TreeBridge: Aligning LLM Embeddings in Industrial Recommender Systems

Yabo Ni[1], Cao Yuanpeng[2], Wenhang Zhou[2], Bangyang Hong[2], Zhongyi Zhang[2], Enlei Cai[2], Kangle Wu[2], Anxiang Zeng[1], Han Yu[1], Xiaoxiao Li[3,4]

[1]College of Computing and Data Science, Nanyang Technological University, Singapore
[2]Shopee Pte Ltd., Singapore
[3]Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada
[4]Vector Institute, Canada
yabo001@e.ntu.edu.sg, {yuanpeng.cao, wenhang.zhou, bangyang.hong, zhangzhongyi, enlei.cai, kangle.wu}@shopee.com, {zeng0118, han.yu}@ntu.edu.sg, xiaoxiao.li@ece.ubc.ca

Abstract

Large language models (LLMs) have shown great potential in enhancing search and recommender systems by providing rich semantic representations from unstructured texts. However, directly integrating LLM embeddings into industrial recommendation pipelines often results in subpar performance due to the semantic and distributional mismatch between pre-trained LLM features and domain-specific, feedback-driven representations. Existing approaches struggle to effectively align LLM embeddings with recommendation objectives, often facing challenges such as label misalignment or the potential loss of semantic diversity during fine-tuning. In this work, we present TreeBridge, a novel framework that introduces a structure-aware generative encoding tree to bridge the semantic gap between LLM embeddings and recommendation tasks. It preserves the external semantic richness of LLM embeddings, while learning label-informed structures that capture user preferences and interaction patterns. This enables the generation of task-adaptive representations without compromising embedding diversity. We further adopt an online-offline hybrid service paradigm to ensure low-latency real-world deployment. TreeBridge has been deployed on the Shopee e-commerce platform, one of the largest online shopping platforms in Southeast Asia serving hundreds of millions of users. Since its deployment in May 2025, it has helped the company achieve a commercially significant 1.55% relative improvement in gross merchandise volume (GMV). The deployment experience demonstrates the effectiveness, scalability, and significant commercial value of TreeBridge.

Introduction

In recent years, LLMs have demonstrated remarkable capabilities in understanding and generating natural language (Devlin et al. 2019; Brown et al. 2020; Touvron et al. 2023), making them attractive sources of semantic representations for downstream tasks. In the domain of search and recommender systems, LLM-generated embeddings offer a promising avenue for enriching user and item representations with external knowledge and deep contextual understanding (Huang et al. 2024; Lin et al. 2023; Zhu et al. 2023). However, directly integrating LLM embeddings into search and recommendation pipelines often leads to suboptimal results. This is primarily due to the semantic and distributional mismatch between pre-trained LLM representations and the highly domain-specific, feedback-driven representations used in practice. Current approaches face three key limitations:

1. Directly using LLM embeddings as frozen features preserves their external semantic knowledge and offers high representation diversity. However, these embeddings are not aligned with the recommendation task's supervised objectives, such as clicks or conversions. As a result, although diverse, they often fail to contribute meaningful predictive signals within the recommendation context.
2. Supervised training on top of LLM embeddings without updating them introduces label misalignment. The embeddings remain in their original semantic space, while the downstream model attempts to map them to task-specific outputs. This misalignment can limit the model capability to fully exploit the semantic content of the embeddings, resulting in limited or no gain over traditional representations.
3. Adapter-based methods aim to bridge this gap by learning lightweight transformation layers to project LLM embeddings into the recommendation label space. While this enhances label alignment and enables end-to-end training, it often sacrifices the inherent diversity and semantic richness of the LLM embeddings, effectively collapsing them into the same latent space as standard recommendation features.

To address these limitations and facilitate the deployment of LLMs in search and recommender systems, we propose a novel framework that introduces a structure-aware generative encoding tree as an intermediate representation. Instead of modifying or fine-tuning the LLM embeddings directly, we construct a tree-based structure that is generated from the LLM embedding space and trained with recommendation-specific labels. This tree structure serves as a semantic bridge that preserves the semantic independence and diversity of LLM embeddings. This design integrates rich, general semantic signals from LLMs into highly customized recommendation pipelines. Empirical evaluation on real-world recommendation benchmarks shows that it sig-

nificantly outperforms existing embedding fusion methods, particularly in cold-start and sparse data scenarios.

