CONTINUOUS-TIME SEQUENTIAL RECOMMENDATION VIA PERSONALIZED GATED NEURAL FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous-time sequential recommendation (CTSR) models continuously evolving user preferences from irregularly occurring interactions. Recent Neural ODE-based approaches show promise in capturing these dynamics, but their reliance on numerical integration significantly increase inference latency, entailing accuracy–efficiency trade-off. To address this limitation, we propose Personalized Gated Neural Flow (PGNF), a solver-free CTSR framework that directly parameterizes a flow map for continuous-time preference modeling. Through a temporal gating mechanism, PGNF evolves item states from a user's interaction history over time in a user-specific manner, then combines them to represent the user's evolving preferences. To complement these personalized dynamics with global collaborative signals, PGNF reflects information from an item co-occurrence graph into the initial item states. This graph captures relative temporal proximity between items within users' interaction histories providing time-aware collaborative information without requiring additional computation cost at inference. Experiments on real-world datasets across diverse domains and scales demonstrate that PGNF achieves the state-of-the-art accuracy, improves computational efficiency, and substantially reduces inference latency compared to ODE-based approaches.

1 INTRODUCTION

Efficient and effective user-item matching is essential in modern large-scale recommendation systems for high-quality user experiences. Sequential recommendation, predicting a next item that the target user may prefer based on her historical interaction sequence, is a representative example. To tackle this task, diverse neural architectures have been adopted, including Recurrent Neural Networks (Hidasi et al., 2016; Hidasi & Karatzoglou, 2018; Yue et al., 2024), Transformers (Vaswani et al., 2017; Kang & McAuley, 2018; Sun et al., 2019; Zhang et al., 2023), and Graph Neural Networks (Kipf & Welling, 2017; Zhang et al., 2022; Ye et al., 2023; Liu et al., 2024b). Each of them captures sequential and collaborative patterns in user interactions.

Conventional sequential recommendation methods, however, primarily model the ordering of interactions, without explicitly taking into account the interval between interactions. Considering the nature of a user-item interaction that happens irregularly, such an approach would be suboptimal in characterizing how user tastes *evolve* over time. Continuous-Time Sequential Recommendation (CTSR) (Bai et al., 2019; Chen et al., 2021; Guo et al., 2022; Qin et al., 2024; Xiao et al., 2025; Fu et al., 2025) formulates the problem to predict an item that a user may prefer *at a specific time in the future*, reflecting the irregular temporal nature of real-world interactions and providing a more faithful formulation for practical recommendation scenarios. Recent work (Guo et al., 2022; Qin et al., 2024; Fu et al., 2025) has shown that Neural Ordinary Differential Equations (Neural ODEs) (Chen et al., 2018), modeling the time derivative of a state with a neural network to infer the evolved state at a target time through numerical integration (Euler, 1763; Runge, 1895; Kutta, 1901), are effective for capturing latent continuous-time user preference dynamics.

Nevertheless, numerical integration incurred by the ODE solver, innate in the Neural ODE-based approaches, inherently introduces an accuracy–efficiency trade-off (Butcher, 1963; 1964). Specifically, small integration steps and many function evaluations are required for high accuracy, and thus these repeated computations in CTSR significantly increase the inference cost and latency, limiting the practical applicability of ODE-based models in real-world recommendation services.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To address this trade-off of Neural ODEs while preserving their ability to model continuous-time dynamics, we propose a solver-free CTSR framework, **P**ersonalized **G**ated **N**eural **F**low (**PGNF**). The key idea of PGNF is to model continuous-time user preference dynamics through *personalized item dynamics*: the user-specific evolution of item states in each user's interaction history. Building on Neural Flows (Biloš et al., 2021), PGNF directly parameterizes a flow map that represents ODE solutions as a function of the initial item state and time, avoiding repeated numerical integration. To realize personalized item dynamics, we introduce a *temporal gating mechanism* that conditions this flow on the temporal context of the user's discrete interaction history. This mechanism personalizes the continuous-time evolution of item states, and PGNF combines the evolved item states at the target time to obtain a time-specific user state, thereby capturing how user preferences evolve in continuous time.

While personalized item dynamics capture the continuous-time evolution of item states in each user context, they do not explicitly capture global collaborative signals among items. To complement these dynamics, we incorporate global collaborative signals from an item co-occurrence graph into the initial item states without additional computation at inference. To reflect the temporal nature of CTSR, we construct this graph based on relative temporal proximity between items, measured through *normalized relative ranking* within each user sequence.

Extensive experiments on real-world datasets from diverse domains and scales show that PGNF achieves state-of-the-art performance in CTSR. PGNF also provides greater computational efficiency, substantially reducing inference latency compared to ODE-based approaches.

2 PRELIMINARIES

Continuous-Time Sequential Recommendation (CTSR). Let us define the user set $\mathcal{U}$, the item set $\mathcal{V}$, and the non-negative time set $\mathcal{T} \coloneqq [0, \infty)$. The CTSR task aims to predict an unknown preference of a user $u \in \mathcal{U}$ at some target time $t \in \mathcal{T}$, given the interaction history of the user u before t. We denote this interaction history, or the *user sequence*, by $\mathcal{S}_u^t$:

$$\mathcal{S}_u^t \coloneqq \{(v, \tau) \in \mathcal{V} \times \mathcal{T} : \text{user } u \text{ interacted with item } v \text{ at time } \tau, \;\; \tau < t\}. \tag{1}$$

Neural Ordinary Differential Equations (ODEs). As a specific class of dynamical systems, an ODE can model a time-varying state. Let us denote the state as $\boldsymbol{x} \in \mathbb{R}^d$, and define the vector field $f : \mathcal{T} \times \mathbb{R}^d \to \mathbb{R}^d$. Without loss of generality, we set the initial time to $t_0 = 0$. Then, the initial value problem (IVP) of an ODE is to find the solution curve $\boldsymbol{x}(t)$ that satisfies

$$\frac{d\boldsymbol{x}}{dt} = f(t, \boldsymbol{x}), \quad \boldsymbol{x}_0 = \boldsymbol{x}(0), \tag{2}$$

where $\boldsymbol{x}_0 \in \mathbb{R}^d$ is the initial state. Building upon this formulation, Neural ODEs (Chen et al., 2018) parameterize the vector field f using a neural network. When f is Lipschitz continuous, the Picard–Lindelöf theorem (Picard, 1890; Lindelöf, 1894) guarantees a unique IVP solution. This solution is numerically approximated using an ODE solver, denoted by $\mathrm{ODESolve}$:

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \int_0^t f(\tau, \boldsymbol{x}(\tau)) d\tau = \mathrm{ODESolve}\left(\boldsymbol{x}_0, t, f\right). \tag{3}$$

Neural Flows. Numerical integration via $\mathrm{ODESolve}$ inherently imposes an accuracy-efficiency trade-off, requiring small step size to mitigate discretization errors. To bypass this, Neural Flows (Biloš et al., 2021) introduce a solver-free paradigm that directly parameterizes the solution curve (*i.e., flow*) rather than the vector field. Here, the IVP solution is explicitly given by a *parameterized flow map* $F : \mathcal{T} \times \mathbb{R}^d \to \mathbb{R}^d$ satisfying the initial condition:

$$\boldsymbol{x}(t) = F(t, \boldsymbol{x}_0), \quad \boldsymbol{x}_0 = \boldsymbol{x}(0) = F(0, \boldsymbol{x}_0). \tag{4}$$

For the flow F to be induced by some vector field, F must be invertible in $\boldsymbol{x} \in \mathbb{R}^d$ and differentiable in $t \in \mathcal{T}$ (Lee, 2013). To satisfy this condition, prior studies (Biloš et al., 2021; Mercatali et al., 2024; Gao et al., 2026) apply the following Theorem 2.1 to the *residual flow map*.

Theorem 2.1 (Invertibility). *Let the residual flow map be defined as $F(t, x) \coloneqq x + \varphi(t) \odot g(t, x)$, where $x \in \mathcal{X}$ is a state, φ is a temporal encoder, and g is a Lipschitz continuous function. If the state space $\mathcal{X}$ is a complete metric space and $\|\varphi\|_\infty \|g\|_L < 1$, then the flow map $F(t, \cdot)$ is invertible, where $\|\cdot\|_\infty$ and $\|\cdot\|_L$ denote the infinity norm and the Lipschitz seminorm operators, respectively.*

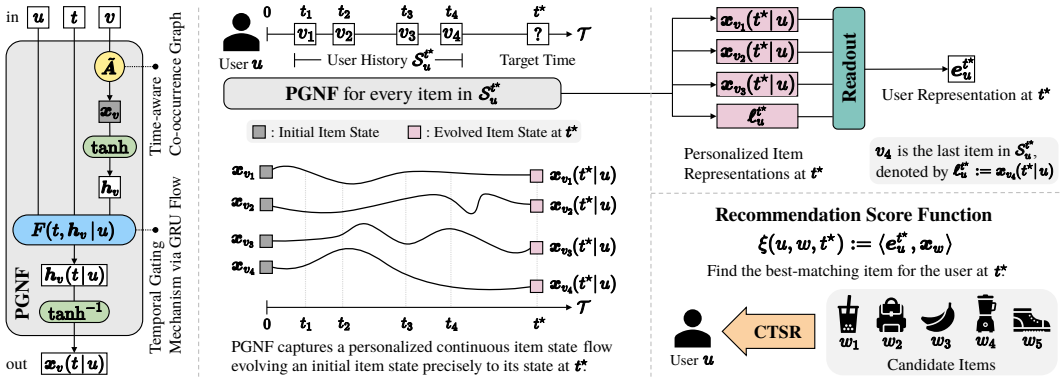

Figure 1: Personalized Gated Neural Flow (PGNF) architecture for CTSR.

