MIC: Model-agnostic Integrated Cross-channel Recommender

Ping Nie*
Tencent
Beijing, China
ping.nie@pku.edu.cn

Yujie Lu*
University of California, Santa Barbara
Santa Barbara, CA, USA
yujielu10@gmail.com

Shengyu Zhang
Zhejiang University
Zhejiang, China
sy_zhang@zju.edu.cn

Ming Zhao
Tencent
Beijing, China
marcozhao@tencent.com

Ruobing Xie†
Tencent
Beijing, China
xrbsnowing@163.com

William Yang Wang†
University of California, Santa Barbara
Santa Barbara, CS, USA
william@cs.ucsb.edu

Yi Ren†
Tencent
Beijing, China
henrybjren@tencent.com

ABSTRACT

Semantically connecting users and items is a fundamental problem for the matching stage of an industrial recommender system. Recent advances in this topic are based on multi-channel retrieval to efficiently measure users' interest on items from the massive candidate pool. However, existing studies are primarily built upon pre-defined retrieval channels, including User-CF (U2U), Item-CF (I2I), and Embedding-based Retrieval (U2I), thus access to the limited correlation between users and items which solely entail from partial information of latent interactions. In this paper, we propose a model-agnostic integrated cross-channel (MIC) approach for the large-scale recommendation, which maximally leverages the inherent multi-channel mutual information to enhance the matching performance. Specifically, MIC robustly models correlation within user-item, user-user, and item-item from latent interactions in a universal schema. For each channel, MIC naturally aligns pairs with semantic similarity and distinguishes them otherwise with more uniform anisotropic representation space. While state-of-the-art methods require specific architectural design, MIC intuitively considers them as a whole by enabling the complete information flow among users and items. Thus MIC can be easily plugged into other retrieval recommender systems. Extensive experiments show that our MIC helps several state-of-the-art models boost their performance on four real-world benchmarks. The satisfactory deployment of the proposed MIC on industrial online services empirically proves its scalability and flexibility.

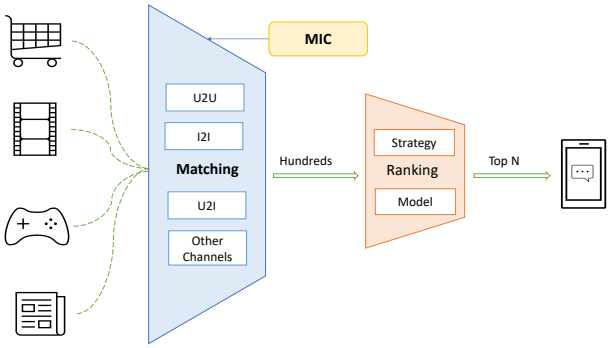

Figure 1: A diagram of a typical two-stage (matching and ranking) recommender system in the real world. MIC can be easily applied in the matching stage.

CCS CONCEPTS

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

KEYWORDS

retrieval recommender, model-agnostic, cross-channel contrastive

ACM Reference Format:
Ping Nie, Yujie Lu, Shengyu Zhang, Ming Zhao, Ruobing Xie, William Yang Wang, and Yi Ren. 2022. MIC: Model-agnostic Integrated Cross-channel Recommender. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557081

*Both authors contributed equally to this research.
†Corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557081

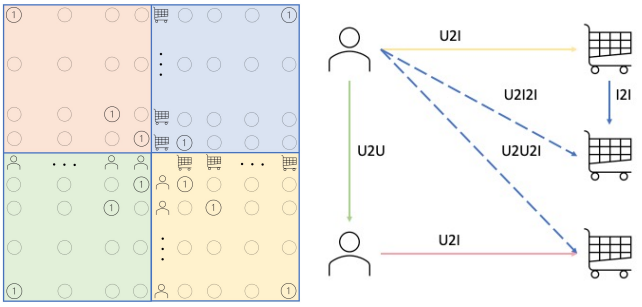

Figure 2: A diagram for multiple connected paths (U2I, I2I, U2U, U2U2I, U2I2I) among users and items. The interactions and correlations are reflected in the left matrix.

1 INTRODUCTION

In this era of information explosion, recommendation services have emerged to match various products with diverse users efficiently. As shown in Figure 1, the matching stage providing the retrieved items list to the ranking stage is the cornerstone and the bottleneck of a typical two-stage industrial recommender system. Figure 2 depicts the commonly used retrieval connected paths: *1)* U2I: Directly recommend items to users. *2)* I2I: Recommend similar items. *3)* U2U: Retrieve similar users. *4)* U2U2I: Recommend items that similar users like based on user-based collaborative filtering. *5)* U2I2I: Recommend similar items based on user interaction history and similar items. These paths finally depict the commonly retrieval channels: **U2U** (by U2U2I path), **U2I** (by U2I path) and **I2I** (by U2I2I path). In this scenario, it is vital to efficiently model user preferences over items to retrieve from large-scale candidate pools; thus, multi-channel retrieval, which efficiently mixes the diversified retrieved items, is a natural and indispensable approach.

However, most previous methods seek to improve the performance of user modeling based on a single channel, thus failing to leverage inherent correlations in the user-based channel, item-based channel, and user-item channel simultaneously. For user channel (U2U), it is common in the industry recommendation systems to use Locality sensitive hashing [13], Paragraph2Vector, [25] and DSSM [20] models to encode user history items and generate similar users. [28] improve the performance of personalization and diversity in item-based collaborative filtering from the item channel (I2I) perspective. [3, 7, 21, 27, 29] are proposed to model dynamic and diversified user preferences based on interaction records from the user-item channel (U2I). For retrieval from multiple sources, [36] propose a hierarchical reinforcement learning framework to recommend heterogeneous items. Nevertheless, the existing method focuses on improving performance based on partial information from each channel, significantly reducing their performance.

We argue that addressing the aforementioned issues in a unified manner is under-explored and points to a new promising direction for developing recommender systems. Models that solely focus on a single angle could learn common relevance between users and items while ignoring the inherent cross-channel information and performing poorly in a real-world scenario. Industrial systems attempt to mitigate such performance reduction by retrieving items based on multiple channels, including various features, strategies, and models. However, existing offline training pipelines are bound to a channel-specific model framework, and the online mixture of multiple channel retrieval is usually controlled by a simple quota mechanism, which leads to two major challenges: *a)* Devising a mechanism to utilize cross-channel information. *b)* Improving item retrieval accuracy and diversity simultaneously in a unified manner. In contrast, our proposed model-agnostic integrated cross-channel (MIC) approach is towards addressing the challenges mentioned above within a universal retrieval recommender system.

