Hierarchical Group-wise Ranking Framework for Recommendation Models

YaChen Yan
yachen.yan@creditkarma.com
Credit Karma
San Francisco, California, USA

Liubo Li
liubo.li@creditkarma.com
Credit Karma
San Francisco, California, USA

Ravi Choudhary
ravi.choudhary@creditkarma.com
Credit Karma
San Francisco, California, USA

ABSTRACT

In modern recommender systems, CTR/CVR models are increasingly trained with ranking objectives to improve item ranking quality. While this shift aligns training more closely with serving goals, most existing methods rely on in-batch negative sampling, which predominantly surfaces easy negatives. This limits the model's ability to capture fine-grained user preferences and weakens overall ranking performance. To address this, we propose a Hierarchical Group-wise Ranking Framework with two key components. First, we apply residual vector quantization to user embeddings to generate hierarchical user codes that partition users into hierarchical, trie-structured clusters. Second, we apply listwise ranking losses to user-item pairs at each level of the hierarchy, where shallow levels group loosely similar users and deeper levels group highly similar users, reinforcing learning-to-rank signals through progressively harder negatives. Since users with similar preferences and content exposure tend to yield more informative negatives, applying ranking losses within these hierarchical user groups serves as an effective approximation of hard negative mining. Our approach improves ranking performance without requiring complex real-time context collection or retrieval infrastructure. Extensive experiments demonstrate that the proposed framework consistently enhances both model calibration and ranking accuracy, offering a scalable and practical solution for industrial recommender systems.

CCS CONCEPTS

• **Computing methodologies**; • **Machine learning**; • **Machine learning approaches**; • **Neural networks**;

KEYWORDS

Recommender Systems, Learning to Rank, Vector Quantization

ACM Reference Format:
YaChen Yan, Liubo Li, and Ravi Choudhary. 2025. Hierarchical Group-wise Ranking Framework for Recommendation Models. 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/XXXXXXX.XXXXXXX

1 INTRODUCTION

Click-through rate (CTR) and conversion rate (CVR) prediction models play a pivotal role in large-scale recommender systems and online advertising. While most modern systems rely on binary classification objectives such as log loss to estimate the likelihood of user actions, enhancing the ranking quality of model predictions has become a critical direction for improving user experience and achieving business goals. In this context, learning-to-rank (LTR) objectives, including pairwise and listwise losses, are widely adopted to better capture users' relative preferences among items.

However, a persistent challenge lies in constructing meaningful item comparisons during training. In particular, existing ranking losses often rely on in-batch negative sampling or uniformly sampled negative pairs, which tend to overemphasize easy negatives while underutilizing more informative, harder negatives. Recent research has shown that sampling negatives based on similarity or gradient-based importance can significantly improve model performance, but often at the cost of increased computational overhead, particularly in real-time environments. Existing context-aware approaches such as JRC [6] and SBCR [10] improve ranking performance by leveraging online ranked list logging. However, these methods require real-time infrastructure and tightly integrated systems, which increases deployment complexity and limits scalability in production environments. Furthermore, CVR models often suffer from sparse in-session user feedback, limiting the effectiveness of context-aware negative sampling based on in-session interactions.

To address these challenges, we propose a novel **Hierarchical Group-wise Ranking Framework** that improves ranking performance without relying on real-time context or nearest-neighbor retrieval. Our approach uses residual vector quantization (RVQ) to learn hierarchical user codes and group user-item pairs into multi-level clusters. Within each group, we apply listwise ranking losses over progressively harder negatives, based on the intuition that users with similar profiles or behaviors yield more informative comparisons. This hierarchical, multi-resolution cross-user sampling provides an efficient and scalable alternative for industrial recommendation systems. The main contributions of this paper can be summarized as follows:

- We propose a residual vector quantization module to encode user embeddings into hierarchical discrete codes, which serve as the foundation for multi-level user grouping. This structure enables dynamic and granular control over the difficulty level of sampled negatives during training.
- We introduce a hierarchical group-wise listwise ranking loss that applies ranking loss within user groups defined at each hierarchical level. By varying the granularity level of grouping, our method progressively surfaces harder negatives,

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

offering an efficient alternative to gradient-based sampling strategies.