Gated Recurrent Unit (GRU) Flow. GRU-ODE (De Brouwer et al., 2019) extends the discrete GRU update to continuous time by defining the vector field with a GRU structure (Chung et al., 2014), making it effective for modeling continuous dynamics from irregular and discrete observations. Building on this idea, Biloš et al. (2021) introduced the GRU Flow as a solver-free counterpart to GRU-ODE. Let $\boldsymbol{h} \in \mathcal{H}^d := (-1, 1)^d$ be a bounded state, and let $\varphi : \mathcal{T} \to \mathbb{R}^d$ be a temporal encoder. GRU Flow is defined by

$$F(t, \boldsymbol{h}) := \boldsymbol{h} + \varphi(t) \odot z(t, \boldsymbol{h}) \odot (c(t, \boldsymbol{h}) - \boldsymbol{h}), \tag{5}$$

where the update gate $z(\cdot, \cdot)$, reset gate $r(\cdot, \cdot)$, and candidate state $c(\cdot, \cdot)$ are given by

$$z(t, \boldsymbol{h}) := \alpha \cdot \sigma(f_z(t, \boldsymbol{h})), \quad r(t, \boldsymbol{h}) := \beta \cdot \sigma(f_r(t, \boldsymbol{h})), \quad c(t, \boldsymbol{h}) := \tanh(f_c(t, r(t, \boldsymbol{h}) \odot \boldsymbol{h})). \tag{6}$$

Note that $\alpha, \beta \in [0, 1)$ are fixed constants, $f_z, f_r, f_c : \mathcal{T} \times \mathcal{H}^d \to \mathbb{R}^d$ are arbitrary neural networks, and $\sigma(\cdot)$ denotes the sigmoid function. Henchforth, whenever the underlying object is clear from context, we use the subscript $*$ as a notational placeholder to be uncluttered. Under suitable conditions, the following results show that GRU Flow maps $\mathcal{H}^d$ into itself and is corestriction-invertible. Note that invertibility should be understood as bijectivity of F_t from $\mathcal{H}^d$ onto its image, since $\mathcal{H}^d$ is not a complete metric space. See Appendix A for rigorous proofs of all theorems.

Theorem 2.2 (Boundedness)**.** *Assume that a temporal encoder satisfies $\varphi(t) \in [0, 1]^d$ for all $t \in \mathcal{T}$. Then, the GRU Flow is closed on $\mathcal{H}^d$. That is, $F(t, \boldsymbol{h}) \in \mathcal{H}^d$ for all $\boldsymbol{h} \in \mathcal{H}^d$.*

Theorem 2.3 (Corestriction Invertibility)**.** *Assume that f_* are Lipschitz continuous with $\|f_*\|_L < 1$, $\varphi(t) \in [0, 1]^d$, and $\alpha(6 + 5\beta) \leq 4$. Then, the GRU Flow map $F_t(\cdot) := F(t, \cdot)$ is invertible onto its image for any fixed $t \in \mathcal{T}$. In other words, its corestriction $\bar{F}_t : \mathcal{H}^d \to F_t(\mathcal{H}^d)$ is bijective.*

3 Methodology

Inspired by the GRU-ODE family (De Brouwer et al., 2019; Zhou & Li, 2021; Guo et al., 2022; Zhao et al., 2023; Jhin et al., 2024) for capturing continuous-time dynamics from discrete observations, we propose **P**ersonalized **G**ated **N**eural **F**low (PGNF), to model a user's evolving preference through the continuous-time dynamics of the items they have interacted with, given the sequence of interaction history $\mathcal{S}_u^t$. Building upon the GRU Flow structure, PGNF evolves each of these item states under the temporal context of user u and combines the evolved states, thereby lifting item-level dynamics into a single time-varying user preference dynamics. Concretely, we instantiate PGNF in four phase: initialization of the item states (Sec. 3.1), temporal encoding (Sec. 3.2), personalized gating for evolving the item states (Sec. 3.3), and combining the evolved item dynamics into a user representation through a readout layer (Sec. 3.4). The resulting representation is then utilized to score candidate items and to formulate the training objective. Fig. 1 illustrates the complete PGNF architecture for continuous-time sequential recommendation.

3.1 Initial Item States with Global Collaborative Signals

We aim to evolve each item up to time t under the personalized context of a specific user. Before we perform user-specific temporal evolution of an item, we initialize its state user-agnostically, encoding its intrinsic collaborative signal across all users. Thereby, this initial state is to capture global

item-level signals that complement the personalized Neural Flow, providing a stable foundation from which user-specific dynamics can evolve.

To effectively capture collaborative item-item relations from all user sequences, we adopt a widely-used item co-occurrence graph-based approach (Zhang et al., 2022; Liu et al., 2024a). Specifically, we construct a time-aware item co-occurrence graph whose edges reflect the temporal proximity between the corresponding item interactions. In this way, the initial item state is expected to capture not only whether two items co-occur in a user sequence, but also how closely they are observed in continuous time. Note that our proposed framework is agnostic to this specific choice, as long as the item embedding can be initialized reflecting the semantic item-item relations.

Modeling Temporal Proximity. To effectively model item collaborative signals, it is crucial to measure the temporal proximity of interactions with those items. However, raw time intervals between interactions are difficult to compare across users due to varying activity frequencies. To enable robust comparison, we quantify temporal proximity relative to each user sequence. Denoting the set of distinct item pairs in $\mathcal{S}_u^t$ by $\mathcal{P}_u^t := \{(v_i, v_j) : (v_i, t_i), (v_j, t_j) \in \mathcal{S}_u^t, v_i \neq v_j\}$, we introduce a *normalized relative ranking* to neutralize scale variance as follows:

$$\pi_u^t : \mathcal{P}_u^t \longrightarrow [0,1], \quad (v_i, v_j) \longmapsto \sum_{(v_a, v_b) \in \mathcal{P}_u^t} \frac{\mathbb{I}\left[\delta_u^t(v_a, v_b) < \delta_u^t(v_i, v_j)\right]}{\max\left\{1,\ |\mathcal{P}_u^t| - 1\right\}}, \tag{7}$$

where we define the temporal distance as $\delta_u^t(v_i, v_j) := |t_i - t_j|$, and $\mathbb{I}$ is an indicator function. Intuitively, this non-parametric formulation maps the absolute temporal gap $\delta_u^t(v_i, v_j)$ to its relative rank among all pairs for user u, where a smaller π_u^t reflects stronger temporal synchrony. Consequently, this ranking effectively captures temporal associations while ensuring scale invariance across users with disparate time scales.

Item Co-occurrence Graph. At a fixed time t, we represent the item co-occurrence graph by an adjacency matrix $\boldsymbol{A}^t \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, whose entries are constructed by iteratively accumulating normalized relative-rankings over all user sequences $\{\mathcal{S}_u^t\}_{u \in \mathcal{U}}$. Specifically, for each user u, we update

$$\boldsymbol{A}_{ij}^t \leftarrow \boldsymbol{A}_{ij}^t + \max\left\{0,\ \lambda - \pi_u^t(v_i, v_j)\right\}, \tag{8}$$

where the hyperparameter $\lambda \in [0, 1]$ controls the graph density; a smaller λ retains only highly synchronized item pairs, while a larger one allows more temporally distant pairs to contribute. After aggregating edge weights across users, we add self-loops and apply row-wise normalization:

$$\boldsymbol{D}_{ii}^t := \sum_{j=1}^{|\mathcal{V}|} (\boldsymbol{A}^t + \boldsymbol{I})_{ij}, \quad \boldsymbol{D}^t := \operatorname{diag}\left(\boldsymbol{D}_{11}^t, \boldsymbol{D}_{22}^t, \cdots, \boldsymbol{D}_{|\mathcal{V}||\mathcal{V}|}^t\right), \quad \tilde{\boldsymbol{A}}^t := (\boldsymbol{D}^t)^{-1}(\boldsymbol{A}^t + \boldsymbol{I}), \tag{9}$$

where $\boldsymbol{I}$ is the identity matrix. Each row of $\tilde{\boldsymbol{A}}^t$ can be interpreted as relative similarity weights between an item and its neighbors, normalized to sum to one.

Initial Item State. We use the initial item state to roughly capture global collaborative signals among items over time without continual updates at each time t. To this end, we define $\tilde{\boldsymbol{A}} := \tilde{\boldsymbol{A}}^{t_{\text{split}}}$, where t_{split} is the training–validation cutoff in a chronological data split, and keep it fixed throughout validation and test inference. We then obtain the item representation $\boldsymbol{x}_{v_i} := \sum_{j=1}^{|\mathcal{V}|} \tilde{\boldsymbol{A}}_{ij} \boldsymbol{v}_j$, enriched with global collaborative signals, by propagating item ID embeddings $\boldsymbol{v}_j \in \mathbb{R}^d$ over the graph. The item ID embeddings are randomly initialized and trained end-to-end with our PGNF. This user-agnostic representation $\boldsymbol{x}_{v_i}$ serves as the initial item state of our PGNF. Once the training is done, the propagation graph can be discarded since $\tilde{\boldsymbol{A}}$ remains fixed, letting $\boldsymbol{x}_{v_i}$ sufficient at inference. This strategy efficiently captures temporal signals without incurring any additional inference overhead.

3.2 Temporal Encoder

The temporal encoder $\varphi(\cdot)$ is designed to satisfy two structural requirements imposed by the GRU Flow. First, it must satisfy $\varphi(0) = \mathbf{0}$, ensuring that the residual flow preserves the initial state. Second, to satisfy boundedness and corestriction invertibility, its output must remain bounded as $\varphi(t) \in [0, 1]^d$. We choose the *absolute hyperbolic tangent* as a simple temporal encoder that satisfies this boundedness condition and is differentiable with respect to non-negative time t:

$$\varphi(t) := [\varphi_1(t), \cdots, \varphi_d(t)]^\top, \quad \varphi_i(t) := |\tanh(\theta_i t)| = \tanh(|\theta_i| t), \tag{10}$$

where $\theta_i \in \mathbb{R}$ are learnable parameters.

3.3 Temporal Gating Mechanism for Personalized Item Dynamics

The GRU-ODE family (De Brouwer et al., 2019; Zhou & Li, 2021; Guo et al., 2022; Zhao et al., 2023; Jhin et al., 2024) uses gating mechanism to balance state retention and updates, enabling effective modeling of continuous-time dynamics from discrete observations. Motivated by this capability, we introduce a *temporal gating* to model personalized item dynamics.

