PATHWISE ALIGNMENT FOR COLLABORATIVE FILTERING: BILATERAL CONDITIONAL FLOW REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Collaborative filtering predicts user-item interactions from implicit feedback. We focus on matrix factorization (MF) trained with the Bayesian Personalized Ranking loss. This loss updates each user embedding only through that user's observed items and sampled negatives, so sparse feedback gives MF few constraints. To constrain each user through other users, we propose **P**athwise **A**lignment for **C**ollaborative **F**iltering (PACF), which trains MF with one network shared by all training pairs. At points between user and item embeddings, a flow matching term trains the network to predict the user-to-item step, reversed and scaled for sampled negatives. A trajectory consistency term pulls each user's predicted path to end at the observed item. Both terms update the embeddings, so the network is discarded after training, thereby maintaining the efficiency of MF at inference. We prove that a network with bounded sensitivity to its input point accurately fits users who share an item only if their embeddings are close, so these users constrain one another. Against eleven discriminative and generative baselines, PACF achieves the highest performance on all seven datasets. Its relative gains over MF are largest where feedback is sparsest. The shared network thus supplies the missing constraints through other users.

1 INTRODUCTION

Collaborative Filtering (CF) learns user and item embeddings from sparse interactions and scores pairs by their inner product (Goldberg et al., 1992; Papadakis et al., 2022; He et al., 2017). Pairwise ranking losses supervise these scores, and inference uses the learned embeddings for static similarity scoring (Hsieh et al., 2017).

A pairwise loss constrains the inner products of observed and sampled pairs, leaving the remaining embedding geometry to the model's inductive bias (Steck et al., 2024). Sparse feedback supplies few such constraints per embedding. Graph encoders use interaction-graph propagation (He et al., 2020), while contrastive objectives impose constraints on the embedding distribution (Wu et al., 2021; Yu et al., 2022).

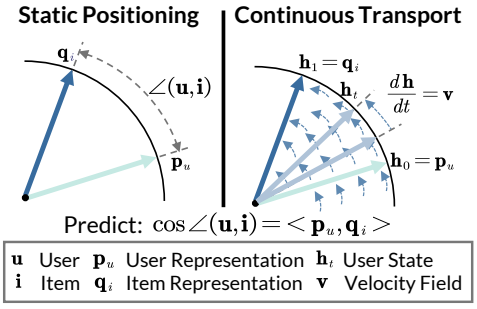


Figure 1: Comparison of supervision in unit space. The left panel shows scalar endpoint supervision. The right panel shows PACF fitting signed displacements at sampled states along user-item segments. PACF thus supervises the whole path from each user to its item instead of one endpoint score.

For Matrix Factorization (MF) (Koren et al., 2009) trained with Bayesian Personalized Ranking (BPR) (Rendle et al., 2009), we study an inductive bias that constrains each user through other users. To that end, **P**athwise **A**lignment for **C**ollaborative **F**iltering (PACF) augments the ranking loss with an auxiliary regression shared by all training pairs. As shown in Fig. 1, PACF places a directional target at sampled positions on the segment between the user and item embeddings of each training pair, pointing toward the item for an observed pair and back toward the user for a sampled negative pair. This **pathwise alignment** learns the shared predictor jointly with the embeddings.

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.

PACF sets this target to the user-to-item displacement at every position, with a sign set by the label, and we call this pair of signed targets **Bilateral Conditional Flow** (BCF). PACF fits one velocity network to these targets under a flow matching loss (Lipman et al., 2023) and ties the integrated endpoint to the observed item with a trajectory consistency term. Both losses update the embeddings, so PACF discards the network after training and ranks candidates with static inner products. When the network's sensitivity to its state input is bounded, PACF accurately fits users who share an item only if their embeddings are close, so each user draws constraints from the other users of its items.

Our contributions cover the training objective, its analysis, and its empirical evaluation:

- PACF reframes collaborative filtering as pathwise alignment, in which a Bilateral Conditional Flow regularizer turns scalar feedback into signed directional targets along user-item segments. One shared network fits these targets, and through gradient amortization its training signal stays in the static embeddings used for inner-product ranking.
- A shared network with bounded state sensitivity provably fits users of the same item accurately only if their embeddings are close, which constrains each user through the others.
- Against eleven discriminative and generative baselines on seven datasets from four domains, PACF achieves the highest performance on every dataset, with its largest relative gains over MF where feedback is sparsest, while maintaining the efficiency of MF at inference.

2 Related Work

Collaborative Filtering. CF learns user and item embeddings from observed interactions (Shi et al., 2014; Koren et al., 2022). MF scores pairs by an inner product, and BPR trains these embeddings to rank an observed item above a sampled negative item (Koren et al., 2009; Bokde et al., 2015; Rendle et al., 2009). Later methods enrich the scorer, as in Neural Collaborative Filtering (NCF) (He et al., 2017), or the representations, through graph propagation (He et al., 2020) and contrastive objectives on augmented views or neighbors (Wu et al., 2021; Yu et al., 2022; Zhang et al., 2025b). DirectAU aligns the embeddings of observed pairs and spreads them uniformly (Wang et al., 2022; Wang & Isola, 2020), while SCCF and RecDCL shape the embeddings through contrastive objectives (Wu et al., 2024; Zhang et al., 2024). Graph-regularized MF and UltraGCN tie the embeddings of related users and items through explicit graph constraints (Rao et al., 2015; Mao et al., 2021). PACF keeps the MF scorer and the BPR loss and adds a displacement regression shared by all training pairs, so one learned fit also constrains each user through the other users who share any of its items.

Flow Matching for Recommendation. Flow matching regresses a velocity field onto target velocities along conditional paths between coupled source and target points (Lipman et al., 2023), and an ordinary differential equation (ODE) integrates the field into trajectories (Chen et al., 2018; Finlay et al., 2020). Each PACF training pair fixes this coupling, so the conditional path becomes the relevant design choice (Peyré & Cuturi, 2019; Pooladian et al., 2023). Generative flow models use the field at inference to transform distributions (Liu et al., 2023; Gat et al., 2024), and recommender systems apply this design to interaction distributions, sequential preferences, and denoised social signals (Li et al., 2026; Huang et al., 2025). FlowCF applies it to binary interaction vectors with a behavior-guided prior (Liu et al., 2025). PACF instead fits the field to signed user-to-item displacements to regularize static embeddings during training, then discards it and scores items by inner products.

3 Preliminaries

3.1 Notations

Let $\mathcal{U}$ and $\mathcal{I}$ denote the sets of users and items, and let $\mathcal{O} \subseteq \mathcal{U} \times \mathcal{I}$ collect the observed interactions, with $\mathcal{I}_u = \{i : (u, i) \in \mathcal{O}\}$ for user u. We map each entity to a shared d-dimensional Euclidean space $\mathcal{M} = \mathbb{R}^d$. Users and items are parameterized by embeddings $\mathbf{P} \in \mathbb{R}^{|\mathcal{U}| \times d}$ and $\mathbf{Q} \in \mathbb{R}^{|\mathcal{I}| \times d}$ with rows $\mathbf{p}_u$ and $\mathbf{q}_i$. PACF is based on MF and randomly initializes $\mathbf{P}$ and $\mathbf{Q}$ as ID embedding tables. Each training pair carries a label $y_{ui} \in \{+1, -1\}$. A positive item i^+ of user u satisfies $(u, i^+) \in \mathcal{O}$ and $y_{ui} = +1$, and a sampled negative item i^- is drawn from the unobserved items $\mathcal{I} \setminus \mathcal{I}_u$ with $y_{ui} = -1$. Each positive pair is trained with K_- sampled negatives. The interpolation time $t \in [0, 1]$ indexes states on the segment from a user embedding at $t = 0$ to an item embedding at $t = 1$.

