PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

Contrastive Flow Matching for Sequential Recommendation

Wangyu Jin[1] , **Yang Xue**[1] , **Wenwen Xia**[1] , **Hongliang He**[1] , **Guanfeng Liu**[2] and **Pengpeng Zhao**[1,*]
[1]School of Computer Science and Technology, Soochow University, China
[2] School of Computing, Macquarie University, Australia
20254027012@stu.suda.edu.cn, 20255227016@stu.suda.edu.cn, wwxia@suda.edu.cn, hlhe2023@suda.edu.cn, guanfeng.liu@mq.edu.au, ppzhao@suda.edu.cn

Abstract

Flow Matching (FM), as a highly promising generative paradigm, has recently been introduced into sequential recommendation. However, the lack of explicit inter-user supervision in existing FM-based method may lead to an averaged flow phenomenon, where the model tends to produce similar velocity directions for different users during item generation. This greatly undermines the model's ability to generate personalized item. To address this issue, we propose a **Con**trastive **F**low **M**atching (ConFM) for sequential recommendation, aiming to inject inter-user separation signals so as to preserve the discriminability of users' velocity directions. Specifically, considering the common design where velocities are derived from predicted item representations, we introduce a Representation Contrastive Regularizer in the representation space, which forces the predicted item embeddings of different users to be separated from each other. In addition, we introduce a Velocity Contrastive Regularizer in the induced velocity space, which directly encourages each user's induced velocity to stay far from other users' velocity targets, thereby better aligning the generation process with the user's own interests. Extensive experiments on four datasets demonstrate that ConFM consistently outperforms strong baselines, and it also achieves notable improvements on long-tail items.

1 Introduction

Sequential recommendation (SR) [Liu *et al.*, 2021; Wang *et al.*, 2022] stands as a cornerstone of modern e-commerce [Wang *et al.*, 2020] and content platforms [Schedl *et al.*, 2018; Hansen *et al.*, 2020], which aims to predict the next item by modeling users' personalized preferences based on their historical interaction sequences. Recently, generative sequential recommendation [Ni *et al.*, 2023; Yuan *et al.*, 2019] has emerged as a promising paradigm that goes beyond discriminative ranking by learning a history-conditioned distribution over next items and generating item representations for retrieval.

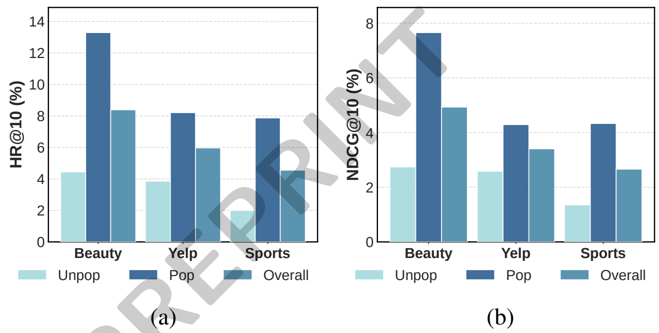

Figure 1: Popularity-stratified diagnosis results of FM4Rec on three datasets. (a) HR@10 and (b) NDCG@10 for Overall, pop, and unpop groups. We split test instances by the popularity of the target item, where *pop* denotes targets within the top 20% most frequent items in the training set and *unpop* denotes the remaining ones; *Overall* reports the full test set.

Among generative approaches, Diffusion Models (DMs) [Ho *et al.*, 2020; Song *et al.*, 2021] have garnered significant attention due to their powerful distribution modeling capabilities. DM-based methods such as DiffuRec [Li *et al.*, 2023], DreamRec [Yang *et al.*, 2023], and InDiRec [Qu and Nobuhara, 2025] cast recommendation as a history-conditioned denoising process that corrupts the target item embedding into noise and then reconstructs it in multiple reverse steps. Despite their success, DM-based recommenders rely on noise-perturbed representations and stochastic multi-step sampling, which may blur fine-grained user intent and introduce additional randomness into generation.

Recently, Flow Matching (FM) [Lipman *et al.*, 2023], due to its stable deterministic generation, has been explored for SR. A representative method, FM4Rec [Liu *et al.*, 2025b] trains a shared history-conditioned predictor on interpolation states between noise and the target next-item embedding, and uses it to define the velocity field for deterministic transport. At inference, starting from noise, the model follows the induced velocity field to progressively obtain a next-item representation for retrieval.

However, existing FM-based recommendation method typ-

* Corresponding author.

ically lacks explicit inter-user separation signals. This deficiency raises a critical concern regarding the model's ability to accurately capture unique user interests and generate personalized items that satisfy individual preferences. Our empirical study on FM4Rec across four datasets validates this concern. As shown in Figure 1, the model performs consistently well on popular targets but deteriorates substantially on unpopular ones. This indicates that, limited by its personalization capacity, the model tends to generate common patterns for all users.

This tendency can be attributed to a critical phenomenon in flow matching known as *averaged flow*, where the learned velocity field (and thus its implied velocity directions) becomes overly similar across different users. Specifically, FM-based models construct an interpolated state by fusing the target item with noise, and utilize a shared predictor to estimate the item representation at this state. Crucially, this estimated representation is then used to estimate the corresponding velocity field for transport. However, due to the lack of inter-user separation guidance, the shared predictor tends to yield overly similar predicted representations for different users. Subsequently, this similarity propagates to the velocity space, causing the derived velocity directions to become averaged across users. Consequently, during inference, these averaged velocities drive the generation toward generic outcomes for all users, resulting in a severe loss of personalization.