Personalized Item Dynamics. Given a user $u \in \mathcal{U}$ and an item $v \in \mathcal{V}$, we first map the initial item state $\boldsymbol{x}_v \in \mathbb{R}^d$ to $\boldsymbol{h}_v \coloneqq \tanh(\boldsymbol{x}_v) \in \mathcal{H}^d$, where the GRU Flow is defined on. We then model the personalized dynamics of the item v through the following gated flow:

$$\boldsymbol{h}_v(t \,|\, u) = F(t, \boldsymbol{h}_v \,|\, u) \coloneqq \boldsymbol{h}_v + \varphi(t) \odot z(t, \boldsymbol{h}_v \,|\, u) \odot (c(t, \boldsymbol{h}_v \,|\, u) - \boldsymbol{h}_v). \tag{11}$$

Here, $F(t, \cdot \,|\, u)$ defines a user-conditioned flow map on $\mathcal{H}^d$, which transports the initial item state $\boldsymbol{h}_v$ to its time-evolved state at time t under the personalized context of user u. After evolving the item state to time t, we remap it back to Euclidean space via $\boldsymbol{x}_v(t \,|\, u) \coloneqq \tanh^{-1}(\boldsymbol{h}_v(t \,|\, u)) \in \mathbb{R}^d$. This $\tanh^{-1}(\cdot)$ domain transformation is well-defined by Theorem 2.2.

Temporal User Context. To condition the item dynamics on user u, we use the latest time at which the user interacted with item v before the target time t. Rather than introducing a separate user embedding, our goal is to represent each user through the items in the user sequence. Accordingly, any item evolved by $F(t, \cdot \,|\, u)$ is assumed to belong to the user sequence $\mathcal{S}_u^t$, which ensures that its latest observation time is well-defined. Formally, we define $s_{uv}^t \in \mathcal{T}$ as

$$s_{uv}^t \coloneqq \max\{\tau \in \mathcal{T} : (v, \tau) \in \mathcal{S}_u^t\}. \tag{12}$$

Note that the user-conditioned flow map is differentiable with respect to t between interactions, but not necessarily when s_{uv}^t changes upon a new interaction with item v. This is analogous to the discrete observation-driven updates in GRU-ODE-Bayes (De Brouwer et al., 2019).

Temporal Gating Mechanism. We now instantiate the gates in the user-conditioned flow map by injecting the temporal user context s_{uv}^t into each gating mechanism. To this end, we summarize the absolute target time t, the latest observation time s, the elapsed time $t - s$, and the initial item state $\boldsymbol{h}$ into a single temporal representation:

$$\eta(t, s, \boldsymbol{h}) \coloneqq \mathrm{agg}(\phi(t), \phi(s), \psi(t - s), \boldsymbol{h}), \tag{13}$$

where $\phi : \mathcal{T} \to \mathbb{R}^d$ encodes absolute time and $\psi : \mathcal{T} \to \mathbb{R}^d$ encodes elapsed time. The agg denotes a temporal-feature aggregation operator, such as concatenation or addition. See Appendices B and C for specific choices of the internal temporal encoders ϕ and ψ, and the aggregation function agg.

Using this aggregated temporal representation, we define the update and reset gates:

$$z(t, \boldsymbol{h}_v \,|\, u) \coloneqq \alpha \cdot \sigma(f_z(t, \boldsymbol{h}_v \,|\, u)), \quad f_z(t, \boldsymbol{h}_v \,|\, u) \coloneqq \tilde{\mu}_z(\eta(t, s_{uv}^t, \boldsymbol{h}_v)), \tag{14}$$

$$r(t, \boldsymbol{h}_v \,|\, u) \coloneqq \beta \cdot \sigma(f_r(t, \boldsymbol{h}_v \,|\, u)), \quad f_r(t, \boldsymbol{h}_v \,|\, u) \coloneqq \tilde{\mu}_r(\eta(t, s_{uv}^t, \boldsymbol{h}_v)), \tag{15}$$

where $\tilde{\mu}_*$ are linear layers equipped with spectral normalization (Miyato et al., 2018), which is used to enforce $\|f_*\|_L < 1$.

An item representation needs to balance preserving its underlying characteristics with adapting to the temporal user context for effective recommendation. To support this balance, we construct the candidate state using two branches: a direct branch (c_1) that accesses the initial item state and a reset-modulated branch (c_2) that selectively adjusts its contribution based on the temporal context.

$$c(t, \boldsymbol{h}_v \,|\, u) \coloneqq \tanh(f_c(t, \boldsymbol{h}_v \,|\, u)), \tag{16}$$

$$f_c(t, \boldsymbol{h}_v \,|\, u) \coloneqq \tilde{\mu}_{c_1}(\eta(t, s_{uv}^t, \boldsymbol{h}_v)) + \tilde{\mu}_{c_2}(\eta(t, s_{uv}^t, r(t, \boldsymbol{h}_v \,|\, u) \odot \boldsymbol{h}_v)). \tag{17}$$

Here, $\tilde{\mu}_{c_1}$ and $\tilde{\mu}_{c_2}$ are equipped with spectral normalization, and we constrain both operators as $\|\tilde{\mu}_{c_1}\|_L, \|\tilde{\mu}_{c_2}\|_L < 1/2$. As shown in Corollary 3.1, this candidate structure preserves corestriction invertibility whenever $\alpha\,(16 + 5\beta) \leq 8$ (see Appendix A for proof). Following Biloš et al. (2021), we set $\alpha = 2/5$ and $\beta = 4/5$, for which the condition holds with equality. Combined with differentiability with respect to t between consecutive temporal context updates, Corollary 3.1 allows PGNF to be interpreted as a flow induced by a time-dependent vector field within each such interval.

Corollary 3.1 (Corestriction Invertibility for PGNF)**.** *Assume that the conditions of Theorem 2.3 hold for the temporal encoder and the update and reset gates, and additionally* $\|\tilde{\mu}_{c_1}\|_L, \|\tilde{\mu}_{c_2}\|_L < 1/2$*, and* $\alpha\,(16 + 5\beta) \leq 8$*. Then,* $F(t, \cdot \,|\, u)$ *is corestriction-invertible for any target time* $t \in \mathcal{T}$*.*

3.4 Readout Layer, Training, and Inference

We derive the personalized dynamics of item v from the perspective of user u. Using the resulting dynamic item states, we construct a user representation $\boldsymbol{e}_u^t \in \mathbb{R}^d$ defined over continuous time t, thereby capturing user preference dynamics. To this end, we first summarize the item states available to user u at time t in the sequence $\mathcal{S}_u^t$ by mean pooling:

$$\boldsymbol{m}_u^t \coloneqq \frac{1}{|\mathcal{S}_u^t|} \sum_{(v,\cdot)\in\mathcal{S}_u^t} \boldsymbol{x}_v(t \,|\, u). \tag{18}$$

As the relative importance of sequence-wide preferences and recent interactions varies across domains, we propose two readout variants, mean and gating readouts. Here, μ_* denotes a linear layer.

Mean Readout. When the sequence summary provides sufficient predictive signal, we simply apply a linear layer to the mean-pooled vector:

$$\boldsymbol{e}_u^t \coloneqq \mu_{\text{mean}}(\boldsymbol{m}_u^t). \tag{19}$$

Gating Readout. When recency is crucial, we explicitly incorporate the most recent interaction. Let $\boldsymbol{\ell}_u^t \in \mathbb{R}^d$ denote the dynamic state of the most recent item in the user sequence $\mathcal{S}_u^t$. We fuse this recent-item state with the sequence context $\boldsymbol{m}_u^t$ via a gating mechanism. Specifically, we compute an information gate $\boldsymbol{p}_u^t$ and apply a sigmoid to perform a convex combination:

$$\boldsymbol{p}_u^t \coloneqq \mu_{\boldsymbol{p}}\big(\boldsymbol{\ell}_u^t \,\|\, \boldsymbol{m}_u^t \,\|\, \boldsymbol{\ell}_u^t - \boldsymbol{m}_u^t \,\|\, \boldsymbol{\ell}_u^t \odot \boldsymbol{m}_u^t\big), \quad \boldsymbol{a}_u^t \coloneqq \sigma(\boldsymbol{p}_u^t) \odot \boldsymbol{\ell}_u^t + \big(\mathbf{1} - \sigma(\boldsymbol{p}_u^t)\big) \odot \boldsymbol{m}_u^t, \tag{20}$$

where $\|$ is the concatenation operator. To retain both input signals alongside their gated combination, we concatenate the gated state with the recent-item state and the sequence summary, followed by a linear projection:

$$\boldsymbol{e}_u^t \coloneqq \mu_{\text{gating}}\big(\boldsymbol{a}_u^t \,\|\, \boldsymbol{\ell}_u^t \,\|\, \boldsymbol{m}_u^t\big). \tag{21}$$

The resulting user state combines sequence-wide preferences with the recency-aware signal from the latest interaction.

Training and Inference. Given a training triplet $(u, v, t) \in \mathcal{D}$ with target item v, we optimize the cross-entropy loss using an inner-product recommendation score:

$$\mathcal{L} \coloneqq -\frac{1}{|\mathcal{D}|} \sum_{(u,v,t)\in\mathcal{D}} \log \frac{\exp(\xi(u,v,t))}{\sum_{w\in\mathcal{V}} \exp(\xi(u,w,t))}, \quad \xi(u,w,t) \coloneqq \langle \boldsymbol{e}_u^t, \boldsymbol{x}_w \rangle. \tag{22}$$

Here, $\boldsymbol{x}_w \in \mathbb{R}^d$ denotes the initial state of candidate item $w \in \mathcal{V}$ and serves as its item representation. At inference time, candidate items are ranked according to $\xi(u, \cdot, t)$.

4 Experiments

To verify the efficacy of our proposed method, we evaluate its recommendation accuracy, scalability, and design factors through ablation studies on three real-world datasets from different domains. We additionally conduct a hyperparameter study to analyze the effects of graph density. All metrics are averaged over five random seeds. More details on experimental setup, standard deviations, and more hyperparameter studies are provided in Appendices D and E.

4.1 Experimental Setup

Datasets. We evaluate our model on datasets spanning different domains and scales: Amazon-Tools (He & McAuley, 2016) and Tmall-Buy (Tianchi, 2018) purchase histories, and Behance (He et al., 2016) artwork likes. To assess future-event prediction from historical data, we split interactions chronologically into training, validation, and test sets at an 8:1:1 ratio with strictly non-overlapping timestamps. Tab. I in Appendix D provides dataset statistics.

Baselines. We evaluate PGNF against state-of-the-art sequential recommendation baselines, including *general* models—SASRec (Kang & McAuley, 2018), FEARec (Du et al., 2023), BSARec (Shin et al., 2024), TV-Rec (Shin et al., 2025), and KERec (Yang et al., 2026)—and, with a particular focus on *continuous-time* models, CTRec (Bai et al., 2019), TiSASRec (Li et al., 2020), CTSMA

Table 1: Overall accuracy on a percentage scale. OOM denotes out of memory.

