Synergizing Implicit and Explicit User Interests: A Multi-Embedding Retrieval Framework at Pinterest

Zhibo Fan*
zb1439@outlook.com
Pinterest
San Francisco, CA, USA

Hongtao Lin
hongtaolin@pinterest.com
Pinterest
San Francisco, CA, USA

Haoyu Chen*
hchen@pinterest.com
Pinterest
San Francisco, CA, USA

Bowen Deng
bdeng@pinterest.com
Pinterest
San Francisco, CA, USA

Hedi Xia†
hxia@pinterest.com
Pinterest
San Francisco, CA, USA

Yuke Yan
yukeyan@pinterest.com
Pinterest
San Francisco, CA, USA

James Li
jamesyili@gmail.com
Pinterest
San Francisco, CA, USA

Abstract

Industrial recommendation systems are typically composed of multiple stages, including retrieval, ranking, and blending. The retrieval stage plays a critical role in generating a high-recall set of candidate items that covers a wide range of diverse user interests. Effectively covering the diverse and long-tail user interests within this stage poses a significant challenge: traditional two-tower models struggle in this regard due to limited user-item feature interaction and often bias towards top use cases. To address these issues, we propose a novel multi-embedding retrieval framework designed to enhance user interest representation by generating multiple user embeddings conditioned on both implicit and explicit user interests. Implicit interests are captured from user history through a Differentiable Clustering Module (DCM), whereas explicit interests, such as topics that the user has followed, are modeled via Conditional Retrieval (CR). These methodologies represent a form of conditioned user representation learning that involves condition representation construction and associating the target item with the relevant conditions. Synergizing implicit and explicit user interests serves as a complementary approach to achieve more effective and comprehensive candidate retrieval as they benefit on different user segments and extract conditions from different but supplementary sources. Extensive experiments and A/B testing reveal significant improvements in user engagements and feed diversity metrics. Our proposed framework has been successfully deployed on Pinterest home feed.

CCS Concepts

• **Information systems** → **Social recommendation**; *Information retrieval*.

Keywords

Recommendation System, Information Retrieval, Two-Tower Model

ACM Reference Format:
Zhibo Fan, Hongtao Lin, Haoyu Chen, Bowen Deng, Hedi Xia, Yuke Yan, and James Li. 2025. Synergizing Implicit and Explicit User Interests: A Multi-Embedding Retrieval Framework at Pinterest. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3711896.3737265

1 Introduction

As one of the largest visual discovery platforms, Pinterest hosts a billion-scale visual content gallery and inspires over 500 million users worldwide. Upon visiting Pinterest, users are immediately presented with a diversified and inspiring home feed (Figure 1, left) designed to help them discover new ideas. To enhance user engagement, it is crucial to surface recommendations that capture multiple user interests at first glance.

Pinterest home feed follows a modern multi-stage recommendation system architecture, consisting of retrieval, ranking, and blending. The retrieval stage focuses on generating candidate items with high recall and low serving latency. It sets the upper bound for the quality of recommendations, as subsequent stages primarily refine and re-rank the retrieved results. Different from Pin search [9] and related Pins [17], home feed retrieval often lacks explicit user browsing intents or contextual cues. Therefore, the key to effective home feed retrieval lies in holistically understanding user interests and ensuring sufficient coverage of relevant content.

Among various retrieval algorithms and strategies, embedding-based retrieval [5, 23] has been proved to be one of the most effective

*Work done at Pinterest.
†Corresponding author.

KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737265

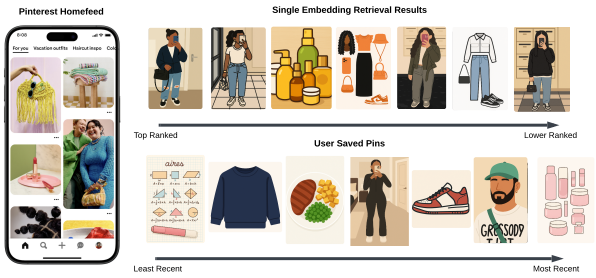

Figure 1: Left: Pinterest home feed screenshot. Right: A failure case of a vanilla two-tower model. The top retrieved candidates from the two-tower model (above) fail to cover user interests, such as food and education, from the user's previously saved Pins (below). Images from user history and recommendation results are iconized to obscure identifiable user information.

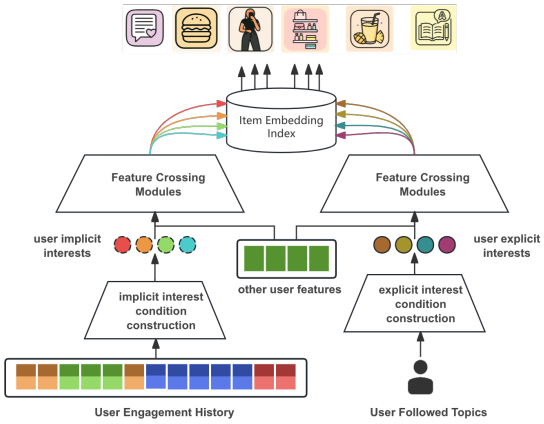

Figure 2: The multi-embedding retrieval framework conditioned on both implicit and explicit user interests.

and widely adopted approaches in industry, where both users and candidate items are encapsulated as latent embedding vectors, with dot-product (or cosine similarity) as the affinity measure. Such a formulation facilitates efficient retrieval via approximate nearest neighbor search (ANNS) [13, 20]. These models are commonly referred to as "Two-Tower Models" [5, 23, 36] due to their architecture, where user and item information are encoded separately in a user tower and an item tower, without interaction before the similarity computation. While effective and widely used, Two-Tower Models can struggle to capture long-tail user interests due to the lack of early-stage feature interaction between user and item features. Empirically, we observed that dominant user interests tend to overwhelm the user embedding, leading to fewer retrieved candidates from the torso and tail of a user's interest distribution. Figure 1 (right) illustrates an example where a single-embedding retrieval model overlooks user interests. This contradicts the primary objective of the retrieval stage–to maximize user interest coverage–and can negatively impact user engagement and retention.