- We integrate this hierarchical ranking objective with standard calibration losses in a multi-task learning framework and demonstrate that our method improves both convergence efficiency and ranking performance across multiple domains, without requiring real-time context collection or retrieval infrastructure. Our framework serves as a plug-in component that introduces negligible additional training overhead and zero serving latency.

2 PRELIMINARIES

The recommendation model with binary relevance is commonly formulated as a binary classification problem optimized with binary logistic loss. The binary logistic loss function is defined as:

$$\mathcal{L}_{\text{logloss}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{1}$$

where $\hat{y}_i$ represents the predicted probability and $y_i \in \{0, 1\}$ is the binary feedback label. To enhance ranking performance, recent approaches incorporate Learning-to-Rank (LTR) losses as auxiliary objectives. The combined objective function $\mathcal{L} = \mathcal{L}_{\text{logloss}} + \lambda\mathcal{L}_{rank}$ integrates binary logistic loss with either pairwise ranking loss (comparing item pairs) or listwise ranking loss (optimizing the entire item list ordering), where λ controls the contribution of the ranking component. In this setting, the ranking loss typically operates on the model's predicted scores over user-item pairs. Negative samples for the ranking loss are commonly drawn uniformly from other user-item pairs within the same training batch (i.e., in-batch negative sampling) to construct contrastive comparisons during training.

Without loss of generality, the notation used throughout this paper for recommendation models with binary relevance is as follows: the training dataset consists of instances (x_u, x_i, y), where x_u and x_i denote user and item features respectively, $y \in \{0, 1\}$ indicates the binary user-item feedback label. The raw user features x_u and item features x_i are processed by their respective networks to obtain user embedding $\mathbf{e}_u$ and item embedding $\mathbf{e}_i$. These embeddings are then fed into a main network that outputs logit s, which is transformed into the predicted probability $\hat{y} = \sigma(s)$ using the sigmoid function σ. The primary optimization objective is the binary logistic loss comparing $\hat{y}$ against ground truth labels y, while a ranking loss encourages correct ordering of items. This general notation serves as the foundation for our proposed model as well, which extends these concepts with additional specialized notation to capture our proposed framework.

3 NEGATIVE SAMPLING CHALLENGES AND THEORETICAL GRADIENT ANALYSIS

While widely adopted, CTR/CVR models that combine binary logistic and ranking losses still face key optimization challenges—namely, inefficient negative sampling and slow convergence. Existing methods often focus on easy negatives that offer limited learning value. Through gradient-based analysis, we show that sampling harder negatives in proportion to their gradient norms significantly improves convergence. This insight motivates our hierarchical groupwise sampling strategy, which efficiently surfaces challenging negatives from similar users during training.

3.1 Limitations of Uniform Sampling

Traditional negative sampling strategies, such as uniform in-batch sampling, often prioritize easy negatives that are trivial for the model to distinguish. This limits the model's ability to learn fine-grained user preference signals and ultimately undermines ranking performance. To address this, recent work has explored retrieving harder negatives via approximate nearest neighbor (ANN) search [8], which improves the model's capacity to capture subtle distinctions. However, ANN-based methods introduce significant computational overhead and are generally better suited to retrieval models than to ranking models, which require capturing complex user-item interactions beyond inner products and supporting online learning. The key challenge is to efficiently surface informative hard negatives during training, without relying on real-time context collection or exhaustive nearest-neighbor search.

3.2 Gradient-Based Sampling Theory

Consider a training batch with positive and negative samples. An importance-weighted stochastic gradient descent (SGD) update for the ranking loss can be expressed as:

$$\theta_{t+1} = \theta_t - \eta\frac{1}{Np^-}\nabla_{\theta_t} l(s^+, s^-) \tag{2}$$

where η is the learning rate. θ_t represents the parameters at iteration t, θ_{t+1} the updated parameters. p^- denotes the probability of selecting a particular negative instance (x_u, x_i, y^-). The scaling factor $\frac{1}{Np^-}$ ensures an unbiased gradient estimate. Following derivations in variance reduction [3, 4], let $g = \frac{1}{Np^-}\nabla_{\theta_t} l(s^+, s^-)$ be the weighted gradient, we can write the convergence rate as:

$$\begin{aligned} E\Delta_t &= \|\theta_t - \theta^*\|^2 - E_{P^-}(\|\theta_{t+1} - \theta^*\|^2) \\ &= 2\eta E_{P^-}(g)^T(\theta_t - \theta^*) - \eta^2 E_{P^-}(g)^T E_{P^-}(g) \\ &\quad - \eta^2 \mathrm{Tr}(V_{P^-}(g)) \end{aligned} \tag{3}$$

where P^- is the negative sampling distribution for a given positive example (x_u, x_i, y^+). This formulation shows that convergence can be improved by selecting negative examples from a distribution that reduces $\mathrm{Tr}(V_{P^-}(g))$, which quantifies the total gradient variance introduced by the negative sampling. The optimal sampling strategy is

$$p^{*-} = \arg\min_{p^-} \mathrm{Tr}(V_{P^-}(g)) \propto \|\nabla_{\theta_t} l(s^+, s^-)\|^2 \tag{4}$$

The above analysis shows that the optimal negative sampling distribution is proportional to the squared gradient norm of each instance, favoring samples that contribute larger updates to the model. This approach is particularly valuable in recommendation systems, where the vast majority of negative examples contribute minimal learning signals. Prioritizing hard negatives with large gradient contributions reduces the variance of gradient estimates and accelerates convergence, aligning with established variance reduction principles in stochastic optimization and motivating our gradient-informed sampling strategy.

4 PROPOSED FRAMEWORK

To operationalize the above theoretical insight, we propose a hierarchical group-wise negative sampling strategy that approximates the optimal distribution without incurring costly computations. Our approach clusters users based on profile or behavioral similarity, and groups the associated user-item samples according to these user clusters across multiple hierarchical levels. This structure enables the model to sample negatives from users of varying similarity: from coarse groups capturing general behavior patterns to fine-grained subgroups reflecting closely shared interests. Intuitively, negatives samples drawn from similar users are more informative, as similar users tend to be exposed to overlapping content and exhibit comparable preferences. By leveraging negative samples from similar users, our method introduces progressively harder negatives that enhance the learning signal throughout the hierarchy. This promotes more effective optimization of ranking loss while maintaining computational efficiency, ultimately improving learning dynamics in CTR/CVR prediction tasks.

We designed the proposed framework, as illustrated in Figure 1 with three main components:

Hierarchical User Code Generation: We quantize each user embedding into a structured sequence of discrete codes using multi-stage residual vector quantization, where each stage uses a codebook to quantize the remaining error from the previous level. This produces a hierarchical code sequence that forms a trie-like structure, where higher levels represent broad semantic groupings and deeper levels capture fine-grained user distinctions.

Hierarchical Group-wise Ranking Objective: Based on the generated user codes, we organize users into nested groups where users sharing the same prefix codes at each level are grouped together, forming a trie-like structure of increasing similarity. We apply listwise ranking loss within each group, computing the loss over groups containing user-item pairs with shared code prefixes. By varying the group depth, we control negative difficulty: shallow levels provide easier negatives from loosely similar users, while deeper levels yield harder negatives from highly similar users. To balance contributions across hierarchy levels, we employ an uncertainty-based weighting scheme, enabling the model to adaptively focus on the most informative hierarchy depths during training.

Multi-Objective Training Strategy: Our training objective combines three components: a primary calibration loss on predictions from the original user embedding, an auxiliary calibration loss on predictions from the quantized embedding (using straight-through estimator to enable gradient flow), and the proposed hierarchical ranking loss. Unlike traditional vector quantization approaches, we omit commitment loss to preserve adaptability in dynamic recommendation settings, instead relying on the auxiliary calibration loss to encourage alignment between original and quantized embeddings while maintaining flexibility for evolving user preferences and behaviors.