	Model	Amazon-Tools					Behance					Tmall-Buy				
		NDCG		HR		MRR	NDCG		HR		MRR	NDCG		HR		MRR
		@10	@20	@10	@20		@10	@20	@10	@20		@10	@20	@10	@20	
General	SASRec	5.14	6.24	9.04	13.42	4.89	14.21	16.12	22.79	30.35	12.83	8.98	10.43	15.04	20.79	8.19
	FEARec	5.74	6.91	10.07	14.75	5.39	20.79	22.32	28.38	34.44	19.47	9.72	11.15	15.88	21.56	8.89
	BSARec	5.35	6.48	9.51	14.01	5.06	21.76	23.34	29.57	35.84	20.40	9.99	11.42	16.27	21.94	9.13
	TV-Rec	6.05	7.20	10.56	15.14	5.63	21.61	23.23	29.81	36.22	20.14	9.75	11.11	15.71	21.12	8.94
	KERec	5.62	6.73	9.73	14.17	5.32	22.04	23.64	30.19	36.53	20.57	9.88	11.30	16.16	21.78	9.00
Continuous-Time	CTRec	4.81	5.66	8.07	11.44	4.63	4.12	5.54	8.24	13.91	4.08	7.20	8.43	11.84	16.72	6.77
	TiSASRec	5.34	6.45	9.47	13.89	5.01	14.66	16.46	23.14	30.29	13.22	9.04	10.49	14.93	20.68	8.33
	CTSMA	4.73	5.77	8.27	12.40	4.56	16.09	17.67	23.59	29.86	14.87	7.63	8.92	12.60	17.71	7.12
	TGSRec	5.20	6.34	9.08	13.64	5.02	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	GNG-ODE	5.27	6.40	9.20	13.70	5.04	18.62	20.15	26.41	32.49	17.23	10.12	11.56	16.60	22.35	9.18
	GDERec	3.71	4.50	6.62	9.76	3.59	5.90	7.71	11.45	18.67	5.55	9.44	10.74	14.98	20.16	8.73
	SS4Rec	5.16	6.28	9.05	13.54	4.90	22.66	24.17	30.33	36.31	21.31	9.30	10.72	15.10	20.77	8.58
	TGODE	5.76	6.96	10.01	14.75	5.45	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	PGNF	**6.59**	**7.88**	**11.37**	**16.49**	**6.15**	**23.66**	**25.35**	**32.19**	**38.86**	**22.12**	**11.15**	**12.63**	**17.97**	**23.87**	**10.13**

(Chen et al., 2021), TGSRec (Fan et al., 2021), GNG-ODE (Guo et al., 2022), GDERec (Qin et al., 2024), SS4Rec (Xiao et al., 2025), and TGODE (Fu et al., 2025).

Metrics. To evaluate top-k recommendation accuracy, we employ NDCG@k, HR@k, and MRR, which are widely adopted metrics in the research community (Tamm et al., 2021). Each metric is computed based on the ranking of the ground-truth item alongside 1,000 sampled negative items, with the top-k variants evaluated at $k \in \{10, 20\}$ and MRR computed over the entire ranking. For conciseness, we report only the results at $k = 20$ for all ablation and hyperparameter studies.

4.2 Accuracy

As demonstrated in Tab. 1, PGNF consistently outperforms existing general and continuous-time sequential recommendation baselines across all datasets and evaluation metrics. Notably, explicit graph-based CTSR models such as TGODE and TGSRec struggle to scale on large datasets due to the extensive computational overhead incurred by attempting to simultaneously capture sequential dependencies and global collaborative signals through the graph topology. To circumvent this scalability bottleneck, our approach injects global collaborative signals directly into the initial item states, effectively bypassing the need for computationally expensive graph operations.

4.3 Scalability

Fig. 2 compares the scalability of PGNF with major baselines across various state dimensions d. Since scalability is primarily governed by d and the number of items, we evaluate multiple values of d on a single dataset (Amazon-Tools) and report batch-level measurements. SS4Rec and TGODE run out of memory (OOM) at $d = 1024$. Following their official implementations, ODE-based models (GNG-ODE and TGODE) use three-step numerical integration.

Runtime Scalability. PGNF achieves the shortest training and inference times among all compared models with particularly pronounced advantages over continuous-time baselines such as GNG-ODE, SS4Rec, and TGODE. These results support the computational efficiency of its solver-free architecture, which avoids the numerical integration required by ODE-based models (GNG-ODE and TGODE), and suggest its suitability for latency-sensitive recommendation settings.

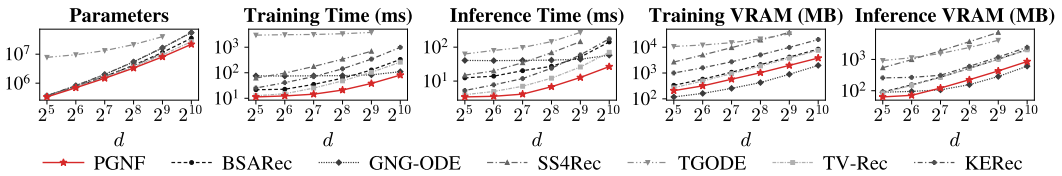

Figure 2: Per-batch scalability on Amazon-Tools across state dimensions d with batch size 512.

Table 2: PGNF component ablation. **ICG**, **TUC**, and **TGM** denote the item co-occurrence graph, temporal user context, and temporal gating mechanism, respectively. The best readout is mean for Amazon-Tools and Tmall-Buy; gating for Behance.

Variant				Amazon-Tools			Behance			Tmall-Buy		
Readout	**ICG**	**TUC**	**TGM**	**NDCG**	**HR**	**MRR**	**NDCG**	**HR**	**MRR**	**NDCG**	**HR**	**MRR**
Mean	✓	✓	✓	**7.88**	**16.49**	**6.15**	10.36	22.99	7.64	**12.63**	**23.87**	**10.13**
Gating	✓	✓	✓	7.42	15.51	5.84	**25.35**	**38.86**	**22.12**	12.31	23.23	9.90
Best	✗	✓	✓	6.44	13.52	5.11	24.80	38.16	21.61	11.56	22.52	9.18
	✓	✗	✓	7.84	16.43	6.12	23.74	37.18	20.54	12.21	23.80	9.64
	✓	✓	✗	7.62	15.85	5.99	21.62	34.64	18.57	12.49	23.64	10.05

Memory Scalability. PGNF achieves memory consumption comparable to GNG-ODE and lower than all other baselines. We attribute this efficiency to its gating mechanism, which provides sufficient expressive capacity with relatively few parameters. In addition, although PGNF uses an item co-occurrence graph, it stores only the item representations obtained after graph propagation during inference. This design avoids redundant memory overhead and facilitates deployment in large-scale or resource-constrained environments.

4.4 Ablation Study

As reported in Tab. 2, we conduct an ablation study to empirically assess the contributions of four key components of our proposed method: (1) item co-occurrence graph, (2) temporal user context, (3) temporal gating mechanism, and (4) readout layer.

Item Co-occurrence Graph. We evaluate the contribution of the item co-occurrence graph by removing graph propagation based on the adjacency matrix $\tilde{\boldsymbol{A}}$, which encodes temporal proximity as normalized relative rankings within user sequences. In this variant, item ID embeddings are used directly as the initial item states without item graph propagation. Remark that item ID embeddings are randomly initialized and trained end-to-end with our PGNF. The resulting accuracy drop highlights the importance of incorporating global collaborative signals into the initial item states to capture relationships among items.

Temporal User Context. Recommendation accuracy decreases when we remove s_{uv}^t, which denotes the user's latest interaction time with a specific item. This result demonstrates that s_{uv}^t, determined by user behavior, enables effective personalization of item dynamics.

Temporal Gating Mechanism. Eliminating the difference term $c(t, h_v|u) - h_v$ removes the temporal gating mechanism. This modification causes the architecture to degenerate into ResNet Flow (Biloš et al., 2021). We observe a consistent accuracy decrease from this structural modification, demonstrating that the GRU Flow architecture utilizing explicit gating operations is an empirically more effective approach than simple residual connections for modeling continuous-time dynamics.

Readout Layer. We observe that the *mean readout* yields the highest accuracy on the Amazon and Tmall, while the *gating readout* achieves optimal results on the Behance by a substantial margin. We conjecture that this discrepancy arises from the distinct interaction characteristics of each domain. *Purchase*-based behaviors on Amazon and Tmall benefit mostly from aggregating the complete, overall context of a user sequence. Conversely, the Behance dataset is dominated by exploratory *like* interactions on artworks. In this environment, short-term temporal recency is significantly more critical, and we infer that the gating mechanism provides a distinct advantage in capturing these recent dynamics.

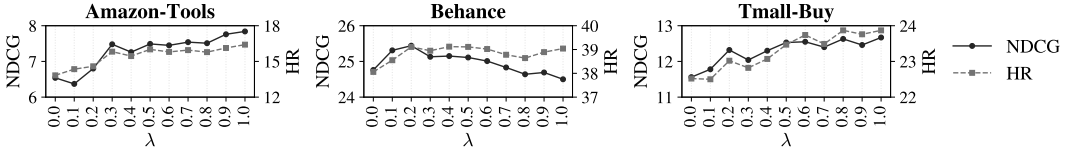

Figure 3: Effect of λ on PGNF accuracy.

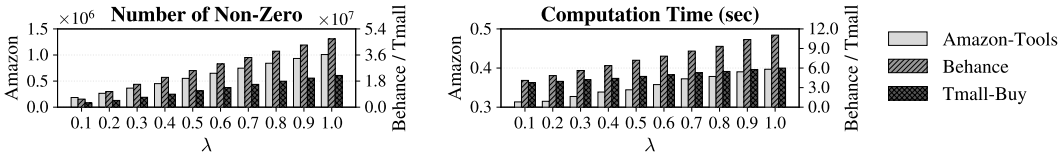

Figure 4: Effect of λ on the size and computation time of the item co-occurrence graph.

4.5 Hyperparameter Study on Graph Density

Accuracy. We analyze in Fig. 3 the impact of the graph density hyperparameter $\lambda \in [0, 1]$ on recommendation accuracy. Note that a larger λ captures collaborative signals between more distant item pairs. Consequently, purchase-based datasets (Amazon and Tmall) benefit overall from higher λ, whereas like-based dataset (Behance) does not. This divergence aligns with the findings of the preceding ablation study on the readout layer.