3.2 Problem Formulation

CF learns user and item embeddings from sparse feedback in the shared space $\mathcal{M}$. Their inner product defines the score $s(u, i)$ that ranks all candidate items for each user at inference:

$$s(u, i) = \langle \mathbf{p}_u, \mathbf{q}_i \rangle. \tag{1}$$

BPR (Rendle et al., 2009) ranks an observed item i^+ above a sampled negative i^- through their score difference and minimizes the expected logistic loss over sampled training triplets (u, i^+, i^-):

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{(u,i^+,i^-)} \left[-\log \sigma\big(s(u, i^+) - s(u, i^-)\big) \right], \tag{2}$$

where σ is the sigmoid. Eq. (2) updates each user embedding only through the scores of that user's observed and sampled pairs, so sparse feedback gives the embedding few constraints.

4 Methodology

The recommendation loss $\mathcal{L}_{\text{rec}}$ in Eq. (2) shapes the user and item embeddings through the scores of observed and sampled pairs. Fig. 2 shows the two losses PACF adds to each user-item segment, namely flow matching toward BCF targets at sampled states and trajectory consistency at the integrated endpoint. Their gradients update the static embeddings that Eq. (1) scores during inference.

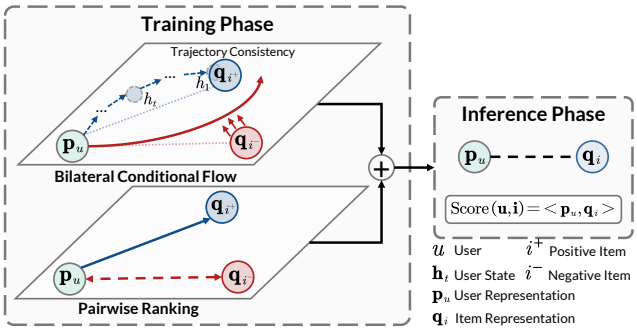


Figure 2: Framework of PACF. Left: **Training** fits the shared velocity network to the signed BCF targets at states $\mathbf{h}_t$ sampled on the reference segment. A trajectory consistency term on positive pairs pulls the RK4 endpoint $\Phi_1(\mathbf{p}_u)$ toward $\mathbf{q}_{i^+}$, so negative pairs enter the pathwise losses only through the flow matching term. Right: **Inference** discards the network and scores by inner product. PACF thus adds cost only during training and ranks with the same efficiency as MF.

4.1 Bilateral Conditional Flow

For a user u and an item i with label $y_{ui} \in \{+1, -1\}$, the intermediate state on the segment at time $t \in [0, 1]$ is the straight-line interpolation:

$$\mathbf{h}_t \equiv \mathbf{h}_t^{(u,i)} = (1-t)\,\mathbf{p}_u + t\,\mathbf{q}_i. \tag{3}$$

Its velocity $\dot{\mathbf{h}}_t = \mathbf{q}_i - \mathbf{p}_u$ is constant along the path. The observed label sets the direction and scale of the signed BCF regression target, which is therefore also constant along the segment:

$$\mathbf{u}_t(\mathbf{h}_t \mid y_{ui}) = \mu(y_{ui})\,(\mathbf{q}_i - \mathbf{p}_u), \quad \text{where } \mu(y) = \begin{cases} +1, & y = +1 \\ -\mu_-, & y = -1 \end{cases}. \tag{4}$$

In Eq. (4), $\mu_- \in [0, 1]$ scales the target on negative pairs. The BCF target points from the user embedding toward the item embedding for positive pairs and back toward the user for negative pairs.

4.2 Pathwise Flow Matching

In PACF, one velocity network $\mathbf{v}_{\boldsymbol{\theta}}$ is shared by all training pairs and regressed onto the BCF target $\mathbf{u}_t(\mathbf{h}_t \mid y_{ui})$ in Eq. (4). On a positive pair this target is the reference-path velocity used by conditional flow matching (Lipman et al., 2023), and on a sampled negative pair it reverses the velocity of that pair's own segment and scales it by μ_-. The flow matching loss is the expected squared error:

$$\mathcal{L}_{\text{flow}} = \mathbb{E}_{t,\,(u,i,y_{ui})} \left[\left\| \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{h}_t, t, \mathbf{q}_i, y_{ui}) - \mathbf{u}_t(\mathbf{h}_t \mid y_{ui}) \right\|_2^2 \right]. \tag{5}$$

In Eq. (5), $t \sim \text{Unif}[0, 1]$ is drawn independently of the training pair, which ranges over each positive pair and its K_- sampled negatives. The target keeps its $(\mathbf{p}_u, \mathbf{q}_i)$ dependence during backpropagation, so $\mathcal{L}_{\text{flow}}$ updates the embeddings as well as $\mathbf{v}_{\boldsymbol{\theta}}$, which serves only as a training-time regressor.

The velocity network is a three-layer residual multilayer perceptron (MLP) with hidden width d_h:

$$\mathbf{z} = \mathrm{GELU}\big(\mathbf{W}_1\,\mathrm{LN}([\mathbf{h}_t;\,\mathbf{q}_i;\,t;\,y_{ui}])\big), \quad \mathbf{v}_{\boldsymbol{\theta}} = \mathbf{W}_3\big(\mathbf{z} + \mathrm{GELU}(\mathbf{W}_2\,\mathbf{z})\big). \tag{6}$$

Layer normalization (LN) acts on the input, and the Gaussian error linear unit (GELU) follows each hidden projection. The trainable weights are $\boldsymbol{\theta} = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3\}$. The skip connection passes the hidden representation $\mathbf{z}$ directly to $\mathbf{W}_3$, and all training pairs share these weight matrices.