In our model specifically, the user embedding $\mathbf{e}_u$ undergoes residual vector quantization to produce the quantized user embedding $\mathbf{e}_u^q$ and user hierarchical codes $\mathbf{c}_u$. When the shared main network uses the quantized user embedding $\mathbf{e}_u^q$ and the item embedding $\mathbf{e}_i$ as inputs, it produces logit s^q and the corresponding predicted probability $\hat{y}^q = \sigma(s^q)$. To avoid conflicting gradients between the dual prediction paths, we stop the gradient flow from the auxiliary loss into the item embedding by applying stop_gradient($\mathbf{e}_i$) when computing $\hat{y}^q$. This ensures that only the user network receives updates from the auxiliary calibration loss, preserving training stability. Both the original and quantized predictions ($\hat{y}$ and $\hat{y}^q$) are optimized using calibration loss against ground truth labels. Additionally, the user hierarchical codes $\mathbf{c}_u$ are used to compute a hierarchical group-wise ranking loss to further enhance ranking performance. These notations ($\mathbf{e}_u^q$, $\mathbf{c}_u$, s^q, $\hat{y}^q$) extend the general framework presented earlier and are specific to our quantization-based approach, introducing additional variables necessary to describe our model's unique architecture and optimization strategy.

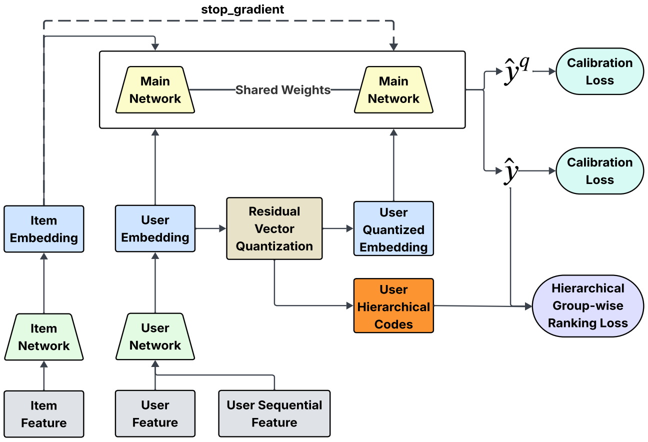

Figure 1: The Architecture of the Proposed Framework

4.1 Hierarchical User Codes Generation

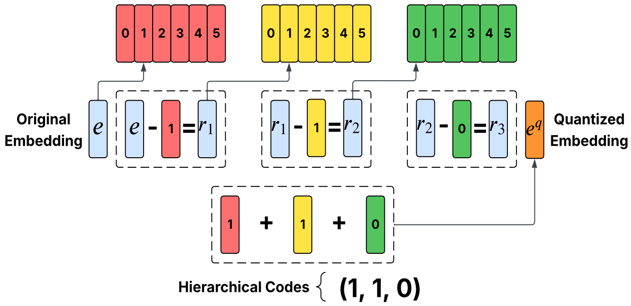

Figure 2: Residual Vector Quantization

To capture structured user similarity and enable efficient group-wise sampling, we discretize user embeddings using a residual vector quantization (RVQ) framework. This process encodes each user into a sequence of discrete codes, referred to as hierarchical user codes, which form the foundation of our multi-resolution user grouping strategy. Figure 2 illustrates the cascaded quantization process and the resulting hierarchical user code structure.

Given a user embedding $\mathbf{e}_u \in \mathbb{R}^d$ produced by the user network (which may include contextual or sequential features), we apply an L-stage residual quantization procedure to obtain a code sequence $\mathbf{c}_u = [\mathbf{c}_{u,1}, \ldots, \mathbf{c}_{u,L}]$. At each stage l, a codebook $C^{(l)} = \{C_1^{(l)}, \ldots, C_K^{(l)}\}$ is used to quantize the residual vector passed down from the previous level. These codebooks are arranged in a cascaded structure, where each level incrementally refines the remaining quantization error:

$$\begin{aligned} \mathbf{r}_u^{(1)} &= \mathbf{e}_u \\ \mathbf{c}_{u,l} &= \arg\min_k \|\mathbf{r}_u^{(l)} - C_k^{(l)}\|_2^2 \\ \mathbf{r}_u^{(l+1)} &= \mathbf{r}_u^{(l)} - C_{\mathbf{c}_{u,l}}^{(l)} \end{aligned} \tag{5}$$

The quantized embedding is reconstructed by summing codebook vectors from all stages:

$$\hat{\mathbf{e}}_u = \sum_{l=1}^{L} C_{\mathbf{c}_{u,l}}^{(l)} \tag{6}$$

To ensure stable learning and effective usage of codebook entries, each codebook is updated using Exponential Moving Average (EMA) strategy. Following each assignment, usage statistics and accumulated residuals are used to softly update the code vectors:

$$C_k^{(l)} \leftarrow m \cdot C_k^{(l)} + (1-m) \cdot \mu_k, \quad N_k \leftarrow m \cdot N_k + (1-m) \cdot n_k \tag{7}$$

where μ_k and n_k represent the average and count of residuals assigned to code k, and m is the EMA decay rate. We also apply Laplace smoothing to the EMA count to avoid instability from rare updates. To prevent representation collapse, we replace infrequently used codes with randomly sampled embeddings from the current batch, following SoundStream [9].

