ESANS: Effective and Semantic-Aware Negative Sampling for Large-Scale Retrieval Systems

Anonymous Author(s)*

Abstract

Industrial recommendation systems typically involve a two-stage process: retrieval and ranking, which aims to match users with millions of items. In the retrieval stage, classic embedding-based retrieval (EBR) methods depend on effective negative sampling techniques to enhance both performance and efficiency. However, existing techniques often suffer from false negatives, high cost for sampling quality and semantic information deficiency. To address these limitations, we propose Effective and Semantic-Aware Negative Sampling (ESANS), which integrates two key components: Effective Dense Interpolation Strategy (EDIS) and Multimodal Semantic-Aware Clustering (MSAC). EDIS generates virtual samples within the low-dimensional embedding space to improve the diversity and density of the sampling distribution while minimizing computational costs. MSAC refines the negative sampling distribution by hierarchically clustering item representations based on multimodal information (visual, textual, behavioral), ensuring semantic consistency and reducing false negatives. Extensive offline and online experiments demonstrate the superior efficiency and performance of ESANS.

CCS Concepts

• **Information systems** → **Retrieval models and ranking**.

Keywords

Recommendation systems, Embedding-based retrieval, Negative sampling

ACM Reference Format:

Anonymous Author(s). 2024. ESANS: Effective and Semantic-Aware Negative Sampling for Large-Scale Retrieval Systems. In *Proceedings of the 2025 International World Wide Web Conference (WWW '25).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Recommendation systems have been widely adopted across diverse domains, including online e-commerce, advertising, short video platforms and delivery services [16, 17, 59], due to their effectiveness in mitigating information overload by providing tailored recommendations from large-scale item collections [18, 19]. Industrial recommendation systems typically involve two stages: retrieval and ranking. The retrieval stage is responsible for retrieving thousands of candidate items, whereas the ranking stage predicts the likelihood of user interaction with these candidates. Considering that retrieval tasks can be formulated as identifying the nearest neighbors in a vector space, substantial research has been devoted to developing high-quality representations for both users and items. Collaborative Filtering (CF) methods [8, 24, 42, 45] address this issue by encoding user preference and item representation into low-dimensional embedding space, based on historical interacted information. With the rapid development of deep learning, neural networks have been widely adopted in personalized recommendation systems [5, 20, 55]. Recently, Embedding-Based Retrieval (EBR) methods [3, 12, 30] have demonstrated significantly better performance compared to traditional CF methods, establishing themselves as the dominant approach in recommendation systems. EBR methods encode user and item information into separate embeddings using parallel neural networks, and these embeddings are trained through the strategy of contrastive learning [15, 36, 44].

EBR methods rely heavily on the contrast between positive and negative samples to produce distinguishable representations. The careful selection of negatives is crucial to enhancing the model's ability to differentiate between relevant and irrelevant items, significantly impacting overall retrieval performance. The classic Uniform

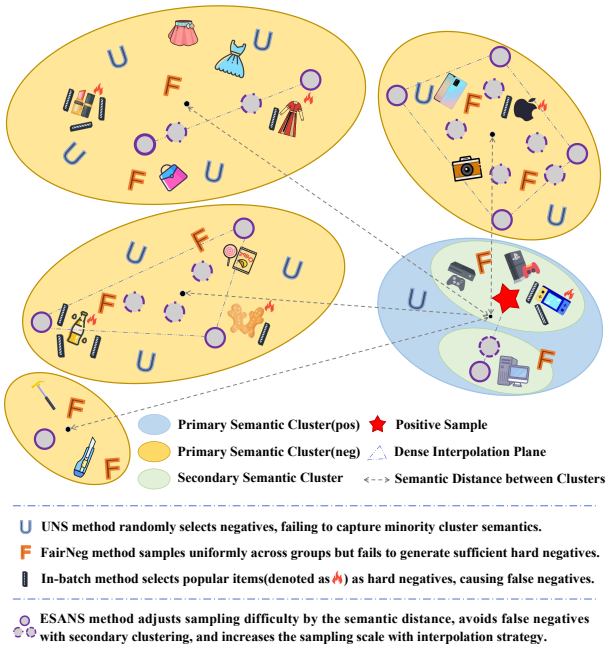

Figure 1: Visual diagram of our ESANS compared with other methods. Each method has sampled ten negatives equally.

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

WWW '25, April 28 - May 02, 2025, Sydney, Australia

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Negative Sampling (UNS) method [23, 44] randomly selects negatives from the item candidate set, providing efficiency but yielding **low-quality samples**. Following this, additive margin [50] and temperature coefficient [33, 51] adjust the contrastive loss function to mine high-quality negatives from naive negatives sampled by UNS. FairNeg [9] reweights negatives in accordence with item group fairness to provide high-quality samples. Adap-τ [4] dynamically adjusts the temperature coefficient for reweighting uniform negatives in accordence with their relevance to user interests. However, these methods fail to introduce more challenging negatives and further expand the scale of sampling which limit the performance of EBR methods. To address this issue, In-batch sampling [7] introduces relatively harder negatives by the in-batch sharing strategy. Airbnb [21] heuristically introduces orders from the same city as harder negatives. MixGCF [27] employs a hop-mixing interpolation technique in Graph Neural Networks (GNNs) to generate virtual hard negatives. However, these methods fail to effectively adjust the difficulty of negatives and distinguish users' potential interests from hard negatives, which may exacerbate the issue of **false negatives** (i.e. items relevant to users' potential interests but incorrectly regarded as negatives). Moreover, existing methods require substantial computational resources to further **improve the sampling quality** (i.e. sufficient hard negatives) [6]. Furthermore, from a contrastive learning perspective, these methods are unable to regulate sampling strategies based on **semantic information** in the real world, rendering the sampling process a black box.

Inspired by recent works in multi-modal learning [35, 40] and vector quantization techniques [47], we propose the **E**ffective and **S**emantical-**A**ware **N**egative **S**ampling (ESANS) to address these challenges in the sampling process. Our method consists of two main components: the first part is Effective Dense Interpolation Strategy (EDIS), and the second part is Multimodal Semantic-Aware Clustering (MSAC). EDIS is devised to generate a sufficient number of virtual samples within the low-dimensional embedding space. More specifically, generating virtual samples among existing negatives creates a more uniform, dense, and diverse sampling distribution. Virtual samples positioned between the positive sample and surrounding negative samples contribute to gradually enhance the discriminatory ability of the neural network. By adjusting the interpolation parameters and strategies, we can control the difficulty of generated negatives and generate sufficient hard negatives. Meanwhile, in contrast to memory banks [22], interpolation within the low-dimensional embedding space leads to minimal computational cost and eliminates the need for extra memory storage.

Nevertheless, EDIS strongly relies on the judicious selection of negative sample anchors. In practice, virtual negative samples generated via interpolation may lack clear semantic information, occasionally producing meaningless samples. For example, interpolating between "iPhone" and "Cola" produces meaningless results, potentially introducing noise. Moreover, interpolating among randomly sampled negative anchors may introduce false negatives, further complicating the training process.