4.3 Trajectory Consistency

The flow matching loss $\mathcal{L}_{\text{flow}}$ in Eq. (5) supervises the field $\mathbf{v}_{\boldsymbol{\theta}}$ only at sampled reference states, while trajectory consistency integrates this field from $\mathbf{p}_u$ to $t = 1$ and supervises the resulting endpoint. For a positive pair (u, i^+), the trajectory starts at $\widetilde{\mathbf{h}}_0 = \mathbf{p}_u$ and follows the neural ODE:

$$\dot{\widetilde{\mathbf{h}}}_t = \mathbf{v}_{\boldsymbol{\theta}}(\widetilde{\mathbf{h}}_t, t, \mathbf{q}_{i^+}, +1), \tag{7}$$

with exact endpoint $\widetilde{\mathbf{h}}_1$. With $(\mathbf{q}_{i^+}, +1)$ fixed, the fixed-step fourth-order Runge–Kutta (RK4) solver (Butcher, 2016) with T steps approximates $\widetilde{\mathbf{h}}_1$ by $\Phi_1(\mathbf{p}_u)$ in all reported experiments. Each RK4 step evaluates the field once at the current state, twice at midpoint estimates, and once at an endpoint estimate. The trajectory consistency loss $\mathcal{L}_{\text{traj}}$ measures the squared distance between this endpoint and the observed item over positive pairs, whose BCF target leads to that item:

$$\mathcal{L}_{\text{traj}} = \mathbb{E}_{(u,i^+)}\left[\left\|\Phi_1(\mathbf{p}_u) - \mathbf{q}_{i^+}\right\|_2^2\right]. \tag{8}$$

The solver backpropagates this endpoint residual through all T RK4 steps to $\mathbf{p}_u$, $\mathbf{q}_{i^+}$, and $\boldsymbol{\theta}$.

4.4 Training Objective

Adding both pathwise losses to the ranking loss in Eq. (2) gives the full training objective:

$$\min_{\{\mathbf{p}_u\},\{\mathbf{q}_i\},\boldsymbol{\theta}} \mathcal{L}_{\text{rec}} + \lambda\mathcal{L}_{\text{flow}} + \beta\mathcal{L}_{\text{traj}}, \tag{9}$$

where $\lambda, \beta \geq 0$ weight the two auxiliary losses. Setting $\lambda = \beta = 0$ recovers BPR and its gradients.

4.5 Static Scoring through Gradient Amortization

Gradient amortization writes the fitting information of both auxiliary losses into the static embeddings through their training gradients, and these embeddings keep it after the velocity network and solver are discarded. The network and the RK4 solver thus add cost only during training.

Proposition 1 (User-Side Gradient Decomposition)**.** *For one training triplet (u, i^+, i^-), let ℓ_{flow} sum its sampled flow losses that depend on $\mathbf{p}_u$. Define $\Delta s = s(u, i^+) - s(u, i^-)$, $\mathbf{r}_u = \Phi_1(\mathbf{p}_u) - \mathbf{q}_{i^+}$, and $\mathbf{J}_\Phi(\mathbf{p}_u) = \partial\Phi_1/\partial\mathbf{p}_u$. With the item embeddings and $\boldsymbol{\theta}$ fixed, the per-sample objective $\ell = -\log\sigma(\Delta s) + \lambda\ell_{\text{flow}} + \beta\|\mathbf{r}_u\|_2^2$ has a negative gradient with one additive term per loss:*

$$-\nabla_{\mathbf{p}_u}\ell = \sigma(-\Delta s)(\mathbf{q}_{i^+} - \mathbf{q}_{i^-}) - \lambda\nabla_{\mathbf{p}_u}\ell_{\text{flow}} - 2\beta\mathbf{J}_\Phi(\mathbf{p}_u)^\top\mathbf{r}_u. \tag{10}$$

The first term moves $\mathbf{p}_u$ along $\mathbf{q}_{i^+} - \mathbf{q}_{i^-}$ with $\sigma(-\Delta s)$, which grows as the score difference falls. The flow term differentiates the sampled regression errors through the state and the signed target, and the transpose of $\mathbf{J}_\Phi(\mathbf{p}_u)$, which describes how the initial user embedding moves the RK4 endpoint, returns $\mathbf{r}_u$ to $\mathbf{p}_u$. Optimization forms each embedding step from their sum, which reduces to the BPR gradient when the flow and endpoint residuals both vanish, so residuals carry all pathwise signal.

5 Algorithmic Analysis

Flow matching updates the embeddings in proportion to the regression residual, through both the state that the network reads and the signed target. Written as a function of the state, the BCF target is $\mu(y)(\mathbf{q}_i - \mathbf{h}_t)/(1-t)$ for $t < 1$, so its sensitivity to the state grows as $|\mu(y)|/(1-t)$ toward the item. One network with bounded sensitivity serves every user of the item, so users whose embeddings differ keep residuals where their states converge, and these residuals feed back into their embeddings.

5.1 Shared Fitting at the Item End

For two users u and v paired with item i under label y, the reference states at time t differ by $(1-t)(\mathbf{p}_u - \mathbf{p}_v)$ and converge to $\mathbf{q}_i$, while the signed targets keep the difference $\mu(y)(\mathbf{p}_v - \mathbf{p}_u)$. A field with state sensitivity $L_{\mathbf{h}}$, its Lipschitz constant in $\mathbf{h}$, changes its output between the two states by at most $L_{\mathbf{h}}(1-t)\|\mathbf{p}_u - \mathbf{p}_v\|_2$, so the remaining target difference appears in their residuals.

Proposition 2 (Item-End Compromise)**.** *Fix an item i, a label y, and $t \in [0,1)$. For a user u paired with i under label y, let $\mathbf{e}_u(t) = \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{h}_t^{(u,i)}, t, \mathbf{q}_i, y) - \mu(y)(\mathbf{q}_i - \mathbf{p}_u)$ be its flow residual. If $\mathbf{v}_{\boldsymbol{\theta}}(\cdot, t, \mathbf{q}_i, y)$ is $L_{\mathbf{h}}$-Lipschitz between the states of two such users u and v, their residuals satisfy:*

$$\|\mathbf{e}_u(t)\|_2 + \|\mathbf{e}_v(t)\|_2 \geq \big(|\mu(y)| - L_{\mathbf{h}}(1-t)\big)_+ \|\mathbf{p}_u - \mathbf{p}_v\|_2. \tag{11}$$

Here $(x)_+ = \max\{x, 0\}$. If the Lipschitz condition holds pairwise on a set $\mathcal{S}$ of such users for all $t \in [0,1)$, with a common constant $L_{\mathbf{h}} > 0$ satisfying $L_{\mathbf{h}} \geq |\mu(y)|$, then $t \sim \mathrm{Unif}[0,1]$ gives the following bound, where $\bar{\mathbf{p}}_{\mathcal{S}}$ is the mean of the users in $\mathcal{S}$, so the right side measures their dispersion:

$$\mathbb{E}_t \sum_{u \in \mathcal{S}} \|\mathbf{e}_u(t)\|_2^2 \geq \frac{|\mu(y)|^3}{6L_{\mathbf{h}}} \sum_{u \in \mathcal{S}} \|\mathbf{p}_u - \bar{\mathbf{p}}_{\mathcal{S}}\|_2^2. \tag{12}$$

Prop. 2 gives a necessary condition for small flow error at fixed state sensitivity, and its bound falls as these embeddings move together or as $L_{\mathbf{h}}$ grows. For $t < 1$ the complete input determines $\mathbf{p}_u$, so a field of unrestricted sensitivity could fit each user separately, and the coupling comes from the bounded state sensitivity of the shared network. The state sensitivity $|\mu(y)|/(1-t)$ of the target exceeds $L_{\mathbf{h}}$ exactly where the coefficient in Eq. (11) is positive. For $y = +1$, $\mathcal{S}$ holds observed users of item i, and for $y = -1$, users for whom i is a sampled negative. App. D proves both results.