To address the above limitations, we propose **Con**trastive **F**low **M**atching for sequential recommendation (**ConFM**). Motivated by the causes of averaged flow, ConFM introduces inter-user separation signals from two perspectives, enabling the model to better preserve user personalization. Specifically, to tackle the issue that predicted item representations at interpolation states become insufficiently distinguishable, we design a Representation Contrastive Regularizer (RCR) that explicitly pushes different users' predicted item representations apart, thereby better retaining user-specific preferences. In addition, we propose a Velocity Contrastive Regularizer (VCR), which directly enforces inter-user separation in the induced velocity space, ensuring that each user follows a distinct generation trajectory. By enforcing user separation in both the representation space and the induced velocity space, ConFM produces more user-specific item generation at inference, thereby enhancing personalized recommendation.

The main contributions of this work are summarized as follows:

- We identify the flow averaging issue in FM-based sequential recommendation through diagnostic experiments and analyze that it arises from the lack of explicit inter-user separation constraints.
- We propose a contrastive flow matching (ConFM) for sequential recommendation, which explicitly introduces inter-user separation regularizer in both the representation space and the velocity space to preserve personalization in generated recommendations.
- Extensive experiments on four public datasets show that ConFM consistently outperforms competitive baselines, with particularly significant gains on long-tail items.

2 Related Work

2.1 Sequential Recommendation

Sequential recommendation aims to predict the next item by learning user preferences from historical interaction sequences. In early work, recurrent neural networks were introduced to sequence modeling. The representative method GRU4Rec [Hidasi *et al.*, 2016] employs gated recurrent units to encode user interaction sequences, regarding the generated hidden states as representations of current preferences. In recent years, to more accurately capture dependencies within user behavioral sequences, self-attention has been widely explored in sequential recommendation. SASRec [Kang and McAuley, 2018] and STOSA [Fan *et al.*, 2022] respectively introduce causal self-attention and structure-aware attention, which better capture the evolving patterns in user interaction sequences. BERT4Rec [Sun *et al.*, 2019] learns sequence representations through introducing bidirectional transformers and masked word prediction. To mitigate data sparsity, contrastive learning has also been introduced to sequence recommendation. S^3-Rec[Zhou *et al.*, 2020] combines multiple self-supervised pre-training tasks to learn more robust sequence representations. CL4SRec [Xie *et al.*, 2022] constructs positive pairs through data augmentation and uses other sequences as negative samples for contrastive learning, thereby promoting discriminative sequence representations.

2.2 Generative Sequential Recommendation

Generative models have begun to be widely used in sequential recommendations with their advantages in modeling uncertainty and capturing the evolution of user interests. The existing methods can generally be classified into two representative routes: one is based on the diffusion model, and the other is based on flow matching.

Diffusion Model-based Methods. Diffusion models have been successfully adapted to sequential recommendation in several recent studies. Early diffusion-based recommenders, such as DiffuRec [Li *et al.*, 2023] and DreamRec [Yang *et al.*, 2023], typically cast next-item prediction as a history-conditioned denoising process. Recently, PreferDiff [Liu *et al.*, 2025c] further improves training efficiency and preference modeling, while DiQDiff [Mao *et al.*, 2025] alleviates the data bias that diffusion models may suffer from during generation. However, diffusion-based recommenders still rely on stochastic, multi-step sampling at inference time, which is computationally expensive and can introduce generation variance that blurs fine-grained user intent. This drawback motivates more efficient and stable generative alternatives, such as flow matching.

Flow Matching-based Methods. Flow matching learns a continuous-time velocity field to deterministically transport samples from a noise prior toward the target distribution [Lipman *et al.*, 2023]. Building on this idea, it has been progressively introduced into social graph recommendation [Huang *et al.*, 2025] and collaborative filtering [Liu *et al.*, 2025a]. In sequential recommendation, FM4Rec [Liu *et al.*, 2025b] performs history-conditioned deterministic transport to generate next-item representations. However, FM4Rec shares a

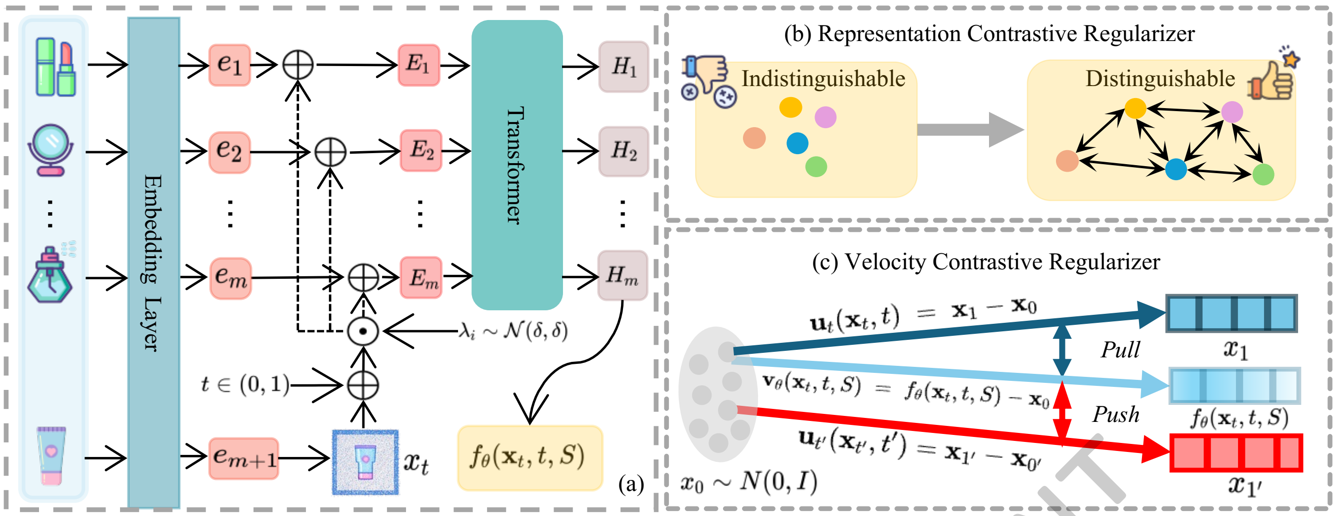

Figure 2: The overall framework of the ConFM.

single estimator across users. As a result, it provides limited objective-level encouragement for user-specific diversity. Therefore, more discriminable models are desirable.