To support real-time service provision in practical e-commerce applications, we further design an online-offline hybrid service paradigm to update the model without disrupting online services. TreeBridge has been deployed from May 2025 on the Shopee e-commerce platform, one of the largest online shopping platforms in Singapore serving hundreds of millions of users across Southeast Asia with large-scale, real-time search and recommendation traffic. So far, it has achieved 1.55% higher in gross merchandise volume (GMV) compared to its predecessor, which is very significant in real-world e-commerce applications.

Application Description

TreeBridge has been incorporated into the personalization recall phase of the Shopee search system. Effective personalized recall directly determines whether the retrieved candidates align with individual user interests, significantly impacting downstream ranking quality and overall user experience. In addition, the recall stage typically operates on an extremely large item pool, often involving millions of items. Given this scale, recall models must balance personalization effectiveness with computational efficiency, as overly complex models are not feasible for real-time retrieval at such large volumes.

Achieving high-quality personalized recall depends heavily on user and item representation learning. Classical representation learning methods primarily rely on statistical features (e.g., co-click rates, purchase frequencies) and handcrafted signals derived from historical interactions (Cheng et al. 2016; Chen et al. 2023; Huang et al. 2013; Li et al. 2022; Ni et al. 2025). User representations are typically aggregated from past behaviors, such as click and purchase histories (Ni et al. 2018; Chang et al. 2023; Cao et al. 2022; Zhou et al. 2019). However, these methods rely heavily on historical data, limiting their effectiveness in cold-start and sparse scenarios. They also lack the ability to fully exploit the rich semantic signals embedded in product titles, descriptions, and user queries, especially for new or low-frequency items.

Recent studies have shown that LLMs excel at extracting semantic information from unstructured text, providing valuable insights beyond what traditional recall features can capture. In the e-commerce context, product descriptions and reviews contain latent information that can greatly enrich user and item embeddings. Existing works (Lin et al. 2023; Zhu et al. 2023) have begun to explore LLM applications in search and recommendation for representation learning.

Some methods directly use LLM embeddings as frozen features to preserve their external semantic knowledge and maintain high representation diversity. For instance, TCF (Li et al. 2023b) leverages GPT-3 (Brown et al. 2020) to generate textual representations of items for text-based collaborative filtering in recommendation tasks. However, these embeddings are not naturally aligned with the supervised objectives of recommendation tasks, such as clicks or conversions. This often results in limited predictive power despite the semantic richness of the features. Adapter-based approaches (Yang et al. 2024; Liu et al. 2025; Song et al. 2024; Jia et al. 2024; Qiu et al. 2024) attempt to mitigate this misalignment by introducing lightweight transformation layers to project LLM embeddings into the recommendation label space, enabling end-to-end training. For example, ILM (Yang et al. 2024) introduces a framework that leverages a Q-Former-based (Li et al. 2023a) item encoder to align collaborative filtering embeddings with language representations, which are then integrated into a frozen LLM to enable conversational recommendation with interleaved item and text inputs.

Although prior work has demonstrated the potential of LLMs in enhancing search and recommender systems, effectively integrating LLM embeddings into personalized recall pipelines in large-scale real-world application remains underexplored. Existing solutions either fail to align with user feedback, struggle with semantic-label mismatch, or compromise the unique advantages of LLM-derived representations. To address these challenges, we propose TreeBridge, which augments traditional statistical feature-driven recall with LLM-based embeddings while preserving their semantic diversity and explicitly aligning them with recommendation labels through a structure-aware generative encoding mechanism. By constructing structure-aware generative encodings based on LLM embeddings, our method enables personalized, semantically rich, and behavior-aligned recall candidate generation that significantly improves the recall results. In the next section, we describe the architecture of the TreeBridge-based AI Engine and its key components in detail.

Use of AI Technology

Overall Architecture

The architecture of TreeBridge is illustrated in Figure 1, consisting of three key components. On the left side is the embedding generation module, which produces two distinct types of embeddings: 1) LLM-based embeddings, which capture rich external semantic information from large-scale pre-trained language models; and 2) feedback-based embeddings, which are learned from recommendation-specific user feedback signals such as clicks and transactions. The top-right part of the framework is the tree building module, where hierarchical tree structures are constructed based on the relationships between the embeddings generated in the first stage. The third part of the framework is the generative prediction module, which focuses on the recall task. In this stage, user features are used as model input, and a transformer-based architecture is employed to extract meaningful user representations.

It is important to note that both the embedding generation and tree building modules operate offline, while only the generative prediction module needs to support online inference to meet real-time recommendation requirements.