The resulting hierarchical code sequence $\mathbf{c}_u = [\mathbf{c}_{u,1}, \ldots, \mathbf{c}_{u,L}]$ defines a multi-level grouping scheme, assigning each user to a path in a trie-like structure. Higher levels represent coarse semantic groupings, while deeper levels capture increasingly fine-grained distinctions. Users sharing longer prefix codes are considered more similar, enabling efficient multi-resolution grouping where upper levels offer diversity and lower levels surface harder negatives from closely related users.

4.2 Hierarchical Group-wise Ranking Objective

The hierarchical user codes $\mathbf{c}_u = [\mathbf{c}_{u,1}, \ldots, \mathbf{c}_{u,L}]$ enable structured negative sampling through multi-level user grouping. At each level l, users sharing the same prefix $(\mathbf{c}_{u,1}, \ldots, \mathbf{c}_{u,l})$ form nested clusters of growing specificity. This structure defines semantically coherent user-item groups at multiple granularities, allowing listwise ranking losses to be applied within groups of progressively similar users—supporting harder negative sampling and more effective ranking supervision.

As illustrated in Figure 3, we recursively partition user-item pairs into finer groups according to these hierarchical prefixes. Within each group, the users' positive items are treated as positive examples, while users' negative items serve as negatives. By varying the group depth l, we effectively control the difficulty of sampled negatives: shallower levels provide easier negatives from loosely similar users, while deeper levels yield harder negatives from highly similar users who share more behavioral or contextual overlap and content exposure.

To train the model on these grouped samples, we adopt the Regression Compatible Listwise Cross Entropy loss (ListCE)[1], which replaces the softmax transformation with the sigmoid function based normalization. This improves compatibility between ranking and calibration losses under binary relevance.

Let $G_1^{(l)}, G_2^{(l)}, \ldots, G_{M_l}^{(l)}$ denote the groups formed at level l, where each group $G_m^{(l)}$ contains user-item pairs sharing the same code prefix. The listwise loss at level l is defined as:

$$\mathcal{L}_{\text{listce}}^{(l)}(s, y) = \frac{1}{M_l} \sum_{m=1}^{M_l} \sum_{i \in G_m^{(l)}} -\tilde{y}_i^{(l,m)} \log \left(\frac{\sigma(s_i)}{\sum_{j \in G_m^{(l)}} \sigma(s_j)} \right) \tag{8}$$

where s_i is the predicted logit for user-item pair i, and $\sigma(s_i)$ represents its sigmoid-transformed score. The corresponding normalized label $\tilde{y}_i^{(l,m)}$ is computed as: $\tilde{y}_i^{(l,m)} = \frac{y_i}{\sum_{j \in G_m^{(l)}} y_j + \epsilon}$, ensuring that labels are normalized within each group $G_m^{(l)}$.

To balance the contribution from different hierarchy levels, we introduce an uncertainty-based weighting scheme [5]. Each level l is associated with a learnable uncertainty parameter σ_l. The total hierarchical ranking loss is defined as:

$$\mathcal{L}_{\text{hierarchical}} = \sum_{l=1}^{L} \left(\frac{1}{2\sigma_l^2} \mathcal{L}_{\text{listwise}}^{(l)}(s, y) + \log \sigma_l \right) \tag{9}$$

This formulation enables the model to adaptively weight ranking loss from each level based on its estimated uncertainty. As a result, the model learns to focus on the most informative hierarchy depths during training, while preserving stable and balanced optimization across levels.