To address this, we propose a multi-embedding retrieval framework (see Figure 2) to guide user embedding generation with both implicit and explicit user interests as conditions. It consists of (A) an implicit interest conditioned model utilizing a Differentiable Clustering Module (DCM) to extract implicit interests from user historical engagements, and (B) a conditional retrieval (CR) [14] model to retrieve candidate items related to explicit user interests, such as followed topics and long-term interests that are otherwise "forgotten" due to user history input limit. Both models generate multiple embeddings per user to represent diverse interests, but each of the models captures a different aspect of user interests to complement the other: implicit interest modeling quickly adapts to users' realtime engagements and benefits active users the most, whereas explicit interest conditioned retrieval not only replenishes overlooked or long-term interests for active users, but also acts as an important candidate source for new and low-signal users. Synergizing both models improves user interest coverage and drives higher online engagement across all user segments.

Conceptually, both models can be framed as conditional representation learning, involving two key aspects:

(1) **Condition construction**: user interests are extracted and encoded into embeddings, and
(2) **Condition association**: engaged candidates are linked to the corresponding interests to form training labels.

Specifically, DCM applies differentiable clustering over the user sequence to form latent user interest embeddings as conditions and associate them with positive labels inside the neural network. On the other hand, CR keeps record of source interests for engaged items at logging time and learns a topic embedding table to encode the conditions.

In summary, our main contributions of this work are as follows.

- We build a multi-embedding retrieval framework at Pinterest, which improves user interest coverage and provides highly engaging and diversified contents by synergizing both implicit and explicit user interests.
- We present our solution for implicit interest modeling, namely DCM, together with our extensive survey on a series of SoTA modeling strategies, and show the superior performance of DCM among others.
- We deploy the framework at Pinterest and verify its effectiveness via extensive online experiments. The deployed system improves major user engagement metrics by a large margin.

2 Related Works

In this section, we focus on discussing prior work on **embedding-based multi-interest retrieval** that is most relevant to our approach. We roughly categorize existing methods into three streams.

Self-Attention. This stream uses multi-head self-attention [10, 24, 26] to derive multiple user interests from the user engagement sequence. Rather than aggregating the multi-head outputs to form a single user embedding, these methods[3, 33] treat the output of each attention head as a distinct embedding to represent one of the user's latent interests.

Interest Token. Another line of research [16, 19, 25, 27, 30, 34] involves learnable model parameters to represent a range of interests explicitly. SINE [25] initializes a set of learnable concept prototype embeddings and activate top-K concepts per user request to attend to the user sequence. MVKE [34] introduces a set of "virtual kernels" to bridge the user tower and item tower. Omitting the multi-task setup, these "virtual kernels" can be interpreted as learnable query interest tokens for attention over user sequence. From this perspective, attention head weights in self-attentive methods [3, 33] can also be viewed as a special form of interest tokens. Additionally, rather than learning the concepts and their representations jointly, Conditional Retrieval (CR) [14] proposes to learn the embeddings for predefined conditions to provide better controllability when "bootstrapping for new use cases". In this paper, we also adopted CR for explicit interest modeling to complement user interest coverage.

Dynamic Routing. Unlike other modeling streams, dynamic routing based approaches [4, 12, 28] do not introduce specific model parameters to represent interests. Instead, they derive interest representations from diverse patterns within the user sequence via dynamic routing [22], which enables personalized interest granularity based on the varying diversity of user history. MIND [12] applies Capsule Networks [22] to extract the interest representations from the user sequence and UMI [4] enhances the routing by incorporating extra attention. Our deployed implicit interest modeling solution follows this line of research.

In addition to these modeling techniques, there are also works establishing multi-interest retrieval frameworks. ComiRec [3] extends self-attention and MIND [12] with a controllable diversification layer to build an implicit user interest modeling framework. Trinity [35] devises a complex system that extracts explicit user interests from different sources with machine learning and heuristics. In this paper, we present a deployed framework utilizing both implicit and explicit user interest models to cover different sources of user interests and boost the performance across all user segments.

3 Method

In this section, we elaborate on our proposed multi-embedding retrieval framework, starting from a problem statement as an overview of the framework. Next, we present the best performing solution, DCM, along with other alternatives for implicit interest modeling with an emphasis on how condition construction and association are implemented. After that, we describe the explicit interest modeling with CR [14]. Finally, we summarize the connection and the supplementary effect of the framework components and present deployment details.

3.1 Problem Statement

The general objective of recommendation models is to predict the ranking score for an item i given the user information u. Specifically, for retrieval stage models, to facilitate efficient retrieval, the score is usually formulated as $f(i|u) \propto \exp(\phi(u)^\top \psi(i))$, where ϕ and ψ are neural functions, aka "towers", that map input features of i and u to lower dimensional normalized vectors. Without cross-tower feature interactions, such a formulation tends to overamplify the head user interests in $\phi(u)$ and result in worse coverage for torso and tail user interests.

Our proposed multi-embedding framework introduces K_{im} implicit and K_{ex} explicit user interest conditions to guide the user embedding generation as shown in Figure 2, where K_{im} and K_{ex} are pre-defined parameters. For a user u with interest condition c, we have more information to tell if they will engage with an item i and thus could revamp the ranking score as $f(i|u,c) \propto \exp(\phi(u,c)^\top \psi(i))$. However, successful conditioning[6, 15] is challenging in two aspects. First, we need to carefully pick or construct the user interest conditions. While explicit interests are usually those selected by users at sign-up or later on, implicit interests need to be mined from user engagement sequence s_u. Second, we need to associate the positive samples i to the right condition c for the model to learn effectively. This can be done at model training time for implicit user interest modeling and at data logging time for explicit user interest modeling, which will be detailed in the following sections. We concentrate on condition construction and condition association as two key elements that illuminate the commonalities among the methods.

3.2 Implicit User Interest Modeling

3.2.1 Capsule Networks for Recommendations. Li et al. proposed MIND [12] to use Capsule Networks in recommendation systems to extract user interest representations from user engagement history. Capsules are groups of neurons whose embedding vectors represent spatial hierarchies of the encoded concepts [22]. Each item e_i in the user sequence can be seen as lower capsules, which are used to derive the higher capsules $\{c_j\}_{j=1}^{K_{im}}$ representing multiple user interests via dynamic routing [22]. Unlike its application in computer vision, Capsule Networks for recommendations have only one routing layer and the bilinear mapping matrix S is shared among the capsules [3, 4, 12, 18], resembling a clustering procedure.