3 Preliminaries

3.1 Problem Formulation

In sequential recommendation, we consider the user set $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ and the item set $\mathcal{I} = \{i_1, i_2, \ldots, i_{|\mathcal{I}|}\}$, where $|\mathcal{U}|$ and $|\mathcal{I}|$ denote the numbers of users and items, respectively. For each user $u \in \mathcal{U}$, the interaction history is an ordered sequence $S = (i_1, i_2, \ldots, i_m)$ of length m. Given S, we recommend the next item by

$$\hat{i}_{m+1} = \arg\max_{i \in \mathcal{I}} \Theta(i \mid S). \tag{1}$$

where $\Theta(i \mid S)$ denotes a history-conditioned scoring function (or evaluation metric) for item i.

3.2 Flow Matching

Flow Matching (FM) is a recent continuous-time generative paradigm that learns a transport from a simple base distribution to the target distribution via a velocity field.

Formally, let $\mathbf{x}_0 \sim p_0$ be a sample from the source distribution, and let $\mathbf{x}_1 \sim p_1$ be a sample from the target distribution. FM constructs an intermediate state at time $t \in [0, 1]$ via linear interpolation:

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\mathbf{x}_1. \tag{2}$$

Along this linear path, the ground-truth velocity is

$$\mathbf{u}_t = \mathbf{x}_1 - \mathbf{x}_0. \tag{3}$$

FM trains a time-dependent velocity field $\mathbf{v}_\theta(\mathbf{x}_t, t)$ by regression:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_1, t}\left[\|\mathbf{v}_\theta(\mathbf{x}_t, t) - \mathbf{u}_t\|_2^2\right]. \tag{4}$$

At inference, we draw an initial sample $\mathbf{x}_0 \sim p_0$ and generate $\hat{\mathbf{x}}_1$ by solving the ODE $d\mathbf{x}_t = \mathbf{v}_\theta(\mathbf{x}_t, t)\, dt$, which is typically integrated using a deterministic Euler method.

4 Methodology

The proposed ConFM framework is motivated by a simple challenge in FM-based sequential recommendation: due to the lack of inter-user separation guidance, a shared predictor tends to produce similar predicted representations, which further leads to averaged velocity directions and ultimately weakens personalized prediction. To address this, ConFM introduces two inter-user contrastive regularizers to the FM-based backbone. The first is the *Representation Contrastive Regularizer*, which promotes mutual distinguishability among users' predicted next-item embeddings within each mini-batch. The second is the *Velocity Contrastive Regularizer*, which pushes each user's induced velocity away from the ground-truth velocity targets of randomly sampled other users. These two regularizers explicitly enforce inter-user separation in both the representation and velocity spaces, enhancing the model's ability to capture user-specific preferences. Figure 2 illustrates the overall framework of ConFM.

4.1 Backbone method of ConFM

In this subsection, we first briefly review FM4Rec [Liu *et al.*, 2025b], the backbone model upon which our ConFM is built, which pioneered the introduction of flow matching to sequential recommendation. FM4Rec learns a history-conditioned generation mechanism that transforms a noisy intermediate state into a next-item representation for ranking. It comprises three key steps, detailed as follows.

Interpolation state construction. For each training instance, given the interaction history $S = (i_1, \ldots, i_m)$ and the target item i_{m+1}, the backbone takes the target item embedding as the target endpoint $\mathbf{x}_1 = \text{Emb}(i_{m+1}) \in \mathbb{R}^d$, where $\text{Emb}(\cdot)$ denotes a trainable embedding layer of item ID. The model then samples a noise endpoint $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and draw $t \sim \text{Uniform}(0, 1)$.

Accordingly, the interpolated state on the straight path between these two endpoints is defined as

$$\mathbf{x}_t = (1 - t)\,\mathbf{x}_0 + t\,\mathbf{x}_1. \tag{5}$$

Finally, the tuple $(\mathbf{x}_t, t, S)$ is fed into the backbone network to produce the next-item representation for training.

Ground-truth velocity and FM objective. Given the interpolation in Eq. (5), the ground-truth velocity along the straight path is

$$\mathbf{u}_t(\mathbf{x}_t, t) \;=\; \mathbf{x}_1 - \mathbf{x}_0. \tag{6}$$

Following the FM4Rec-style instantiation, the history-conditioned predictor generates an endpoint estimate $\hat{\mathbf{x}}_1 = f_\theta(\mathbf{x}_t, t, S)$. Instead of employing a separate head for velocity regression, the predicted velocity field is defined as the induced displacement from the noise endpoint:

$$\mathbf{v}_\theta(\mathbf{x}_t, t, S) \;=\; f_\theta(\mathbf{x}_t, t, S) - \mathbf{x}_0. \tag{7}$$

The model is then optimized via the standard flow matching regression objective, which minimizes the difference between the predicted and ground-truth velocities:

$$\begin{aligned}\mathcal{L}_{\mathrm{FM}} &= \mathbb{E}_{(S,i_{m+1}),\,\mathbf{x}_0,\,t}\Big[\big\|\mathbf{v}_\theta(\mathbf{x}_t, t, S) - \mathbf{u}_t(\mathbf{x}_t, t)\big\|_2^2\Big] \\ &= \mathbb{E}_{(S,i_{m+1}),\,\mathbf{x}_0,\,t}\Big[\big\|f_\theta(\mathbf{x}_t, t, S) - \mathbf{x}_1\big\|_2^2\Big],\end{aligned} \tag{8}$$

Predictor parameterization. To instantiate the history-conditioned predictor $f_\theta(\mathbf{x}_t, t, S)$, a causal Transformer backbone is adopted that conditions each historical token on the interpolation state $\mathbf{x}_t$ and the time t.