To address these deficiencies, we propose the MSAC method to optimize the sampling space by integrating the real-world semantic information. Firstly, we propose a multimodal-aligned technique to fuse multi-perspective item information from visual, textual and behavioral perspectives. Subsequently, a two-level vector quantized clustering approach is employed to assign semantic representations into multiple secondary clusters. Consequently, we can mitigate the issue of false negatives by selecting hard negatives from the same primary cluster as the positive sample, while ensuring they belong to a different secondary cluster. Additionally, we dynamically calibrate the sampling probabilities for each negative cluster to control the difficulty of negatives and refine the sampling quality. It is worth noting that this calibration is precisely guided by the semantic distance between the cluster centers of positives and negatives. This allows us to adjust the difficulty of the sampling process by increasing the sampling probabilities of clusters that are semantically similar to the positive cluster. Once the MSAC is introduced, EDIS based on semantics can be performed within the well-established semantic clusters. More specifically, we can ensure that the interpolated outcomes remain confined within the convex hull of that cluster. This intrinsic constraint preserves a measurable degree of semantic consistency and **"real-world applicability"** in the interpolated samples. Furthermore, interpolation between positives and hard negatives is also employed to generate additional high-quality hard negatives. Figure 1 shows the comparison between our ESANS and other methods. Our contributions can be summarized as follows:

- We propose a novel and effective sampling approach called ESANS, which provides explicit semantics guidance for interpolation negative sampling. Moreover, ESANS effectively enhances the diversity and richness of negative samples and allows for controllable negative sample difficulty, thereby boosting performance.
- We propose a general multimodal-aligned clustering approach that captures the multi-perspective similarities among candidate items on e-commerce platforms, thereby enabling a more refined semantic description in the interpolation space and eliminating false negative instances in the hard negative sampling process.
- We provide both extensive offline and online experiments to demonstrate the effectiveness and the efficiency of ESANS.

2 Related Work

This section presents a brief review of the relevant literature, specifically addressing techniques for negatives re-weighting, heuristic negative sampling, and model-based negative sampling.

Negatives Re-weighting. UNS [23, 44] represents the foundational negative sampling method, where negative samples are uniformly drawn from the entire dataset. The simplicity of UNS's algorithmic design provides substantial efficiency gains. Nevertheless, it exhibits notable deficiencies in the quality of negative samples. UMA2 [33] computes the sampling probabilities of random negative samples according to the current model and subsequently employs the Inverse Probability Weighting (IPW) technique to assign loss weights to these negative samples. The method proposed by [43] implements position-weighted approach for negative samples, where the weight is determined by the sample's ranking position. These approaches mine high-quality negatives from naive negatives sampled by UNS, which fails to introduce more challenging negatives.

Heuristic Negative Sampling. Heuristic negative sampling algorithms primarily define the sampling distribution by predefined

heuristic rules. Popularity-biased Negative Sampling (PNS) [7] utilizes item popularity as the sampling probability. Airbnb [21] applies personalized negative sampling within the same city, assuming bookings in the same location exhibit similar patterns. While this approach enhances the sampling process, it solely focuses on similarity-based sampling, neglecting sampling bias. CBNS [54] employs in-batch negative sampling and expands the negative sample set by incorporating previously trained items. The method [57] incorporates estimated item frequency into the batch softmax cross-entropy loss to reduce sampling bias within the batch. MNS [56] integrates UNS with in-batch negative sampling, adopting a hybrid strategy. While these methods enhance sampling quality, they introduce popularity bias, aggravating the Sample Selection Bias (SSB) issue. Our method enhances sampling quality via a multimodal-aligned clustering algorithm and dense interpolation negative sampling, while also mitigating sampling bias.

Model-based Negative Sampling. Model-based negative sampling algorithms are highly effective at selecting high-quality negative samples. Model-based scoring methods are demonstrated by Dynamically Negative Sampling (DNS) [58] and ESAM [10], where the current model scores samples and selects the highest-scoring ones as negative samples. Adversarial learning methods also contribute to sampling improvements. MixGCF [27] employs a hop-mixing technique to synthesize hard negative samples by leveraging the user-item graph structure and the aggregation mechanism of Graph Neural Networks (GNNs). IRGAN [53] utilizes two recommendation models, a discriminator and a generator, trained adversarially. AdvIR [38] and RNS [13] further optimize IRGAN's structure, improving both efficiency and performance. The Adap-τ [4] adaptively adjusts the temperature coefficient of the loss function by calculating the loss for each user and the corresponding random negative samples. This method leverages personalized user preferences to effectively identify hard negative samples. FairNeg [9] enhances the sampling distribution by fairly sampling from groups and then reweighting based on their relevance to the user. Our method precisely controls the difficulty of negatives, improving sampling quality and eliminating false negatives without increasing the complexity of the retrieval model.

3 Methodology

In this section, we formulate the problem and describe our proposed framework specifically, as well as introducing the detailed process of our negative sampling method.

3.1 Problem Formulation

The primary objective of the retrieval stage in industrial recommendation systems is to efficiently retrieve a potentially relevant subset of items from a large item pool $\mathcal{I}$ for each user $u \in \mathcal{U}$. In pursuit of this objective, each instance can be represented by a tuple $(\mathcal{B}_u, \mathcal{P}_u, \mathcal{I}_i)$ where $\mathcal{B}_u$ denotes the sequence of user historical behaviors, $\mathcal{P}_u$ denotes the basic profile of user u, $\mathcal{I}_i$ denotes the information of target item such as item id and category id. In the classical two-tower architecture [50] of the EBR models, users and items are separated into two individual encoders to reduce online computational complexity. We can define the user encoder as f_{user} and the item encoder as g_{item}, so we have:

$$\begin{aligned} \mathbf{u}_u &= f_{user}(\mathcal{B}_u, \mathcal{P}_u) \\ \mathbf{v}_i &= g_{item}(\mathcal{I}_i) \end{aligned} \tag{1}$$

where $\mathbf{u}_u \in \mathbb{R}^{d_k \times 1}$ is the output vector of the user encoder called user embedding, and $\mathbf{v}_i \in \mathbb{R}^{d_k \times 1}$ is the output vector of the item encoder called item embedding. K denotes the dimension of output embeddings. Finally, the relevance of a user-item pair can be estimated by a scoring function:

$$s(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v} \tag{2}$$

3.2 Overall Framework

As previously discussed, existing methods fail to balance sampling quality, bias, and efficiency simultaneously. To address these limitations, we designed ESANS, as illustrated in Figure 2. ESANS consists of two main components:

- **Multimodal Semantic-Aware Clustering(MSAC)**, which performs hierarchical clustering based on visual, textual, and behavior based representations to optimize the sampling process by integrating semantic information. Our proposed method addresses the limitations of unclear anchor semantics, improves sampling quality, and reduces the risk of introducing false negatives.
- **Effective Dense Interpolation Strategy(EDIS)**, which employs linear interpolation among existing samples within the same semantic cluster to make sure the semantic consistency. Our proposed method works with minimal computational cost, enhances the diversity and richness of negative samples, and facilitates the controllable difficulty of hard negative samples.