First, Capsule Networks instantiate potential cluster centroids c_j via Gaussian random initialization [12, 22]. At each routing step, the routing weights from e_i to c_j are derived as

$$b_{ij} \leftarrow \frac{exp(c_j^\top S e_i)}{\sum_k exp(c_k^\top S e_i)}, \tag{1}$$

and then update c_j as a weighted sum of all the item representations with closer items having higher contributions,

$$c_j \leftarrow squash(\sum_i b_{ij} S e_i) \tag{2}$$

with

$$squash(v) = \frac{||v||_2^2}{1 + ||v||_2^2} \frac{v}{||v||_2} \tag{3}$$

as a non-linear activation that enables the L2 norm of the vector to represent the existence of a cluster centroid. This procedure is repeated for several steps until cluster centroids converge.

3.2.2 Implicit Condition Construction. In this section, we describe the implicit interest condition construction method we adopt, namely **Differentiable Clustering Module (DCM)** as shown in Figure 3. It differs from MIND [12] in terms of the initialization and routing procedure, which we deem essential to clustering quality.

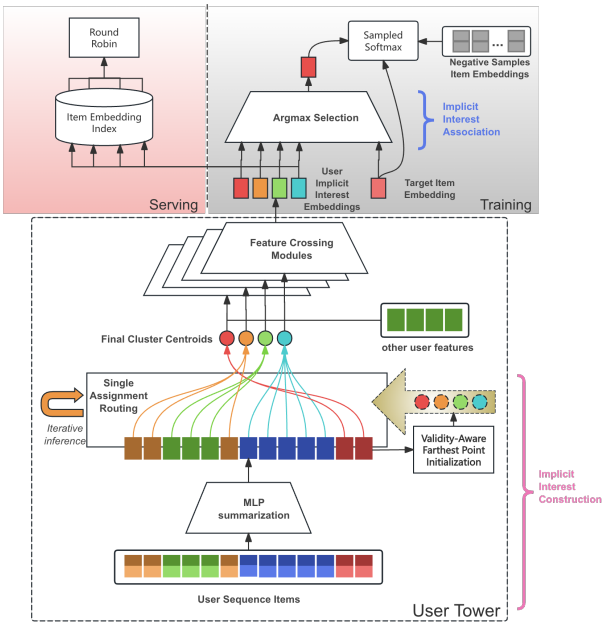

Figure 3: Implicit user interest modeling with the Differentiable Clustering Module. It differs from vanilla Capsule Network in the Validity-Aware Farthest Point Initialization and Single Assignment Routing to construct the implicit interest conditions. Condition association is performed via the argmax selection. We use these embeddings to retrieve candidates and do a round-robin merge at serving time.

At Pinterest, we use a set of features for user engaged items, including pretrained features [37] and categorical inputs. Different from MIND [12] that pools and linearly projects the inputs with S, we learn the item representation e_i from N input features f_{in} (n denotes the n-th input item level feature) via an MLP summarization layer activated by $GELU$ [7]:

$$e_i = W_2^\top(GELU(W_1^\top \cdot concat(f_{i1}, f_{i2}, \ldots, f_{iN}))) \tag{4}$$

where W_1 and W_2 are weights for a 2-layer MLP.

Initialization is known to be a crucial component for clustering algorithms to converge. We term our initialization heuristic as **Validity-Aware Farthest Point Initialization**, which ensures great coverage and diversity of the cluster centroids. Farthest Point Initialization (FPI) [1, 31] assumes that given m intialized cluster centroids $\{c_j\}_{j=1}^m$, we select the next cluster centroid iteratively as the item indexed by

$$i^* = \arg\min_i \max_j^m c_j^\top e_i. \tag{5}$$

We first randomly select an item e_i as the first cluster centroid and repeat the above procedure K_{im} times until we have all the cluster centroids. Equation 5 ensures that the initialized centroids are diversified as each iteration tries to minimize the maximum similarity between the next centroid and any given centroids.

However, it is important to incorporate validity filtering in the industry setup. In practice, certain input features significantly influence the output item embeddings. When these features are missing or deemed invalid, the resulting output embeddings can fall out of distribution. This misalignment critically affects our clustering algorithm, as the initialization process may incorrectly choose out-of-distribution items as cluster centroids. Without validity filtering, we observe significant regression on model performance and retrieval quality. With the validity filtering, Equation 5 can be rewritten with a validity indicator $\mathbf{I}_{valid}$ that masks out invalid items if they miss any feature inputs or indicate negative actions:

$$i^* = \arg\max_i \min_j^m -\mathbf{I}_{valid}(e_i) c_j^\top e_i \tag{6}$$

To further encourage the cluster centroids to diverge from each other after initialization, we incorporate the **Single-Assignment Routing** mechanism [18]. Recall the vanilla routing procedure in Equation 1 assigns each item e_i to every cluster centroid c_j by a non-zero routing weights b_{ij}. When two cluster centroids are in close proximity, nearby items may contribute similarly to both centroids, potentially causing them to converge even further over time. Here we describe the single-assignment routing procedure from Lu [18], which diverges the cluster centroids effectively by simply masking out non-maximum entries in the routing weights:

$$b_{ij} \leftarrow \mathbf{I}(c_j^\top e_i = \max_k c_k^\top e_i) \frac{exp(c_j^\top e_i)}{\sum_k exp(c_k^\top e_i)} \tag{7}$$

We provide extensive ablation study and visualization in Section 4.4 to show the importance of each component of DCM.

3.2.3 Implicit Condition Association. While the above sections describe condition construction for implicit user interests, we need to associate the target item to its corresponding interest. Given the j-th output user embedding $o_u^j = \phi(u, c_j)$ and the item embeddings $o_{y_i} = \psi(i)$, we achieve the condition association by our training objective, which involves one single user embedding that maximizes affinity to the target item in the sampled softmax loss [2, 36].

$$j^* = \arg\max_j {o_u^j}^\top o_{y_i} \tag{8}$$

$$\mathcal{L}_{SSM} = -log \frac{exp({o_u^{j^*}}^\top o_{y_i} - logp_{y_i})}{exp({o_u^{j^*}}^\top o_{y_i} - logp_i) + \sum_{k \in \mathcal{B}} exp({o_u^{j^*}}^\top o_{y_k} - logp_{y_k})} \tag{9}$$

where y_i is the target item, y_k represents in batch negative samples from $\mathcal{B}$, p_{y_i} and p_{y_k} represent streaming frequency estimation [36] for y_i and y_k. Different from UMI[4], we only use $o_u^{j^*}$ to compute the similarity for negative samples for computational efficiency, as the embedding layers and feature crossing modules are memory-intensive and occupy significant GPU resources.