Embedding Generation

We first elaborate the embedding generation module, which plays a foundational role in representing items from hetero-

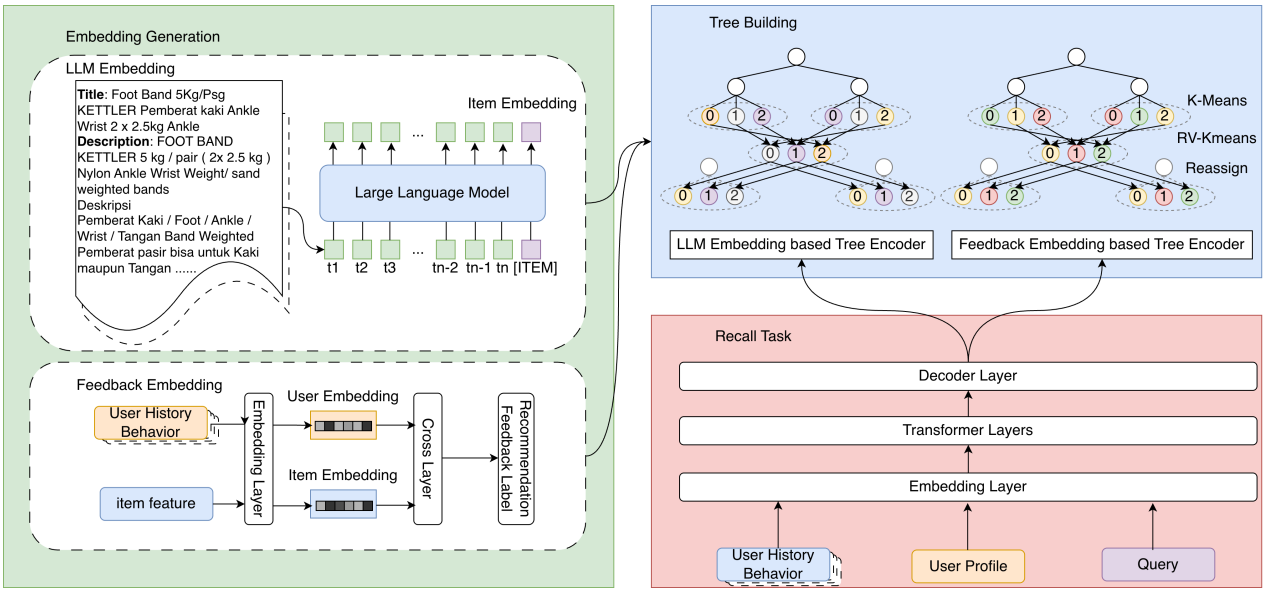

Figure 1: The overall TreeBridge framework.

geneous perspectives and serves as the basis for the subsequent tree construction and generative modeling. The quality and diversity of these embeddings are crucial for building expressive and semantically meaningful tree structures.

LLM-based embeddings For each item $i \in \mathcal{I}$, we concatenate its textual attributes (title, tags, description) into a single sequence T_i and prepend a domain-agnostic prompt. A special token `[ITEM]` is appended to the sequence, producing the token list $\left[t_1, t_2, \ldots, t_m, \texttt{[ITEM]}\right]$, where m represents the length of text tokens. This design follows the item-level input construction strategy in HLLM (Chen et al. 2024), enabling the model to generate meaningful representations based on rich item metadata. The Item LLM processes this sequence and the final-layer hidden state corresponding to `[ITEM]` is extracted as the item embedding. This design compresses heterogeneous textual information into a dense vector while preserving semantic nuances essential for downstream retrieval. Keywords are treated analogously to items and share the same Item LLM backbone. Sharing the Item LLM ensures semantic alignment between keyword and item representations while avoiding additional parameter overhead.

Feedback-based Embeddings The feedback-based embeddings are designed to incorporate collaborative filtering(CF) information into the item representations by capturing user preferences from observed interaction signals. These embeddings are learned using a classic Deep Structured Semantic Model (DSSM) framework (Huang et al. 2013), a widely adopted approach for modeling user-item interactions in recommender systems.

By leveraging this supervised signal from explicit user feedback, the DSSM-based embedding captures collaborative filtering patterns and user preferences grounded in real behavioral data. This contrasts with LLM embeddings, which primarily encode external semantic knowledge and contextual information.