3.3 Multimodal Semantic-Aware Clustering

Most existing negative sampling methods tend to ignore the semantic correlations among samples. Against this deficiency, our MSAC is proposed to capture the multi-perspective similarities among items and incorporate explicit semantics into the negative sampling process.

3.3.1 Multimodal-aligned Technique. When users browse items on the e-commerce platform, they primarily perceive items through three views: visual images, descriptive text, and collaborative filtering recommendations. To generate a comprehensive description of items, it is necessary to consider these views concurrently. The visual representations $\mathcal{R}_{\mathcal{I}}$ and textual representations $\mathcal{R}_{\mathcal{T}}$ can be pretrained by specific encoders [28, 31] in advance. The behavior-based representations $\mathcal{R}_{\mathcal{G}}$ can be pretrained using graph representation learned based on a substantial number of user behaviors. Given a mini-batch of N items, we design multimodal-aligned encoders for each view.

$$\begin{aligned} \mathcal{M}_{\mathcal{I}} &= H_{\mathcal{I}}(\mathcal{R}_{\mathcal{I}}) \in \mathbb{R}^{N \times d_m} \\ \mathcal{M}_{\mathcal{T}} &= H_{\mathcal{T}}(\mathcal{R}_{\mathcal{T}}) \in \mathbb{R}^{N \times d_m} \\ \mathcal{M}_{\mathcal{G}} &= H_{\mathcal{G}}(\mathcal{R}_{\mathcal{G}}) \in \mathbb{R}^{N \times d_m} \end{aligned} \tag{3}$$

where H_* denotes the encoder of each view, $\mathcal{M}_*$ denotes the output embedding of each view, d_m denotes the output dimension of each multimodal-aligned encoder.

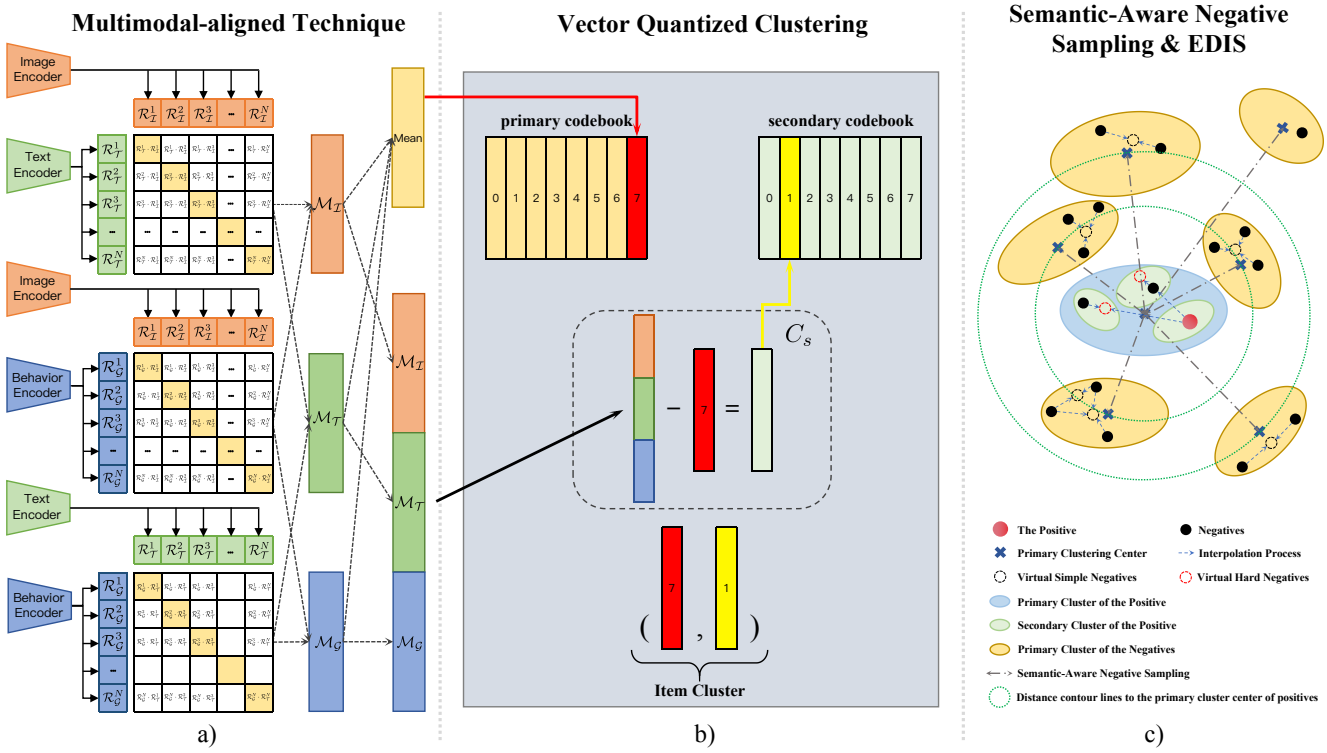

Figure 2: Our proposed ESANS framework. a) Multimodal-aligned Technique. b) Vector Quantized Clustering with Cascaded Codebooks. c) Semantic-Aware Negative Sampling & Effective Dense Interpolation Strategy (EDIS).

Inspired by the Contrastive Language-Image Pre-training (CLIP) [40], We propose a multimodal alignment method to fuse item representations from three perspectives. Given a dataset of $\mathcal{M}_*$ that consists of a collection of output embeddings $\{\mathcal{M}_{\mathcal{I}}^i, \mathcal{M}_{\mathcal{T}}^i, \mathcal{M}_{\mathcal{G}}^i\}_{i=1}^N$, we contrast congruent and incongruent pairs across any two modalities. For instance, we sample from the joint distribution of image-text modals $\mathbf{x}_{\mathcal{I}-\mathcal{T}} \backsim \mathbf{P}(\mathcal{M}_{\mathcal{I}}, \mathcal{M}_{\mathcal{T}})$ or $\mathbf{x}_{\mathcal{I}-\mathcal{T}} = \{\mathcal{M}_{\mathcal{I}}^i, \mathcal{M}_{\mathcal{T}}^i\}$, which we call positive samples. We sample from the product of marginals, $\mathbf{y}_{\mathcal{I}-\mathcal{T}} \backsim \mathbf{P}(\mathcal{M}_{\mathcal{I}})\mathbf{P}(\mathcal{M}_{\mathcal{T}})$ or $\mathbf{y}_{\mathcal{I}-\mathcal{T}} = \{\mathcal{M}_{\mathcal{I}}^i, \mathcal{M}_{\mathcal{T}}^j\}$, which we call negative samples. Multimodal-aligned encoders are optimized to correctly select a single positive sample $\mathbf{x}_{\mathcal{I}-\mathcal{T}}$ out of the set $\mathcal{S} = \{\mathbf{x}_{\mathcal{I}-\mathcal{T}}, \mathbf{y}_{\mathcal{I}-\mathcal{T}}^1, ..., \mathbf{y}_{\mathcal{I}-\mathcal{T}}^{N-1}\}$ which contains $N-1$ negative samples:

$$
\begin{aligned}
\mathcal{L}_{align}^{\mathcal{I}-\mathcal{T}} &= -\mathop{\mathbb{E}}_{\mathcal{S}}[log\frac{h(\mathbf{x}_{\mathcal{I}-\mathcal{T}})}{h(\mathbf{x}_{\mathcal{I}-\mathcal{T}}) + \sum_{i=1}^{N-1} h(\mathbf{y}_{\mathcal{I}-\mathcal{T}}^i)}] \\
\mathcal{L}_{align}^{\mathcal{I}-\mathcal{G}} &= -\mathop{\mathbb{E}}_{\mathcal{S}}[log\frac{h(\mathbf{x}_{\mathcal{I}-\mathcal{G}})}{h(\mathbf{x}_{\mathcal{I}-\mathcal{G}}) + \sum_{i=1}^{N-1} h(\mathbf{y}_{\mathcal{I}-\mathcal{G}}^i)}] \\
\mathcal{L}_{align}^{\mathcal{G}-\mathcal{T}} &= -\mathop{\mathbb{E}}_{\mathcal{S}}[log\frac{h(\mathbf{x}_{\mathcal{G}-\mathcal{T}})}{h(\mathbf{x}_{\mathcal{G}-\mathcal{T}}) + \sum_{i=1}^{N-1} h(\mathbf{y}_{\mathcal{G}-\mathcal{T}}^i)}]
\end{aligned}
\tag{4}
$$

where $h(\cdot)$ is the cosine similarity operation after exponentiation, $\mathcal{L}_{align}^{\mathcal{I}-\mathcal{T}}$ is the alignment loss between visual and textual modals, $\mathcal{L}_{align}^{\mathcal{I}-\mathcal{G}}$ is the alignment loss between visual and behavior-based modals, $\mathcal{L}_{align}^{\mathcal{G}-\mathcal{T}}$ is the alignment loss between behavior-based and textual modals.

3.3.2 Vector Quantized Clustering with Cascaded Codebooks. While aligning $\mathcal{M}_{\mathcal{I}}, \mathcal{M}_{\mathcal{T}}, \mathcal{M}_{\mathcal{G}}$ into the same embedding space, we simultaneously quantize these representations into several clusters with cascaded codebooks, as illustrated in Figure 2. Specifically, the primary codebook is designed to effectively differentiate coarse-level item representations, while the secondary codebook enhances this distinction by refining the differentiation of fine-grained item representations, especially when significant disparities persist among aligned representations across partial modalities.

The ***primary codebook*** $C_p = \{z_p^k\}_{k=1}^{K_p}$ consists of K_p codewords [29] and the dimension of each codeword is d_m. The clustering stage is conducted by calculating the mean of the aligned embeddings:

$$
\mathcal{R}_p^i = \frac{1}{3}(\mathcal{M}_{\mathcal{I}}^i + \mathcal{M}_{\mathcal{T}}^i + \mathcal{M}_{\mathcal{G}}^i) \tag{5}
$$

Subsequently $\mathcal{R}_p = \{\mathcal{R}_p^i\}_{i=1}^N$ is quantized by assigning it to the nearest codeword within the primary codebook. We denote that the nearest codeword to $\mathcal{R}_p^i$ is $C_p^i = \arg\min_k \|\mathcal{R}_p^i - z_p^k\|$.

In the ***secondary codebook***, we compute the residual between $\{\mathcal{M}_{\mathcal{I}}, \mathcal{M}_{\mathcal{T}}, \mathcal{M}_{\mathcal{G}}\}$ and the primary corresponding codeword $z_p^{C_p^i}$. These residuals are concatenated to a vector $\mathcal{R}_s^i$, which is used to

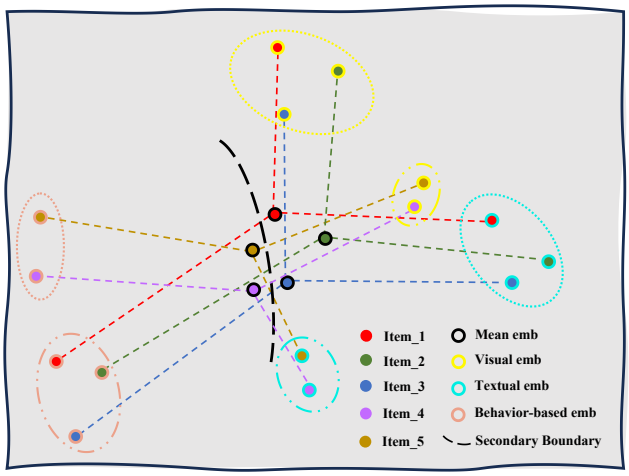

Figure 3: The visualization of items in the representation space during secondary clustering. Although item 1-3 and item 4-5 have similar mean embeddings, but in each view thier embeddings differ significantly, resulting in their assignment to different secondary clusters. By leveraging three modalities, clustering accuracy is significantly enhanced.

describe the modal-specific information between different items.

$$\mathcal{R}_s^i = [\mathcal{M}_{\mathcal{I}}^i - z_p^{C_p^i}; \mathcal{M}_{\mathcal{T}}^i - z_p^{C_p^i}; \mathcal{M}_{\mathcal{G}}^i - z_p^{C_p^i}] \tag{6}$$

The advantages of using information from three modalities for secondary clustering are illustrated in Figure 3. Similar to the primary clustering, we select the codeword closest to $\mathcal{R}_s = \{\mathcal{R}_s^i\}_{i=1}^N$ from another codebook $C_s = \{z_s^k\}_{k=1}^{K_s}$, where K_s denotes the number of codewords in the secondary codebook. The nearest secondary codeword to $\mathcal{R}_s^i$ is recorded as $C_s^i = \arg\min_k \|\mathcal{R}_s^i - z_s^k\|$.

Once we have all cluster indice for an item, the clustering loss can be defined as:

$$\mathcal{L}_{\text{SQ}} = \sum_{i=1}^{N} \|\mathcal{R}_p^i - z_p^{C_p^i}\|^2 + \sum_{i=1}^{N} \|\mathcal{R}_s^i - z_s^{C_s^i}\|^2 \tag{7}$$

Finally, the loss function for multimodal-aligned clustering is given by Equation 8:

$$\mathcal{L} = \beta_1 \mathcal{L}_{align}^{\mathcal{I}-\mathcal{T}} + \beta_2 \mathcal{L}_{align}^{\mathcal{I}-\mathcal{G}} + \beta_3 \mathcal{L}_{align}^{\mathcal{G}-\mathcal{T}} + \mathcal{L}_{\text{SQ}} \tag{8}$$

3.3.3 Semantic-Aware Negative Sampling. Based on the above framework, we divide the whole set of candidate items into multiple semantic clusters. Then we introduce the semantic-aware negative sampling which includes simple negative sampling and hard negative sampling. In simple negative sampling, we select primary clusters for each positive sample based on the following probability formula, ensuring that none of these selected clusters are the same as the primary cluster of the positive sample.

$$\begin{aligned} Q(C_p = i) &= \frac{1}{d(z_p^i, z_p^+)^\gamma}, i \neq + \\ P(C_p = i) &= \frac{Q(C_p = i)}{\sum_{j \neq +} Q(C_p = j)} \end{aligned} \tag{9}$$

where $d(\cdot,\cdot)$ measures the distance between primary codewords using an inner-product operation, which is subsequently normalized to a range from 0 to 1. z_p^+ is the primary cluster of the positive sample, $Q(C_p = i)$ is the unnormalized sampling probability of similar primary clusters with γ, $P(c_p = i)$ is the normalized sampling probability of primary cluster z_p^i. Then, we randomly select samples from each cluster which enhances the diversity of negative samples. After being encoded by the item tower [25], the embedding set of simple negative samples can be represented as V_s:

$$\begin{aligned} V_s &= \{V_s^1, ..., V_s^k, ..., V_s^{m_c}\} \\ V_s^k &= \{\mathbf{v}_s^{(k-1)m_o+1}, ..., \mathbf{v}_s^{km_o}\} \end{aligned} \tag{10}$$

where V_s^k is the embedding set of the simple negative samples in k-th cluster, m_c is the number of selected clusters and m_o is the number of selected samples in each cluster. In this way, we dynamically adjust the difficulty of the simple negatives as well as mitigate group-level sampling biases.