Computational Overhead. Fig. 4 reports the graph construction time and matrix size across different λ. Increasing λ increases graph density, leading to linear increases in the number of non-zero elements and graph construction time. Nevertheless, the additional graph construction time and storage requirements remain manageable, with each sparse element occupying only 20 bytes and total graph storage requiring less than 1 GB during training. Moreover, neither graph construction nor graph storage is required during inference, eliminating both sources of overhead. Separately, since sequence length quadratically increases intra-sequence pairs, Behance incurs a larger graph size and higher construction latency than Tmall, despite containing fewer unique users and items.

5 Related Work

Neural Differential Equations. Neural ODEs (Chen et al., 2018) and their extensions—neural differential equations (NDEs) (Jia & Benson, 2019; Liu et al., 2019; Kidger et al., 2020)—model continuous dynamical systems across time-dependent domains: time series forecasting (Rubanova et al., 2019; De Brouwer et al., 2019; Norcliffe et al., 2021), robotics (Yildiz et al., 2021; Kasaei et al., 2023), physics (Lee & Parish, 2021; Zakwan et al., 2023), and image generation (Song et al., 2021; Lipman et al., 2023). However, their reliance on numerical solvers requires numerous integration steps, hindering inference speed. To address this limitation, Neural Flows (Biloš et al., 2021) directly parameterize the flow map, bypassing numerical integration altogether. This solver-free paradigm has since been extended to time series forecasting (Mercatali et al., 2024; Gao et al., 2026), stochastic flow maps (Kiyohara et al., 2025; Pan et al., 2026; McCallum et al., 2026).

Continuous-Time Sequential Recommendation. By explicitly modeling irregular interaction timestamps, continuous-time sequential recommendation advances prediction accuracy across three main mathematical paradigms. First, point processes statistically model user action timing and intensity; for example, CTRec (Bai et al., 2019) and CTSMA (Chen et al., 2021) utilize Hawkes processes (Hawkes, 1971) and survival analysis (Aalen et al., 2008) to estimate continuous-time interaction probabilities. Second, ODEs capture latent preference evolution as continuous dynamics rather than discrete steps, as instantiated by GNG-ODE (Guo et al., 2022), GDERec (Qin et al., 2024), and TGODE (Fu et al., 2025). Third, state space models (SSMs) (Hamilton, 1994; Gu & Dao, 2023) efficiently capture continuous long-term dependencies over irregular intervals, with SS4Rec (Xiao et al., 2025) employing variable timestep discretization to process temporal gaps without constant-interval assumptions. In contrast to these frameworks, TGSRec (Fan et al., 2021) uniquely relies on a temporal graph attention mechanism.

6 Conclusion

In this work, we presented PGNF, a solver-free framework for CTSR. Our results show that user preference dynamics can be modeled effectively without numerical integration, substantially reducing inference latency. Incorporating global collaborative signals into the initial item states complements personalized dynamics without inference-time overhead. Experiments on real-world datasets demonstrate the effectiveness of PGNF in terms of recommendation accuracy and scalability. We hope these findings encourage further research into solver-free approaches to recommendation.

AI USE STATEMENT

Scope of Use. We use generative AI tools to refine author-written proofs; create and modify figures; assist in implementing the proposed model and baseline methods and integrating their codebases into a unified experimental framework; draft and edit manuscript text; and search, summarize, and analyze relevant literature and information. For proofs, AI assistance is limited to improving the rigor, clarity, and readability of author-written text. We develop all core ideas, logical structures, proof strategies, and substantive mathematical arguments and derivations.

Verification and Responsibility. We review all AI-assisted work and independently verify the revised proofs. We test the integrated code for consistent implementation and evaluation, check literature summaries and retrieved information against original sources, and review text and figures for accuracy, clarity, consistency, and appropriate attribution. We take full responsibility for the final content and all AI-assisted artifacts.

REFERENCES

Odd O Aalen, Ørnulf Borgan, and Håkon K Gjessing. *Survival and event history analysis: a process point of view*. Springer, 2008.

Ting Bai, Lixin Zou, Wayne Xin Zhao, Pan Du, Weidong Liu, Jian-Yun Nie, and Ji-Rong Wen. Ctrec: A long-short demands evolution model for continuous-time recommendation. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2019.

Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta mathematicae*, 1922.

Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

John C Butcher. Coefficients for the study of runge-kutta integration processes. *Journal of the Australian Mathematical Society*, 1963.

John C Butcher. On runge-kutta processes of high order. *Journal of the Australian Mathematical Society*, 1964.

Chao Chen, Haoyu Geng, Nianzu Yang, Junchi Yan, Daiyue Xue, Jianping Yu, and Xiaokang Yang. Learning self-modulating attention in continuous time space with applications to sequential recommendation. In *Proc. of the International Conference on Machine Learning (ICML)*, 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Xinyu Du, Huanhuan Yuan, Pengpeng Zhao, Jianfeng Qu, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor S Sheng. Frequency enhanced hybrid attention network for sequential recommendation. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2023.

Leonhard Euler. De integratione aequationum differentialium. *Novi Commentarii Academiae Scientiarum Petropolitanae*, 1763.

Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. Continuous-time sequential recommendation with temporal graph collaborative transformer. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*, 2021.

Haoyan Fu, Zhida Qin, Shixiao Yang, Haoyao Zhang, Bin Lu, Shuang Li, Tianyu Huang, and John CS Lui. Time matters: Enhancing sequential recommendations with time-guided graph neural odes. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2025.

Mengzhou Gao, Kaiwei Wang, and Pengfei Jiao. One-step graph-structured neural flows for irregular multivariate time series classification. In *Proc. of the International Conference on Machine Learning (ICML)*, 2026.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Jiayan Guo, Peiyan Zhang, Chaozhuo Li, Xing Xie, Yan Zhang, and Sunghun Kim. Evolutionary preference learning via graph nested gru ode for session-based recommendation. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*, 2022.

James D Hamilton. State-space models. *Handbook of econometrics*, 1994.

Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 1971.

Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proc. of the ACM International Conference on World Wide Web (WWW)*, 2016.

Ruining He, Chen Fang, Zhaowen Wang, and Julian McAuley. Vista: A visually, socially, and temporally-aware model for artistic recommendation. In *Proc. of the ACM Conference on Recommender Systems (RecSys)*, 2016.

Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*, 2018.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2016.

Sheo Yon Jhin, Seojin Kim, and Noseong Park. Addressing prediction delays in time series forecasting: A continuous gru approach with derivative regularization. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2024.

Junteng Jia and Austin R Benson. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *Proc. of the IEEE International Conference on Data Mining (ICDM)*, 2018.

Mohammadreza Kasaei, Keyhan Kouhkiloui Babarahmati, Zhibin Li, and Mohsen Khadem. A data-efficient neural ode framework for optimal control of soft manipulators. In *Annual Conference on Robot Learning*, 2023.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2017.

Naoki Kiyohara, Edward Johns, and Yingzhen Li. Neural stochastic flows: Solver-free modelling and inference for sde solutions. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Wilhelm Kutta. *Beitrag zur näherungsweisen Integration totaler Differentialgleichungen*. Teubner, 1901.

John M Lee. Smooth manifolds. In *Introduction to smooth manifolds*. Springer, 2013.

Kookjin Lee and Eric J Parish. Parameterized neural ordinary differential equations: Applications to computational physics problems. *Proceedings of the Royal Society A*, 2021.

Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In *Proc. of the International Conference on Web Search and Data Mining (WSDM)*, 2020.

Ernest Lindelöf. Sur l'application des méthodes d'approximations successives à l'étude des intégrales réelles des équations différentielles ordinaires. *Journal de mathématiques pures et appliquées*, 1894.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2023.

Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural sde: Stabilizing neural ode networks with stochastic noise. *arXiv preprint arXiv:1906.02355*, 2019.

Yaokun Liu, Xiaowang Zhang, Minghui Zou, and Zhiyong Feng. Attribute simulation for item embedding enhancement in multi-interest recommendation. In *Proc. of the International Conference on Web Search and Data Mining (WSDM)*, 2024a.

Yuxi Liu, Lianghao Xia, and Chao Huang. Selfgnn: Self-supervised graph neural networks for sequential recommendation. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2024b.

Sam McCallum, Zander W Blasingame, Timothy Herschell, Niklas Rindtorff, Alexander Tong, and James Foster. Strong stochastic flow maps. *arXiv preprint arXiv:2606.01086*, 2026.

Giangiacomo Mercatali, Andre Freitas, and Jie Chen. Graph neural flows for unveiling systemic interactions among irregularly sampled time series. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.

Alexander Norcliffe, Cristian Bodnar, Ben Day, Jacob Moss, and Pietro Liò. Neural ode processes. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2021.

Zhengkai Pan, Peter Potaptchik, Wenxi Yao, Michael S. Albergo, and Jakiw Pidstrigach. Itô maps for any-step sdes. *arXiv preprint arXiv:2606.11156*, 2026.

Émile Picard. Memoire sur la theorie des equations aux derivees partielles et la methode des approximations successives. *Journal de Mathématiques pures et appliquées*, 1890.

Yifang Qin, Wei Ju, Hongjun Wu, Xiao Luo, and Ming Zhang. Learning graph ode for continuous-time sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2024.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Carl Runge. Über die numerische auflösung von differentialgleichungen. *Mathematische Annalen*, 1895.

Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. An attentive inductive bias for sequential recommendation beyond the self-attention. In *Proc. of the AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

Yehjin Shin, Jeongwhan Choi, Seojin Kim, and Noseong Park. Tv-rec: Time-variant convolutional filter for sequential recommendation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2021.

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 *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*, 2019.

Yan-Martin Tamm, Rinchin Damdinov, and Alexey Vasilev. Quality metrics in recommender systems: Do we calculate metrics consistently? In *Proc. of the ACM Conference on Recommender Systems (RecSys)*, 2021.

Tianchi. Ijcai-16 brick-and-mortar store recommendation dataset, 2018. URL `https://tianchi.aliyun.com/dataset/dataDetail?dataId=53`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Wei Xiao, Huiying Wang, Qifeng Zhou, and Qing Wang. Ss4rec: Continuous-time sequential recommendation with state space models. *arXiv preprint arXiv:2502.08132*, 2025.

Yueting Yang, Yao Li, Rongmei Zhao, and Shenggen Ju. Coupling global context with kinematic evolution for sequential recommendation. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2026.

Yaowen Ye, Lianghao Xia, and Chao Huang. Graph masked autoencoder for sequential recommendation. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2023.