Let $S = (i_1, \ldots, i_m)$ denote the interaction history. For the k-th interacted item i_k, its embedding is denoted by $\mathbf{e}_k = \mathrm{Emb}(i_k)$. The input at position k is formed by fusing $\mathbf{e}_k$ with the injected state:

$$\mathbf{E}_k(\mathbf{x}_t, t) = \mathbf{e}_k + \boldsymbol{\lambda}_k \odot \big(\mathbf{x}_t + \mathbf{t}\big), \boldsymbol{\lambda}_k \sim \mathcal{N}(\delta, \delta), \tag{9}$$

where $\odot$ denotes element-wise multiplication, $\mathbf{t}$ is a learned embedding of the scalar time t, and δ controls the magnitude of the stochastic injection.

A stack of n_1 causal self-attention blocks is then applied to encode the noised sequence and obtain contextualized hidden states:

$$[\mathbf{h}_1, \ldots, \mathbf{h}_m] = \mathrm{Dec}_1\Big([\mathbf{E}_1(\mathbf{x}_t, t), \ldots, \mathbf{E}_m(\mathbf{x}_t, t)]\Big). \tag{10}$$

On top of these states, an additional n_2-layer stack is applied, and the last-token output is used as the predictor output

$$f_\theta(\mathbf{x}_t, t, S) = \mathrm{LAST}(\mathrm{Dec}_2([\mathbf{h}_1, \ldots, \mathbf{h}_m]))\,. \tag{11}$$

The output $f_\theta(\mathbf{x}_t, t, S)$ is taken as the predicted next-item representation $\hat{\mathbf{x}}_1$.

4.2 Representation Contrastive Regularizer

As discussed in Section 1, for FM-based methods that derive the velocity field by predicting the next item at interpolated states, the root cause of the average flow issue—which degrades personalization—is that all users share a single predictor without inter-user separation constraints. Even when the input is conditioned on user history, the model tends to output an average pattern across users to minimize the global training loss. Consequently, the predicted embeddings for many users become increasingly similar, leading to a severe degradation in representation separability.

To explicitly enforce inter-user distinguishability, we introduce a Representation Contrastive Regularizer (RCR). Specifically, RCR operates within each mini-batch by penalizing high similarities between the predicted item embeddings of different users, thereby preventing the shared predictor from collapsing into a generic pattern.

Formally, consider a mini-batch of B training instances $\{(S_b, i_{m+1,b})\}_{b=1}^{B}$. For each instance b, we sample $\mathbf{x}_{0,b} \sim \mathcal{N}(0, \mathbf{I})$ and $t_b \sim \mathrm{Uniform}(0,1)$, construct $\mathbf{x}_{t,b}$ by Eq. (5), and obtain the predicted next-item representation

$$\hat{\mathbf{x}}_{1,b} = f_\theta(\mathbf{x}_{t,b}, t_b, S_b). \tag{12}$$

We then define an inter-user similarity penalty using a log-sum-exp aggregator, and the RCR term is defined as

$$\mathcal{L}_{\mathrm{RCR}} = \frac{1}{B}\sum_{b=1}^{B} \log\left(\sum_{b'=1, b'\neq b}^{B} \exp(\mathrm{sim}(\hat{\mathbf{x}}_{1,b}, \hat{\mathbf{x}}_{1,b'}))\right). \tag{13}$$

where $\mathrm{sim}(\cdot,\cdot)$ denote a similarity function, such as cosine similarity. Minimizing $\mathcal{L}_{\mathrm{RCR}}$ encourages each $\hat{\mathbf{x}}_{1,b}$ to be less similar to the other predictions in the mini-batch. As a result, the model tends to push the predicted embeddings of different users farther apart, making the predictions easier to distinguish.

4.3 Velocity Contrastive Regularizer

RCR promotes separation among prediction embeddings, which can be viewed as an indirect way of separating velocities through representations. However, this remains insufficient because a user's velocity vector is not determined by the prediction alone. Instead, it is computed jointly from the prediction and a randomly sampled state. Given the random sampling of interpolation states, separation only in the embedding space does not guarantee that users' velocity directions are also well separated.

To this end, we propose the Velocity Contrastive Regularizer (VCR), which explicitly constrains the velocity field derived from the user's state. By aligning the predicted velocity with the user's own ground-truth direction while repelling it from the ground-truth velocities of randomly sampled negative users, VCR ensures a clear separation between the user's generated trajectory and the paths of others

Formally, for a mini-batch of size B, we construct an interpolated state $\mathbf{x}_{t,b}$ for each instance b and obtain the backbone prediction $\hat{\mathbf{x}}_{1,b} = f_\theta(\mathbf{x}_{t,b}, t_b, S_b)$.

We denote the backbone-predicted velocity field by $\mathbf{v}_\theta(\mathbf{x}_{t,b}, t_b, S_b)$, and the corresponding target velocity by $\mathbf{u}_{t,b}$. The proposed velocity contrastive regularizer is defined as

$$\mathcal{L}_{\mathrm{VCR}} = -\frac{1}{B}\sum_{b=1}^{B} \|\mathbf{v}_\theta(\mathbf{x}_{t,b}, t_b, S_b) - \mathbf{u}_{t,b'}\|_2^2\,, \tag{14}$$

where $\mathbf{u}_{t,b'}$ denotes the target velocity of a randomly sampled other user (instance) in the mini-batch.