In this work, we focus on capturing correlations among users and items across multiple channels with a single model in a unified schema. To achieve this, we first found that it is possible to use one model such as Comirec [3] for three-channel retrieval: U2I, U2U, I2I. Then we designed cross-channel contrastive learning techniques to boost a single model's performance on three channels. We introduce cross-channel contrastive learning techniques into our unified framework with learnable and configurable settings to handle the dynamic and uncertain nature when connecting users and items. In particular, we randomly perturb the fields of each instance and perform dropout in the embedded feature space. The objective is to learn the representations by leveraging a contrastive learning loss to maximize the similarity between the embeddings of two versions of the same instance. User and item representations are learned in their own semantic space via **intra-channel** contrastive loss with the user-user (U-U) contrastive and the item-item (I-I) contrastive training setting. To further connect users and items, we intuitively perform a non-linear projection to learn additional users and items representations in a common semantic space via **inter-channel** user-item (U-I) contrastive loss. The relevance between users and items is measured as the cosine similarity between their vectors in a shared space. Finally, We built a unified score function to generate top-N items from U2U, I2I, and U2I retrieved items.

MIC can realize efficient multi-channel retrieval to capture the co-evolving diversified and dynamic users and item representations in an integrated schema. Since the cross-channel learning module is independent of the encoders and the embedding layer is adaptable to sparse and dense features of users and items, MIC achieves a model-agnostic performance boost by simply switching the encoder to other retrieval models as shown in Figure 3. To summarize, the main contributions of this work are as follows:

- We formulate the matching stage of recommendation as connecting user and item from multiple channels and propose a model-agnostic MIC architecture based on integrated cross-channel user and item representation learning techniques.
- We address the aforementioned long-standing challenges in recommendation in a unified manner via a cross-channel contrastive aggregation mechanism. MIC mitigates the uncertainty of co-evolving user-item correlations and alleviates the seesaw effect between retrieval accuracy and diversity. To the best of our knowledge, this is the first work that proves it is possible to simultaneously utilize U2I, U2U, and U2I channels to improve retrieval accuracy and diversity.
- Compared with the existing method, MIC shows superior effectiveness and efficiency performance on four public datasets.

MIC can also be incorporated into other matching stage recommenders to boost their performance.
- We deployed MIC on the Tencent News platform, and the satisfactory online A/B test results on million-scale users and items confirm the efficiency and effectiveness of MIC practically.

2 APPROACH

2.1 Problem Formulation

In a typical recommendation scenario, we have a set of users and a set of items which can be denoted as $U = \{u_1, u_2, ..., u_{|U|}\}$ and $V = \{v_1, v_2, ..., v_{|V|}\}$, respectively. Let $X_u = \{x_1^u, x_2^u, ..., x_{|X_u|}^u\}$ denote the sequence of interacted items from user $u \in U$ sorted in a chronological order: x_t^u denotes the item that the user u has interacted with item at time step t. Given the user historical behaviors, the goal of the sequential recommendation task considered in this paper is to retrieve a subset of items from the pool V for each user in U such that the user is most likely to interact with the recommended items. Specifically, each instance is represented by a tuple (X_u, F_u, F_v), where X_u denotes the interactions records of user u, F_u denotes the fields of features of the user u including user ID, gender and age. F_v denotes the fields of features of target item v including the information of item ID, item keywords. MIC learns a function f and g for the representations of users and items respectively as

$$\overrightarrow{e_u} = f(X_u, F_u), \overrightarrow{e_v} = g(F_v) \tag{1}$$

where $\overrightarrow{e_u} \in \mathbb{R}^{d\times 1}$ denotes the representation vector of user u, and d is the dimension. $\overrightarrow{e_v} \in \mathbb{R}^{d\times 1}$ denotes the representation vector of item v. When user representation vector and item representation vector are learned, top-N items are recommended according to the likelihood function p as:

$$p(i|U, V, X) = \lambda_{u2v} * p(\overrightarrow{e_u}, \overrightarrow{e_v}) + \lambda_{u2u} * p(\overrightarrow{e_u}, U, X) + \lambda_{v2v} * p(\overrightarrow{e_v}, X) \tag{2}$$

where N is the predefined number of items to be retrieved. $\overrightarrow{e_v}$ is the embedding of item v from a set of items V. λ_{u2v}, λ_{u2u} and λ_{v2v} represent the balance factor for each inference channel U2I, U2U and I2I respectively. We use Grid Search to choose λ_{u2v}, λ_{u2u} and λ_{v2v} as 1:1:1. As we mainly focus on improving the performance in the matching stage of classical industrial recommender systems, Our framework outputs the probabilities for all the items, representing how likely the specific user will engage with these items, and retrieves top-N candidate items.

2.2 Datastore and Inference Procedure

When the MIC is trained, we can predict all users' and items' representation in the training dataset and build a user Datastore and an item Datastore. In the user Datastore, we define the key-value pair $(\overrightarrow{e_u}, u)$ where the key $\overrightarrow{e_u}$ is the vector representation of the value user u. In the item Datastore, the key-value pair is $(\overrightarrow{e_v}, v)$ from the item v representation $\overrightarrow{e_v}$. We also build an interaction Datastore with key-value pairs (u, X_u) where the key is the user ID, and the value is the user interaction history.

At test time, given the user u with interaction history and features, we get user representations $\overrightarrow{e_u}$ from $f(X_u, F_u)$. MIC uses e_u to retrieve N items from item Datastore (U2I) and m similar users from user Datastore. For each similar user, we obtain their interaction history from the interaction Datastore (U2U). We also search similar items according to the user's history from item Datastore (I2I). After U2I, U2U, and I2I channels' search, we have a set of candidate items with counting scores $V_C = \{(v_i, s_i)\}$, where s_i is the repeated number of the retrieved item i. If a specific item is retrieved from more similar users or more similar user interactions, then the counting score will be larger. The counting scores directly considers the contribution of U2U and I2I channel. The size of V_C is often larger than N and much smaller than $|V|$. MIC calculates each item's probability with user embedding, item embedding, and item counting numbers.

$$Score_{Basic}(v_i) = p(\overrightarrow{e_{v_i}, u}), Score_{MIC}(v_i) = \frac{exp(s_i)}{\sum_{j\in|V_C|} exp(s_j)} \tag{3}$$

$$g(i, j) = \delta(C(i)! = C(j)), Score_{Div}(v_i) = \sum_{i\in V_C} \sum_{j\in V_C} g(i, j) \tag{4}$$

$$Score = Score_{Basic} + \lambda_{mic} Score_{MIC}(v_i) + \lambda_{div} Score_{Div}(v_i) \tag{5}$$

where λ_{mic} represents the adjustable factor to aggregate items from different channels and $\lambda_{diversity(div)}$ to control retrieved items' diversity. Similar to ComiRec [3], we control retrieved items' diversity according to item category. We use Grid Search to choose λ_{mic} as 0.5 and $\lambda_{diversity}$ as 0.2. C denotes the category of the specific item. After MIC scored each item to the current user according to U2I, U2U, and I2I channels results, we choose top N items from V_C.