In hard negative sampling strategy, we randomly select partially similar samples within the positive primary cluster. Then, we consider samples in the same secondary cluster as false negatives and remove these samples from the hard negative samples set. The output embedding set of hard negative samples can be represented as $V_h = \{\mathbf{v}_h^1, \mathbf{v}_h^2, ..., \mathbf{v}_h^{m_h}\}$, where m_h is the number of selected samples in the positive primary cluster.

3.4 Effective Dense Interpolation Strategy

By employing our negative sampling process, we obtain negative sample clusters and randomly selected negative sample anchors for each cluster. It's a well-established principle [6] that increasing the negative sampling size can enhance the performance of the EBR models. However, the process mentioned above does not guarantee a sufficient sampling size for each cluster. To solve this problem, we propose a parameter-adaptive negative sampling augmentation technique based on the linear interpolation to increase the number of negative samples. The detailed interpolation process is applied to both simple negative samples and hard negative samples, which is illustrated in Figure 2.

3.4.1 Interpolation on Simple Negative Samples. Suppose we select n_o negative anchors ($2 \leq n_o \leq m_o$) from the k-th cluster. The output item embeddings are reordered as $V_{s_k} = \{\mathbf{v}_{s_k}^1, ..., \mathbf{v}_{s_k}^{n_o}\}$. Each vector in the embedding set is selected once as the anchor vector $\mathbf{v}_{s_k}^a$, and generate the virtual negative samples similar to the embedding set by linear interpolation:

$$\begin{aligned} \tilde{\mathbf{v}}_{s_k}^a &= \sum_{i=1}^{n_o} \alpha_i \mathbf{v}_{s_k}^i \\ \alpha_j &= \frac{d(\mathbf{v}_{s_k}^a, \mathbf{v}_{s_k}^j)^\eta}{\sum_{j=1}^{n_o} d(\mathbf{v}_{s_k}^a, \mathbf{v}_{s_k}^j)^\eta} \end{aligned} \tag{11}$$

where $\tilde{\mathbf{v}}_{s_k}^a$ denotes the virtual negative sample obtained by linear interpolation, $d(\cdot,\cdot)$ is the inner-product operation to measure the embedding distance between item vectors, η is designed to adjust the magnitude of impact resulting from surrounding vectors, α_j is the adaptive parameter to fuse negative samples. Our method ensures that each virtual sample is proximate to the anchor and can be disturbed by other negative samples in terms of similarity. The quantity m_c^v of virtual samples in each primary cluster is proportional to $O(m_o^2)$, which can be solve as follow:

$$m_c^v = (m_o + 2)(m_o - 1)/2 \propto O(m_o^2) \tag{12}$$

In this way, we efficiently enhance the diversity and richness of negative samples.

3.4.2 Interpolation on Hard Negative Samples. As mentioned in the previous explanation, the hard negative samples are selected from the same primary cluster but different secondary cluster. The interpolation on hard negative samples is proposed to further augment the quantity of samples and facilitate the controllable difficulty of hard negative samples. We denote the output embedding of positive sample as $\mathbf{v}_+$. We conduct the linear interpolation between $\mathbf{v}_+$ and each existing hard negative sample $\mathbf{v}_h^a$:

$$\tilde{\mathbf{v}}_h^a = \lambda \mathbf{v}_+ + (1-\lambda)\mathbf{v}_h^a \tag{13}$$

where λ is a hyperparameter used to adjust the difficulty of hard negatives during the training process. When $0 < \lambda < 1$, virtual hard negatives are generated between positive samples and existing hard nagetives which provides more challenging samples. When $\lambda < 0$, virtual hard negatives are easier than existing hard nagetives which provides relatively simple samples. By this strategy, we enhance the challenge of discriminating the classification boundary and incorporate the stochastic uncertainty into the model which also improves its generalization performance.

3.5 Model Learning

Following the widely used EBR method, Deep Structured Semantic Model (DSSM) [26, 50], we can optimize the similarity between user embeddings $\mathbf{u}$ and item embeddings $\mathbf{v}$ by contrastive learning method. The objective function applied is the InfoNCE loss, which is defined as follows:

$$\mathcal{L}_{sm} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{\mathbf{u}_i^T \mathbf{v}_i^+/\tau}}{e^{\mathbf{u}_i^T \mathbf{v}_i^+/\tau} + \sum_{i=1}^{N_s} e^{\mathbf{u}_j^T \mathbf{v}_i^{j-}/\tau}} \tag{14}$$

Where N_s is the number of negatives effectively sampled by our ESANS method, $\mathbf{v}_i^{j-}$ denotes the j-th negative sample of the i-th positive sample $\mathbf{v}_i^+$, $\mathbf{u}^T\mathbf{v}$ is also called as the target logit [39] of the i-th positive sample, τ is the temperature hyperparameter used to adjust the distribution of logits.

The pseudocode for the algorithm is presented in Figure 4.

4 Offline Experiments

In this section, we conduct offline experiments on three real-world datasets to demonstrate the effectiveness and efficiency of our proposed method. The first two are public datasets, while the third is an industrial dataset. The descriptions and statistics of the two public datasets and the industrial dataset are detailed in Tables 1 and 2, respectively. Additionally, we perform an ablation study of our modules and address the following research questions:

- **RQ1**: How does our ESANS perform compared to other state-of-the-art models?
- **RQ2**: What is the impact of each component on the overall model's performance?
- **RQ3**: What is the effect of the hyper-parameters on the performance of our model?