3.2.4 Other Implicit User Interest Modeling Approaches. We also surveyed and explored a wide range of existing methods for implicit interest modeling at Pinterest, including self-attention [3, 10], interest token based [34], and PinnerFormer subsequence (PFS)[1] embeddings. We briefly introduce our implementation and share

[1]More details on PFS are included in the Appendix.

the empirical learnings from online results.

Condition Construction. As introduced in Section 2, we adopt the multi-head self-attention module [3] to construct the implicit interest conditions for the self-attention method. For interest token based approach, we omit the multi-task setup and remove the virtual kernel gating in MVKE [34], making the condition construction process equivalent to applying a single-head transformer decoder over the user sequence with learnable query interest tokens.

For PFS, we generate multiple user embeddings by splitting user sequences into subsequences of different topics, which are generated by clustering 10 million random Pins into $K = 32$ clusters using K-means on their Pinsage [37] embeddings. Then we assign sequence items to different topics based on their Pinsage similarity to the K cluster centroids and group them by their assigned topics. This results in at most K subsequences per user, which are then processed by the PinnerFormer [21] user sequence model to generate different user embeddings[2]. These embeddings can be used directly to retrieve candidates related to the topics, or as implicit conditions in a retrieval model. Empirically, using these subsequence embeddings as conditions performs better than using them for direct retrieval.

Condition Association. Unlike the Dynamic Routing approach, if we use the *argmax* operator to associate the positive item to conditions constructed by self-attention and interest token based approaches, the conditions will often collapse during training and the model will degenerate into a single-embedding retrieval model. We argue that when the interest condition representation involves random initialization, using a deterministic *argmax* for embedding selection induces a "winner-takes-all" effect, where only one embedding receives meaningful gradients and the shared parameters collapse all embeddings into near-identical representations. The Dynamic Routing approach, however, constructs the condition representation by clustering the user sequence. When coupled with the modifications we have on DCM, it results in more diverse conditions by construction and prevents the model from collapsing. To prevent embedding collapsing for self-attention and interest token based models, we use a Straight-Through Gumbel Softmax [8] to enable gradient flow to all implicit interest representations, which is the key to their convergence.

3.3 Explicit User Interest Modeling

In addition to the implicit user interests extracted from user engagement sequence, we incorporate explicit user signals, such as followed topics, into our framework. Similar to many social media platforms, Pinterest users can follow topics (e.g., food, pets, and women fashion) during sign-up, with the flexibility to add or remove them at any time. Followed topics serve as clear indicators of user interests, which can be leveraged in retrieval.

In this section, we describe how we leverage Conditional Retrieval (CR) [14] for explicit interest modeling. Similarly, CR can be broken down into condition construction and association steps. By leveraging conditional user engagement data, we can perform better at explicit interest modeling, as shown in the "Explicit Interest Association" path in Figure 4.

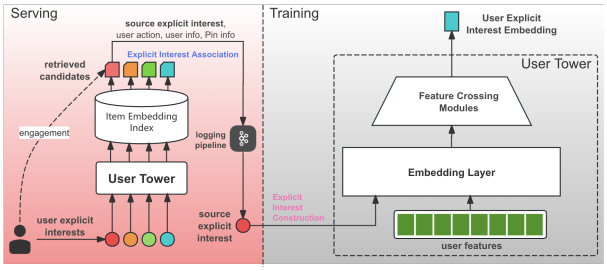

Figure 4: Explicit user interest modeling with Conditional Retrieval. The condition is constructed by embedding explicit signals (e.g., followed topics). Condition association is performed at serving time by tracking the source interests of engagements, which serve as the conditional inputs for training.

3.3.1 Explicit Condition Construction. We modify the user tower by incorporating the condition information directly into the embedding layer. The condition embedding is then processed through feature crossing layers, allowing the model to capture higher-order feature interactions between the user and the condition.

3.3.2 Explicit Condition Association. In Lin et al. [14], the goal is to retrieve items related to a given topic using only generic user-item engagement data. To bootstrap new use cases, they leverage existing item-to-topic signals and randomly sample a topic as the condition for the user tower. Essentially, the condition association is performed at training time. While effective for bootstrapping, this approach comes with a limitation: as one item can belong to multiple topics, the sampled topic may not overlap with users' followed topics or true intents, creating potential gaps between training and serving.

In the Pinterest home feed, we previously employed an inverted index retriever that fetches items based on users' followed topics. We collect user engagement data powered by this retriever and form training data (condition, user, item) tailored for explicit interest modeling. In other words, our condition association is performed at user action logging time, ensuring better alignment with users' followed topics. With the superior performance of CR, we subsequently replaced the inverted index retriever while keeping the same condition association process at logging time.

3.3.3 Explicit Relevance Filter. Ensuring condition relevance for explicit interest modeling is crucial for exposing users to their expected topics and improving the quality of the training data for explicit interest modeling. While CR already retrieves highly relevant items among embedding-based retrievers [14], we enhance this further by applying post-filtering based on item-to-topics signals as a guardrail for condition relevance, resulting in a considerable increase in engagements.

[2]More details in the Appendix.

3.4 Connections between Implicit and Explicit User Interest Modeling

Implicit and explicit interest models complement each other at retrieval stage. Implicit interests are learned from user engagement sequence. Due to challenges in serving super long sequences within the retrieval model, implicit interest modeling usually captures shorter term interests. Explicitly followed topics complement the implicit ones in the following ways: 1) They may provide additional long-term and/or long-tail interests. 2) They are especially useful for less active users with limited user sequences. 3) They are more controllable and explainable, making it easy to inject heuristics to the retrieval pipeline. In practice, the average overlap of candidates retrieved from explicit and implicit interests conditioned user embeddings is **only 3.2%**, indicating strong complementarity within our framework.

In terms of modeling techniques, they share the same philosophy but differ in implementation. The crux of the two models lies in two critical components: condition construction and accurate condition association. In terms of condition construction, the implicit interest modeling utilizes DCM to extract conditional signals through the clustering of user sequences. In contrast, explicit interest modeling involves the straightforward embedding of auxiliary input conditions. Regarding condition association, implicit interest modeling achieves this internally through the application of the argmax operator, while explicit interest modeling accomplishes this during the data logging phase. Both models effectively reframe the ranking score from $f(i|u)$ to $f(i|u, c)$.