5.2 Complexity

Space. The embedding tables $\mathbf{P}$ and $\mathbf{Q}$ contain $O((|\mathcal{U}| + |\mathcal{I}|)d)$ parameters. The input and output projections of the residual MLP in Eq. (6) contribute $O(dd_h)$ parameters, and its hidden projection contributes $O(d_h^2)$. Total parameter storage is $O((|\mathcal{U}| + |\mathcal{I}|)d + d_h(d + d_h))$. The shared field adds a cost independent of user and item counts. Training also stores optimizer states and activations for differentiating the field and RK4 steps. Deployment retains only the embedding tables.

Training time. One field evaluation costs $O(d_h(d + d_h))$. For a batch of B positive pairs with K_- negatives per positive, flow matching evaluates $B(1 + K_-)$ sampled states. Trajectory consistency integrates only the positive pairs, using four field evaluations per RK4 step and T steps per pair. These losses therefore require $B(1 + K_- + 4T)$ field evaluations and cost $O((1 + K_- + 4T)B\, d_h(d + d_h))$ beyond the BPR loss. Backpropagation through this fixed computation graph has the same asymptotic order. For example, $K_- = 4$ and $T = 4$ give 21 field evaluations per positive pair. For fixed model dimensions, K_-, and T, the cost of one pass is linear in the number of training positive pairs.

Inference time. PACF discards the velocity network and solver after training. Scoring uses the inner product of the static embeddings in Eq. (1), costing $O(d)$ per user-item pair and $O(|\mathcal{I}|d)$ per user. Deployment therefore uses the same scoring operations as MF at the same embedding dimension.

6 Experiments

6.1 Baselines

We compare PACF with eleven baselines from two families. *Discriminative Models*: **BPR-MF** (Rendle et al., 2009) learns pairwise rankings, **NCF** and **NeuMF** (He et al., 2017) use a multi-layer perceptron and a hybrid of linear and non-linear branches as the interaction function, **LightGCN** (He et al., 2020) propagates signals through neighborhood aggregation, **SGL** (Wu et al., 2021) and **SimGCL** (Yu et al., 2022) add contrastive objectives through graph augmentation and embedding noise, and **LightCCF** (Zhang et al., 2025b) learns neighborhood aggregation through a contrastive objective. *Generative Models*: **MultiVAE** (Liang et al., 2018) and **RecVAE** (Shenbin et al., 2020) model implicit feedback with variational autoencoders, **DiffRec** (Wang et al., 2023) recovers interaction vectors from noise by diffusion, and **FlowCF** (Liu et al., 2025) fits binary interaction vectors by flow matching with a behavior-guided prior, which is the closest flow-based baseline to PACF.

6.2 DATASETS

We evaluate PACF on seven public datasets from four domains, and Tab. 1 summarizes their statistics. For movies, we use **MovieLens-latest-small** (ML)[1] (Harper & Konstan, 2015). Food and Dance denote Bili_Food and Bili_Dance from **NineRec**[2] (Zhang et al., 2025a), used for short-video recommendation. The **Yelp2018** (Yelp)[3] benchmark covers business recommendation (He et al., 2020). From the 2014 **Amazon review data**[4] (McAuley et al., 2015; He & McAuley, 2016), we use the 5-core subsets of the Baby, "Sports and Outdoors", and "Clothing, Shoes and Jewelry" categories for e-commerce, denoted Baby, Sports, and Clothing. We treat every observed interaction as positive implicit feedback, and pair each positive item with $K_- = 4$ negative items sampled uniformly from the unobserved pool. All datasets keep users and items with at least five observed interactions.

Table 1: Statistics of the seven datasets used in our experiments. Sparsity is the fraction of user-item pairs without any observed interaction in the data.

Dataset	#Users	#Items	#Interactions	Sparsity
ML	610	3,650	90,274	95.95%
Food	5,990	1,125	36,482	99.46%
Dance	10,231	1,676	80,086	99.53%
Yelp	31,668	38,048	1,561,406	99.87%
Baby	19,445	7,050	161,276	99.88%
Sports	35,598	18,357	296,337	99.95%
Clothing	39,387	23,033	278,677	99.97%

6.3 EXPERIMENTAL SETUP

Evaluation Protocols and Metrics. For leave-one-out evaluation (He et al., 2017), we reserve each user's latest interaction for testing and the second latest for validation. Over all candidate items, Recall@10 measures whether the held-out item enters the top ten, and Normalized Discounted Cumulative Gain (NDCG@10) also rewards higher placement. Main-comparison scores are five-seed means. Validation NDCG@10 selects each ablation control in its own search, so every control is tuned for its design choice, and the PACF entries in these ablations repeat the main comparison.

Hyperparameter Configuration. For the main comparison, we use Bayesian optimization with a budget of 500 trials for each method and dataset. We tune the learning rate over $[10^{-5}, 10^{-1}]$ on a logarithmic scale, and weight decay over $[10^{-8}, 10^{-4}]$. The embedding dimension d and the hidden width d_h of the velocity network are both chosen from $\{64, 128, 256, 512\}$. For the flow dynamics, we search for the number of integration steps $T \in \{1, 4, 8, 16\}$. We tune the regularization weights $\lambda \in [0.1, 1.0]$ and $\beta \in [0, 0.5]$, and the negative target scale $\mu_- \in [0, 1]$. We tune each baseline over the ranges recommended in its original paper. App. C gives the ablation protocols, initialization, hardware, selected settings, training and inference costs, and five-seed standard deviations.

6.4 PERFORMANCE ANALYSIS

PACF achieves the best result on every dataset in Tab. 2. It keeps the scorer and the ranking loss of BPR-MF, so this advantage lies entirely in the embeddings that its pathwise losses train. LightCCF or SimGCL holds the second-best result in every column, and both share information across users through graph aggregation and a contrastive objective. PACF exceeds them with the plain MF scorer, so its cross-user constraints act only during training and the deployed model stays an inner-product scorer. Every generative baseline trails both methods on the three sparse Amazon datasets.

The gain over BPR-MF rises steadily from ML, the densest dataset, to Clothing, the sparsest. BPR constrains a user only through its own interactions, which sparsity removes, while the shared fit draws on every user of the same items. Clothing also gives PACF its widest relative margin over the strongest baseline. PACF adds constraints rather than scorer capacity, and on sparse data the constraints matter more. NeuMF adds a learned nonlinear interaction to MF and stays close to BPR-MF on the three Amazon datasets, where PACF, with the plain inner product, leads by a wide

[1] https://grouplens.org/datasets/movielens/latest/

[2] https://github.com/westlake-repl/NineRec

[3] https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset

[4] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html

Table 2: Comparison on seven datasets, measured by Recall@10 (Recall) and NDCG@10 (NDCG) in percent. **Bold** and underline mark the best and second-best results for each dataset and metric. PACF has the best result in every column, and its relative gain over BPR-MF grows as sparsity rises.