4.3 Multi-Objective Training Strategy

4.3.1 Objective Function Formulation. Our overall training objective integrates three components: a primary calibration loss on the predicted click-through probability from the original user embedding, a secondary calibration loss from the quantized user embedding, and a hierarchical group-wise listwise ranking loss. Formally, the total loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{loss}} &= \mathcal{L}_{\text{logloss}}(\hat{y}, y) \\ &+ \lambda \mathcal{L}_{\text{logloss}}(\hat{y}^q, y) \\ &+ \mathcal{L}_{\text{hierarchical}} \end{aligned} \tag{10}$$

The first term, $\mathcal{L}_{\text{logloss}}(\hat{y}, y)$, serves as the primary loss for calibrating the model's click-through probability prediction using the original user embedding $\mathbf{e}_u$. This component ensures that the model produces well-calibrated probability estimates suitable for real-world serving, ensuring compatibility with recommendation systems that consume predicted probabilities.

The second term, $\mathcal{L}_{\text{logloss}}(\hat{y}^q, y)$, introduces an auxiliary calibration loss applied to predictions derived from the quantized user embedding $\mathbf{e}_u^q$, obtained via residual vector quantization. Importantly, to enable backpropagation through the non-differentiable quantization operation, we apply a straight-through estimator (STE):

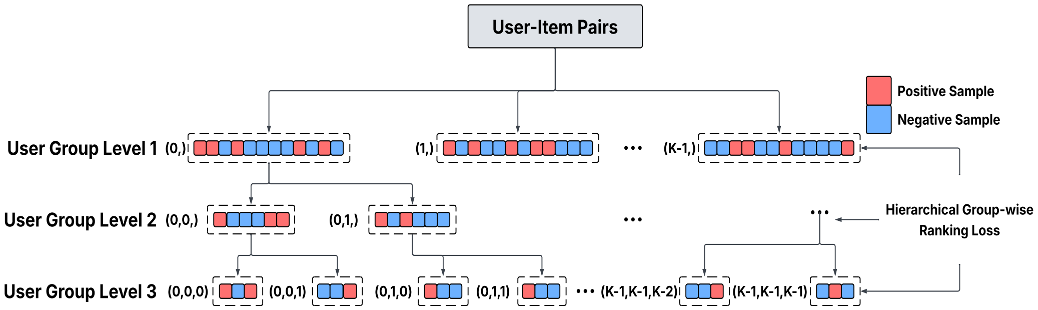

Figure 3: Hierarchical Group-wise Ranking Framework. A trie-structured approach organizes user-item pairs into multi-level user groups based on shared code prefixes. Positive (red) and negative (blue) examples are drawn within each group to supervise ranking objectives. Varying the group depth enables learning to rank over increasingly fine-grained user similarity.

$$\mathbf{e}_u^q = \mathbf{e}_u + \text{stop\_gradient}(\hat{\mathbf{e}}_u - \mathbf{e}_u) \tag{11}$$

Here, $\hat{\mathbf{e}}_u = \sum_{l=1}^{L} C_{\mathbf{c}_{u,l}}^{(l)}$ represents the quantized embedding. This formulation makes $\mathbf{e}_u^q$ match $\hat{\mathbf{e}}_u$ in the forward pass, while gradients flow through $\mathbf{e}_u$ in the backward pass, enabling user encoder training with quantization-based regularization.

Although $\mathbf{e}_u^q$ is not used during serving, this auxiliary loss regularizes the shared user network by encouraging it to produce embeddings that are not only predictive but also structurally compressible and cluster-aware. Specifically, it helps align the user embedding space with the quantized codebook space, promoting smoother transitions and more stable quantization behavior. This alignment facilitates better codebook utilization and supports dynamic clustering under shifting user distributions. As a result, it improves training stability by maintaining a semantically meaningful and adaptive latent structure.

The final component, $\mathcal{L}_{\text{hierarchical}}$, is the proposed hierarchical group-wise listwise ranking loss. It utilizes trie-structured user codes generated via residual vector quantization to group user-item pairs into semantically coherent user clusters at multiple levels of granularity. By applying the ranking loss within these groups, the model receives user cluster-aware ranking supervision using progressively harder negatives drawn from similar users. This structure improves ranking quality without incurring the cost of real-time context collection or explicit nearest-neighbor retrieval.