Tree Encoder Generation

To convert dense embeddings into compact product codes, we propose a hybrid approach that combines hierarchical balanced clustering with Residual-Vector Clustering. The method simultaneously guarantees near-uniform leaf-node sizes, which is critical for low-latency online retrieval, and preserves semantic coherence within each code.

The Tree Encoder Generation process consists of three key stages: Balanced k-means, RV-Kmeans Refinement, and Code Reassignment, each of which will be described in detail below.

1. **Balanced k-means.** We first perform balanced k-means producing a tree whose leaf nodes contain at most τ items each. This step strictly controls capacity, preventing any leaf from becoming too large for efficient beam search. From top to bottom, each node is partitioned into k clusters $C_1, \ldots, C_k$, each with up to τ items. If the number of items is not divisible by τ, empty placeholders are inserted to maintain balance.
2. **RV-Kmeans Refinement.** For an item i with vector representation V_i, let $C_{i,l}$ denote its clustering assignment at layer l of balanced k-means, where $l = 0, \ldots, d_{\max} - 1$ and $d_{\max}$ is the total number of codebook layers. The residual vector at layer l is defined as $RV_{i,l} = V_i - \text{Centroid}(C_{i,l-1})$. Unlike balanced k-means, which recursively re-clusters items within each cluster from the previous layer, our approach performs clustering at each layer using the residual vectors $RV_{i,l}$ of all items, thereby generating a new codebook at every stage.
3. **Code Reassignment.** Finally, for each code in the codebook from stage 1, we map it to the nearest code in the

corresponding layer of the codebook produced in stage 2. This reassignment preserves the clustering effectiveness of balanced k-means while enhancing the semantic consistency of codes within each layer of the codebook.

Multi-Tree Multi-Task Generation Structure

To further leverage the semantic diversity of LLM embeddings and incorporate task-specific feedback, we propose a Multi-Tree Multi-Task Generation Structure. In this design, multiple trees are independently constructed for different types of item embeddings, allowing the model to preserve semantic distinctions across embeddings and build structure-aware representations for each. At the same time, all trees share a unified set of user features, and the generative prediction is trained with real feedback signals for different tree encoders. This enables the integration of user interaction patterns into the retrieval stage while maintaining the expressiveness of the original embeddings.

Specifically, we construct two complementary trees, each capturing a distinct aspect of item representation. One tree is built from embeddings derived from user behavior modeling, effectively encoding collaborative filtering signals that reflect users' historical interactions. The other tree is constructed using embeddings generated by LLMs, which encapsulate richer semantic meaning and general world knowledge. By maintaining separate trees for different embedding spaces, we preserve the external semantic expressiveness of LLMs while enabling structured modeling tailored to specific signal types.

To generate codes from these trees, we formulate code prediction as a multi-task generative learning problem. Instead of predicting a single code per item, the model learns to predict codes corresponding to multiple semantic spaces, each aligned with a different tree. This design allows the model to integrate heterogeneous signals and adapt to multiple recall objectives simultaneously. The overall loss function is a weighted sum of individual losses for each task, defined as follows:

$$\begin{cases} \mathcal{L}_{\text{LLM}} = -\sum_{l=1}^{L} \log P\Big(s_i^{(l)} \mid \text{s}_{\text{[Keyword]}}, s_i^{[1:l]}, z_u\Big) \\ \mathcal{L}_{\text{CF}} = -\sum_{l=1}^{L} \log P\Big(s_i^{(l)} \mid \text{s}_{\text{[Keyword]}}, s_i^{[1:l]}, z_u\Big) \\ \mathcal{L} = \lambda_1 \mathcal{L}_{\text{LLM}} + \lambda_2 \mathcal{L}_{\text{CF}} \end{cases} \tag{1}$$

Where $s_i^{(l)}$ indicate the tree node at the l-th layer and $s_{[Keyword]}$ denotes the start token, we use the keyword embedding as start token to adapt into search task. And z_u indicate the user information. $\mathcal{L}_{\text{LLM}}$ denotes the loss for predicting codes from the LLM-based tree, and $\mathcal{L}_{\text{CF}}$ is the loss for code prediction based on the CF-based tree. The weights λ_1 and λ_2 are hyperparameters used to balance the contributions of each task. This joint optimization encourages the shared layers of the generative model to learn robust and generalized representations that integrate information from diverse domains and explicitly align with user queries and preferences, resulting in recall outcomes that better meet user needs in search applications.