	ML		**Food**		**Dance**		**Yelp**		**Baby**		**Sports**		**Clothing**	
Method	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR-MF	5.59	2.66	6.91	3.36	6.62	3.29	2.77	1.37	1.42	0.67	1.48	0.74	0.64	0.30
NeuMF	6.18	3.11	7.38	3.55	6.66	3.31	3.11	1.54	1.43	0.68	1.49	0.74	0.65	0.31
NCF	6.39	3.24	7.57	3.63	6.72	3.37	3.19	1.60	1.68	0.82	1.48	0.75	0.68	0.32
LightGCN	6.52	3.16	7.44	3.60	6.19	3.09	3.26	1.62	1.37	0.66	1.34	0.67	0.58	0.27
SGL	6.72	3.35	9.08	4.48	9.23	4.83	4.23	2.10	2.16	1.11	2.29	1.19	1.30	0.69
SimGCL	6.13	2.98	9.46	4.71	10.17	5.17	4.40	2.19	2.58	1.32	3.45	1.82	2.56	1.35
LightCCF	7.38	3.56	9.56	4.68	10.21	5.18	4.46	2.21	2.50	1.40	3.48	1.91	2.67	1.47
MultiVAE	6.27	3.09	8.36	4.09	8.37	4.24	3.54	1.77	1.67	0.80	1.36	0.69	1.49	0.77
RecVAE	6.82	3.14	8.02	3.93	8.74	4.56	3.62	1.81	1.61	0.80	1.60	0.81	1.47	0.79
DiffRec	6.82	3.54	7.34	3.53	8.65	4.26	3.08	1.53	1.57	0.72	1.58	0.85	0.65	0.30
FlowCF	7.31	3.45	8.36	4.31	9.05	4.79	3.56	1.72	1.79	0.91	1.98	1.08	1.22	0.66
PACF	**7.48**	**3.73**	**9.75**	**4.79**	**10.32**	**5.25**	**4.49**	**2.23**	**2.63**	**1.42**	**3.69**	**1.97**	**2.88**	**1.59**

margin. PACF also exceeds FlowCF on every dataset, and unlike FlowCF its flow regularizes the same user and item embeddings that the inner product later scores for every candidate item.

6.5 Ablation Study

6.5.1 Comparison with Endpoint Regularization

Penalizing the length of each user-to-item displacement is the most direct way to use it, and Tab. 3 tests this alternative. **EP**$^+$ adds the squared distance of each positive pair to **BPR**, and **EP**$^\pm$ also adds a margin against sampled negatives (Hsieh et al., 2017). Both penalties improve on BPR, so pulling users toward their items already adds useful constraints to MF.

Table 3: Comparison of PACF with BPR and two endpoint penalties on ML and Food, measured by Recall@10 (Recall) and NDCG@10 (NDCG) in percent. PACF regresses one shared field along each user-item segment, while both penalties act only on endpoint distances.

Dataset	Metric	BPR	EP$^+$	EP$^\pm$	PACF
ML	Recall	6.06	6.89	6.72	7.48
	NDCG	3.05	3.14	3.28	3.73
Food	Recall	7.45	8.94	9.19	9.75
	NDCG	3.60	4.45	4.59	4.79

On positive pairs, EP$^+$ is the special case of PACF in which the network predicts nothing. The flow error of a positive pair is then exactly its squared distance, and the integrated endpoint stays at the user embedding, so both pathwise losses collapse to the EP$^+$ penalty. The field changes where each user is pulled. EP$^+$ pulls every user straight toward each of its items, one pair at a time, while PACF pulls through the residual of one shared fit. Near an item, the states of all its users nearly coincide, so the shared network predicts roughly their average displacement, and the residual of each user becomes its offset from the average of its co-users. This residual pulls the user toward that average and shrinks the spread of users around the item that Prop. 2 bounds from below.

Users who share an item thus move toward one another, so an item adopted by one of them scores higher for the others. EP$^\pm$ adds a push away from sampled negatives within each pair, while the negative pairs of PACF join the same shared fit through their reversed and scaled targets.

6.5.2 Sampling Positions Along the User-Item Path

Fig. 3 fixes the time of the reference state in Eq. (5), where $t{=}0$ places it at the user end, $t{=}0.5$ at the midpoint, and **Grid** at an anchor drawn from three evenly spaced interior points, while PACF samples the whole segment. Moving the support from the user end to the midpoint, nearer the item, helps.

The BCF target is constant along a segment, so the support only chooses the state the network reads. At the user end every user presents its own embedding, and the network fits each displacement separately. Toward the item, the states of co-users converge while their targets stay apart. At the midpoint these states lie half as far apart as the embeddings. Next to the item, the target changes with

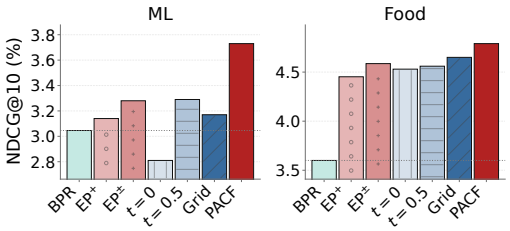


Figure 3: Sampling supports of t in $\mathcal{L}_{\text{flow}}$ on ML and Food, shown with BPR and both endpoint penalties and measured by NDCG@10 in percent. The dotted line marks the BPR level. PACF samples t uniformly over the segment.

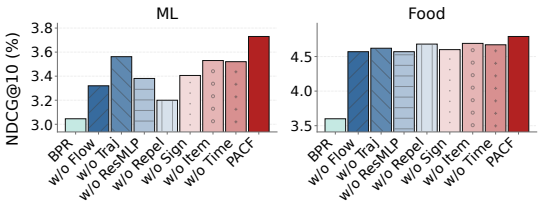


Figure 4: Removal of one PACF component at a time on ML and Food, measured by NDCG@10 in percent against BPR. PACF feeds the complete input $(\mathbf{h}_t, \mathbf{q}_i, t, y_{ui})$ to one residual field and trains it with both pathwise losses on all training pairs.

the state faster than the network can follow, and Prop. 2 bounds their residuals from below by the spread of these users. Uniform sampling keeps a fixed share of reference states in this region.

6.5.3 Choice of Flow-Matching Target

Fig. 5 compares four targets for $\mathcal{L}_{\text{flow}}$. **Const** uses the signed displacement in Eq. (4). **Pull** points from the sampled state toward the item under both labels. **Grad** uses the signed score gradient with respect to the user embedding, and **NormGrad** normalizes it to unit length.

Const gives the highest control NDCG on both datasets. Const retains each user's full displacement as its reference state approaches the item. Users paired with the same item and label present nearby states with distinct targets, creating the residuals that Prop. 2 bounds. Grad and NormGrad assign these users one common target, while Pull makes both labels attractive. The target therefore determines which user differences the shared field must reconcile and how the two labels shape the embeddings.