2.3 Overall Architecture

Figure 3 gives an overview of our proposed MIC model in each component. MIC is composed of 1) Perturbation Mining module: Perturbing data samples via Dropout Layer and Field Mask Embedding Layer, and retrieving similar samples via Nearest Neighbor Mining to construct contrastive positive pairs. 2) Encoder Module: Encoding the user and item features into inherent representations; Replaceable with existing encoders from retrieval baselines. 3) Cross-channel Contrastive module: Maximally leveraging the inherent mutual information in multiple channels via contrastive loss from user-user, item-item, and user-item space. In each channel module, the objective is to pull similar samples and push away dissimilar ones.

2.4 Perturbating and Mining

Contrastive learning method encourages positive pairs to have similar representations while negative pairs to have dissimilar representations. In the scenario of our unified framework, we consider both users and items as the anchor and generate pseudo views of each instance for comparison. We also leverage retrieved nearest neighbors to support the augmented sample views further.

2.4.1 Multi-level Perturbation. Data augmentation has been proved effective and widely used in contrastive prediction tasks without changing the architecture [4]. We devise a simple augmentation method to decouple from the neural network architecture. For users,

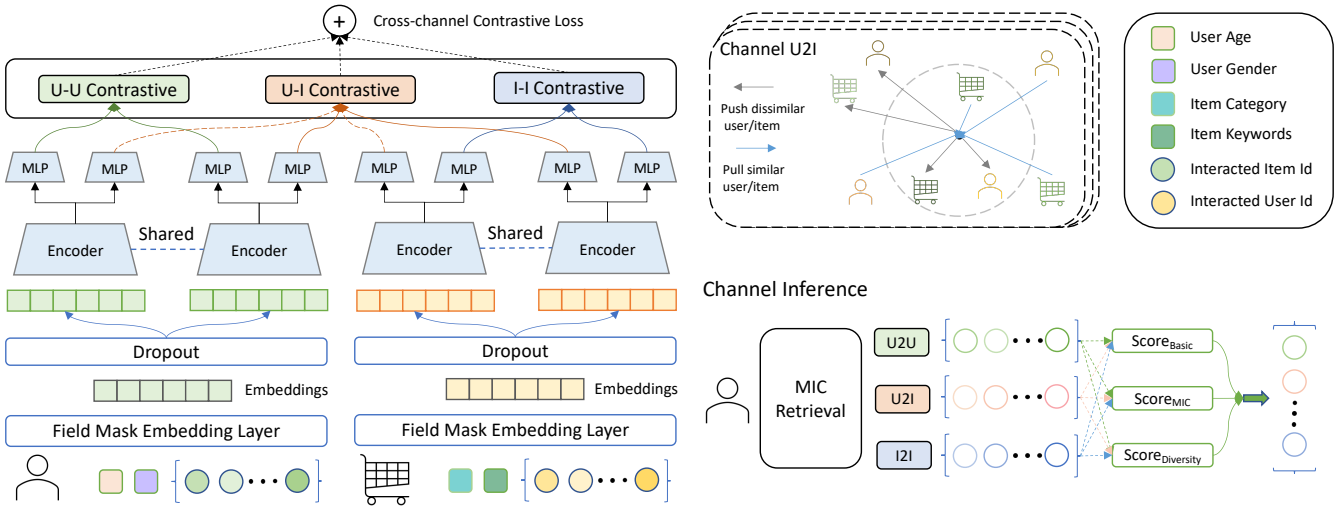

Figure 3: Overview of model-agnostic integrated cross-channel recommenders (MIC). The perturbations is performed in both field level and embeded features level. The user-item (U2I), user-user (U2U) and item-item (I2I) modules are aggregated to calculate cross-channel contrastive loss. In Inference stage, MIC applies aggregation over items retrieved from three channels and compute Score_{Basic}, Score_{MIC} and $\text{Score}_{Diversity}$ for final recommendation reference.

we randomly masked the user fields, including attributes (Id, gender, age) and interaction sequence (item Id). Similarly, we randomly masked attributes (item Id, keywords) and each item's interaction records (user Id). In addition to the field-level perturbations, the dropout is performed in the embedded features space. When only perturbation-based view augmentation is available, we treat the other $2(N-1)$ augmented examples within a minibatch as negative examples.

2.4.2 Nearest Neighbor Mining. We observe limited views generated by augmentation. First, view augmentation is limited to origin instance and fail to provide diversified samples. Second, effective augmentation is difficult to devise, refine, and evaluate in some scenarios. Finally, the augmentation method suffers from the balance between providing diversified views and maintaining semantic consistency.

In addition to augmentation, we argue that it's necessary to leverage information from a retrieval angle of view. For users, we retrieve the anchor user's k-nearest neighbor (kNN) in the representation space as the extension of user positive pairs. Besides, we adopt k-means++ to cluster the users and choose users from different clusters as hard negative samples. For items, both positive and hard negative samples are mined in the representation space in the same manner as users. At the interaction level, we use users to retrieve items and items to retrieve users. Before that, we project user and item representation in the same space. The same retrieval is then applied in this joint user-item representation space. Note that our sample selection pool is highly flexible. All the parameters, including the number of nearest-neighbor, number of clusters, and number of masked attributes, are tuned during training and adaptable to manual modification. Thus MIC maintains scalability and robust temporal efficacy in fast-speed transforming online changes.

2.5 Cross-channel Contrastive Estimation

Many works [18] directly optimize by forcing $click(u,v) = 1$ in diagonal and $click(u,v) = 0$ in other positions. However, these forcing methods assume the deterministic correlation between user and items, which is always not true in the real world. The real-world environment is always stochastic (*e.g.* diversified and dynamic user behaviors), where deterministic functions can only predict the average. On the other hand, contrastive estimation is an energy-based model. Instead of setting the cost function to be zero only when the prediction and the observation are the same, the energy-based model assigns low cost to all compatible prediction-observation pairs. Thus, the contrastive estimation can handle the stochasticity by its nature [26]. Inspired by recent contrastive learning algorithms [4], we propose to train these models by maximizing agreement between the anchor and augmented views via a contrastive loss. We randomly sample a minibatch of N user-item pairs (u,i). For the unified model, augmented users and items and the mined samples in the support set are defined as positive examples. Following SimCLR [4], we treat the other $2(N-1)$ real representation within a minibatch as negative examples. We use cosine similarity to denote the distance between two representation (u,v), that is $\texttt{sim}(u,v) = \mathbf{u}^T \cdot \mathbf{v}/||\mathbf{u}|| \cdot ||\mathbf{v}||$. The loss function for a positive pair of examples (u,v) is defined as:

$$\mathcal{L}_{uv} = -\log \frac{\exp(\texttt{sim}(u,v_i)/\tau)}{\sum_{\substack{j=1 \\ j\neq i}}^{N} \exp(\texttt{sim}(u,\tilde{v_j})/\tau)} - \log \frac{\exp(\texttt{sim}(v,u_i)/\tau)}{\sum_{\substack{j=1 \\ j\neq i}}^{N} \exp(\texttt{sim}(v,\tilde{u_j})/\tau)} \tag{6}$$

where τ denotes a temperature parameter that is empirically chosen as 0.1.