3.5 Deployment

For implicit interest modeling, we infer K_{im} implicit user interests from the user engagement sequence. The corresponding user embeddings are used to perform ANN search from a per-computed item index. We assign different retrieval budgets based on the sum of their cluster routing weights $\sum_i b_{ij}$, interpreted as cluster importance. This prevents over-fetching candidates from torso and tail interests that deteriorate online performance. For explicit interest modeling, we randomly sample K_{ex} followed topics and assign equal retrieval budgets for each conditional user embedding.

The retrieved candidates for each user embedding are then merged using a round-robin strategy with deduplication, ensuring a balanced mix of candidates from different interest conditions. This approach prevents over-representation of any single interest and delays ranking decisions to later stages for more refined personalization.

Since each user tower is only computed once per request, the increased model complexity is acceptable for online serving. Empirically, p90 latency for the embedding-based retrieval increases from 150ms to 205ms, remaining below responsiveness limits. In addition, to balance the serving cost and user engagements, we also reduce the number of candidates to retrieve per embedding and keep an on-par budget as the single embedding retrieval.

4 Experiments

In this section, we present our extensive experiments on the proposed multi-embedding retrieval framework, including comparisons and ablations of implicit and explicit interest modeling through both offline and online experiments. Additionally, we provide a visual case study to analyze the effectiveness of each component.

4.1 Experiment Setup

4.1.1 Dataset and Metrics. Following previous work [32], we compile the training dataset from a 15-day window of user engagement logs, using the first 14-day window for training and 15th day for evaluation. The dataset contains 6 billion engagement records from 160 million users. For retrieval tasks, we formulate a composite prediction task that incorporates multiple positive actions, such as clicks, repins (saves), etc. Label sampling and weighting are applied to align with business objectives, but we omit the details for brevity to maintain the paper's focus.

For offline evaluation, we construct a retrieval corpus with 1 million of the most engaged items from the dataset. Each record in the evaluation dataset contains a single positive item to retrieve, and we measure performance using hit rate (**HR**) at various ranking thresholds. To compute the score of an item with multiple embeddings, we follow previous work [3, 12] to take the maximum score of each item among all user embeddings. In practice, since the positive item may not always be present in the constructed corpus, we compute HR by comparing the score of the positive item against the ranked items within the corpus.

For online metrics, we report relative improvements on: (1) **HF Repins**, which is the repin (save) volume on home feed and serves as one of the most important metrics for this surface [32], and (2) **Adopted Pincepts (A-Pincepts)**[3] as the online engagement diversity indicator. We also report them across user segments (core[4] and non-core users).

4.1.2 Implementation Details. All models in the experiments are trained on the same dataset using sampled softmax loss with logQ correction [36] over the in-batch negatives. For both implicit and explicit interest modeling, the feature crossing layer in Figure 3 and Figure 4 is a DHEN [38] module that ensembles Transformers [26], Parallel Mask Net [29], and MLPs.

In online comparison experiments, we fetch the same number of candidates and merge the candidates with round-robin before passing them to the ranking model, unless otherwise specified. We set K_{im} = 7, K_{ex} = 5, and the overall retrieval budgets for implicit and explicit models are $O(1k)$.

4.1.3 Competitor Methods. Since our framework consists of two models, we conduct separate comparisons for implicit and explicit interest modeling against existing methods. For implicit interest modeling, we evaluate DCM against representative multi-interest retrieval methods from different modeling streams, as described in Section 3.2.4, including MIND [12], self-attention [10], interest token-based [34], and our in-house PinnerFormer Subsequence (PFS) based models. For explicit interest modeling, we report its performance comparison against inverted index-based retrieval, a widely adopted industrial solution for relevance-oriented retrieval, as well as CR with conditions directly extracted from item attributes [14].

[3] Pincepts is an internal annotation system to identify fine-grained interests for items. Adoption is defined as having 3 or more qualified actions in a week on a Pincept.
[4] We define users with >4 repins in 28 days as core users.

Table 1: Offline evaluation for implicit interest modeling.

Method	HR@100	HR@1000
Self-Attention [3, 10]	0.167	0.470
Interest Token [34]	0.180	0.474
MIND [12]	0.175	0.464
DCM	**0.185**	**0.476**

Table 2: Offline evaluation for explicit interest modeling.

Method	filtered HR@100	filtered HR@1000	HR@100
CR w/ item interest	0.164	0.541	0.139
CR w/ source interest	**0.191**	**0.565**	**0.145**

Table 3: Online comparison for explicit interest modeling. Colored numbers (blue or red) are statistically significant and gray numbers are non-statistically-significant.

control	inverted index	inverted index	item interest
enabled	CR w/ filter	CR w/o filter	source interest
HF Repins ↑	+0.56%	+0.3%	+0.98%
NC-HF Repins ↑	+1.13%	+0.46%	+3.04%
A-Pincepts ↑	+0.42%	+0.37%	+0.32%
NC-A-Pincepts ↑	+0.44%	+0.42%	+1.03%

4.2 Offline Evaluation

In this section, we report the offline comparison results for implicit and explicit interest modeling. Table 1 presents the offline results for implicit interest modeling, where DCM outperforms the other methods in both HR@100 and HR@1000. The comparison between DCM and MIND [12] also validates that the modifications upon Capsule Networks [22] in DCM improve the retrieval quality.

Offline results of differently trained Conditional Retrieval for explicit interest modeling are shown in Table 2. We use different condition association strategies that either directly extract interest conditions from item attributes [14] (termed as **CR w/ item interest**) or perform condition association at the time of action logging (termed as **CR w/ source interest**). Since explicit interest modeling applies term-based post-filtering at serving time, we replicate this step offline and report **filtered HR** to approximate the online treatment. Additionally, we include HR without filtering for completeness. CR w/ source interest outperforms CR w/ item interest across all metrics, with a particularly strong advantage when filters are applied.

4.3 Online Experiments

Given the well-known empirical challenge of offline-online metric discrepancy [11] especially for industrial retrieval models, we make design choices mainly based on online A/B testing for our framework. We obtain these metrics from experiments over a 2-week period.