Cagatay Yildiz, Markus Heinonen, and Harri Lähdesmäki. Continuous-time model-based reinforcement learning. In *Proc. of the International Conference on Machine Learning (ICML)*, 2021.

Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. Linear recurrent units for sequential recommendation. In *Proc. of the International Conference on Web Search and Data Mining (WSDM)*, 2024.

Muhammad Zakwan, Loris Di Natale, Bratislav Svetozarevic, Philipp Heer, Colin N Jones, and G Ferrari Trecate. Physically consistent neural odes for learning multi-physics systems. *IFAC-PapersOnLine*, 2023.

Yipeng Zhang, Xin Wang, Hong Chen, and Wenwu Zhu. Adaptive disentangled transformer for sequential recommendation. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2023.

Yixin Zhang, Yong Liu, Yonghui Xu, Hao Xiong, Chenyi Lei, Wei He, Lizhen Cui, and Chunyan Miao. Enhancing sequential recommendation with graph contrastive learning. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2022.

Xuanle Zhao, Duzhen Zhang, Han Liyuan, Tielin Zhang, and Bo Xu. Ode-based recurrent model-free reinforcement learning for pomdps. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Fan Zhou and Liang Li. Forecasting reservoir inflow via recurrent neural odes. In *Proc. of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

APPENDIX

A MATHEMATICAL DETAILS

Remark 1. For any mapping $f : \mathcal{X} \to \mathcal{Y}$ between metric spaces $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_{\mathcal{Y}})$, its Lipschitz seminorm is defined by

$$\|f\|_L \coloneqq \sup_{x_1 \neq x_2} \frac{d_{\mathcal{Y}}(f(x_1), f(x_2))}{d_{\mathcal{X}}(x_1, x_2)}. \tag{23}$$

This seminorm is the tight Lipschitz constant. Indeed, let $\mathcal{C}_f$ denote the set of all $C \geq 0$ satisfying

$$d_{\mathcal{Y}}(f(x_1), f(x_2)) \leq C d_{\mathcal{X}}(x_1, x_2) \quad \forall x_1, x_2 \in \mathcal{X}. \tag{24}$$

With the convention $\inf \emptyset = \infty$, every $C \in \mathcal{C}_f$ is an upper bound of the ratios defining $\|f\|_L$, so $\|f\|_L \leq \inf \mathcal{C}_f$. Conversely, if $\|f\|_L < \infty$, then the definition of supremum gives

$$d_{\mathcal{Y}}(f(x_1), f(x_2)) \leq \|f\|_L d_{\mathcal{X}}(x_1, x_2) \quad \forall x_1, x_2 \in \mathcal{X}, \tag{25}$$

and hence $\|f\|_L \in \mathcal{C}_f$, which gives $\inf \mathcal{C}_f \leq \|f\|_L$. Therefore, $\|f\|_L = \inf \mathcal{C}_f$. In particular, $\|f\|_L < 1$ if and only if f admits a Lipschitz constant $C < 1$.

Theorem 2.1 (Invertibility). *Let the residual flow map be defined as $F(t, x) \coloneqq x + \varphi(t) \odot g(t, x)$, where $x \in \mathcal{X}$ is a state, φ is a temporal encoder, and g is a Lipschitz continuous function. If the state space $\mathcal{X}$ is a complete metric space and $\|\varphi\|_\infty \|g\|_L < 1$, then the flow map $F(t, \cdot)$ is invertible, where $\|\cdot\|_\infty$ and $\|\cdot\|_L$ denote the infinity norm and the Lipschitz seminorm operators, respectively.*

Proof. Let $t \in \mathcal{T}$ be an arbitrary but fixed time. We define the intermediate map $q(x) \coloneqq F(t, x) - x$. Given the global condition $\|\varphi\|_\infty \|g\|_L < 1$ and the Lipschitz continuity of the function g with respect to x, we can show that q is a contractive mapping as follows:

$$\begin{aligned} \|q(x_1) - q(x_2)\| &\leq \|\varphi(t)\|_\infty \|g(t, x_1) - g(t, x_2)\| && (26) \\ &\leq \|\varphi\|_\infty \|g\|_L \|x_1 - x_2\|. && (27) \end{aligned}$$

Eq. (27) follows from the definition of the infinity norm $\|\varphi(t)\|_\infty \leq \|\varphi\|_\infty$ and the fact that g has a Lipschitz seminorm $\|g\|_L$. By the assumption $\|\varphi\|_\infty \|g\|_L < 1$, the Lipschitz seminorm of q is strictly less than 1 for any time t.

Now, consider the equation $F(t, x) = x^\star$ for an arbitrary target state $x^\star$. Let us define an operator $T(x) \coloneqq x^\star - q(x)$. Since $\|T(x_1) - T(x_2)\| = \|q(x_1) - q(x_2)\|$, the operator $T(\cdot)$ is also a contractive mapping.

Since $\mathcal{X}$ is a complete metric space, there exists a unique fixed point x such that $T(x) = x$ by the contraction mapping theorem (Banach, 1922). Because $T(x) = x$ is equivalent to $F(t, x) = x^\star$ by definition, this guarantees that for any arbitrary target $x^\star$ (surjectivity), there exists exactly one corresponding input x (injectivity). Consequently, the residual flow map $F(t, \cdot)$ is bijective, which is the necessary and sufficient condition for the existence of an inverse function. □

Theorem 2.2 (Boundedness). *Assume that a temporal encoder satisfies $\varphi(t) \in [0, 1]^d$ for all $t \in \mathcal{T}$. Then, the GRU Flow is closed on $\mathcal{H}^d$. That is, $F(t, \boldsymbol{h}) \in \mathcal{H}^d$ for all $\boldsymbol{h} \in \mathcal{H}^d$.*

Proof. Fix a coordinate $i \in \{1, \ldots, d\}$. Define $\boldsymbol{h} \coloneqq (h_1, \ldots, h_d)$ and $\gamma_i(t, \boldsymbol{h}) \coloneqq \varphi_i(t) \cdot z_i(t, \boldsymbol{h})$. Since $\alpha, \beta \in [0, 1)$, we have $\varphi_i(t) \in [0, 1]$, $z_i(t, \boldsymbol{h}) \in (0, 1)$, and $\gamma_i(t, \boldsymbol{h}) \in [0, 1)$. Moreover, the i-th coordinate of $F(t, \boldsymbol{h})$ is given by

$$\begin{aligned} F_i(t, \boldsymbol{h}) &= h_i + \gamma_i(t, \boldsymbol{h}) \cdot (c_i(\boldsymbol{h}) - h_i) && (28) \\ &= (1 - \gamma_i(t, \boldsymbol{h})) \cdot h_i + \gamma_i(t, \boldsymbol{h}) \cdot c_i(\boldsymbol{h}). && (29) \end{aligned}$$

Thus, $F_i(t, \boldsymbol{h})$ is a convex combination of $h_i, c_i(t, \boldsymbol{h}) \in (-1, 1)$. Therefore, $F_i(t, \boldsymbol{h}) \in (-1, 1)$ and so, the GRU Flow is closed on $\mathcal{H}^d$. □

Theorem 2.3 (Corestriction Invertibility). *Assume that f_* are Lipschitz continuous with $\|f_*\|_L < 1$, $\varphi(t) \in [0,1]^d$, and $\alpha(6+5\beta) \leq 4$. Then, the GRU Flow map $F_t(\cdot) \coloneqq F(t,\cdot)$ is invertible onto its image for any fixed $t \in \mathcal{T}$. In other words, its corestriction $\bar{F}_t : \mathcal{H}^d \to F_t(\mathcal{H}^d)$ is bijective.*

Proof. Let a time $t \in \mathcal{T}$ be fixed. Note that the Lipschitz seminorm $\|\cdot\|_L$ is understood with respect to the spectral norm $\|\cdot\|_2$. This proof is a rigorous reformulation of prior work (Biloš et al., 2021). In particular, since $\mathcal{H}^d$ is not a complete metric space, we can not prove the invertibility of F by applying the contraction mapping theorem (Banach, 1922). To bypass this limitation, we prove that F is injective, while defining the corestriction of F to its image $F(\mathcal{H}^d)$ so that surjectivity holds by construction. Consequently, this corestriction is bijective, and hence F is invertible on its image.

Step 1. We first induce the Lipschitz boundedness for reset gate and update gate, respectively.

$$\|r(t,\boldsymbol{h}_1) - r(t,\boldsymbol{h}_2)\|_2 = \beta\|\sigma(f_r(t,\boldsymbol{h}_1)) - \sigma(f_r(t,\boldsymbol{h}_2))\|_2 \tag{30}$$

$$\leq \beta \underbrace{\|\sigma\|_L}_{=\frac{1}{4}} \|f_r(t,\boldsymbol{h}_1) - f_r(t,\boldsymbol{h}_2)\|_2 = \frac{\beta}{4}\|f_r(t,\boldsymbol{h}_1) - f_r(t,\boldsymbol{h}_2)\|_2 \tag{31}$$

$$\leq \frac{\beta}{4}\|f_r\|_L\|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2. \tag{32}$$

Moreover, we also have

$$\sigma(\cdot) \in (0,1) \implies r_i(t,\boldsymbol{h}) \in (0,\beta) \quad \forall i \in \{1,\dots,d\} \implies \|r(t,\boldsymbol{h})\|_\infty < \beta. \tag{33}$$

Similarly, we obtain

$$\|z(t,\boldsymbol{h}_1) - z(t,\boldsymbol{h}_2)\|_2 \leq \frac{\alpha}{4}\|f_z\|_L\|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2, \quad \|z(t,\boldsymbol{h})\|_\infty < \alpha. \tag{34}$$

Step 2. We also induce the Lipschitz boundedness for reset-modulated state $m(t,\boldsymbol{h}) \coloneqq r(t,\boldsymbol{h}) \odot \boldsymbol{h}$.

$$m(t,\boldsymbol{h}_1) - m(t,\boldsymbol{h}_2) = r(t,\boldsymbol{h}_1) \odot \boldsymbol{h}_1 - r(t,\boldsymbol{h}_2) \odot \boldsymbol{h}_2 \tag{35}$$

$$= r(t,\boldsymbol{h}_1) \odot (\boldsymbol{h}_1 - \boldsymbol{h}_2) + (r(t,\boldsymbol{h}_1) - r(t,\boldsymbol{h}_2)) \odot \boldsymbol{h}_2. \tag{36}$$