Similarly, for user-user and item-item model, the loss function for a positive pair of examples $(\tilde{u},u)$ and $(\tilde{v},v)$ is defined as:

$$\mathcal{L}_{uu} = -\log \frac{\exp(\texttt{sim}(u_k, \tilde{u_k})/\tau)}{\sum_{\substack{j=1 \\ j \neq k}}^{N} \exp(\texttt{sim}(u_k, u_j)/\tau)} \tag{7}$$

$$\mathcal{L}_{vv} = -\log \frac{\exp(\texttt{sim}(v, \tilde{v_i})/\tau)}{\sum_{\substack{j=1 \\ j \neq i}}^{N} \exp(\texttt{sim}(v, v_j)/\tau)} \tag{8}$$

The basic logistic loss by comparing the cosine similarity of users and items to predict y_i are computed as below:

$$\mathcal{L}_{basic} = -\frac{1}{N} \sum_{i} [y_i \log \hat{y_i} + (1 - y_i \log(1 - \hat{y_i}))] \tag{9}$$

2.6 Integrated Model

The user-item (U2I), user-user (U2U) and item-item (I2I) modules are aggregated to calculate cross-channel contrastive loss. We use the Adam optimizer to train our method. The objective function for training our model is to minimize the following cross-channel contrastive loss:

$$\mathcal{L} = \lambda \mathcal{L}_{basic} + (1 - \lambda)(\mathcal{L}_{uv} + \mathcal{L}_{vv} + \mathcal{L}_{uu}) \tag{10}$$

where λ is set to 0.7, each channel weight is $1:1:1$ after parameter optimization in our experiments. MIC can achieve the optimum trade-off across multiple channels by selecting the value of hyper-parameter λ and channel weight. During training, the total loss is computed across all positive pairs in a mini-batch.

2.7 Model-agnostic Plugin

MIC can also be treated as a plug-in to other matching stage recommenders by simply switching the encoder. MIC incorporate the perturbation and mining module in the item-side and add a cross-channel contrastive learning module on top of the retrieval baselines. Since the cross-channel learning module is independent of the encoders and the embedding layer is adaptable to sparse and dense features of users and items, MIC is highly flexible and achieves a model-agnostic performance boost in retrieving items from multiple channels efficiently.

2.8 Cross-channel Inference

During the inference phase of MIC, we get user and item representation from the user and item side encoder, respectively. For the U2I channel, we directly use the user vector to retrieve K_1 nearest neighbor from the whole item pool. For the U2U channel, we search M_1 similar users from the training dataset and retrieve K_2 items from M_1 similar users' history by considering the weight of similar users and user-item vector cosine similarity. For the I2I channel, we use the user's history to find M_2 relevant items within the whole item vector space for each history item and retrieve K_3 items by considering the weight of similar items and user-item vector cosine similarity. Finally, according to the final score in Equation 5, we rank top N items from multiple channels ($K_1 + K_2 + K_3$).

2.9 Online Deployment

We have deployed MIC on a well-known platform named Tencent News. Tencent News is one of the most popular news recommendation software, which has more than 300 million active users per month. The online architecture of Tencent News mainly consists of the retrieval stage and ranking stage widely used in the industry. The retrieval stage aims to quickly search hundreds of candidates from the entire news corpus (containing million-level news) efficiently, while the ranking stage aims to score news items accurately. MIC is deployed on the retrieval stage and an embedding-based recall model. We train and update our MIC model hourly.

Once MIC is trained, we infer all item vectors in the corpus and users' vectors of the current hour. Item vectors and user vectors are used to search similar items and users offline. Similarities of each item and user, user's interaction history are stored in Redis [1]. Item vectors are also used to build the item Faiss [2] search index. MIC is also served online for real-time user representation generating. When a user request comes, MIC first builds Redis key with userID and last M interaction items (we keep M_3 similar items for each item), then get M_1 similar users (we keep last M_2 clicked items for each user). For each similar user, MIC gets their clicked items from Redis. So we get $K_2 = M_1 \times M_2$ items from similar users (U2U) and $K_3 = M \times M_3$ items from similar items (I2I). We package real-time user features and generate user representation from MIC online serving then the representation is used to search top K_1 items from item Faiss index. Finally, the $K_1 + K_2 + K_3$ items are aggregated according to final score in Equation 5 and top N items are recalled. The fast nearest neighbor retrieval of Faiss search time is T1 (less than ten milliseconds). The time cost of similar users and items from Redis is T2 (less than ten milliseconds). The aggregation step time T3 (less than ten milliseconds). The whole time cost is acceptable for online serving in our system.

While our MIC model is updated hourly, we are still able to capture the real-time user preferences. We build real-time MIC request features including real-time click history for each user representation. User and corresponding similar users' interactions stored in Redis are also updated (in seconds) in real-time. So we can use real-time clicked items to get similar items from I2I channel and similar users' real-time interactions from U2U channel. Since the time cost is also related to similar users and similar items, the hyper-parameters should be adjusted to satisfy the online serving time requirements. In our system, $M_1 = 40$, $M_2 = 60$, $M_3 = 50$, $M = 30$ and $N = 200$.

3 EXPERIMENTS

In this section, we first cover the experimental settings of the dataset, evaluation metrics, parameter settings, and competitors. Then we report the results of extensive offline and online experiments with in-depth analysis to verify the effectiveness of MIC. We conduct experiments to investigate the following research questions:

- Research Question 1 (RQ1): How does MIC perform on large public recommendation datasets (Book, Taobao, Movielens, Steam)?
- Research Question 2 (RQ2): How does MIC perform in real-word News Recommendations System?
- Research Question 3 (RQ3): Are different components and losses essential in MIC?

[1] https://redis.io/
[2] https://github.com/facebookresearch/faiss

Table 1: Performance of four public datasets: Amazon Book, Taobao, Movielens and Steam. Results of three retrieval baselines and the proposed MIC are reported over three metrics: Recall, NDCG and Hit Rate. Gain represents the performance gain of X+MIC over vanilla X model.