We conduct online experiments on modeling and serving variants for explicit interest modeling in Table 3. We report metrics for all users and non-core users to showcase its strength on non-core users. We attach an "NC-" prefix in Table 3 to indicate metrics over non-core users. Applying post-filtering for Conditional Retrieval (CR w/ filter) performs the best among other modeling or serving variants. When comparing against the same model without filtering (CR w/o filter), the notable gains demonstrate that enforcing interest relevance is helpful for this component, even when explicit context relevance is not a must for home feed. We also compare using the item interest from item attributes with the source interest from action logging as conditions. There is an increasing trend in all metrics if we use source interest, especially on HF Repins. It shows that appropriate condition association is crucial to improve engagements effectively. In addition, all metrics on non-core users are amplified, indicating that explicit interest modeling is adept at non-core users.

Table 4: Online comparison for implicit interest modeling.

Methods	HF Repins ↑		A-Pincepts ↑	
	all	core	all	core
Self-Attention [3, 10]	+0.83%	+1.01%	+0.44%	+0.63%
Interest Token [34]	+0.68%	+0.95%	+0.21%	+0.19%
MIND [12]	+0.43%	+0.56%	+0.04%	+0.15%
PFS	+0.47%	+1.02%	+0.32%	+0.46%
DCM	**+0.86%**	**+1.23%**	**+0.46%**	**+0.87%**

Table 5: Online lift of the proposed framework.

Control	Sitewide Repins ↑	HF Repins ↑	A-Pincepts ↑
+ Framework	**+0.48%**	**+1.09%**	**+0.81%**

Table 6: Ablation Study on DCM Model Architecture.

VA-FPI	SAR	#Cluster	HR@100	HF Repins ↑	A-Pincept ↑
×	×	7	0.175	+0.43%	+0.04%
×	✓	7	0.176	+0.37%	+0.30%
✓	×	7	0.175	-0.10%	+0.17%
✓	✓	2	0.176	-0.53%	+0.25%
✓	✓	4	0.180	-0.15%	+0.29%
✓	✓	7	**0.185**	**+0.86%**	**+0.46%**

We then present the results for implicit interest modeling in Table 4. It is worth noting that these experiments are conducted with post-filtered CR launched to production. Thus, the metric gains in Table 4 show a complementary effect of implicit interest-based retrieval. Consistent with offline results, DCM outperforms other methods in both HF Repins and A-Pincepts, while the self-attention-based method performs comparably. Notably, MIND [12] relatively underperforms compared to other approaches without DCM's modifications, further validating the effectiveness of centroid divergence in improving representation quality. We also report metrics for core users, as implicit interest modeling tends to benefit active users more due to its reliance on user sequence modeling. All methods show greater improvements on core users. Additionally, diversity metrics such as A-Pincepts exhibit strong correlation with HF Repins, reinforcing that improving diversity and user interest coverage at the retrieval stage benefits the overall system.

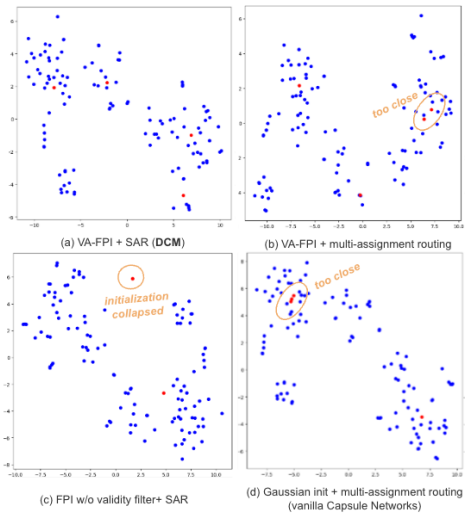

Figure 5: Comparison between DCM and Capsule Networks with 4 clusters. These embeddings are sampled from the same user and visualized with t-SNE, embeddings may locate differently across figures due to random projection.

From the above discussion, we can see that implicit and explicit interest modeling show a complementary effect on different user segments. To measure the joint effect of the framework, we run a retrospective experiment and report the overall gains achieved by our framework in Table 5. Synergizing implicit and explicit interests increases feed diversity and user engagements by a large margin, including sitewide repins, which demonstrates cascading impact beyond the directly applied surface.

4.4 Ablation Study

We run extensive ablation studies both offline and online to verify the design of DCM, including Validity Aware Farthest Point Initialization (VA-FPI), Single Assignment Routing (SAR) [18], and number of clusters. As shown in Table 6, the joint effect of VA-FPI and SAR on enhancing the cluster centroid divergence improves model capability and online metrics significantly. We also visualize the clustering procedure inside the neural network as shown in Figure 5. We take the PinSage [37] embeddings from the user sequence and extract 4 cluster centroids to inspect the clustering process. Without validity filtering, some cluster centoids will collapse to one point as shown in Figure 5(c). From Figure 5(a) to (d), by removing SAR [18] and VA-FPI, the cluster centroids are getting more concentrated, indicating suboptimal diversity to capture user interests holistically.

We also study the impact of different number of clusters. As shown in Table 6, less number of clusters significantly deteriorates online performance, and even decreases metrics when there are only 2 clusters. This is possibly due to missing major interest coverage with less number of clusters. Similar to MIND [12], we try to reduce number of clusters adaptively for different users, but results in a -0.12% change in HF Repins compared with serving a fixed number of clusters. In addition, taking the top ranked candidates among all embeddings [3, 12] is also experimented online, but leading to a marginal decrease in the number of repin users by 0.35%. Given these results, we finalize our design choice as outlined in Section 3.

4.5 Case Study

In Figure 6, we present a case study based on a randomly selected user. For privacy reasons, we iconize the original images in the user history and recommendation feed to obscure identifiable user information. The user's previously saved Pins are shown in the top right corner, and two example Pins retrieved from each embedding source are displayed separately for explicit interest modeling and implicit interest modeling. Note that this is the same user as in Figure 1 where we have shown that single embedding retrieval fails to cover the user's interests holistically. Results from explicit interest modeling (CR) are labeled with the source interest provided to the model, while results from implicit interest modeling (DCM) are manually annotated by the Pins' common interest taxonomy. The case clearly illustrates that the interests reflected in the user's saved Pins are well covered by our framework, and the retrieved candidates exhibit high relevance and personalization. Notably, while DCM overlooks the interest in "education", it is successfully recovered by CR, showcasing the supplementary effect of the components in our framework. In addition, DCM shows personalized interest granularity beyond human-defined interest taxonomy. For example, "beauty" related Pins from DCM range from make-up to cosmetic organization (coarse-grained), while "photography" Pins are more fine-grained with a common theme of "candid youthful moments and lifestyle".