Application Development and Deployment

Here, we first describe the offline experiment setup, including datasets, evaluation metrics, and implementation details. Then, we conduct a series of experiments to answer the following research questions in order to support the deployment decision of TreeBridge by Shopee management:

- **RQ1:** Can the proposed structure-aware generative encoding tree effectively bridge the semantic gap between general-purpose LLM embeddings and task-specific recommendation objectives?
- **RQ2:** How does TreeBridge perform compared to state-of-the-art recall-oriented recommendation methods, including those based on LLM embedding integration and adapter tuning?
- **RQ3:** Does TreeBridge provide significant advantages in cold-start scenarios where user or item interactions are limited?

Experiment Settings

Datasets The raw data used in our experiments was collected from Shopee, a large-scale e-Commerce platform, and includes 3 months of online trading records. Each record in the dataset is a user behavior log that includes item information , user information, keyword information, and interaction information. In our recall task, the click and purchase behavior are considered as the label, the remaining attributes are used as features.

Evaluation Metrics We adopt *HitRate@K* as the primary evaluation metrics to assess the recall quality of our method. This metric reflects the proportion of users for whom at least one relevant item appears within the top-K results, which aligns with the candidate generation objective in our production recommender system. Specifically, we report results for $K = 50$ and $K = 100$, denoted as *HitRate@50* and *HitRate@100*, to match the candidate list size typically used in real-world deployment. The formal definition of HitRate@K is as follows:

$$\text{HitRate@K} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} I\left[\min_{r \in \mathcal{R}_q} \text{rank}_r \leq K\right], \tag{2}$$

where $\mathcal{Q}$ denotes the set of users, $\mathcal{R}_q$ is the set of relevant items for user q, and $I[\cdot]$ is the indicator function that equals 1 if at least one relevant item is ranked within the top-K positions, and 0 otherwise.

Implementation Details For each type of item embedding, we constructed a four-level codebook with capacities of 300, 100, 50, and 10 for each respective layer, where each leaf node contains 1 to 10 items. The LLM embedding is based on a 1B-parameter HLLM model that has been fine-tuned on real production environment datasets. The generative prediction model employs a single-layer Transformer with 128 dimensions and 8 heads to extract features from user behavioral sequences, while the decoder component utilizes a single-layer Transformer with 128 dimensions and 8 heads for code generation.

Comparison Baselines

To comprehensively evaluate the effectiveness of our proposed method, we compare it against a diverse set of strong baselines from three major categories:

- **Traditional recall-based methods:** These methods rely on supervised learning over manually designed features and collaborative signals. Representative models include:
  - **CDSSM** (Shen et al. 2014): Classic dual-tower semantic model.
  - **MGDSPR** (Li et al. 2021): Multi-granularity sequential model for personalized retrieval.
- **Tree-based recall algorithms:** These approaches structure the item space using hierarchical trees to improve retrieval efficiency and scalability. We include:
  - **RecForest** (Feng et al. 2022): A hierarchical retrieval framework that builds and learns multiple item trees in parallel.
- **LLM embedding-based methods:** These methods aim to utilize pre-trained large language model representations for recommendation, either through direct integration or task-specific adaptation. We consider:
  - **LLMEmb** (Liu et al. 2024): Directly uses frozen LLM embeddings.
  - **EASE** (Qiu et al. 2024): Linear fine-tuning aligning LLM embeddings with recommendation signals.
  - **PRECISE** (Song et al. 2024): Adapter-based model bridging LLM knowledge with supervised objectives.

RQ1: Bridging the Semantic Gap between LLM Embeddings and Recommendation Objectives

To investigate whether the proposed structure-aware generative encoding tree effectively bridges the semantic gap between general-purpose LLM embeddings and task-specific recommendation objectives, we design a set of ablation experiments focusing on different ways of utilizing LLM embeddings.

Specifically, we compare the following variants:

- **LLM-only**: Directly uses frozen LLM embeddings as input features for recall, without any structural transformation or alignment. This variant retains the raw semantic space of the LLM but lacks adaptation to recommendation-specific objectives.
- **CF Tree-only**: Constructs a recall tree based solely on traditional collaborative filtering (CF) embeddings derived from user-item interaction feedback. This variant serves as a strong behavior-based baseline, without incorporating any LLM semantics.
- **LLM Tree-only**: Builds a recall tree exclusively using LLM embeddings, applying tree-based encoding to structure the semantic space. This variant examines whether tree transformation alone can effectively align LLM features with recommendation objectives, without reliance on CF-based embeddings.