Datasets	@N	Metrics	Baselines			X+MIC					
			DNN	Gru4Rec	ComiRec	DNN	**Gain**	Gru4Rec	**Gain**	ComiRec	**Gain**
Amazon Book	@20	Recall	5.608	5.877	6.634	5.934	**5.81%**	6.0141	**2.33%**	7.457	**12.41%**
		NDCG	5.371	5.835	6.023	5.836	**8.66%**	5.992	**2.69%**	6.195	**2.86%**
		Hit Rate	12.291	12.545	13.423	12.828	**4.37%**	12.997	**3.60%**	15.124	**12.67%**
	@50	Recall	8.885	8.908	10.2574	9.3066	**4.75%**	9.411	**5.65%**	11.55	**10.90%**
		NDCG	6.594	6.915	7.217	7.077	**7.32%**	7.105	**2.75%**	7.889	**9.31%**
		Hit Rate	18.709	18.949	19.231	19.373	**3.55%**	19.535	**3.09%**	22.790	**18.51%**
Taobao	@20	Recall	3.319	4.132	5.065	3.531	**6.39%**	4.442	**7.50%**	5.642	**11.39%**
		NDCG	12.493	15.449	19.324	13.481	**7.91%**	17.995	**16.48%**	21.221	**9.82%**
		Hit Rate	28.417	32.033	38.429	29.592	**4.13%**	36.661	**14.45%**	41.878	**8.97%**
	@50	Recall	5.075	6.118	7.115	5.278	**4.00%**	6.377	**4.23%**	7.861	**10.48%**
		NDCG	14.263	16.084	20.635	15.187	**6.48%**	18.999	**18.12%**	22.509	**9.08%**
		Hit Rate	39.31	42.114	48.094	40.324	**2.58%**	45.551	**8.16%**	51.607	**7.30%**
Movielens	@20	Recall	12.251	12.993	13.001	12.508	**2.10%**	13.012	**0.15%**	13.322	**2.47%**
		NDCG	36.249	37.033	37.207	36.898	**1.79%**	37.603	**1.54%**	38.186	**2.63%**
		Hit Rate	71.688	72.344	73.772	73.841	**3.00%**	74.308	**2.71%**	76.551	**3.77%**
	@50	Recall	23.028	24.447	25.043	23.875	**3.68%**	25.003	**2.27%**	25.927	**3.53%**
		NDCG	38.756	39.888	41.099	40.003	**3.22%**	41.309	**3.56%**	42.109	**2.46%**
		Hit Rate	87.245	89.705	90.138	88.907	**1.90%**	90.111	**0.45%**	91.391	**1.39%**
Steam	@20	Recall	2.901	2.672	2.753	3.117	**7.45%**	2.839	**6.25%**	3.009	**9.30%**
		NDCG	4.702	4.557	5.284	4.992	**6.17%**	5.703	**25.15%**	5.503	**4.14%**
		Hit Rate	10.308	9.928	11.044	10.554	**2.39%**	10.422	**4.98%**	11.333	**2.62%**
	@50	Recall	3.671	4.432	5.021	4.288	**16.81%**	4.775	**7.74%**	5.123	**2.03%**
		NDCG	5.077	4.997	6.23	5.779	**13.83%**	5.413	**8.32%**	6.671	**7.08%**
		Hit Rate	12.031	11.089	13.149	12.608	**4.80%**	12.307	**10.98%**	14.388	**9.42%**

- Research Question 4 (RQ4): How does MIC alleviate the seesaw phenomenon between retrieval accuracy and diversity: Can MIC achieve high retrieval accuracy and diversity simultaneously?
- Research Question 5 (RQ5): How does contrastive learning modules (UU,UI,II) help improve the embedding space and recall performance for corresponding U2U, U2I, I2I channel?

3.1 Dataset and Metric

We used four large benchmark datasets, Amazon Book, Taobao, Movielens, and Steam. The statistics are shown in 2.

- `Amazon Books`([17]): This dataset contains product reviews and metadata from Amazon, including 142.8 million reviews product metadata and links.
- `Steam`: This dataset contains more than 40k games from the steam shop with detailed data, including reviews and information about which games were bundled together.
- `Taobao`[43]: This dataset contains user behaviors recorded by Taobao recommendation system, consisting of users' clicks, item ID, item category, and timestamp.
- `Movielens-1M`[15]: One of the currently released MovieLens datasets, which contains 1,000,209 movie ratings from 6,040 users across 3,900 movies.

Table 2: Statistics of the Datasets.

Dataset	users	items	interactions
Amazon Books	459,133	313,966	8,898,041
Steam	2,567,538	15,474	7,793,069
Taobao	976,779	1,708,530	85,384,110
MovieLens-1M	6,040.	3,900	1,000,209

To compare the performance of different models, we use three metrics **Recall@N**, **NDCG@N**(Normalized Discounted Cumulative Gain), and **HR@N**, where N is set to 20, 50 respectively as metrics for evaluation. The details of our evaluation metrics are as below:

- `Recall`: Number of corrected recommended items divided by the total number of all recommended items.

$$Recall@N = \frac{1}{|U|} \sum_{u \in U} \frac{|\hat{I}_{u,N} \cap I_u|}{|I_u|} \tag{11}$$

where $\hat{I}_{u,N}$ denotes the set of top-N recommended items for user u and I_u is the set of testing items for user u.

- `Normalized Discounted Cumulative Gain`(NDCG): NDCG measures the percentage of correct recommended items, considering the positions of correct recommended items.

$$DCG@N = \frac{1}{|U|} \sum_{u \in U} \sum_{r \in R} \frac{\delta_N(r)}{log_2(i_r + 1)}, \tag{12}$$

$$NDCG@N = \frac{DCG@N}{IDCG@N} \tag{13}$$

  where i_r is the index of r in R. $\delta_N(\cdot)$ is an indicator function which returns 1 if item r is in top-N recommendation, otherwise 0. IDCG is the DCG of ideal ground-truth list which refers to the descending ranking of ground-truth list in terms of predicted scores.
- `Hit Rate`(HR): This measures the percentage of at least one item is correctly recommended to and interacted by corresponding user.

In all these three metrics, a higher value implies better performance. Besides, we adopt a per-user average for each metric. We track Recall, NDCG, Hit Rate of the Development split during training. Then we keep models with the best Recall Rate on Development split during experiments for a fair comparison.

3.2 Parameter Settings

We implement baselines and our proposed model in the same settings for fairness. The implementation is based on Tensorflow for offline experiments. The dimension of the collaborative embedding is set as 128. Batch size is set to 1024 on a single NVIDIA P40 GPU. The learning rate is set to 0.001, and the dropout rate is set to 0.2. The temperature parameter is empirically chosen as 0.1. We utilize Xavier and Adam algorithms in the experiments to initialize and optimize the parameters of the models.