4.4 Overall Objective and Inference

Overall training objective. Our proposed ConFM is built upon an FM-based backbone and further introduces two contrastive regularizers. Therefore, the overall training objective consists of two parts: (1) Following FM4Rec, we adopt its backbone training objectives, which we denote by $\mathcal{L}_{\text{base}}$, including the standard flow matching loss $\mathcal{L}_{\text{FM}}$, an auxiliary mean-squared error loss $\mathcal{L}_{\text{MSE}}$, and a cross-entropy loss $\mathcal{L}_{\text{CE}}$; (2) Two inter-user contrastive regularizers, namely the representation contrastive regularizer $\mathcal{L}_{\text{RCR}}$ for encouraging distinguishable next-item representations across users and the velocity contrastive regularizer $\mathcal{L}_{\text{VCR}}$ for encouraging distinguishable velocity directions across users. Putting them together, the overall objective is

$$\mathcal{L} = \mathcal{L}_{\text{base}} + \lambda_{\text{R}}\mathcal{L}_{\text{RCR}} + \lambda_{\text{V}}\mathcal{L}_{\text{VCR}}, \tag{15}$$

where λ_{R} and λ_{v} control the strengths of the representation-level and velocity-level contrastive regularizer, respectively.

Inference. At inference time, given a user history S, we start from a noise endpoint $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively refine it to obtain a next-item representation. Following the FM4Rec-style instantiation, the velocity field at time t is

$$\mathbf{v}_\theta(\mathbf{x}_t, t, S) = f_\theta(\mathbf{x}_t, t, S) - \mathbf{x}_0. \tag{16}$$

Given a total number of inference steps q, we set the step size as $\Delta t = \frac{1}{q}$ and update $\mathbf{x}_t$ with an Euler solver:

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \mathbf{v}_\theta(\mathbf{x}_t, t, S)\,\Delta t \tag{17}$$

where t takes values $0, \Delta t, \ldots, 1 - \Delta t$. After q updates, we obtain $\mathbf{x}_1$ as the generated next-item representation, which is then scored against candidate item embeddings for retrieval.

4.5 Complexity Analysis

We analyze the time complexity of ConFM for training and inference. Let B, n, d, N, and q denote the batch size, sequence length, representation dimension, number of Transformer blocks, and the number of Euler update steps at inference, respectively.

Training complexity. For each mini-batch, The computational cost of causal self-attention and feed-forward networks is $O\big(N \cdot B \cdot (n^2d + nd^2)\big)$. In addition, RCR adds an $O(B^2d)$ cost for in-batch pairwise similarities, while VCR adds an $O(Bd)$ cost for vector-level operations. Therefore, the overall training complexity can be summarized as $O\big(N \cdot B \cdot (n^2d + nd^2) + B^2d\big)$.

Inference complexity. Since RCR and VCR are introduced only during training, ConFM incurs no additional inference overhead beyond one computation per Euler update step; therefore, its inference time complexity is $O\big(q \cdot N \cdot B \cdot (n^2d + nd^2)\big)$.

Overall, compared with the computational cost of causal self-attention and feed-forward networks, RCR and VCR introduce only lightweight additive overhead, indicating that they can improve performance without significantly affecting the training and inference efficiency.

5 Experiments

In this section, we conduct comprehensive experiments to investigate the following research questions:

Dataset	# Users	# Items	# Actions	Sparsity
Yelp	30,431	20,033	316,354	99.95%
Sports	35,598	18,357	296,337	99.95%
Toys	19,412	11,924	167,597	99.93%
Beauty	22,363	12,101	198,502	99.93%

Table 1: The statistics of the datasets.

- **RQ1:** How does ConFM perform compared with representative baseline methods for sequential recommendation?
- **RQ2:** Can ConFM alleviate the performance degradation on unpopular targets observed in recommenders?
- **RQ3:** How do the two proposed components, VCR and RCR, contribute to the recommendation performance of ConFM?
- **RQ4:** How sensitive is ConFM to different hyperparameter settings?

5.1 Experimental Settings

Datasets. To evaluate the effectiveness of ConFM, we conduct experiments on four widely used datasets: Beauty, Sports, Toys, and Yelp. Beauty, Sports, and Toys are category-specific subsets from the Amazon review corpus [McAuley *et al.*, 2015]. Yelp is a widely adopted benchmark for business recommendation. Following prior practice [Kang and McAuley, 2018], we filter out inactive users and infrequent items with fewer than five interactions. The detailed statistics are summarized in Table 1.

Baselines. We compare ConFM with representative baselines from three categories of sequential recommendation models: **(1) Traditional recommenders:** GRU4Rec [Hidasi *et al.*, 2016], Caser [Tang and Wang, 2018], SASRec [Kang and McAuley, 2018], and BERT4Rec [Sun *et al.*, 2019] predict the next item with discriminative sequence models (e.g., RNN/CNN/Transformer), which can capture dependency patterns in user behavior sequences. **(2) Contrastive-based methods:** CL4SRec [Xie *et al.*, 2022] and CaDiRec [Cui *et al.*, 2024] adopt data augmentation and contrastive objectives to alleviate representation degeneration and data sparsity in sequential recommendation. **(3) Generative recommenders:** DreamRec [Yang *et al.*, 2023], DiQDiff [Mao *et al.*, 2025], DiffuRec [Li *et al.*, 2023], and FM4Rec [Liu *et al.*, 2025b] utilize generative modeling to model item distributions or generate the next-item representation conditioned on interaction sequences.

Evaluation Settings. In our experiments, we evaluate recommendation performance using HR@K and NDCG@K, with results reported for $K \in \{5, 10, 20\}$. All datasets are split into training, validation, and test sets under the leave-one-out protocol. For a fair comparison, we perform full-sort evaluation over the entire item set without negative sampling.