By the submultiplicativity of Hadamard products,

$$\|m(t,\boldsymbol{h}_1) - m(t,\boldsymbol{h}_2)\|_2 \leq \underbrace{\|r(t,\boldsymbol{h}_1)\|_\infty}_{\text{Eq. (33)}} \|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2 + \underbrace{\|\boldsymbol{h}_2\|_\infty}_{<1} \underbrace{\|r(t,\boldsymbol{h}_1) - r(t,\boldsymbol{h}_2)\|_2}_{\text{Eq. (32)}} \tag{37}$$

$$\leq \beta\left(1 + \frac{1}{4}\|f_r\|_L\right)\|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2. \tag{38}$$

Step 3. Next, we induce the Lipschitz boundedness for candidate state by $\|\tanh\|_L = 1$:

$$\|c(t,\boldsymbol{h}_1) - c(t,\boldsymbol{h}_2)\|_2 = \|\tanh(f_c(t,m(t,\boldsymbol{h}_1))) - \tanh(f_c(t,m(t,\boldsymbol{h}_2)))\|_2 \tag{39}$$

$$\leq \|\tanh\|_L\|f_c\|_L \underbrace{\|m(t,\boldsymbol{h}_1) - m(t,\boldsymbol{h}_2)\|_2}_{\text{Eq. (38)}} \tag{40}$$

$$\leq \beta\|f_c\|_L\left(1 + \frac{1}{4}\|f_r\|_L\right)\|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2. \tag{41}$$

Step 4. We finally induce the Lipschitz boundedness for difference term $q(t,\boldsymbol{h}) \coloneqq c(t,\boldsymbol{h}) - \boldsymbol{h}$.

$$\|q(t,\boldsymbol{h}_1) - q(t,\boldsymbol{h}_2)\|_2 = \|c(t,\boldsymbol{h}_1) - \boldsymbol{h}_1 - c(t,\boldsymbol{h}_2) + \boldsymbol{h}_2\|_2 \tag{42}$$

$$\leq \underbrace{\|c(t,\boldsymbol{h}_1) - c(t,\boldsymbol{h}_2)\|_2}_{\text{Eq. (41)}} + \|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2 \tag{43}$$

$$\leq \left(1 + \beta\|f_c\|_L\left(1 + \frac{1}{4}\|f_r\|_L\right)\right)\|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2. \tag{44}$$

On the other hand, we trivially gain

$$c(t,\boldsymbol{h}) \in (-1,1)^d,\ \boldsymbol{h} \in (-1,1)^d \implies q(t,\boldsymbol{h}) \in (-2,2)^d \implies \|q(t,\boldsymbol{h})\|_\infty < 2. \tag{45}$$

Step 5. We combine all inequalities to induce the Lipschitz boundedness for GRU residual term except for temporal encoder φ. By defining $g(t, \boldsymbol{h}) \coloneqq z(t, \boldsymbol{h}) \odot q(t, \boldsymbol{h})$, we obtain

$$g(t, \boldsymbol{h}_1) - g(t, \boldsymbol{h}_2) = z(t, \boldsymbol{h}_1) \odot (q(t, \boldsymbol{h}_1) - q(t, \boldsymbol{h}_2)) + (z(t, \boldsymbol{h}_1) - z(t, \boldsymbol{h}_2)) \odot q(t, \boldsymbol{h}_2). \quad (46)$$

Again, by the triangular inequality,

$$\begin{aligned}
&\|g(t, \boldsymbol{h}_1) - g(t, \boldsymbol{h}_2)\|_2 && (47)\\
&\leq \underbrace{\|z(t, \boldsymbol{h}_1)\|_\infty}_{\text{Eq. (34)}} \underbrace{\|q(t, \boldsymbol{h}_1) - q(t, \boldsymbol{h}_2)\|_2}_{\text{Eq. (44)}} + \underbrace{\|q(t, \boldsymbol{h}_2)\|_\infty}_{\text{Eq. (45)}} \underbrace{\|z(t, \boldsymbol{h}_1) - z(t, \boldsymbol{h}_2)\|_2}_{\text{Eq. (34)}} && (48)\\
&< \alpha \left(1 + \beta \|f_c\|_L \left(1 + \frac{1}{4}\|f_r\|_L\right)\right) \|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2 + 2 \cdot \frac{\alpha}{4} \|f_z\|_L \|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2 && (49)\\
&= \alpha \left(1 + \beta \|f_c\|_L \left(1 + \frac{1}{4}\|f_r\|_L\right) + \frac{1}{2}\|f_z\|_L\right) \|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2. && (50)
\end{aligned}$$

Step 6. Now we prove that *if the residual term is contractive, i.e.,* $\|R(t, \cdot)\|_L < 1$*, then* F *is injective on* $\mathcal{H}^d$. Rewrite the GRU Flow as follows:

$$F(t, \boldsymbol{h}) = \boldsymbol{h} + \underbrace{\varphi(t) \odot g(t, \boldsymbol{h})}_{=: R(t, \boldsymbol{h})}. \quad (51)$$

Letting $F(t, \boldsymbol{h}_1) = F(t, \boldsymbol{h}_2)$, we have

$$\begin{aligned}
\|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2 &= \|F(t, \boldsymbol{h}_1) - F(t, \boldsymbol{h}_2) + R(t, \boldsymbol{h}_2) - R(t, \boldsymbol{h}_1)\|_2 && (52)\\
&= \|R(t, \boldsymbol{h}_1) - R(t, \boldsymbol{h}_2)\|_2 \leq \underbrace{\|R(t, \cdot)\|_L}_{=: C} \cdot \|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2. && (53)
\end{aligned}$$

This implies $\|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2 \leq C\|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2$, thus $(1 - C)\|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2 \leq 0$. Since $1 - C > 0$, we conclude $\boldsymbol{h}_1 = \boldsymbol{h}_2$. Therefore, F is injective on $\mathcal{H}^d$.

Constructive Result. Since $\varphi(t) \in [0, 1]^d$ and Eq. (50), we obtain

$$\begin{aligned}
\|R(t, \cdot)\|_L &< \underbrace{\|\varphi(t)\|_\infty}_{\leq 1} \alpha \left(1 + \beta \underbrace{\|f_c\|_L}_{<1} \left(1 + \frac{1}{4} \underbrace{\|f_r\|_L}_{<1}\right) + \frac{1}{2} \underbrace{\|f_z\|_L}_{<1}\right) && (54)\\
&< \alpha \left(1 + \beta \left(1 + \frac{1}{4}\right) + \frac{1}{2}\right) = \alpha \left(\frac{3}{2} + \frac{5}{4}\beta\right) \leq 1. && (55)
\end{aligned}$$

Hence, the GRU Flow F is injective on $\mathcal{H}^d$. Since $\mathcal{H}^d$ is not a complete metric space, we consider the following corestriction of F_t:

$$\bar{F}_t : \mathcal{H}^d \longrightarrow F_t(\mathcal{H}^d), \quad \boldsymbol{h} \longmapsto F(t, \boldsymbol{h}). \quad (56)$$

By construction, $\bar{F}_t$ is surjective. Moreover, since F_t is injective, $\bar{F}_t$ is also injective. Therefore, $\bar{F}_t$ is bijective and thus admits an inverse, $\bar{F}_t^{-1} : F_t(\mathcal{H}^d) \to \mathcal{H}^d$.

From a neural-network perspective, this corestriction is natural. The goal is not to invert an arbitrary future state $\boldsymbol{h}(t) \in \mathcal{H}^d$ at time t back to an initial state $\boldsymbol{h}(0)$, but rather to evolve an initial state into a task-relevant state $\boldsymbol{h}(t) \in F_t(\mathcal{H}^d)$ at a suitable time t for the downstream task. □

Corollary 3.1 (Corestriction Invertibility for PGNF)**.** *Assume that the conditions of Theorem 2.3 hold for the temporal encoder and the update and reset gates, and additionally* $\|\tilde{\mu}_{c_1}\|_L, \|\tilde{\mu}_{c_2}\|_L < 1/2$*, and* $\alpha\,(16 + 5\beta) \leq 8$*. Then,* $F(t, \cdot \,|\, u)$ *is corestriction-invertible for any target time* $t \in \mathcal{T}$.

Proof. The proof follows Theorem 2.3, with only the candidate-state bound modified to account for the two branches of f_c. Fix a time $t \in \mathcal{T}$ and introduce $s \coloneqq s_{uv}^t$, $\eta(\boldsymbol{h}) \coloneqq \eta(t, s, \boldsymbol{h})$ for simplicity. For all three aggregation schemes (concatenation, addition, and shift), the dependence of η on $\boldsymbol{h}$ is affine with an identity state component. Therefore, we obtain

$$\|\eta(\boldsymbol{h}_1) - \eta(\boldsymbol{h}_2)\|_2 = \|\boldsymbol{h}_1 - \boldsymbol{h}_2\|_2 \implies \|\eta\|_L = 1. \quad (57)$$

From Theorem 2.3, the reset-modulated state $m(t, \boldsymbol{h}) = r(t, \boldsymbol{h}) \odot \boldsymbol{h}$ satisfies

$$\|m\|_L < \frac{5}{4}\beta. \tag{58}$$

Thus, the PGNF candidate pre-activation satisfies

$$\|f_c\|_L < \|\tilde{\mu}_{c_1}\|_L + \|\tilde{\mu}_{c_2}\|_L \|m\|_L < \frac{1}{2} + \frac{5}{8}\beta. \tag{59}$$

Since $\|\tanh\|_L = 1$, the same bound holds for the candidate state $c(t, \boldsymbol{h} \,|\, u)$. Using this bound in the remaining steps of Theorem 2.3 gives

$$\|R(t, \cdot \,|\, u)\|_L < \alpha \left(2 + \frac{5}{8}\beta\right) \leq 1 \implies \|R(t, \cdot \,|\, u)\|_L < 1. \tag{60}$$

By the same argument as in Theorem 2.3, $F(t, \cdot \,|\, u)$ is also corestriction-invertible. □

B Internal Temporal Encoders

We present the internal temporal encoder $\phi : \mathcal{T} \to \mathbb{R}^d$. The encoder $\psi : \mathcal{T} \to \mathbb{R}^d$ is defined independently but follows the same construction, so we omit its description. The learnable parameters are denoted by $\omega_i \in \mathbb{R}$. While the outermost temporal encoder $\varphi(t)$ must strictly satisfy the boundedness condition, the internal encoder do not necessarily need to meet this constraint. This relaxation is mathematically justified because they are fixed with respect to Lipschitz boundedness and effectively cancel out when computing the differences. Consequently, we can employ a straightforward *linear encoder* which independently scales the elapsed time for each dimension:

$$\phi(t) \coloneqq [\phi_1(t), \cdots, \phi_d(t)]^\top, \quad \phi_i(t) \coloneqq \omega_i t. \tag{61}$$

Note that we define an *identity encoder* by setting $\omega_i = 1$ for all $i \in \{1, \ldots, d\}$.