3.3 Competitors

3.3.1 Retrieval Baselines. YoutubeDNN [7] is one of the predominant deep learning models based on collaborative filtering systems incorporating text and image information which have been successfully applied under the industrial scenario. Gru4Rec [19] is a session-based recommender using Recurrent Neural Networks. ComiRec [3] is a novel controllable multi-interest framework which can be used in sequential recommendation.

3.3.2 MIC as Plugin. As MIC is can also be treated as a model-agnostic plugin, we implement a series of variants with MIC adapted to other retrieval models denoted as $X + MIC$.

3.3.3 MIC Variants. Our unified model MIC co-learns user and item representation in both shared and their own semantic space. The retrieval model considers mutual information across multiple channels, including use-user, item-item, and user-item channel, simultaneously in an integrated framework.

In addition, we provide three representative variants as MIC-UI,MIC-UU, and MIC-II with single-channel contrastive loss. For MIC-UI, we add user-item contrastive training on top of ComiRec as a variant of our proposed MIC. This variant can capture the information behind the interaction and match the users to appropriate items from the user-item channel. For MIC-UU, we add user-user contrastive training on top of ComiRec as a variant of our proposed MIC. This variant is capable of clustering users and matching similar users to each other from the user channel. For MIC-II, we add item-item contrastive training on top of ComiRec as a variant of our proposed MIC. This variant is capable of clustering items and matching similar items to each other from the item channel. All compositional ablation results of each contrastive setting are reported in Table 4.

3.4 Model-agnostic Gain (RQ1)

The model performance for the retrieval stage recommender system is shown in Table 1. We conduct extensive experiments to dissect the effectiveness of our proposed model-agnostic integrated cross-channel (MIC) model. In the baseline performance comparison experiment, the MIC is implemented in a full mode with weighted UI, UU, and II contrastive loss. All these models are running on the four datasets introduced above: Amazon Book, Taobao, Movielens and Steam. We plug our MIC into prevalent retrieval baselins: : YouTube DNN, Gru4Rec and ComiRec.. As shown in Table 1, MIC enhanced models (X+MIC) consistently achieve a significant performance gain on all evaluation metrics than the retrieval baselines over four datasets. In particular, $ComiRec + MIC$ gain 10.90%, 9.31%, 18.51% over vanilla ComiRec model in Recall@50, NDCG@50 and Hit Rate@50 respectively over Amazon Book.

3.5 Online A/B Test(RQ2)

We further conduct an online A/B test to evaluate MIC in real-world scenarios. We have deployed MIC on Tencent News Video Recommendation scenarios as stated in Sec 2.9. MIC is deployed as a matching model in the retrieval stage, with the remained modules in the whole system unchanged. The online recall baseline is an ensemble model containing tens of retrieval models (embedding-based, rule-based, hot-based, etc.). In the online A/B test, we focus on four metrics, including the Exposure Page Viewed Ratio (EPV), Average Play Percentage of each viewed video, Average Duration, and Average Viewed Video of each user in our platform daily. The A/B test was conducted from October 1st, 2021 to October 15th, 2021, and the user number in the experiments group and baseline group is about 1 million. The experimental scenario is Tencent News Video recommendation. We report the improvements percentages of MIC in Table 3 from which we can know that: 1) MIC achieves significant improvements on Average Viewed Video and Average Duration, which means the recommended videos are more attractive to each user. At the same time, the Average Play Percentage of each video is also improved, which means that MIC provides more precise video to users. 2) The EPV ratio of MIC is about 25%, the most effective recall model among all models (the second place recall model's EPV ratio is about 8%).

3.6 Ablation Study (RQ3)

We conduct ablation experiments of contrastive loss modules and inference channel modules for our proposed MIC enhanced ComiRec [3]. Results of variants with various cross-channel contrastive loss settings and various inference channels settings over Amazon Book are reported in Table 4. −UU, −UI, −II represents the Full Model without U-U, U-I, I-I contrastive modules respectively. −Perturbation

Table 3: Online A/B Test Results. We report the relative performance gain of MIC over Baseline in online A/B experiments.

#Scenario	EPV ratio	Average Play Percentage ↑	Average Duration ↑	Average Viewed Video ↑
Video Recommendation	25.00%	+3.51%	+1.26%	+1.85%

Table 4: Ablation Performance of MIC Variants over ComiRec on Amazon Book dataset with Metric@50.

Modules	Settings	Recall	NDCG	Hit Rate	Diverstiy
	Full Model	11.554	7.889	22.790	49.511
Contrastive Loss	-UU	10.556	7.689	21.132	44.021
	-UI	10.347	6.462	21.273	42.483
	-II	11.096	7.089	22.668	46.796
	-Perturbation	8.415	5.346	16.590	34.188
	-Mining	10.176	6.098	20.727	41.983
Inference Channel	-U2U channel	11.148	7.688	22.076	45.478
	-U2I channel	11.484	7.825	22.571	45.603
	-I2I channel	11.316	7,758	22.443	41.709

and −Mining represents the Full Model without perturbation and Nearest Neighbor Mining module. −U2U, −U2I, −I2I represents the Full Model without consideration of retrieved items from U2U, U2I, I2I channel respectively during inference. We observe performance drop over Recall@50, NDCG@50, HitRate@50 and Diversity in these variants compared with Full Model in Table 1. This implies the essential role of each module setting in the Full Model.

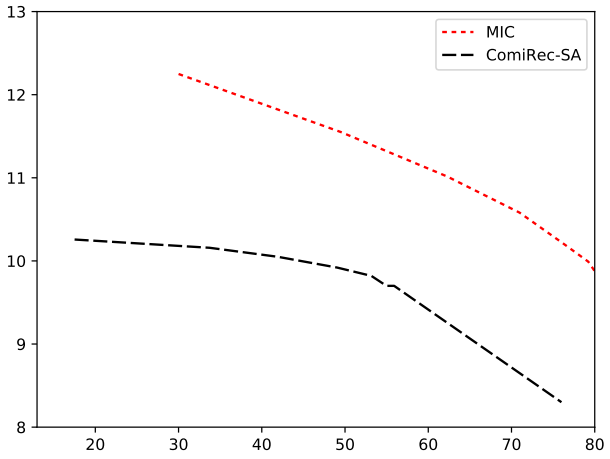

Figure 4: Retrieval Accuracy and Diversity Balance. We compare ComiRec-SA (Black) and MIC enhanced ComiRec-SA (Red) over Amazon Book with Recall@50 (x-axis) and Diversity (y-axis).