Implementation Details. All experiments are conducted on a server equipped with eight NVIDIA Tesla V100 GPUs. For all baseline methods, we use their official

Dataset	Metric	GRU4Rec	Caser	SASRec	BERT4Rec	CL4SRec	CaDiRec	DreamRec	DiQDiff	DiffuRec	FM4Rec	ConFM	Improv.
Yelp	HR@5	1.52	1.42	1.44	1.96	3.58	2.27	2.17	3.67	3.44	<u>4.12</u>	**4.29**	+3.97%
	HR@10	2.48	2.54	2.51	3.39	4.95	3.79	2.62	5.52	5.20	<u>5.97</u>	**6.24**	+4.63%
	HR@20	3.71	4.06	4.39	5.64	6.89	6.12	2.99	8.40	7.91	<u>9.24</u>	**9.40**	+1.70%
	NDCG@5	0.91	0.80	0.85	1.21	2.78	1.46	1.76	2.58	2.39	<u>2.81</u>	**3.08**	+9.54%
	NDCG@10	1.24	1.13	1.19	1.67	3.22	1.95	1.91	3.17	2.96	<u>3.41</u>	**3.71**	+8.85%
	NDCG@20	1.45	1.56	1.66	2.23	3.71	2.53	2.00	3.90	3.64	<u>4.23</u>	**4.50**	+6.52%
Sports	HR@5	1.62	1.54	1.68	2.17	2.09	2.64	2.18	2.82	2.71	<u>3.10</u>	**3.32**	+7.24%
	HR@10	2.58	2.61	2.67	3.59	3.34	4.13	2.57	3.89	3.86	<u>4.55</u>	**4.81**	+5.56%
	HR@20	4.21	3.99	4.22	6.04	5.30	6.13	2.98	5.67	5.60	<u>6.60</u>	**6.85**	+3.72%
	NDCG@5	1.03	1.14	1.10	1.43	1.35	1.76	1.72	1.99	1.93	<u>2.19</u>	**2.28**	+4.23%
	NDCG@10	1.42	1.35	1.41	1.90	1.75	2.24	1.85	2.33	2.30	<u>2.66</u>	**2.76**	+3.82%
	NDCG@20	1.86	1.78	1.80	2.51	2.24	2.75	1.95	2.78	2.74	<u>3.17</u>	**3.28**	+3.21%
Toys	HR@5	0.97	1.66	4.86	2.74	4.11	5.38	5.49	5.29	5.32	<u>5.84</u>	**6.10**	+4.55%
	HR@10	1.76	2.70	7.10	4.50	6.03	7.77	6.50	7.02	7.11	<u>7.80</u>	**8.17**	+4.69%
	HR@20	3.01	4.20	10.08	6.88	8.27	<u>10.85</u>	7.11	9.31	9.68	10.67	**11.03**	+1.70%
	NDCG@5	0.59	1.07	3.23	1.74	2.40	3.61	4.26	3.97	4.02	<u>4.33</u>	**4.50**	+3.92%
	NDCG@10	0.84	1.41	3.95	2.31	3.20	4.38	4.59	4.54	4.60	<u>4.96</u>	**5.16**	+4.03%
	NDCG@20	1.16	1.79	4.70	2.91	3.50	5.16	4.75	5.11	5.25	<u>5.68</u>	**5.88**	+3.53%
Beauty	HR@5	1.80	2.51	3.51	3.60	4.11	4.91	4.70	5.26	5.50	<u>5.81</u>	**5.95**	+2.48%
	HR@10	2.84	3.42	5.77	6.01	6.03	7.16	5.57	7.26	7.78	<u>8.39</u>	**8.55**	+1.94%
	HR@20	4.27	6.43	9.40	9.84	8.27	10.24	6.35	10.03	10.69	<u>11.62</u>	**11.86**	+2.08%
	NDCG@5	1.16	1.45	2.32	2.16	2.40	3.27	3.62	3.81	4.00	<u>4.10</u>	**4.21**	+2.75%
	NDCG@10	1.50	2.26	3.04	3.00	3.02	3.99	3.90	4.46	4.74	<u>4.93</u>	**5.05**	+2.35%
	NDCG@20	1.86	2.98	3.95	3.91	3.59	4.77	4.10	5.16	5.47	<u>5.74</u>	**5.88**	+2.38%

Table 2: Overall performance comparison across four benchmark datasets, presented as percentages (%). We highlight the highest-performing metric values in bold and the second-best values in underlined. Improv. indicates the relative improvement over the strongest competing baseline.

open-source implementations and adopt the best hyperparameter settings reported or recommended by the original authors for fair comparison. For our ConFM, we train the model with Adam using an initial learning rate of 0.001 and a batch size of 512. The embedding dimension is set to 128, the maximum sequence length is 50, and dropout rates are 0.1 on model representations and 0.3 on the item embedding layer. The weights of VCR and RCR are tuned on the validation set via grid search over $\lambda_V \in \{1e^{-3}, 5e^{-3}, 1e^{-2}, 5e^{-2}, 1e^{-1}, 5e^{-1}\}$ and $\lambda_R \in \{1e^{-3}, 5e^{-3}, 1e^{-2}, 5e^{-2}, 1e^{-1}, 2e^{-1}, 3e^{-1}, 4e^{-1}, 5e^{-1}, 7e^{-1}\}$.

5.2 Overall Performance (RQ1)

To answer RQ1, we compare ConFM with representative baselines on four datasets and report the results in Table 2. From the table, we have the following observations: (1) Contrastive-based methods generally perform better than traditional sequential recommenders. This suggests that contrastive supervision introduces additional representation-level training signals beyond next-item prediction, enabling the model to learn more discriminative and noise-robust user representations under sparse feedback, and consequently enhancing recommendation accuracy. (2) Generative recommenders further perform better than contrastive-based methods. In general, they learn a history-conditioned generative process in the latent space to produce next-item embeddings. This continuous formulation enables smoother preference evolution modeling and can better capture complex and multi-modal user preferences by explicitly modeling uncertainty and producing more expressive retrieval representations. (3) Our ConFM achieves the best performance on four datasets across all evaluation metrics. Compared with the strongest baseline FM4Rec, ConFM achieves an average relative improvement of 5.87%, 4.63%, 3.74%, and 2.33% on Yelp, Sports, Toys, and Beauty, respectively.