Algorithm 1 : ESANS

Input: A large item pool $\mathcal{I}$ with multimodal representation $(\mathcal{R}_\mathcal{I}, \mathcal{R}_\mathcal{T}, \mathcal{R}_\mathcal{G})$.
Training set $\mathcal{R} = \{(u,i) \in (\mathcal{U},\mathcal{I})\}$, embedding dimension K;
Output: Multimodal-aligned Model $(H_\mathcal{I}, H_\mathcal{T}, H_\mathcal{G})$.
2 Vector Quantization Codebook (C_p, C_s).
Final user embeddings $\{\mathbf{u}_u | u \in \mathcal{U}\}$ and item embeddings $\{\mathbf{v}_i | i \in \mathcal{I}\}$;

```
Initialize (H_I, H_T, H_G), (C_p, C_s), {u_u | u ∈ U} and {v_i | i ∈ I};
for t = 1 To T do
    Sample a mini-batch I_batch ∈ I of size B;
    Get the output embedding after aligning (M_I^batch, M_T^batch, M_G^batch) from Equation (3);
    Get the embedding (R_p^batch, R_s^batch) from Equation (5) and Equation (6);
    Update Models (H_I, H_T, H_G) and codebooks (C_p, C_s) based on Loss function (8).
end for
for t = 1 To T do
    Sample a mini-batch R_batch ∈ R of size B;
    for each (u, i) ∈ R_batch do
        Get the prime C_p^i and secondary cluster C_s^i of item i;
        Sample m_c prime clusters base on the Equation (9);
        Uniformly sample m_o items from each prime cluster k and m_h hard negatives;
        for k = 1 To m_c do
            for j = 1 To n_o do
                Get the virtual hard negatives ṽ_{s_k}^a based on the Equation (11);
            end for
        end for
        for j = 1 To m_h do
            Get the virtual hard negatives ṽ_h^a based on the Equation (13);
        end for
        Update embeddings (u_u, v_i) based on gradient w.r.t. (14);
    end for
end for
```

Figure 4: The pseudocode of our proposed algorithm.

Table 1: Statistics of Public Datasets.

Dataset	Amazon	PixelRec
#User	247,446	29,845,039
#Item	88,408	408,374
#Interaction	2,146,317	195,755,320

4.1 Experimental Setup

Dataset.

- **Amazon Review.** It was first introduced by Van Gysel et al. [48, 49] and has become a benchmark dataset for evaluating product recommendation methods [14, 34, 46]. We select the Electronics subset which products a sufficient number of user reviews and includes comprehensive metadata, such as product titles and categories. The textual features are extracted by sentence-transformers [41] from [60] and the visual features are extracted and published in [37].
- **Pixel-Rec.** This dataset [11] is derived from a global online video platform which captures approximately 200 million user consumption from September 2021 to October 2022. It focuses on content-driven recommendations spanning diverse categories such as food, games, fashion, and makeup. The textual and visual

features of these contents have already been extracted using PixelNet, a network proposed concurrently with Pixel-Rec.

- **Industrial Dataset.** We establish the offline dataset by collecting the users' sequential behaviors and feedback logs from Alibaba's international e-commerce platform, Lazada. The dataset comprises four categories, each representing a distinct Southeast Asian country, labeled from #A1 to #A4.

Table 2: Statistics of the Industrial Dataset.

Scenarios	#A1	#A2	#A3	#A4
#User	4,930,611	24,931,581	17,037,221	15,914,765
#Item	2,163,338	4,268,324	2,905,716	3,067,253
#Click	61,579,472	336,744,161	149,341,806	163,611,291
#Impression	2,353M	9,648M	4,274M	6,134M

Graph Construction. Due to space limitation, the introduction of the behavior-based graph construction is provided in Appendix Section A.

Baselines. We compared our ESANS with five representative negative sampling methods based on the classical two-tower architecture. The methods are as follows:

- **UNS** [23, 44]: A widely used negative sampling approach involves randomly selecting instances from a uniform distribution.
- **PNS** [7]: A negative sampling method that adjusts the sampling distribution based on item popularity.
- **Debiased MNS** [56, 57]: A method that integrates UNS with in-batch negative sampling, and introduces a technique to address the oversampling issue of popular items.
- **MixGCF** [27]: A method synthesizes hard negatives between negatives and positives in a graph-based model. We adapt this to a two-tower structure to generate virtual hard negatives in the item representation space.
- **FairNeg** [9]: A method that improves item group fairness by adaptively adjusting the distribution of negative samples at the group level.
- **Adap-τ** [4]: A method that adjusts the temperature coefficient of the loss function by the embedding similarity between users and corresponding negatives.

Evaluation Metrics. For the evaluation metrics in recommendation tasks, we follow [2, 52] and use Recall@K for each group based on the Top-K recommendation results. Finally, the Recall@K is averaged over all users.

Parameter settings. We divide users in each public dataset into three subsets: training, validation, and testing, with a ratio of 8:1:1. As for the Industrial Dataset, we set aside the instances from the final day for testing, while using the preceding instances for training. For each user in the training set, we employ their first k actions to predict the $(k+1)$-th action. To ensure computational manageability, we limit the length of user behavior sequences to 10 for the Amazon Review dataset, 32 for the Pixel-Rec dataset and 64 for the Industrial dataset. Due to space limitation, additional implementation details are provided in Appendix Section B.

4.2 Performance Comparison (RQ1)

Table 3 summarizes the overall performance of our ESANS as well as the baselines on both industrial and public datasets, with the best results emphasized in bold and the second-best results underlined. It is noteworthy that ESANS consistently outperforms all baseline methods across the aforementioned datasets, achieving an average improvement of up to **15.32%** in recall@50 and **10.73%** in recall@200 compared to its base method UNS. PNS generally outperforms UNS across most datasets, indicating that boosting the sampling possibility for popular items improves sampling quality. However, it is worth noting that PNS does not exceed UNS performance in the #A3 dataset, which might be attributed to the introduced popularity bias. Once the challenge of popularity bias is addressed, the debiased MNS Sampling method outperforms UNS and PNS across all datasets and outperforms other baselines on #A4. MixGCF introduces virtual hard negatives by hop-mixing interpolation which achieves similar performance with the debiased MNS and proves the feasibility of hard negatives augmentation. However, the interpolation process fails to consider semantics and yields noisy negatives, so it is outperformed by our method. FairNeg is another work conducted to reduce the sampling bias via adjusting the group-level negative sampling distribution which provides the best recommendation utility on Pixel-Rec and #A2 in all baselines. However, this work determines the groups by the only item attribute view which is not comprehensive and thus is surpassed by our method. Ada-τ is proposed to design a learnable τ, which enables the adaptive adjustment of the difficulty level for negatives. This work outperforms other baseline models on Amazon Elecs. However, Ada-τ fails to provide incremental information by deriving more challenging negatives during the training process so that it is beaten by our method.

In summary, our method effectively addresses the inherent limitations of these methods and achieves SOTA performance across all datasets in terms of retrieval efficiency. It is worth noting that the MSAC module is actually detached from the training process of DSSM and the EDIS module is only applied to the output of the deep neural network. Therefore, our method does not introduce the additional computational complexity for either offline training and online serving.

4.3 Ablation Study (RQ2)

To investigate the effectiveness of each component in the proposed model, in this subsection, we conduct a series of ablation studies on the #A2 industrial datasets as follows:

- **w/o MSAC**, removes the Multimodal Semantic-Aware Clustering before the Interpolation-based negative sampling.
- **w/o EDIS**, removes the Effective Dense Interpolation Strategy employed in both simple negative sampling and hard negative sampling strategies.
- **w/o Multimodal Aligning**, removes the textual and visual modalities and reserves the behavior-based modality for further clustering.
- **w/o Secondary Codebook**, removes the secondary codebook in the Vector Quantized Clustering, thereby invalidating the interpolation-based hard negative sampling.