C Aggregation Functions

We specify the concrete choices of the temporal-feature aggregation operator agg.

Concatenation. The concatenation method appends the encoded temporal quantities to the item state. Given that the predefined temporal encoders ϕ and ψ map the inputs to a d-dimensional space, the concatenated representation is formulated as

$$\eta(t, s, \boldsymbol{h}) \coloneqq \phi(t) \parallel \phi(s) \parallel \psi(t - s) \parallel \boldsymbol{h} \in \mathbb{R}^{4d}. \tag{62}$$

Addition. The addition combines the item state and temporal encodings by element-wise addition. This option assumes that the all item state and time encoding lie in the same d-dimensional space:

$$\eta(t, s, \boldsymbol{h}) \coloneqq \phi(t) + \phi(s) + \psi(t - s) + \boldsymbol{h} \in \mathbb{R}^d. \tag{63}$$

Shift. The shift conditions the item state on both absolute and relative temporal information. The temporal context is constructed by concatenating these encodings:

$$\boldsymbol{\kappa} \coloneqq \phi(s) \parallel \phi(t) \parallel \psi(t - s) \in \mathbb{R}^{3d}. \tag{64}$$

A temporal shift is then computed using a learnable linear map:

$$\boldsymbol{\nu} \coloneqq \mu_{\mathrm{shift}}(\boldsymbol{\kappa}) \in \mathbb{R}^d. \tag{65}$$

Finally, the temporally conditioned representation of the item state $\boldsymbol{h} \in \mathbb{R}^d$ is defined as

$$\eta(t, s, \boldsymbol{h}) \coloneqq \boldsymbol{h} + \boldsymbol{\nu} \in \mathbb{R}^d. \tag{66}$$

D EXPERIMENTAL DETAILS

To facilitate reproducibility, we provide dataset statistics, hardware and software configurations, baseline implementation details, and hyperparameter search protocols.

Experimental Setup. Both the proposed model and all baselines are implemented using PyTorch 2.10, accelerated by CUDA 12.8. The entire set of experiments is performed on a single NVIDIA A6000 GPU. For all baseline models, we utilize their official implementations. If the source code is publicly unavailable, we rigorously re-implement the models based on the detailed descriptions in their respective papers.

The hyperparameter search spaces for the baselines strictly follow the configurations proposed by the original paper. When computationally tractable, we conduct an exhaustive grid search. However, for continuous search spaces or scenarios involving prohibitive computational costs, we employ Bayesian optimization (Snoek et al., 2012) for a minimum of 50 iterations to identify the optimal configurations.

Datasets. We conduct experiments across three distinct domains listed in Tab. I to demonstrate practical applicability. Specifically, the Amazon-Tools (He & McAuley, 2016) and Tmall-Buy (Tianchi, 2018) datasets comprise *purchase* histories, whereas the Behance (He et al., 2016) dataset records explicit *like* interactions on artworks.

To rigorously evaluate the model's ability to predict future events using historical data, we chronologically partition the dataset into training, validation, and test sets at an 8:1:1 interaction volume ratio. To prevent any temporal overlap, we ensure that these sets contain strictly disjoint timestamps, indicating simultaneous interactions are never divided across different splits. Following standard conventions in sequential recommendation, a user representation at target time t is encoded using their complete preceding history $\mathcal{S}_u^t$. Importantly, this formulation avoids temporal data leakage by strictly relying only on historical signals.

Table I: Statistics of the preprocessed datasets.

Dataset	Users	Items	Interactions	Avg. Length (±SD)	Density	Total Time	Filtering
Amazon-Tools	16,638	10,217	134,476	8.08 (± 5.86)	0.079%	5371 days	5-core
Behance	23,724	29,794	687,070	28.96 (±60.30)	0.097%	158 days	5-core
Tmall-Buy	125,553	58,058	2,064,290	16.44 (± 8.43)	0.028%	152 days	10-core

Hyperparameter Search. Tab. II presents the hyperparameter search space and the optimal configurations for each dataset. We use Bayesian optimization to identify the optimal values of the tunable hyperparameters. For consistency, we also include the temporal encoder φ, which is fixed to Tanh.

Table II: Hyperparameter search space and configurations.

Hyperparameter	Search Space	Best Value		
		Amazon	Behance	Tmall
Graph density (λ)	{0.0, 0.1, 0.2, . . . , 1.0}	1.0	0.2	0.8
State dimension (d)	{32, 64, 128, 256, 512}	64	128	256
Dropout rate	[0.0, 0.3]	0.1680	0.2493	0.0645
Readout	{Mean, Gating}	Mean	Gating	Mean
Aggregation (agg)	{Concat, Add, Shift}	Add	Shift	Add
Temporal encoder (φ)	{Tanh}	Tanh	Tanh	Tanh
Temporal encoder (ϕ, ψ)	{Tanh, Linear, Identity}	Tanh	Tanh	Linear

For optimization, we use Adam with a learning rate of 0.001 and a weight decay of 0.0001, with a batch size of 1024. We fix the update and reset gate scalers, α and β, to $2/5$ and $4/5$, respectively. These scalers control the contractivity of the GRU Flow architecture, and fixing them at moderate values avoids vanishing gradients without degrading performance when the remaining hyperparameters are properly tuned.

Standard Deviations. All experimental results are reported as averages over five random seeds (2024, 2025, 2026, 2027, and 2028). Here, we present the corresponding standard deviations, which were omitted from the main tables due to space limitations.

Table III: Standard deviation of Tab. 1

	Model	Amazon-Tools					Behance					Tmall-Buy				
		NDCG		HR		MRR	NDCG		HR		MRR	NDCG		HR		MRR
		@10	@20	@10	@20		@10	@20	@10	@20		@10	@20	@10	@20	
General	SASRec	0.14	0.14	0.16	0.30	0.13	0.21	0.20	0.28	0.25	0.19	0.05	0.05	0.11	0.13	0.04
	FEARec	0.06	0.10	0.04	0.27	0.09	0.16	0.16	0.12	0.14	0.18	0.12	0.14	0.21	0.33	0.10
	BSARec	0.17	0.13	0.25	0.18	0.14	0.06	0.05	0.09	0.03	0.07	0.15	0.16	0.23	0.24	0.13
	TV-Rec	0.12	0.14	0.16	0.29	0.13	0.13	0.15	0.08	0.16	0.16	0.05	0.05	0.08	0.08	0.04
	KERec	0.10	0.12	0.17	0.28	0.09	0.19	0.20	0.22	0.31	0.20	0.08	0.08	0.13	0.15	0.06
Continuous-Time	CTRec	0.11	0.10	0.24	0.21	0.06	0.20	0.15	0.33	0.31	0.14	0.06	0.05	0.10	0.07	0.04
	TiSASRec	0.17	0.13	0.40	0.41	0.10	0.27	0.24	0.13	0.10	0.29	0.05	0.06	0.09	0.13	0.04
	CTSMA	0.12	0.16	0.12	0.24	0.15	0.17	0.14	0.16	0.08	0.17	0.03	0.01	0.06	0.05	0.02
	TGSRec	0.07	0.11	0.20	0.42	0.05	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	GNG-ODE	0.15	0.12	0.28	0.30	0.11	0.15	0.11	0.17	0.14	0.15	0.08	0.09	0.12	0.17	0.07
	GDERec	0.10	0.10	0.13	0.22	0.08	0.06	0.05	0.12	0.15	0.04	0.10	0.11	0.14	0.20	0.09
	SS4Rec	0.26	0.29	0.34	0.45	0.26	0.15	0.16	0.19	0.29	0.18	0.09	0.10	0.16	0.19	0.08
	TGODE	0.20	0.22	0.33	0.41	0.18	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	PGNF	0.06	0.04	0.17	0.12	0.05	0.05	0.07	0.10	0.20	0.05	0.16	0.12	0.13	0.17	0.15

Table IV: Standard deviation of Tab. 2.

Variant				Amazon-Tools			Behance			Tmall-Buy		
Readout	ICG	TUC	TGM	NDCG	HR	MRR	NDCG	HR	MRR	NDCG	HR	MRR
Mean	✓	✓	✓	0.04	0.12	0.05	0.16	0.38	0.11	0.12	0.17	0.15
Gating	✓	✓	✓	0.18	0.26	0.16	0.07	0.20	0.05	0.07	0.10	0.08
Best	✗	✓	✓	0.12	0.25	0.10	0.06	0.21	0.07	0.34	0.55	0.30
	✓	✗	✓	0.05	0.14	0.03	0.11	0.17	0.11	0.48	0.50	0.48
	✓	✓	✗	0.17	0.46	0.09	0.08	0.24	0.10	0.12	0.24	0.09

E HYPERPARAMETER STUDY ON STATE DIMENSION

We investigated the effect of state dimension on model performance. As shown in Fig. I, increasing the state dimension does not consistently improve performance, suggesting potential overfitting due to overparameterization. This effect may be partly attributed to the absence of explicit L_2-regularization during training.

Our model substantially outperforms the existing state-of-the-art models while using same or smaller state dimensions: $d = 64$, matching TV-Rec on Amazon-Tools; $d = 128$, matching SS4Rec on Behance; and $d = 256$, half of the $d = 512$ used by GNG-ODE on Tmall-Buy. This suggests that the gains arise from the model design rather than simply increased representational capacity.

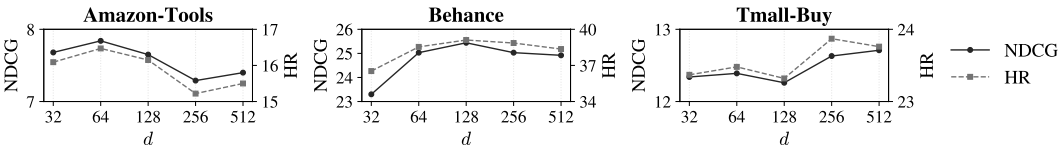

Figure I: Effect of the state dimensions d on PGNF accuracy.