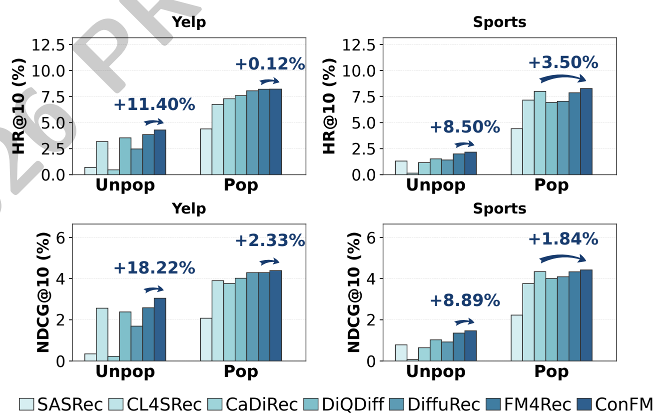

Figure 3: Popularity-stratified Performance. The arrow annotations indicate the relative improvement of our method over the best-performing baseline in each setting.

These improvements are mainly due to two aspects: (i) By pushing the generated next-item embeddings away from each other, ConFM enhances inter-user separability, leading to more distinctive prediction representations; (ii) ConFM suppresses the shared estimator from producing overly similar velocity directions, thereby alleviating the averaged flow effect and improving personalized generation.

5.3 Popularity-stratified Performance (RQ2)

To address RQ2, we conduct a popularity-stratified experiment on the Yelp and Sports datasets. Following previous work [Cheng *et al.*, 2024], we treat the top 20% most frequent

Dataset	Metric	w/o VCR	w/o RCR	w/ Triplet	ConFM
Yelp	HR@10	6.03	5.17	6.20	**6.24**
	NDCG@10	3.52	2.86	3.70	**3.71**
Sports	HR@10	4.70	3.41	4.44	**4.81**
	NDCG@10	2.77	1.80	2.59	**2.76**
Toys	HR@10	8.10	8.04	7.98	**8.17**
	NDCG@10	5.15	5.09	5.06	**5.16**
Beauty	HR@10	8.39	8.35	8.22	**8.55**
	NDCG@10	4.97	4.92	4.83	**5.05**

Table 3: Results(%) of ablation experiments.

items in the training set as popular items and the remaining ones as unpopular items. Test instances are then grouped according to the popularity of their target items. Figure 3 illustrates the performance of our method compared to the most representative baselines from each category. From these results, we make the following observations: First, all methods exhibit significant performance degradation on the unpopular group, indicating that predicting such items demands robust preference modeling capabilities. Second, our ConFM brings clear relative gains on unpopular items: it achieves average relative improvements of 14.81% and 8.70% on Yelp and Sports, respectively. Simultaneously, it demonstrates highly competitive performance on the popular group, indicating that ConFM enhances performance on unpopular items without sacrificing the accuracy for popular ones.

These results suggest that introducing such inter-user separation designals helps produce more distinctive and personalized generated representations, ultimately providing users with more personalized item recommendations.

5.4 Ablation Study (RQ3)

To answer RQ3, we conduct an ablation study to verify the effectiveness of each proposed component in ConFM. Specifically, we construct three variants of ConFM: (i) w/o VCR, which removes the velocity contrastive regularizer from the training objective; (ii) w/o RCR, which removes the representation contrastive regularizer; and (iii) w/ Triplet, which replaces the velocity contrastive regularizer with a triplet-based contrastive objective. The results are reported in Table 3.

As shown in Table 3, all variants perform worse than ConFM. Specifically, the performance drop of w/o VCR suggests that a shared estimator without inter-sample separation signals can make generation trajectories less distinctive. This weakens fine-grained personalization, thereby degrading the final performance. Moreover, the larger degradation of w/o RCR suggests that strengthening inter-user discriminability at the representation level is crucial for improving the overall metrics. Finally, the performance drop of w/ Triplet may be attributed to the relatively weaker supervision provided by a generic triplet objective: compared with the more direct MSE regression supervision, the triplet loss only imposes relative distance constraints and offers less explicit optimization guidance, which can lead to less stable model performance.

5.5 Sensitivity Analysis (RQ4)

To answer RQ4, we further evaluate the sensitivity of ConFM to the two regularizer weights, λ_V and λ_R, on Yelp and Sports. The results are reported in Figure 4.

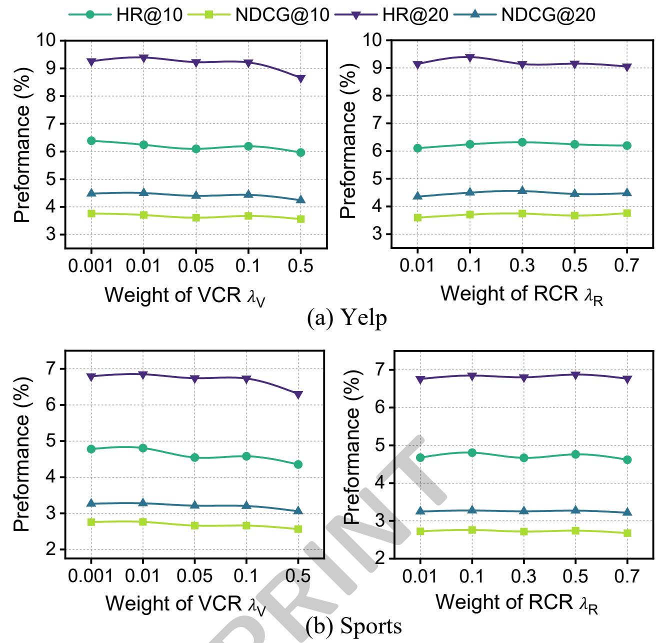

Figure 4: The sensitivity of ConFM to λ_V and λ_R.