Method	Click		Order	
	H@50	H@100	H@50	H@100
LLM-only	0.1943	0.2565	0.2264	0.2725
CF Tree-only	0.2644	0.2922	0.2615	0.2853
LLM Tree-only	<u>0.3250</u>	<u>0.3605</u>	<u>0.3200</u>	<u>0.3615</u>
TreeBridge	**0.3346**	**0.4061**	**0.3567**	**0.4306**

Table 1: Ablation study results on Shopee.

Table 1 presents the performance comparison of the evaluated variants. We observe the following:

Directly using LLM embeddings alone (LLM-only) results in limited recommendation effectiveness, with Click and Order Hitrate@100 scores at 0.2565 and 0.2725 respectively, reflecting a notable gap between raw semantic representations and user feedback signals. Incorporating LLM embeddings into the collaborative filtering-based tree model (comparing the final TreeBridge to Recforest CF Tree-only) substantially enhances performance, with Click Hitrate@100 improving by 39.1% and Order Hitrate@100 by 51.0%. This highlights the crucial role of semantic information in complementing behavior-based embeddings. Applying tree-based encoding directly on LLM embeddings (Recforest LLM Tree-only) already achieves significant gains over the purely ID-based tree variant, increasing Click and Order Hitrate@100 by 23.4% and 26.7% respectively. This indicates that structured transformation effectively aligns semantic embeddings with recommendation objectives, enriching user-item representations. Our full model TreeBridge, which integrates multiple encoding trees and leverages a multi-task learning framework, achieves the best overall results. It further improves over the LLM tree-only variant by 12.6% (Click) and 19.1% (Order) in Hitrate@100, demonstrating that combining semantic and behavioral signals through generative encoding trees effectively bridges the semantic gap.

RQ2: Comparison with State-of-the-Art Recall Methods

To evaluate the effectiveness of our proposed approach in leveraging LLM embeddings for recommendation recall, we compare against a set of state-of-the-art baselines that also incorporate LLM-based representations, as shown in table **??**.

Among traditional recall models, CDSSM (Shen et al. 2014) and MGDSPR (Li et al. 2021) depend on manually crafted features and behavioral interaction modeling. MGDSPR benefits from sequential modeling and outperforms CDSSM, but both remain limited in capturing semantic signals, yielding Hitrate@100 values below 0.32 on both click and order tasks.

The LLM embedding-based group includes LLMEmb (Liu et al. 2024), EASE (Qiu et al. 2024), and PRECISE (Song et al. 2024). LLMEmb serves as a simple baseline by directly utilizing frozen LLM embeddings, while EASE introduces lightweight linear adaptation to

Method	Click		Order	
	H@50	H@100	H@50	H@100
CDSSM	0.1762	0.2434	0.1986	0.2773
MGDSPR	0.2039	0.2653	0.2426	0.3119
LLMEmb	0.1943	0.2565	0.2264	0.2725
EASE	0.2319	0.3043	0.2713	0.3514
PRECISE	0.2441	0.3233	0.2901	0.3754
RecForest	0.2644	0.2922	0.2615	0.2853
TreeBridge	**0.3346**	**0.4061**	**0.3567**	**0.4306**

Table 2: Performance Comparison of Baseline Methods on Shopee Dataset.

better align embeddings with recommendation tasks. PRECISE further enhances this line of work with adapter-based fine-tuning, resulting in the best performance within this group. On Shopee, it achieves a Hitrate@100 of 0.3754 on the order task, outperforming all non-LLM methods.

RecForest represents the tree-based approaches, using learned hierarchical item structures to improve retrieval. Despite not utilizing language models, RecForest shows competitive results, especially on the click prediction task, thanks to its efficient space partitioning and representation learning.

Our method consistently outperforms all baselines across different metrics. It achieves the highest Hitrate@100 values of 0.4061 for clicks and 0.4306 for orders, with relative improvements over the best-performing baseline (PRECISE) reaching approximately 8.2% and 14.7% respectively. These results demonstrate the effectiveness of our dual-space design, which integrates LLM semantics with collaborative signals through a generative tree-based retrieval framework. The improvements confirm the benefits of joint modeling and alignment in capturing both intent-level semantics and user behavior patterns for large-scale recommendation.

RQ3: Effectiveness of TreeBridge in Cold-Start Scenarios

In this section, we evaluate the effectiveness of TreeBridge in cold-start scenarios, focusing on new item recommendation on the Shopee dataset. New items typically lack historical interaction data, which limits the effectiveness of traditional collaborative filtering methods.