5 Conclusion

In this paper, we introduce the multi-embedding retrieval framework at Pinterest, which synergizes Differentiable Clustering Module (DCM) for implicit interest modeling and Conditional Retrieval (CR) for explicit interest retrieval to capture diverse user interests and enhance user engagements. We frame both models as conditional representation learning, introducing distinct mechanisms for condition construction and association to better encode and leverage user interests as conditions. Our extensive offline and online experiments, including A/B testing at Pinterest, demonstrate significant improvements in user engagement and content diversity, validating the effectiveness of our framework in a real-world production environment.

Acknowledgments

We would like to say thank you to our colleagues - Jay Adams, Raymond Hsu, and Dylan Wang for the support and suggestions on this work.

References

[1] David Arthur and Sergei Vassilvitskii. 2006. *k-means++: The advantages of careful seeding.* Technical Report. Stanford.
[2] Yoshua Bengio and Jean-Sébastien Senécal. 2003. Quick training of probabilistic neural nets by importance sampling. In *International Workshop on Artificial Intelligence and Statistics.* PMLR, 17–24.
[3] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings*

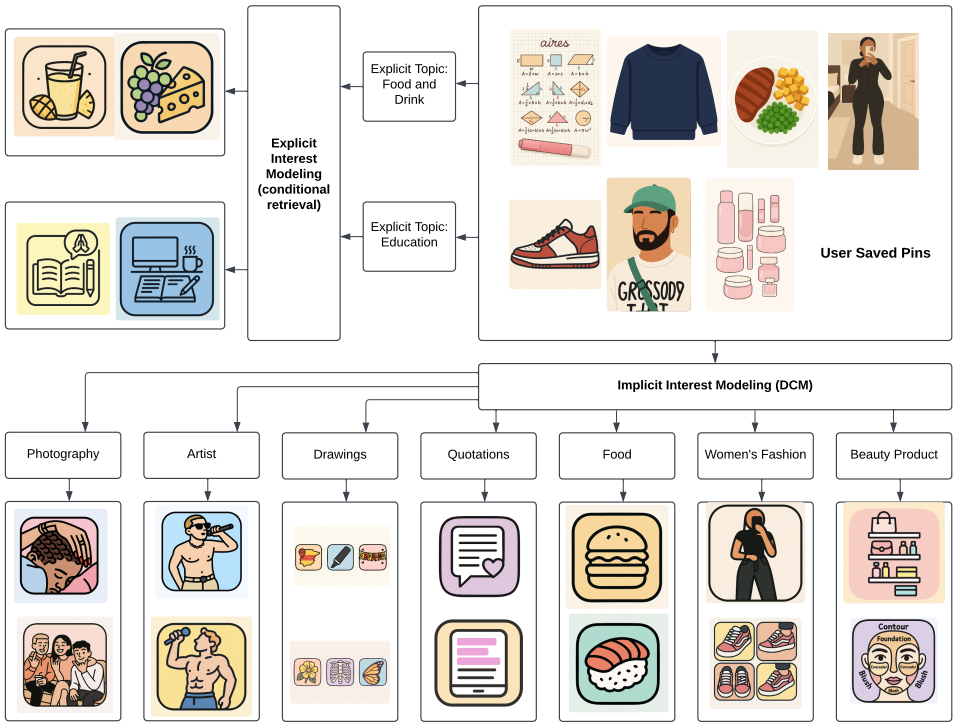

Figure 6: Case study of a Pinterest user. For privacy reasons, we iconize the original images in the user history and recommendation feed to obscure identifiable user information. We show the user's saved Pins and recommendations from implicit interest modeling and explicit interest modeling separately, with each box containing 2 examples for one embedding. Candidates from implicit interest modeling are manually annotated with a common interest term within the retrieval set, while candidates from explicit interest modeling are labeled with the source interest.

of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2942–2951.

[4] Zheng Chai, Zhihong Chen, Chenliang Li, Rong Xiao, Houyi Li, Jiawei Wu, Jingxu Chen, and Haihong Tang. 2022. User-aware multi-interest learning for candidate matching in recommenders. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1326–1335.

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

[6] Shangfeng Dai, Haobin Lin, Zhichen Zhao, Jianying Lin, Honghuan Wu, Zhe Wang, Sen Yang, and Ji Liu. 2021. POSO: personalized cold start modules for large-scale recommender systems. *arXiv preprint arXiv:2108.04690* (2021).

[7] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

[8] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[9] Yushi Jing, David Liu, Dmitry Kislyuk, Andrew Zhai, Jiajing Xu, Jeff Donahue, and Sarah Tavel. 2015. Visual search at pinterest. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1889–1898.

[10] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[11] Karl Krauth, Sarah Dean, Alex Zhao, Wenshuo Guo, Mihaela Curmei, Benjamin Recht, and Michael I Jordan. 2020. Do offline metrics predict online performance in recommender systems? *arXiv preprint arXiv:2011.07931* (2020).

[12] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2615–2623.

[13] Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Mingjie Li, Wenjie Zhang, and Xuemin Lin. 2019. Approximate nearest neighbor search on high dimensional data—experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering* 32, 8 (2019), 1475–1488.

[14] Hongtao Lin, Haoyu Chen, Jaewon Yang, and Jiajing Xu. 2024. Bootstrapping Conditional Retrieval for User-to-Item Recommendations. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 755–757.

[15] Chi Liu, Jiangxia Cao, Rui Huang, Kuo Cai, Weifeng Ding, Qiang Luo, Kun Gai, and Guorui Zhou. 2024. CRM: Retrieval Model with Controllable Condition. *arXiv preprint arXiv:2412.13844* (2024).

[16] Chi Liu, Jiangxia Cao, Rui Huang, Kai Zheng, Qiang Luo, Kun Gai, and Guorui Zhou. 2024. KuaiFormer: Transformer-Based Retrieval at Kuaishou. *arXiv preprint arXiv:2411.10057* (2024).