4.3.2 Clustering Adaptability. Traditional vector quantization frameworks such as VQ-VAE [7] typically employ a commitment loss (e.g., $\|\mathbf{e}_u - \hat{\mathbf{e}}_u\|^2$) to explicitly align the continuous embedding with its quantized counterpart. However, we omit this component in our framework due to the dynamic nature of real-time recommendation settings, where user embeddings must continually adapt to evolving preferences, behaviors, and contextual signals.

Enforcing a commitment loss would constrain user embeddings to remain near static quantized representations, limiting their ability to transition across clusters and adapt in response to new interactions. Instead, we allow the primary calibration loss $\mathcal{L}_{\text{logloss}}(\hat{y}, y)$ to guide representation learning, maintaining expressiveness and adaptability. To softly encourage alignment between the embedding and its quantized version, we incorporate an auxiliary calibration loss $\mathcal{L}_{\text{logloss}}(\hat{y}^q, y)$ applied to the quantized prediction. This auxiliary loss serves as a task-driven regularizer, encouraging meaningful and quantization-friendly embeddings without rigid constraints.

Our approach supports flexible and generalizable representation learning under streaming or non-stationary conditions, aligning with insights from recent work on real-time indexing [2], which similarly avoids commitment loss to preserve adaptability.

5 EXPERIMENTS

In this section, we evaluate our framework on two large-scale public datasets, demonstrating that hierarchical group-wise ranking significantly improves model performance. During experiments, we focus on evaluating the effectiveness of our proposed models and answering the following questions.

- **Q1**: How does our proposed framework perform on ranking tasks? Is it effective and efficient in extremely high-dimensional and sparse data settings?
- **Q2**: How well does our framework handle user cold-start scenarios with limited interaction history? Can it maintain robust performance when user signals are sparse?

5.1 Experiment Setup

5.1.1 Datasets. We evaluate our proposed model using two publicly available real-world datasets commonly utilized in research: KuaiRand and Taobao. The data is randomly divided into three subsets: 70% for training, 10% for validation, and 20% for testing. We applied stratified sampling to ensure that each user has positive samples in every data subset.

- **KuaiRand**[1] is a recommendation dataset collected from the video-sharing mobile app Kuaishou.
- **Taobao**[2] is a Taobao E-commerce dataset released Alibaba.

5.1.2 Evaluation Metrics. For evaluating the ranking performance, we adopt LogLoss and AUC metric and further compute the Group

[1] https://kuairand.com/

[2] https://tianchi.aliyun.com/dataset/649

AUC (GAUC) to measure the goodness of intra-user ranking ability. Group AUC calculates the AUC for each user and aggregates them using a weighted average based on the number of impressions, capturing per-user ranking quality and better reflecting real-world performance.

5.2 Model Performance Comparison (Q1)

Table 1: Performance Comparison of Different Ranking Objectives on KuaiRand and Taobao Datasets.

Objective	**KuaiRand**			**Taobao**		
	LogLoss	AUC	GAUC	LogLoss	AUC	GAUC
LogLoss	0.5735	0.7510	0.6911	0.2011	0.6420	0.5708
LogLoss + PairwiseLogistic	0.5723	0.7524	0.6921	0.2002	0.6435	0.5728
LogLoss + SoftmaxCE	0.5727	0.7520	0.6920	0.2005	0.6428	0.5720
LogLoss + ListCE	0.5709	0.7537	0.6932	0.1995	0.6443	0.5734
JRC	0.5713	0.7533	0.6930	0.1993	0.6540	0.5732
GroupCE (proposed)	**0.5681**	**0.7556**	**0.6953**	**0.1982**	**0.6556**	**0.5745**

The overall performance of different losses is listed in Table 1. We have the following observations in terms of objective function effectiveness:

- **LogLoss** serves as the baseline objective and yields the lowest performance across both datasets, demonstrating the limitations of using a pure calibration loss without ranking supervision.
- **LogLoss + PairwiseLogistic** and **LogLoss + SoftmaxCE** show consistent improvements over the baseline, highlighting the benefit of incorporating pairwise or listwise ranking losses into model training.
- **LogLoss + ListCE** and **JRC** achieve further gains, demonstrating that listwise ranking objectives with calibration-compatible designs lead to stronger overall performance.
- **GroupCE** achieves the best performance across all metrics. These results validate the effectiveness of our hierarchical group-wise ranking strategy, which enables progressively harder negative sampling through structured user clustering.