As shown in Table **??**, directly using frozen LLM embeddings (LLMEmb) yields only modest improvements over MGDSPR, and still underperforms traditional deep semantic models such as CDSSM in order hitrate, suggesting that raw semantic embeddings alone cannot fully capture user purchase preferences for unseen items. In contrast, approaches that combine LLM embeddings with structured encoding and task adaptation, such as PRECISE and our proposed TreeBridge, achieve notable gains. TreeBridge delivers the highest performance, reaching an order Hitrate@50 of 0.1402 and Hitrate@100 of 0.1502, representing a relative improvement of 29.8% over the strongest baseline (PRECISE) in Hitrate@50.

Method	H@50	H@100
RecForest	0.0427	0.0438
CDSSM	0.0930	0.1345
MGDSPR	0.0703	0.1042
LLMEmb	0.0817	0.1081
EASE	0.0945	0.1166
PRECISE	0.1080	0.1407
TreeBridge	**0.1402**	**0.1502**

Table 3: Performance Comparison of Baseline Methods on New Item Recommendation

These results demonstrate that while LLM embeddings provide rich semantic information, their potential for cold-start recommendation is only fully realized when integrated with task-aligned encoding and multi-task learning. By aligning semantic representations with ranking objectives, TreeBridge effectively mitigates cold-start challenges for new item recommendation.

Application Use and Payoff

TreeBridge has been deployed in Shopee e-commerce platform since May 2025 to support the Shopee search personalization recall service. We report the online A/B test results of TreeBridge and the previously adopted approach since the deployment of TreeBridge in May 2025. We use the GMV as the deployment evaluation metrics. The previous approach is a RecForest (Feng et al. 2022) model with behavior features, similar in structure to TreeBridge but lacking LLM-generated features and multi-tree structure. The results indicate that TreeBridge achieved a 1.55% improvement in GMV compared to the previous approach on average, demonstrating significant positive business impact.

Conclusions

In this work, we present the design and deployment experience of TreeBridge, a novel framework designed to bridge the semantic gap between general-purpose LLM embeddings and task-specific recommendation objectives through a structure-aware generative encoding tree. It preserves the semantic richness of LLMs, while aligning with user behavior signals, thereby enabling more effective personalization in the recall phase of large-scale search systems. For practical deployment, we adopt an online-offline hybrid service paradigm that integrates LLM-derived features without compromising system latency or scalability. We conduct extensive offline experiments on both large-scale e-commerce datasets and public benchmarks to verify the effectiveness of our method and provide strong evidence to convince the senior management of Shopee to adopt TreeBridge. It has also been successfully deployed in Shopee's production environment, demonstrating both significant offline gains and substantial commercial impact. These results confirm the effectiveness, efficiency, and deployability of our framework in real-world recommender systems.

References

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cao, Y.; Zhou, X.; Feng, J.; Huang, P.; Xiao, Y.; Chen, D.; and Chen, S. 2022. Sampling Is All You Need on Modeling Long-Term User Behaviors for CTR Prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, 2974–2983. New York, NY, USA: Association for Computing Machinery. ISBN 9781450392365.

Chang, J.; Zhang, C.; Fu, Z.; Zang, X.; Guan, L.; Lu, J.; Hui, Y.; Leng, D.; Niu, Y.; Song, Y.; and Gai, K. 2023. TWIN: TWo-stage Interest Network for Lifelong User Behavior Modeling in CTR Prediction at Kuaishou. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, 3785–3794. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701030.

Chen, J.; Chi, L.; Peng, B.; and Yuan, Z. 2024. HLLM: Enhancing Sequential Recommendations via Hierarchical Large Language Models for Item and User Modeling. arXiv:2409.12740.

Chen, Y.; Huzhang, G.; Zeng, A.; Yu, Q.; Sun, H.; Li, H.-Y.; Li, J.; Ni, Y.; Yu, H.; and Zhou, Z. 2023. Clustered Embedding Learning for Recommender Systems. In *Proceedings of the ACM Web Conference 2023*, WWW '23, 1074–1084. New York, NY, USA: Association for Computing Machinery. ISBN 9781450394161.

Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; Anil, R.; Haque, Z.; Hong, L.; Jain, V.; Liu, X.; and Shah, H. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, 7–10. New York, NY, USA: Association for Computing Machinery. ISBN 9781450347952.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *indegoProceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, 4171–4186.

Feng, C.; Li, W.; Lian, D.; Liu, Z.; and Chen, E. 2022. Recommender forest for efficient retrieval. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713871088.