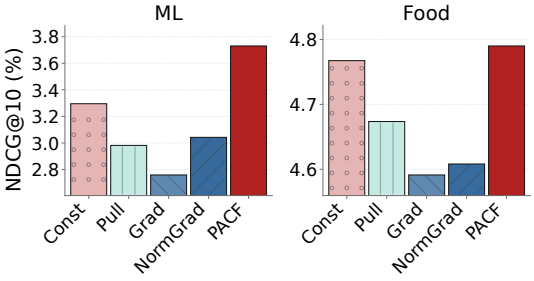


Figure 5: Comparison of flow-matching targets on ML and Food, measured by NDCG@10 in percent. The four target variants share one tuning protocol. PACF reports the separately tuned main result.

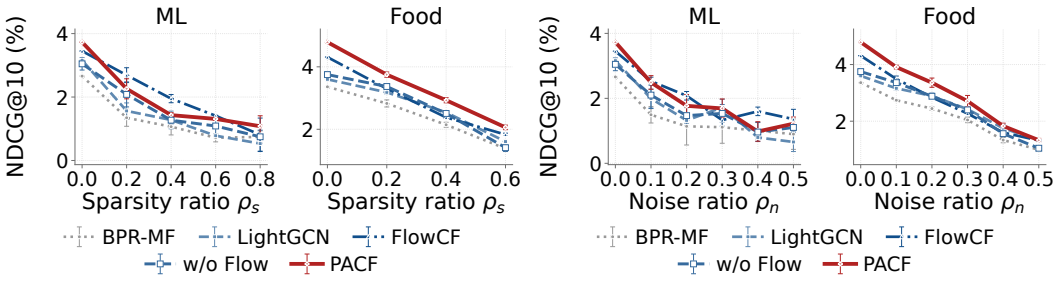
6.5.4 Ablation Study of Components

Fig. 4 removes one component of PACF at a time. The network needs the state, the item, and the time together, because the time converts the remaining distance to the item back into the full displacement. **w/o Item** removes the item input, so the network must infer the item from a state that mixes user and item, and **w/o Time** removes the time, so one remaining displacement fits a nearby user early and a distant user late. Either way the residuals encode missing inputs instead of user differences. **w/o ResMLP**, an equally sized plain MLP, gives the largest drop among the field designs. The BCF target is linear in the embeddings up to the sign $\mu(y)$, so the residual skip carries the linear displacement, and the nonlinear branch absorbs the sign and the correction where co-user segments cross.

The label tells the network which way the target points. **w/o Sign** zeroes the label, so a positive and a negative pair near the same item present similar inputs with opposite targets. **w/o Repel** drops negative-pair regression, so sampled negatives act only through BPR. **w/o Flow** drops the fit of velocities at single states, and **w/o Traj** drops the check of where the composed velocities lead, along which local errors accumulate or cancel. Tab. 8 retunes both removals over five seeds, and each loss alone recovers most of the gain of PACF over BPR-MF while both stay below the full model.

6.5.5 Inference Readout

We compare three readouts of PACF. The static readout a is the inner product of the embeddings and repeats the PACF result of Tab. 2. The endpoint readouts integrate the network from the user toward



(a) Sparsity Sweep.

(b) Replacement Noise Sweep.

Figure 6: Robustness of PACF and baselines to interaction removal in panel a and item replacement in panel b on ML and Food, measured by NDCG@10 in percent. Error bars show seed standard deviations. PACF stays above **w/o Flow** at every level and leads all compared models on Food.

each candidate under a positive label on separately trained checkpoints, one per seed, and a_Φ and a_{dist} take the inner product and the negative distance between this endpoint and the candidate.

The static readout ranks best in every cell of Tab. 4. Trajectory consistency trains each trajectory to reach its conditioning item, so the better PACF fits it, the closer every endpoint lies to the candidate. a_Φ then approaches the squared norm of the candidate, a_{dist} approaches one value for all candidates, and only the residual user information ranks candidates. a_Φ, which adds the item norm, thus ranks below a_{dist}. Eq. (10) returns this residual to $\mathbf{p}_u$, so the trajectory only trains the static readout.

Table 4: Static and endpoint readouts of PACF on ML and Food, measured by Recall@10 and NDCG@10 in percent as five-seed mean $\pm$ std. The static inner product ranks best in every cell, so PACF ranks with the static MF scorer that its two pathwise losses train.

	ML		Food	
Readout	Recall	NDCG	Recall	NDCG
a	7.48 ± 0.41	3.73 ± 0.03	9.75 ± 0.08	4.79 ± 0.05
a_Φ	0.59 ± 0.19	0.23 ± 0.08	2.08 ± 0.06	0.95 ± 0.01
a_{dist}	1.05 ± 0.25	0.50 ± 0.15	4.68 ± 0.58	2.42 ± 0.26

6.6 Robustness Analysis

Fig. 6 perturbs only the training interactions, by removing a fraction ρ_s of them or replacing the item of a fraction ρ_n with a random unobserved item, and all models keep their clean-data configurations.

Removal thins the feedback of most users, and PACF stays above **w/o Flow** at every removal level. PACF leads all compared models on Food and regains the lead on ML at the strongest removal ratio ρ_s. Every retained pair still enters the shared regression that couples the users of its item.

Replacement corrupts the segment, BCF target, and trajectory endpoint of each affected pair. PACF stays above **w/o Flow** at every noise level on both datasets and leads throughout Food. On ML, the densest dataset, PACF gains the least over BPR in the main comparison and loses its lead at most noise levels. With about 150 interactions per user against about six on Food, dense feedback already constrains each ML user through its own observed pairs, leaving less for other users to add.

7 Conclusion

For sparse implicit feedback, PACF augments BPR-MF with Bilateral Conditional Flow regularization, in which one shared velocity network regresses signed displacement targets along user-item segments and a trajectory consistency term ties the integrated endpoint of each observed pair to its item. Through gradient amortization, both losses shape the static embeddings. Our analysis shows how residuals drive these updates and gives a necessary condition at fixed state sensitivity that links the flow error of users sharing an item and label to their dispersion. PACF achieves the highest performance on all seven datasets while keeping the MF scorer and gaining most on sparse data.

AI USE STATEMENT

Generative AI tools assisted with literature search, manuscript writing, review, and LaTeX formatting. The authors reviewed and verified all content and take responsibility for the manuscript.

REPRODUCIBILITY STATEMENT

Sec. 6.3 and Appendix C.1 provide the experimental setup and implementation details. Appendix B.1 gives the training algorithm, and Appendix D presents the theoretical assumptions and proofs.

REFERENCES

Dheeraj Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. Matrix factorization model in collaborative filtering algorithms: A survey. *Procedia Computer Science*, 49:136–146, 2015.

John C Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, 3 edition, 2016.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *NeurIPS*, 31, 2018.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ODE: The world of Jacobian and kinetic regularization. In *ICML*, pp. 3154–3164. PMLR, 2020.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *NeurIPS*, 37:133345–133385, 2024.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *NeurIPS*, 38:75460–75482, 2025.

David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

F Maxwell Harper and Joseph A Konstan. The MovieLens datasets: History and context. *TIIS*, 5(4): 1–19, 2015.

Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, pp. 507–517, 2016. doi: 10.1145/2872427.2883037.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, pp. 173–182, 2017.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. LightGCN: Simplifying and powering graph convolution network for recommendation. In *ACM SIGIR*, pp. 639–648, 2020.

Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. Collaborative metric learning. In *WWW*, pp. 193–201, 2017.

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*, pp. 25474–25485, 2025.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Yehuda Koren, Steffen Rendle, and Robert Bell. Advances in collaborative filtering. In Francesco Ricci, Lior Rokach, and Bracha Shapira (eds.), *Recommender Systems Handbook*, pp. 91–142. Springer, 3rd edition, 2022.

Li Li, Mingyue Cheng, Yuyang Ye, Zhiding Liu, and Enhong Chen. FlowRec: Prior-informed flow matching for efficient sequential recommendation generation. In *DASFAA*, pp. 206–222, 2026. doi: 10.1007/978-981-92-0363-5_13.

Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *WWW*, pp. 689–698, 2018.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

Chengkai Liu, Yangtian Zhang, Jianling Wang, Rex Ying, and James Caverlee. Flow matching for collaborative filtering. In *ACM SIGKDD*, pp. 1765–1775, 2025. doi: 10.1145/3711896.3736967.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. UltraGCN: Ultra simplification of graph convolutional networks for recommendation. In *CIKM*, pp. 1253–1262, 2021.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52, 2015. doi: 10.1145/2766462.2767755.

Harris Papadakis, Antonis Papagrigoriou, Costas Panagiotakis, Eleftherios Kosmas, and Paraskevi Fragopoulou. Collaborative filtering recommender systems taxonomy. *Knowledge and Information Systems*, 64(1):35–74, 2022.

Dogyun Park, Sojin Lee, Sihyeon Kim, Taehoon Lee, Youngjoon Hong, and Hyunwoo J Kim. Constant acceleration flow. *NeurIPS*, 37:90030–90060, 2024.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019.

Gabriel Peyré and Marco Cuturi. Computational optimal transport with applications to data sciences. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *International Conference on Machine Learning*, pp. 28100–28127. PMLR, 2023.

Nikhil Rao, Hsiang-Fu Yu, Pradeep Ravikumar, and Inderjit S Dhillon. Collaborative filtering with graph information: Consistency and scalable methods. In *NeurIPS*, 2015.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*, pp. 452–461, 2009.

Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. RecVAE: A new variational autoencoder for top-N recommendations with implicit feedback. In *WSDM*, pp. 528–536, 2020.

Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM CSUR*, 47(1):1–45, 2014.

Harald Steck, Chaitanya Ekanadham, and Nathan Kallus. Is cosine-similarity of embeddings really about similarity? In *WWW Companion*, pp. 887–890, 2024. doi: 10.1145/3589335.3651526.

Cédric Villani. *Optimal transport: Old and new*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2009.

Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. Towards representation alignment and uniformity in collaborative filtering. In *KDD*, pp. 1816–1825, 2022.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, pp. 9929–9939. PMLR, 2020.

Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. Diffusion recommender model. In *ACM SIGIR*, pp. 832–841, 2023.

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *ACM SIGIR*, pp. 726–735, 2021.

Yihong Wu, Le Zhang, Fengran Mo, Tianyu Zhu, Weizhi Ma, and Jian-Yun Nie. Unifying graph convolution and contrastive learning in collaborative filtering. In *KDD*, pp. 3425–3436, 2024.

Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1294–1303, 2022. doi: 10.1145/3477495.3531937.

Dan Zhang, Yangliao Geng, Wenwen Gong, Zhongang Qi, Zhiyu Chen, Xing Tang, Ying Shan, Yuxiao Dong, and Jie Tang. RecDCL: Dual contrastive learning for recommendation. In *WWW*, pp. 3655–3666, 2024.

Jiaqi Zhang, Yu Cheng, Yongxin Ni, Yunzhu Pan, Zheng Yuan, Junchen Fu, Youhua Li, Jie Wang, and Fajie Yuan. NineRec: A benchmark dataset suite for evaluating transferable recommendation. *TPAMI*, 47(7):5256–5267, 2025a.

Yu Zhang, Yiwen Zhang, Yi Zhang, Lei Sang, and Yun Yang. Unveiling contrastive learning's capability of neighborhood aggregation for collaborative filtering. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1985–1994, 2025b. doi: 10.1145/3726302.3730111.

A Limitations and Future Work

Broader Impact. PACF serves recommendations with the MF scorer and discards the velocity network after training, and Tab. 13 reports its measured latency and memory. PACF learns from observed interactions, so a deployment should evaluate the exposure and subgroup-fairness effects that collaborative filtering inherits from these data and mitigate them where they arise.

Design Scope. PACF uses straight reference paths in a Euclidean embedding space. By App. D.1, this choice uniquely minimizes Euclidean kinetic energy. It gives the constant signed target $\mu(y_{ui})(\mathbf{q}_i - \mathbf{p}_u)$, and lets the complete flow input identify $\mathbf{p}_u = (\mathbf{h}_t - t\mathbf{q}_i)/(1-t)$ for $t < 1$. The gradient and item-end analyses treat free user and item embeddings, matching the deployed MF scorer.

Future Work. Non-Euclidean path families and state-dependent targets widen this design space, and each changes the target function in Eq. (29) on which the analysis rests. A graph encoder ties each embedding to its neighbors, so extending the analysis to graph-based recommenders requires joint gradients through the propagation layers, the flow map, and the shared field. Lower-order solvers can reduce training cost, and comparing their trajectory error and ranking accuracy with RK4 would quantify the role of the final stage at $t = 1$ in Remark 5. Prop. 2 gives a necessary condition that shared fitting at the item end places on the embeddings, and characterizing the full embedding regularizer induced by this coupling is a natural next step toward a sufficient condition.

B Method Details

B.1 Training Algorithm

Alg. 1 draws a mini-batch $\mathcal{B} = \{(u, i^+, \{i_k^-\}_{k=1}^{K_-})\}$ and computes the three losses in Eq. (9). Adam jointly updates the velocity network and static embeddings. Inference uses only the embeddings.

For each positive pair, K_- sampled negatives contribute separate terms to $\mathcal{L}_{\text{flow}}$. The trajectory loss integrates only the positive condition because its endpoint is an observed training item.

Algorithm 1 Algorithm of PACF.