Table 4 presents the performance of these ablation experiments. Firstly, we can observe that adopting Multimodal-aligned Clustering Algorithm improves recall@50 by **6.41%** and recall@200 by

Table 3: Performance Comparison across baselines. The last column(AVG) denotes the average improvement of sampling methods across all datasets. The last row(RI) denotes the relative improvement of our ESANS over UNS.

Method	Amazon Elecs		Pixel-Rec		#A1		#A2		#A3		#A4		AVG	
	Recall@50	Recall@200	Recall@50	Recall@200	Recall@50	Recall@200	Recall@50	Recall@200	Recall@50	Recall@200	Recall@50	Recall@200	Recall@50	Recall@200
UNS	0.1633	0.4512	0.0737	0.1522	0.4276	0.6087	0.3108	0.5155	0.3859	0.6478	0.3857	0.6277	0.2912	0.5005
PNS	0.1695	0.4696	0.0783	0.1587	0.4332	0.6401	0.3370	0.5318	0.3803	0.6405	0.3883	0.6321	0.2978	0.5121
debiased MNS	0.1874	0.4726	0.0801	0.1655	0.4549	0.6562	0.3597	0.5647	0.3970	0.6513	0.4074	0.6549	0.3144	0.5275
MixGCF	0.1963	0.4759	0.0794	0.1631	0.4593	0.6577	0.3621	0.5703	0.4012	0.6574	0.3924	0.6483	0.3151	0.5288
FairNeg	0.1792	0.4705	0.0836	0.1782	0.4790	0.6688	0.3669	0.6052	0.4043	0.6687	0.3983	0.6501	0.3186	0.5403
Adap-t	0.2018	0.4873	0.0763	0.1684	0.4694	0.6757	0.3538	0.5883	0.4097	0.6625	0.4046	0.6437	0.3193	0.5377
ESANS (ours)	**0.2135**	**0.4948**	**0.0908**	**0.1828**	**0.4862**	**0.6918**	**0.3887**	**0.6216**	**0.4176**	**0.6732**	**0.4182**	**0.6609**	**0.3358**	**0.5542**
RI	+30.74%	+9.66%	+23.20%	+20.11%	+13.70%	+13.65%	+25.06%	+20.58%	+8.21%	+3.92%	+8.43%	+5.29%	+15.32%	+10.73%

Table 4: Ablation Study on the #A2 Dataset.

Method	#A2	
	Recall@50	Recall@200
Ours	**0.3887**	**0.6216**
w/o MSAC	0.3653	0.5976
w/o EDIS	0.3788	0.6134
w/o Multimodal Aligning	0.3802	0.6163
w/o Secondary Codebook	0.3724	0.6082

4.02%, which proves that the semantic clustering algorithm employed for interpolation significantly improves sampling quality. The dense interpolation respectively brings a **2.61%** and a **1.34%** improvement for recall@50 and recall@200, which demonstrates the efficient of our sample augment strategy. Besides, the Multimodal clustering performs better than the Unimodal clustering(**2.24%** on recall@50 and **0.86%** on recall@200), which superiority the multi-view representations. The interpolation-based hard negative sampling conducted by the secondary codebook also shows the improvement(**4.38%** on recall@50 and **2.20%** on recall@200), which proves the feasibility of selecting hard negative samplings with heuristics semantic constraint.

4.4 Hyperparameters Sensitivity Analysis (RQ3)

In this section, we investigate the sensitivity of our model's hyperparameters, specifically the number of primary clusters K_p and the number of secondary clusters K_s. These experiments are carried out on the #A1-#A4 industrial datasets, employing five distinct values for K_p (100, 200, 300, 400, 500) and K_s (5, 10, 15, 20, 25). Figure 5 illustrates the performance of these hyperparameter tuning experiments. We observe that the model's performance stays consistently high when K_p is increased from 200 to 500. However, reducing K_p to 100 leads to a slight decrease in performance. This observation encourages us to consider a higher value for K_p to further enhance the intra-cluster semantic consistency. K_s shows optimal prediction performance between 5 to 15. We recommend to set a relatively small value for K_s in order to minimize the occurrence of false negatives.

5 Online Experiments

To further validate the effectiveness of our approach, we conducted an online A/B test on an e-commerce recommendation platform from September 13 to 19, 2024. The control group used a two-tower

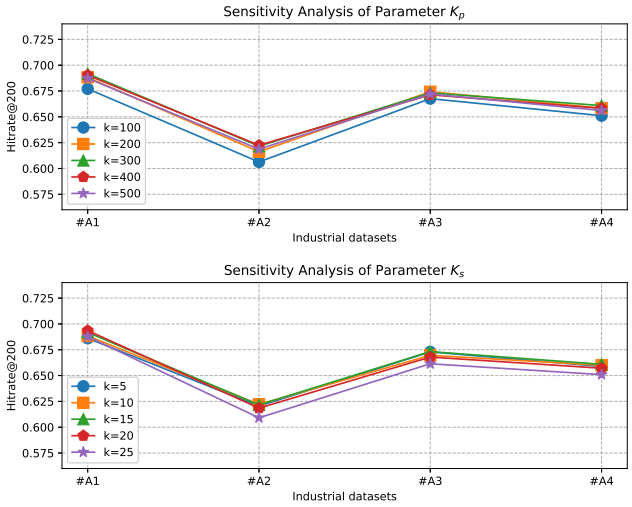

Figure 5: The performance of ESANS under different hyperparameters(K_p and K_s) on #A1-#A4 industrial datasets.

model with debiased mixed negative sampling (MNS) [56], while the experiment group applied our proposed method. Both groups consisted of 30% randomly selected users. Specifically, we observed **2.83%** increase in the **Advertising Revenue**, **1.19%** increase in the **Click-Through-Rate(CTR)** and **1.94%** increase in the **Gross Merchandise Volume(GMV)**. The results of the online experiment once again confirm the efficiency and effectiveness of our method ESANS in negative sampling for recommendation systems.

6 Conclusion

In this study, we proposed a novel negative sampling method, Effective and Semantic-Aware Negative Sampling (ESANS), which integrates an Effective Dense Interpolation Strategy (EDIS) and Multimodal Semantic-Aware Clustering (MSAC). Extensive experiments demonstrated that ESANS significantly improves sampling quality and efficiency compared to baselines. Specifically, EDIS improves the diversity and density of the sampling distribution. MSAC enhances semantic consistency and reduces false negatives. These modules advance the effectiveness of negative sampling in recommendation systems. For future work, we will pursue two directions. The first is to further optimize the multimodal representations based on MSAC. The second direction is to design a more complex interpolation strategy among the outputs of hidden layers.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: a system for Large-Scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*. 265–283.

[2] Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the gru: Multi-task learning for deep text recommendations. In *proceedings of the 10th ACM Conference on Recommender Systems*.

[3] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[4] Jiawei Chen, Junkang Wu, Jiancan Wu, Xuezhi Cao, Sheng Zhou, and Xiangnan He. 2023. Adap-τ: Adaptively modulating embedding magnitude for recommendation. In *Proceedings of the ACM Web Conference 2023*.