Huang, C.; Yu, T.; Xie, K.; Zhang, S.; Yao, L.; and McAuley, J. 2024. Foundation models for recommender systems: A survey and new perspectives. *arXiv preprint arXiv:2402.11143*.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *indegoProceedings of the 22nd ACM International Conference on Information & Knowledge Management*, CIKM '13, 2333–2338. New York, NY, USA: Association for Computing Machinery. ISBN 9781450322638.

Jia, J.; Wang, Y.; Li, Y.; Chen, H.; Bai, X.; Liu, Z.; Liang, J.; Chen, Q.; Li, H.; Jiang, P.; et al. 2024. Knowledge adaptation from large language model to recommendation for practical industrial application. *indegoarXiv preprint arXiv:2405.03988*.

Li, H.-Y.; Ni, Y.; Zeng, A.; Yu, H.; and Miao, C. 2022. Prior-Guided Transfer Learning for Enhancing Item Representation in E-commerce. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11): 12387–12395.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *indegoInternational conference on machine learning*, 19730–19742. PMLR.

Li, R.; Deng, W.; Cheng, Y.; Yuan, Z.; Zhang, J.; and Yuan, F. 2023b. Exploring the Upper Limits of Text-Based Collaborative Filtering Using Large Language Models: Discoveries and Insights. arXiv:2305.11700.

Li, S.; Lv, F.; Jin, T.; Lin, G.; Yang, K.; Zeng, X.; Wu, X.-M.; and Ma, Q. 2021. Embedding-based Product Retrieval in Taobao Search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, 3181–3189. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383325.

Lin, J.; Dai, X.; Xi, Y.; Liu, W.; Chen, B.; Zhang, H.; Liu, Y.; Wu, C.; Li, X.; Zhu, C.; et al. 2023. How can recommender systems benefit from large language models: A survey. *arXiv preprint arXiv:2306.05817*.

Liu, Q.; Wu, X.; Wang, W.; Wang, Y.; Zhu, Y.; Zhao, X.; Tian, F.; and Zheng, Y. 2024. LLMEmb: Large Language Model Can Be a Good Embedding Generator for Sequential Recommendation. arXiv:2409.19925.

Liu, Y.; Cao, J.; Wang, S.; Wen, S.; Chen, X.; Wu, X.; Yang, S.; Liu, Z.; Gai, K.; and Zhou, G. 2025. LLM-Alignment Live-Streaming Recommendation. arXiv:2504.05217.

Ni, Y.; Ou, D.; Liu, S.; Li, X.; Ou, W.; Zeng, A.; and Si, L. 2018. Perceive Your Users in Depth: Learning Universal User Representations from Multiple E-commerce Tasks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, 596–605. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355520.

Ni, Y.; Wu, Y.; Li, J.; Zeng, A.; Yu, H.; and Li, X. 2025. Dynamic masking-based feature interaction modeling for e-commerce click-through rate prediction. *Engineering Applications of Artificial Intelligence*, 157: 111184.

Qiu, Z.; Zhu, J.; Chen, Y.; Cai, G.; Liu, W.; Dong, Z.; and King, I. 2024. EASE: Learning Lightweight Semantic Feature Adapters from Large Language Models for CTR Prediction. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, 4819–4827. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704369.

Shen, Y.; He, X.; Gao, J.; Deng, L.; and Mesnil, G. 2014. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, 373–374. New York, NY, USA: Association for Computing Machinery. ISBN 9781450327459.

Song, C.; Shen, C.; Gu, H.; Wu, Y.; Yi, L.; Wen, J.; and Chen, C. 2024. PRECISE: Pre-training Sequential Recommenders with Collaborative and Semantic Information. *indegoarXiv preprint arXiv:2412.06308*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971.

Yang, L.; Subbiah, A.; Patel, H.; Li, J. Y.; Song, Y.; Mirghaderi, R.; and Aggarwal, V. 2024. Item-Language Model for Conversational Recommendation. *indegoarXiv preprint arXiv:2406.02844*.

Zhou, G.; Mou, N.; Fan, Y.; Pi, Q.; Bian, W.; Zhou, C.; Zhu, X.; and Gai, K. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 5941–5948.

Zhu, Y.; Yuan, H.; Wang, S.; Liu, J.; Liu, W.; Deng, C.; Dou, Z.; and Wen, J.-R. 2023. Large language models for information retrieval: A survey. *arXiv preprint arXiv:2308.07107*.