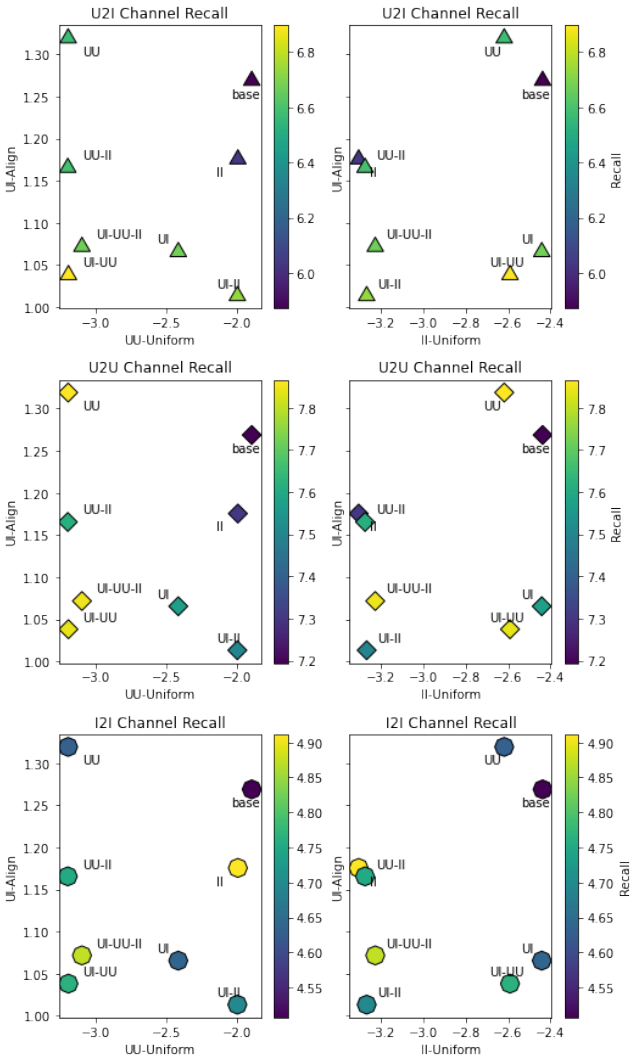

Figure 5: Visualization of User and Item Representation in U2I, U2U and I2I channel over Alignment and Uniformity Metrics of UI-Align, UU-Uniform and II-Uniform and Recall Performance.

3.7 Retrieval Accuracy and Diversity (RQ4)

There is a Seesaw Effect between retrieval performance and retrieval diversity. We can also observe in Comirec that a better diversity score degrades Recall. To mitigate this phenomenon, MIC aggregates retrieved items from three channels (U2U, U2I, I2I). To investigate whether MIC mitigates this, we conduct experiments to compare MIC and ComiRec on the Amazon Book dataset. Results are visualized in Figure 4. We can observe that MIC (Red Line) achieves consistent retrieval performance and diversity gain over

ComiRec (Black Line). This indicates that MIC successfully leverages the information to simultaneously improve retrieval performance and diversity. MIC alleviates the Seesaw Effect and achieves the balance between retrieval accuracy and diversity.

3.8 Qualitative Results (RQ5)

While we care about the integrated cross-channel performance of MIC, we still want to see how does contrastive learning modules (UU,UI,II) help improve the embedding space and recall performance for corresponding U2U, U2I, I2I channel. We analyze the agreement between user representations, item representations, and final recall performance by the Alignment and Uniformity Metrics [34] (lower is better) of UI-Align, UU-Uniform, and II-Uniform. UI-Align measures the alignment between user and target item representation, UU-Uniform and II-Uniform measure the uniformly distributing of user and item representation, respectively. As shown in Figure 5, bright yellow denotes better Recall performance. Each point is marked with corresponding contrastive settings: UI-UU-II means three contrastive learning objects were added, and Base means none contrastive learning objects were considered. For U2I Channel (first row in Figure 5), the Recall performance is very sensitive to UI-align, and in no doubt, UI-align gets better when UI contrastive learning is considered. For U2U Channel (second row), UU-Uniform starts to play more important roles besides UI-align. We can find the best recall scores in the bottom left of the "UI-Align, UU-Uniform" graph in U2U Channel Recall. Besides, U2U-Uniform would be better if we added contrastive learning between users. For I2I Channel (third row), II-Uniform senses to be more important than UI-Align. The "UI-align, II-Uniform" graph shows that the best Recall appears in the lowest II-Uniform other than the lowest UI-align. We observe that if we can simultaneously acquire more aligned user-item representation, and more uniformed user-user, item-item representations, we can push the integrated model's U2I, U2U, and I2I channel performance to the next stage. MIC is one of this type of model-agnostic integrated cross-channel model for recommendations.

4 RELATED WORKS

4.1 Recommendation

Recommendation system can be divided into mainly two categories, content-based recommendation and collaborative filtering. Collaborative filtering techniques is composed of user-based algorithms [39], item-based algorithms [9] and model-based algorithms [23]. Previous studies [18, 40, 42] achieve significant progress based on the idea of user modeling and collaborative recommendation.

Besides collaborative filtering, content-based filtering (e.g. DSSM [21]) is another critical class of recommender systems. Pure content-based only rely on the feature of users and items, thus ignoring the common preferences shared among similar users and common properties among similar items. With the emergence of distributed representation learning, user embeddings obtained by neural networks are widely used. [5] employs RNN-GRU to learn user embeddings from the temporal ordered review documents. [31] utilizes Stacked Recurrent Neural Networks to capture the evolution of contexts and temporal gaps. [12] proposes the framework GraphRec to jointly capture interactions and opinions in the user-item graph. Due to the intrinsic drawback of both pure content-based and collaborative recommendations, the hybrid model concept is proposed to combine them and benefit each other. Commonly used hybrid recommendation algorithms include weighted hybrid recommendation algorithm, cross-harmonic recommendation algorithm, and meta-model mixed recommendation algorithm [2]. Dai *et al.*proposed a dynamic recommendation algorithm [8] that combines the convolutional neural network and multivariate point process by learning the co-evolutionary model of user-commodity implied features. Nevertheless, though these hybrid algorithms seek to combine multi-source data, they failed to consider user-user, item-item, and user-item coevolution and relatedness in a unified framework.

4.2 Contrastive Learning

Contrastive Learning is a framework to learn representations that obey similarity constraints in a dataset typically organized by similar and dissimilar pairs. Hadsell et al. [14] first proposed to learn representations by contrasting positive pairs against negative pairs. Some studies [32, 35, 37] utilize a memory bank to store the instance class representation vector. Other work explored the use of in-batch samples for negative sampling instead of a memory bank [10, 22, 37] Recently, SimCLR [4] and MoCo [6, 16] achieved state-of-the-art results in self-supervised visual representation learning, closing the gap with supervised representation learning. Contrastive training is further explored in visual representation learning [30, 33, 38] and views mining [1, 11]. Leveraging nearest sample to produce pro views of sample mining is also proved effective in machine translation [41] and language models [24]

5 CONCLUSION

In this paper, we propose a model-agnostic integrated cross-channel (MIC) approach, semantically connecting users and items for the matching stage of a typical industrial recommender system by maximally leveraging the inherent multi-channel mutual information. Specifically, MIC models correlation across user-item (U2I), user-user (U2U), and item-item (I2I) channels via intra and inter cross-channel contrastive modules. MIC naturally aligns users and items with semantic similarity and distinguishes them otherwise in each channel. Extensive experiments show that our MIC helps several popular retrieval models boost performance on four real-world benchmarks. By deploying on industrial Tencent News platform with millions of users and conducting online experiments, we confirm the scalability and flexibility of the proposed method.