[5] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*.

[7] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[8] Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. 2012. SVDFeature: a toolkit for feature-based collaborative filtering. *The Journal of Machine Learning Research* (2012).

[9] Xiao Chen, Wenqi Fan, Jingfan Chen, Haochen Liu, Zitao Liu, Zhaoxiang Zhang, and Qing Li. 2023. Fairly adaptive negative sampling for recommendations. In *Proceedings of the ACM Web Conference 2023*.

[10] Zhihong Chen, Rong Xiao, Chenliang Li, Gangfeng Ye, Haochuan Sun, and Hongbo Deng. 2020. Esam: Discriminative domain adaptation with non-displayed items to improve long-tail performance. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[11] Yu Cheng, Yunzhu Pan, Jiaqi Zhang, Yongxin Ni, Aixin Sun, and Fajie Yuan. 2024. An Image Dataset for Benchmarking Recommender Systems with Raw Pixels. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*.

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

[13] Jingtao Ding, Yuhan Quan, Xiangnan He, Yong Li, and Depeng Jin. 2019. Reinforced Negative Sampling for Recommendation with Exposure Data.. In *IJCAI*.

[14] Lu Fan, Qimai Li, Bo Liu, Xiao-Ming Wu, Xiaotong Zhang, Fuyu Lv, Guli Lin, Sen Li, Taiwei Jin, and Keping Yang. 2022. Modeling user behavior with graph convolution for personalized product search. In *Proceedings of the ACM Web Conference 2022*.

[15] Wenqi Fan, Tyler Derr, Yao Ma, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep adversarial social recommendation. In *28th International Joint Conference on Artificial Intelligence, IJCAI 2019*.

[16] Wenqi Fan, Tyler Derr, Xiangyu Zhao, Yao Ma, Hui Liu, Jianping Wang, Jiliang Tang, and Qing Li. 2021. Attacking black-box recommendations via copying cross-domain user profiles. In *2021 IEEE 37th international conference on data engineering (ICDE)*.

[17] Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. 2020. A graph neural network framework for social recommendations. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[18] Wenqi Fan, Yao Ma, Dawei Yin, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep social collaborative filtering. In *Proceedings of the 13th ACM Conference on Recommender Systems*.

[19] Wenqi Fan, Xiangyu Zhao, Xiao Chen, Jingran Su, Jingtong Gao, Lin Wang, Qidong Liu, Yiqi Wang, Han Xu, Lei Chen, et al. 2022. A comprehensive survey on trustworthy recommender systems. *arXiv preprint arXiv:2209.10117* (2022).

[20] Suyu Ge, Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. Graph enhanced representation learning for news recommendation. In *Proceedings of the web conference 2020*.

[21] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

[23] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*.

[24] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*.

[25] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[26] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*.

[27] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. 2021. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

[28] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849* (2020).

[29] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[30] 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*.

[31] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557* (2019).

[32] Xiaochen Li, Xin Song, Pengjia Yuan, Xialong Liu, and Yu Zhang. 2022. Soft Retargeting Network for Click Through Rate Prediction. *arXiv preprint arXiv:2206.01894* (2022).

[33] Jiazhen Lou, Hong Wen, Fuyu Lv, Jing Zhang, Tengfei Yuan, and Zhao Li. 2022. Re-weighting negative samples for model-agnostic matching. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[34] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Jie Zhou. 2023. Exploring false hard negative sample in cross-domain recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*.

[35] Muhammad Arslan Manzoor, Sarah Albarri, Ziting Xian, Zaiqiao Meng, Preslav Nakov, and Shangsong Liang. 2023. Multimodality Representation Learning: A Survey on Evolution, Pretraining and Its Applications, Vol. 20. Association for Computing Machinery, New York, NY, USA.

[36] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems* (2013).

[37] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*.

[38] Dae Hoon Park and Yi Chang. 2019. Adversarial sampling and training for semi-supervised information retrieval. In *The World Wide Web Conference*.

[39] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548* (2017).

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*.

[41] N Reimers. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP* (2019).

[42] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*.

[43] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*.

[44] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[45] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*.

[46] Yongxiang Tang, Wentao Bai, Guilin Li, Xialong Liu, and Yu Zhang. 2022. CROLoss: towards a customizable loss for retrieval models in recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*.

[47] Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in Neural Information Processing Systems* (2017).

[48] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2016. Learning latent vector spaces for product search. In *Proceedings of the 25th ACM international on conference on information and knowledge management*.

[49] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2017. Semantic entity retrieval toolkit. *arXiv preprint arXiv:1706.03757* (2017).
[50] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters* (2018).
[51] Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.
[52] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*.
[53] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*.
[54] Jinpeng Wang, Jieming Zhu, and Xiuqiang He. 2021. Cross-batch negative sampling for training two-tower recommenders. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*.
[55] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*.
[56] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion proceedings of the web conference 2020*.
[57] 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*.
[58] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*.
[59] Xiangyu Zhao, Haochen Liu, Wenqi Fan, Hui Liu, Jiliang Tang, and Chong Wang. 2021. Autoloss: Automated loss function search in recommendations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.
[60] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. 2023. Bootstrap latent representations for multi-modal recommendation. In *Proceedings of the ACM Web Conference 2023*.

A Graph Construction

For each dataset, we pretrain a heterogeneous graph network [32] based on user behaviors. The types of graph nodes include user, item, and its side information (brand / category / price features for Amazon Review dataset, tag / statistical features for Pixel-Rec dataset and brand / shop / category for industrial datasets). The graph edges include: 1) user-item edge. If user u clicks item i, there is an edge between u and i. 2) user-side information edge. If user u clicks an item with side information v (e.g., shop), there is an edge between u and v. 3) item-item edge. If item i and item j are adjacent in user behavior sequence and the time interval between item i and item j is within 60 seconds, there is an edge between i and j. 4) item-side information edge. If item i has a side info v, there is an edge between i and v.

B Parameter Settings

In this section, we elaborate on the parameter settings for the implementation of our algorithm. The training process is implemented using a distributed TensorFlow[1] platform, consisting of 10 parameter servers and 40 workers with 12 CPUs per worker. In the negative sampling process, for each in-batch positive sample, we randomly select $m_c = 2$ clusters and then draw $m_o = 5$ negative samples from each of these clusters. In contrast, the baseline model selects 10 negative samples randomly for each positive sample. These negatives are sampled based on an online sampling framework in the training process and shared across the batch. Additionally, the interpolation coefficient λ of hard negatives is set to 0.1 for a harder interpolation and -0.1 for a easier interpolation. The rest of the hyperparameters settings are demonstrated in Table 5.

Table 5: Hyper-parameter settings of Our ESANS.

Hyper-parameter	Choice
B	512
τ	0.05
d_k	64
d_m	512
K_p	300
K_s	15
β_1	2.0
β_2	2.0
β_3	2.0
η	0.6
Optimizer	Adam
Learning rate	0.0002
Amazon modal emb size	Img: 4096, Text:384, Graph:128
PixelRec modal emb size	Img: 1024, Text:1024, Graph:1024
#A1-#A4 modal emb size	Img: 1024, Text:1024, Graph:1024