5.3 Cold Start Capability (Q2)

To assess the model's robustness in user cold-start scenarios, we evaluate its performance on user cohorts with limited interaction history. Since our model leverages hierarchical user codes for structured contrastive learning, we hypothesize that it can capture discriminative patterns at the cluster level, enabling it to maintain ranking quality even when individual user signals are sparse.

Table 2: Performance Comparison in Cold-Start Scenarios on KuaiRand Dataset.

Objective	**KuaiRand (Cold)**			**KuaiRand (Warm)**		
	LogLoss	AUC	GAUC	LogLoss	AUC	GAUC
LogLoss	0.6189	0.7298	0.6718	0.5683	0.7454	0.6945
LogLoss + PairwiseLogistic	0.6171	0.7305	0.6735	0.5670	0.7461	0.6955
LogLoss + SoftmaxCE	0.6175	0.7302	0.6730	0.5674	0.7458	0.6952
LogLoss + ListCE	0.6137	0.7308	0.6732	0.5662	0.7469	0.6962
JRC	0.6145	0.7308	0.6738	0.5664	0.7475	0.6968
GroupCE (proposed)	**0.6115**	**0.7320**	**0.6786**	**0.5636**	**0.7489**	**0.6986**

We stratify users by the number of impressions in the KuaiRand training set into cold (≤ 20) and warm ($20 - 50$) groups, and assess model performance on the corresponding users in the test set. As shown in Table 2, our proposed GroupCE framework consistently outperforms baselines across both user segments, with the most notable GAUC gains observed in the cold-start group. These results indicate that hierarchical clustering introduces effective user-level priors, enabling better ranking performance even with limited user history, a promising direction for addressing cold-start challenges in recommendation systems.

6 CONCLUSION

We propose a Hierarchical Group-wise Ranking Framework that enhances CTR/CVR model performance by leveraging residual vector quantization to construct hierarchical user clusters for structured, cluster-aware ranking supervision. By grouping user-item pairs into semantically coherent clusters, our framework enables efficient and effective hard negative sampling without requiring real-time context collection or incurring the computational overhead of cross-user retrieval. We introduce a hierarchical listwise ranking loss to model item ordering across varying levels of user similarity and complement it with calibration objectives applied to both original and quantized embeddings. Extensive experiments across multiple datasets demonstrate consistent improvements in both ranking accuracy and calibration quality, particularly under sparse user behavior scenarios. The proposed framework provides a scalable and generalizable solution for industrial recommendation systems and opens new directions for ranking optimization via quantization-based hierarchical clustering, enabling efficient learning-to-rank under limited user feedback.

REFERENCES

[1] Aijun Bai, Rolf Jagerman, Zhen Qin, Le Yan, Pratyush Kar, Bing-Rong Lin, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2023. Regression compatible listwise objectives for calibrated ranking with binary relevance. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4502–4508.

[2] Xingyan Bin, Jianfei Cui, Wujie Yan, Zhichen Zhao, Xintian Han, Chongyang Yan, Feng Zhang, Xun Zhou, Qi Wu, and Zuotao Liu. 2025. Real-time Indexing for Large-scale Recommendation by Streaming Vector Quantization Retriever. *arXiv preprint arXiv:2501.08695* (2025).

[3] Tyler B Johnson and Carlos Guestrin. 2018. Training deep models faster with robust, approximate importance sampling. *Advances in Neural Information Processing Systems* 31 (2018).

[4] Angelos Katharopoulos and François Fleuret. 2018. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*. PMLR, 2525–2534.

[5] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7482–7491.

[6] Xiang-Rong Sheng, Jingyue Gao, Yueyao Cheng, Siran Yang, Shuguang Han, Hongbo Deng, Yuning Jiang, Jian Xu, and Bo Zheng. 2023. Joint optimization of ranking and calibration with contextualized hybrid model. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4813–4822.

[7] Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems* 30 (2017).

[8] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[9] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. 2021. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2021), 495–507.

[10] Shunyu Zhang, Hu Liu, Wentian Bao, Enyun Yu, and Yang Song. 2024. A Self-boosted Framework for Calibrated Ranking. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6226–6235.