REFERENCES

[1] Mehdi Azabou, Mohammad Gheshlaghi Azar, Ran Liu, Chi-Heng Lin, Erik C. Johnson, Kiran Bhaskaran-Nair, Max Dabagia, Keith B. Hengen, William Gray-Roncal, Michal Valko, and Eva L. Dyer. 2021. Mine Your Own vieW: Self-Supervised Learning Through Across-Sample Prediction. *ArXiv* abs/2102.10106 (2021).

[2] Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. 2012. TasteWeights: a visual interactive hybrid recommender system. In *RecSys '12*.

[3] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable Multi-Interest Framework for Recommendation. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2020).

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 1597–1607. http://proceedings.mlr.press/v119/chen20j.html

[5] T. Chen, R. Xu, Y. He, Y. Xia, and X. Wang. 2016. Learning User and Product Distributed Representations Using a Sequence Model for Sentiment Analysis. *IEEE Computational Intelligence Magazine* 11, 3 (2016), 34–44.

[6] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. 2020. Improved Baselines with Momentum Contrastive Learning. *ArXiv* abs/2003.04297 (2020).

[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep Neural Networks for YouTube Recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. New York, NY, USA.

[8] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Recurrent Coevolutionary Latent Feature Processes for Continuous-Time Recommendation. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems* (Boston, MA, USA) *(DLRS 2016)*. Association for Computing Machinery, New York, NY, USA, 29–34. https://doi.org/10.1145/2988450.2988451

[9] Mukund Deshpande and George Karypis. 2004. Item-based top-N recommendation algorithms. *ACM Trans. Inf. Syst.* 22 (2004), 143–177.

[10] Carl Doersch and Andrew Zisserman. 2017. Multi-task self-supervised visual learning. In *ICCV*.

[11] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. 2021. With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations. arXiv:2104.14548 [cs.CV]

[12] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 417–426. https://doi.org/10.1145/3308558.3313488

[13] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 1999. Similarity Search in High Dimensions via Hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB '99)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 518–529.

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, Vol. 2. IEEE, 1735–1742.

[15] F. M. Harper and J. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5 (2015), 19:1–19:19.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.

[17] Ruining He and Julian McAuley. 2016. Ups and Downs. *Proceedings of the 25th International Conference on World Wide Web* (Apr 2016). https://doi.org/10.1145/2872427.2883037

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. *Proceedings of the 26th International Conference on World Wide Web* (2017).

[19] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. *CoRR* abs/1511.06939 (2016).

[20] 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. ACM International Conference on Information and Knowledge Management (CIKM). https://www.microsoft.com/en-us/research/publication/learning-deep-structured-semantic-models-for-web-search-using-clickthrough-data/

[21] 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. *Proceedings of the 22nd ACM international conference on Information & Knowledge Management* (2013).

[22] Xu Ji, João F Henriques, and Andrea Vedaldi. 2019. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*. 9865–9874.

[23] Zhenyan Ji, Weina Yao, Wei Wei, Houbing Song, and Huaiyu Pi. 2019. Deep Multi-Level Semantic Hashing for Cross-Modal Retrieval. *IEEE Access* 7 (2019), 23667–23674.

[24] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through Memorization: Nearest Neighbor Language Models. In *International Conference on Learning Representations (ICLR)*.

[25] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 32)*, Eric P. Xing and Tony Jebara (Eds.). PMLR, Bejing, China, 1188–1196. https://proceedings.mlr.press/v32/le14.html

[26] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. 2006. A tutorial on energy-based learning. *Predicting structured data* 1, 0 (2006).

[27] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Pipei Huang, Huan Zhao, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-Interest Network with Dynamic Routing for Recommendation at Tmall. arXiv:1904.08030 [cs.IR]

[28] Houyi Li, Zhihong Chen, Chenliang Li, Rong Xiao, Hongbo Deng, Peng Zhang, Yongchao Liu, and Haihong Tang. 2021. Path-based Deep Network for Candidate Item Matching in Recommenders. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021).

[29] Yujie Lu, Sheng-Yu Zhang, Yingxuan Huang, Luyao Wang, Xinyao Yu, Zhou Zhao, and Fei Wu. 2021. Future-Aware Diverse Trends Framework for Recommendation. *Proceedings of the Web Conference 2021* (2021).

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*.

[31] Lakshmanan Rakkappan and Vaibhav Rajan. 2019. Context-Aware Sequential Recommendations WithStacked Recurrent Neural Networks. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 3172–3178. https://doi.org/10.1145/3308558.3313567

[32] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive Multiview Coding. In *ECCV*.

[33] Jianren Wang, Yujie Lu, and Hang Zhao. 2020. CLOUD: Contrastive Learning of Unsupervised Dynamics. arXiv:2010.12488 [cs.RO]

[34] Tongzhou Wang and Phillip Isola. 2020. Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. In *International Conference on Machine Learning*. PMLR, 9929–9939.

[35] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*. 3733–3742.

[36] Ruobing Xie, Shaoliang Zhang, Rui Wang, Feng Xia, and Leyu Lin. 2021. Hierarchical Reinforcement Learning for Integrated Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 5 (May 2021), 4521–4528. https://ojs.aaai.org/index.php/AAAI/article/view/16580

[37] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*. 6210–6219.

[38] Xin Yuan, Zhe L. Lin, Jason Kuen, Jianming Zhang, Yilin Wang, Michael Maire, Ajinkya Kale, and Baldo Faieta. 2021. Multimodal Contrastive Training for Visual Representation Learning. In *CVPR*.

[39] Zhi-Dan Zhao and Mingsheng Shang. 2010. User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop. *2010 Third International Conference on Knowledge Discovery and Data Mining* (2010), 478–481.

[40] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (2017).

[41] Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021. Adaptive Nearest Neighbor Machine Translation. In *ACL/IJCNLP*.

[42] Yin Zheng, Bangsheng Tang, Wenkui Ding, and Hanning Zhou. 2016. A Neural Autoregressive Approach to Collaborative Filtering. In *ICML*.

[43] Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. 2018. Learning Tree-Based Deep Model for Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 1079–1088. https://doi.org/10.1145/3219819.3219826