For λ_V, when varying it from small to large values, the performance on both datasets shows a clear increase–then–decrease trend. A moderate λ_V consistently yields the best results, suggesting that an appropriate degree of inter-flow separation helps the model generate more personalized trajectories. However, overly large λ_V leads to noticeable degradation, indicating that excessively strong inter-flow constraints may overly separate the generation trajectories and interfere with fitting the main recommendation objective, thereby hurting overall performance. For λ_R, varying it over a wide range leads to relatively mild performance fluctuations on both Yelp and Sports. This indicates that encouraging separation among the predicted embeddings yields relatively stable gains, and the model is comparatively less sensitive to the exact choice of λ_R. One possible reason is that λ_R mainly acts on the output embedding space to enlarge the differences among generated embeddings, without altering the trajectory-level dynamic generation process; therefore, changes in its weight tend to have a smoother impact on the overall performance.

6 Conclusion

In this paper, we identify a key challenge in FM-based sequential recommendation: during the generation process, the averaged flow phenomenon can make the generated item representations less distinguishable. To mitigate this issue, we propose ConFM, which introduces explicit inter-user separation into flow matching training by separating predicted representations and induced velocity directions, thereby promoting more distinguishable generation. Extensive experiments on four public datasets demonstrate that ConFM effectively alleviates the indistinguishable generation issue and consistently outperforms strong baseline.

Acknowledgements

This research was partially supported by the NSFC (62376180, 62506254), the National Key Research and Development Program of China (2023YFF0725002), the NSF of Jiangsu Province (BK20250822), Suzhou Science and Technology Development Program (SYG202328), the NSF of Jiangsu Higher Education Institutions (25KJB520042) and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

Contribution Statement

Wangyu Jin and Yang Xue contributed equally to this work.

References

[Cheng *et al.*, 2024] Mingyue Cheng, Qi Liu, Wenyu Zhang, Zhiding Liu, Hongke Zhao, and Enhong Chen. A general tail item representation enhancement framework for sequential recommendation. *Frontiers of Computer Science*, 18(6):186333, 2024.

[Cui *et al.*, 2024] Ziqiang Cui, Haolun Wu, Bowei He, Ji Cheng, and Chen Ma. Context matters: Enhancing sequential recommendation with context-aware diffusion-based contrastive learning. In *CIKM*, pages 404–414, 2024.

[Fan *et al.*, 2022] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. Sequential recommendation via stochastic self-attention. In *WWW*, pages 2036–2047, 2022.

[Hansen *et al.*, 2020] Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. Contextual and sequential user embeddings for large-scale music recommendation. In *RecSys*, pages 53–62, 2020.

[Hidasi *et al.*, 2016] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *ICLR*, 2016.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

[Huang *et al.*, 2025] Yinxuan Huang, Ke Liang, Zhuofan Dong, Xiaodong Qu, Tianxiang Wang, Yue Han, Jingao Xu, Bin Zhou, and Ye Wang. Flow matching for denoised social recommendation. In *ICML*, 2025.

[Kang and McAuley, 2018] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *ICDM*, pages 197–206, 2018.

[Li *et al.*, 2023] Zihao Li, Aixin Sun, and Chenliang Li. Diffurec: A diffusion model for sequential recommendation. *TOIS*, 42(3):1–28, 2023.

[Lipman *et al.*, 2023] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

[Liu *et al.*, 2021] Zhiwei Liu, Ziwei Fan, Yu Wang, and Philip S Yu. Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer. In *SIGIR*, pages 1608–1612, 2021.

[Liu *et al.*, 2025a] Chengkai Liu, Yangtian Zhang, Jianling Wang, Rex Ying, and James Caverlee. Flow matching for collaborative filtering. In *KDD*, pages 1765–1775, 2025.

[Liu *et al.*, 2025b] Feng Liu, Lixin Zou, Xiangyu Zhao, Min Tang, Liming Dong, Dan Luo, Xiangyang Luo, and Chenliang Li. Flow matching based sequential recommender model. In *IJCAI*, pages 3108–3116. ijcai.org, 2025.

[Liu *et al.*, 2025c] Shuo Liu, An Zhang, Guoqing Hu, Hong Qian, and Tat-Seng Chua. Preference diffusion for recommendation. In *ICLR*, 2025.

[Mao *et al.*, 2025] Wenyu Mao, Shuchang Liu, Haoyang Liu, Haozhe Liu, Xiang Li, and Lantao Hu. Distinguished quantized guidance for diffusion-based sequence recommendation. In *WWW*, pages 425–435, 2025.

[McAuley *et al.*, 2015] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52, 2015.

[Ni *et al.*, 2023] Shuang Ni, Wei Zhou, Junhao Wen, Linfeng Hu, and Shutong Qiao. Enhancing sequential recommendation with contrastive generative adversarial network. *Information Processing & Management*, 60(3):103331, 2023.

[Qu and Nobuhara, 2025] Yuanpeng Qu and Hajime Nobuhara. Intent-aware diffusion with contrastive learning for sequential recommendation. In *SIGIR*, pages 1552–1561, 2025.

[Schedl *et al.*, 2018] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi. Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval*, 7(2):95–116, 2018.

[Song *et al.*, 2021] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

[Sun *et al.*, 2019] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*, pages 1441–1450, 2019.

[Tang and Wang, 2018] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*, pages 565–573, 2018.

[Wang *et al.*, 2020] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. Time to shop for valentine's day: Shopping occasions and sequential recommendation in e-commerce. In *WSDM*, page 645–653, 2020.

[Wang *et al.*, 2022] Yu Wang, Hengrui Zhang, Zhiwei Liu, Liangwei Yang, and Philip S Yu. Contrastvae: Contrastive variational autoencoder for sequential recommendation. In *CIKM*, pages 2056–2066, 2022.

[Xie *et al.*, 2022] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. In *ICDE*, pages 1259–1273. IEEE, 2022.

[Yang *et al.*, 2023] Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. *NeurIPS*, 36:24247–24261, 2023.

[Yuan *et al.*, 2019] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In *WSDM*, pages 582–590, 2019.

[Zhou *et al.*, 2020] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM*, pages 1893–1902, 2020.