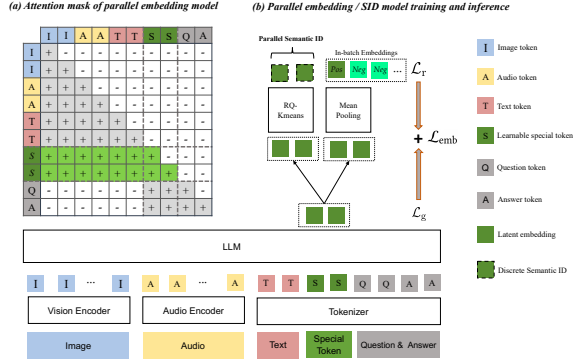


Figure 4: Parallel embedding model and SID construction. (a) Attention mask of the parallel embedding / SID model, where “+” indicates visible tokens and “-” indicates masked ones. (b) Training and inference of the parallel embedding / SID model.

This section provides supplementary details of the parallel embedding model used in the main text. We build the embedding model on Qwen2.5-Omni [17], following the collaborative representation learning paradigm of CREM [8] and CoMa [6]. The model is

trained to support both video retrieval and copy generation with a shared set of embeddings. To obtain compact latent representations from multimodal inputs, we prepend a set of learnable compression tokens to the original input sequence. In our implementation, we use 8 compression tokens, so that each video is encoded into 8 parallel embeddings.

As illustrated in Figure 4(b), the encoder is optimized with a hybrid objective:

$$\mathcal{L}_r = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(\text{sim}(\mathbf{q}_i, \mathbf{p}_i)/\tau)}{\sum_{j=1}^B \exp(\text{sim}(\mathbf{q}_i, \mathbf{p}_j)/\tau)}, \quad (8)$$

$$\mathcal{L}_g = -\frac{1}{B} \sum_{i=1}^B \sum_{t=1}^{T_i} \log P(x_{i,t} | x_{i,<t}), \quad (9)$$

$$\mathcal{L}_{\text{emb}} = \lambda \mathcal{L}_r + (1 - \lambda) \mathcal{L}_g. \quad (10)$$

Here, $\mathcal{L}_r$ is an in-batch InfoNCE retrieval loss and $\mathcal{L}_g$ is the standard autoregressive language modeling loss. B denotes the batch size, $\mathbf{q}_i$ and $\mathbf{p}_i$ denote a matched query–passage pair, $\text{sim}(\cdot, \cdot)$ is the similarity function, τ is a temperature hyperparameter, and $\lambda \in [0, 1]$ balances the two objectives. The corresponding attention mask is shown in Figure 4(a).

Semantic ID construction. The resulting parallel embeddings are discretized into SIDs following the procedure described in Sec. 3.1. In practice, we use $M = 8$ embedding slots and $K = 512$ centroids for each slot-specific codebook. We adopt K-means for its simplicity and scalability.