[17] David C Liu, Stephanie Rogers, Raymond Shiau, Dmitry Kislyuk, Kevin C Ma, Zhigang Zhong, Jenny Liu, and Yushi Jing. 2017. Related pins at pinterest: The evolution of a real-world recommender system. In *Proceedings of the 26th international conference on world wide web companion*. 583–592.

[18] Sitong Lu. 2024. 360 多兴趣召回 MIND 实战优化 [Multi-interest retrieval at 360: MIND practical optimization]. https://mp.weixin.qq.com/s/Hy9yZ8yOF2FwQ9FIn3DQTw Accessed: Dec. 30, 2024.

[19] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *Advances in neural information processing systems* 32 (2019).

[20] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018), 824–836.

[21] Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. 2022. Pinnerformer: Sequence modeling for user representation at pinterest. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 3702–3712.

[22] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *Advances in neural information processing systems* 30 (2017).

[23] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. Learning semantic representations using convolutional neural networks for web

search. In *Proceedings of the 23rd international conference on world wide web*. 373–374.
[24] Yi Sun and Yuri M Brovman. 2024. CoActionGraphRec: Sequential Multi-Interest Recommendations Using Co-Action Graphs. *arXiv preprint arXiv:2410.11464* (2024).
[25] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*. 598–606.
[26] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).
[27] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence.
[28] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1001–1010.
[29] Zhiqiang Wang, Qingyun She, and Junlin Zhang. 2021. Masknet: Introducing feature-wise multiplication to CTR ranking models by instance-guided mask. *arXiv preprint arXiv:2102.07619* (2021).
[30] Zhikai Wang and Yanyan Shen. 2023. Incremental learning for multi-interest sequential recommendation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 1071–1083.
[31] Lemeng Wu, Xingchao Liu, and Qiang Liu. 2021. Centroid transformers: Learning to abstract with attention. *arXiv preprint arXiv:2102.08606* (2021).
[32] Xue Xia, Pong Eksombatchai, Nikil Pancha, Dhruvil Deven Badani, Po-Wei Wang, Neng Gu, Saurabh Vishwas Joshi, Nazanin Farahpour, Zhiyuan Zhang, and Andrew Zhai. 2023. Transact: Transformer-based realtime user action model for recommendation at pinterest. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5249–5259.
[33] Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep multi-interest network for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2265–2268.
[34] Zhenhui Xu, Meng Zhao, Liqun Liu, Lei Xiao, Xiaopeng Zhang, and Bifeng Zhang. 2022. Mixture of virtual-kernel experts for multi-objective user profile modeling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4257–4267.
[35] Jing Yan, Liu Jiang, Jianfei Cui, Zhichen Zhao, Xingyan Bin, Feng Zhang, and Zuotao Liu. 2024. Trinity: Syncretizing Multi-/Long-Tail/Long-Term Interests All in One. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6095–6104.
[36] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM conference on recommender systems*. 269–277.
[37] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.
[38] Buyun Zhang, Liang Luo, Xi Liu, Jay Li, Zeliang Chen, Weilin Zhang, Xiaohan Wei, Yuchen Hao, Michael Tsang, Wenjun Wang, et al. 2022. DHEN: A deep and hierarchical ensemble network for large-scale click-through rate prediction. *arXiv preprint arXiv:2203.11014* (2022).

A Feature Crossing Module

While we illustrate the condition construction and association for implicit and explicit interest modeling in details previously, here we elaborate the detailed architecture of the feature crossing modules in our retrieval model. Both implicit and explicit interest modeling share the same architecture in each tower.

Our feature crossing module is based on DHEN[38], an ensembling framework that can parallelize and stack arbitrary submodules for latent feature interactions. We use a DHEN with two hierarchies:

- 1st hierarchy: a 2-layer Transformer encoder[26] (256 hidden dims, 4 heads), and a 2-layer MLP (1024 hidden dims) in parallel
- 2nd hierarchy: a 4-block parallel mask net[29] (128 hidden dims, 0.5 projection ratio), and a 2-layer MLP (1024 hidden dims) in parallel

The output of the parallel submodules in each hierarchy are summed and feed into the next layer. The input features are splitted and projected into feature fields with equal dimensions to facilitate field-wise feature interaction in a transformer, while the MaskNet and MLPs take in the concatenated feature fields.

B PinnerFormer Subsequnce (PFS)

Here we elaborate the implementation of our in-house PinnerFormer[21] Subsequence (PFS) model as an alternative implicit interest modeling approach that we explored. The core idea is, rather than taking in a full user history sequence into the model, PFS takes in a subsequence that belongs to a similar concept and generate an embedding given each subsequence for retrieval. We first collect a set of $O(1M)$ items in an offline workflow and run K-means clustering based on PinSage[37] to get 32 distinct concept clusters. For the items in each user history sequence, they are then mapped into these 32 clusters and splitted into subsequences accordingly. At training time, the model is trained using the same DenseAllAction[21] loss except that the input subsequence corresponds to the target item's concept cluster.

While these embeddings generated from PFS can be directly used to serve as a retrieval model, however, we found that empirically serving a two-tower model with these subsequence embeddings as input performs better. Due to the serving complexity for the deep Transformer architecture of PFS, we populate the PFS outputs as a user signal that can be taken in as a feature for the two-tower models. The two-tower model is trained in the same way as described in Section3.2, except that the DCM outputs being replaced with these PFS embeddings. We hypothesize the reason DCM performs better than PFS is due to (1) the clustering of PFS is not end-to-end trained such that the embeddings for the items cannot adapt to the model's training objectives, (2) the signal is not consumed in a real-time manner, and (3) the clusters are pre-defined concepts, whereas the cluster granularity in DCM can be adaptive to user history.

C Non-core User Performance Breakdown

Here we provide a detailed breakdown in Table7 of non-core user performance lift by the explicit interest modeling to justify its effectiveness across non-core user cohorts. Note that new users only take up less than 10% of the total non-core user segments, therefore the results here mainly illustrate the positive trend of user experience as they do not surpass a 0.05 p-value threshold due to the small population. These results provide a granular view of the real-world impact on non-core users and demonstrate the effectiveness of explicit interest modeling on users of less activities.

Table 7: Detailed breakdown of explicit interest modeling across non-core user cohorts.

User cohort	Casual	Marginal	Resurrected	New
HF Repins	+1.12%	+1.46%	+1.00%	+0.56%*
A-Pincepts	+0.23%	+0.05%	+0.67%	+0.65%*