Input: Dataset $\mathcal{O}$, batch B, weights λ, β, negative scale μ_-, RK4 steps T, negatives K_-.
Initialize: Embeddings $\mathbf{P}, \mathbf{Q}$ and velocity network $\mathbf{v}_{\boldsymbol{\theta}}$.
Output: Trained embeddings $\mathbf{P}, \mathbf{Q}$.
while not converged **do**
  Sample B positive pairs (u, i^+) from $\mathcal{O}$.
  Draw K_- uniform negatives $i_k^- \in \mathcal{I} \setminus \mathcal{I}_u$ for each pair.
  // 1. Pairwise ranking loss
  Compute $\mathcal{L}_{\text{rec}}$ via Eq. (2).
  // 2. Bilateral flow matching
  Sample times $t \sim \text{Unif}[0, 1]$ of size B.
  for $(u, i, y) \in \{(u, i^+, +1)\} \cup \{(u, i_k^-, -1)\}_{k=1}^{K_-}$ **do**
    Form $\mathbf{h}_t$ via Eq. (3) and $\mathbf{u}_t$ via Eq. (4).
    Accumulate $\mathcal{L}_{\text{flow}}$ via Eq. (5).
  end for
  // 3. Trajectory consistency
  Integrate with RK4: $\Phi_1(\mathbf{p}_u) \leftarrow \text{RK4}(\mathbf{p}_u, \mathbf{q}_{i^+}, \mathbf{v}_{\boldsymbol{\theta}}, +1, T)$.
  Compute $\mathcal{L}_{\text{traj}}$ via Eq. (8) for the positive item i^+.
  // Joint update
  Update $\{\mathbf{p}_u\}, \{\mathbf{q}_i\}, \boldsymbol{\theta}$ by minimizing Eq. (9).
end while

B.2 Trajectory Integration

PACF uses RK4 to compute the endpoint in Eq. (8). For a fixed positive condition, write the field as:

$$\mathbf{f}_t(\mathbf{h}) = \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{h}, t, \mathbf{q}_{i^+}, +1). \tag{13}$$

One RK4 step over $t \in [0, 1]$ combines four evaluations of the shared velocity network $\mathbf{f}_t$:

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}_0(\mathbf{p}_u), \\ \mathbf{k}_2 &= \mathbf{f}_{1/2}(\mathbf{p}_u + \tfrac{1}{2}\mathbf{k}_1), \\ \mathbf{k}_3 &= \mathbf{f}_{1/2}(\mathbf{p}_u + \tfrac{1}{2}\mathbf{k}_2), \\ \mathbf{k}_4 &= \mathbf{f}_1(\mathbf{p}_u + \mathbf{k}_3), \\ \Phi_1(\mathbf{p}_u) &= \mathbf{p}_u + \tfrac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4). \end{aligned} \tag{14}$$

The stages $\mathbf{k}_2$ to $\mathbf{k}_4$ evaluate the field at states generated by earlier predictions, which can leave the reference segment. Flow matching penalizes each reference-state residual separately, while trajectory consistency penalizes only the endpoint, where stage errors add up or cancel. The solver sets how far the two terms differ. A forward Euler step gives $\Phi_1(\mathbf{p}_u) = \mathbf{p}_u + \mathbf{f}_0(\mathbf{p}_u)$, so the positive endpoint loss equals flow matching at $t = 0$. RK4 composes four predictions per step, and its last stage evaluates the shared field at the item end $t = 1$, where Remark 5 shows it couples the users of each item.

C Additional Experiments

This appendix first fixes the protocols behind every supplementary result. The analyses then follow the pathwise signal of PACF from the learned field to the embedding updates, the role of each component, and the static scorer, and close with robustness, cost, and the selected settings.

C.1 Experimental Protocols

Implementation. All methods are implemented in PyTorch 2.1 (Paszke et al., 2019) with CUDA 12.1 on a workstation with an Intel Xeon Gold 6244 CPU, 395 GB RAM, and three NVIDIA Quadro RTX 6000 GPUs with 24 GB of memory each. Each experiment runs on one GPU, and Bayesian optimization runs its trials in parallel across the three GPUs. PACF uses MF ID embeddings as its backbone. Xavier-uniform initialization sets the user table, the item table, and every layer of $\mathbf{v}_{\boldsymbol{\theta}}$, every bias starts at zero, and both tables are optimized directly as free model parameters.

Main comparison. Each method receives 500 Bayesian optimization trials with the same early stopping. PACF searches the space of Sec. 6.3, and each baseline the ranges of its original paper. PACF and the pairwise discriminative baselines pair each training positive with $K_- = 4$ uniform negatives from $\mathcal{I} \setminus \mathcal{I}_u$, and the other baselines keep their original objectives and sampling rules.

Endpoint controls. The endpoint controls in Tab. 3 keep the MF embeddings and the BPR loss and replace the shared regression of PACF with penalties on distances between users and items. For a training triplet (u, i^+, i^-), they use the mean squared coordinate distances $D_\pm = \|\mathbf{p}_u - \mathbf{q}_{i^\pm}\|_2^2/d$ and add endpoint penalties to the BPR ranking loss while keeping the inner-product scorer:

$$\begin{aligned} \mathcal{L}_{\mathrm{EP}^+} &= \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{E}}\, \mathbb{E}[D_+], \\ \mathcal{L}_{\mathrm{EP}^\pm} &= \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{E}}\, \mathbb{E}[D_+ + \gamma_{\mathrm{E}} \max\{0, m_{\mathrm{E}} + D_+ - D_-\}]\,. \end{aligned} \tag{15}$$

Here λ_{E} weights endpoint regularization, γ_{E} weights the negative margin term, and m_{E} sets its distance margin, each searched over $[0, 1]$. EP$^+$ tunes only the weight λ_{E}. EP$^\pm$ also penalizes every negative item whose distance to the user exceeds that of the positive item by less than m_{E}.

Flow-based controls. The sampling and component controls regress onto $\mathbf{q}_{i^+} - \mathbf{p}_u$ for positives and $-(\mathbf{q}_{i^-} - \mathbf{p}_u)$ for negatives. They average the positive and negative errors separately over batch entries and embedding coordinates and add the negative error with a weight $\omega_- \in [0, 1]$, a loss weight of these controls distinct from the target scale μ_- of PACF. The sampling controls keep reference-state regression and the RK4 trajectory term and draw one time per training triplet, shared by its positive and negative flow terms. The point controls fix this time at 0 or 0.5. Grid draws it uniformly from $\{0.25, 0.5, 0.75\}$, which matches the number of reference-state evaluations of uniform sampling. These draws set only the flow-matching states, and the trajectory term keeps RK4 integration.

w/o Sign sets the label coordinate of the network input in Eq. (6) to zero in flow matching and trajectory integration and keeps the signed targets of Eq. (4), so the field receives the label only through its targets. **w/o Repel** trains with $\mathcal{L}_{\mathrm{rec}}$, positive flow matching, and positive trajectory consistency, so sampled negatives act only through BPR. The first control removes the label input and keeps signed negative regression, and the second removes signed negative regression.

Search budgets. On ML and Food, each endpoint control and each fixed-support sampling control receives 150 trials, the BPR reference of Tab. 3 and Figs. 3 and 4 receives a separate 200-trial search, and each component removal receives 500 trials. The ablation search space tunes $d \in \{32, 64, 128, 256, 512\}$, a log-uniform learning rate over $[10^{-5}, 10^{-1}]$, and log-uniform weight decay over $[10^{-8}, 10^{-3}]$, together with the coefficients of the endpoint controls. The flow-based controls also search the outer flow and trajectory weights over $[0.1, 1]$ and $[0, 0.5]$, and the sampling controls also tune $d_h \in \{64, 128, 256, 512\}$ and $T \in \{1, 4, 8, 16\}$. Each control search uses seed 2024, a training batch size of 2,048, an evaluation batch size of 4,096, and up to 500 epochs per trial, and it reports the test score of the configuration that validation NDCG@10 selects, independently of the full-model settings in Tab. 14. The PACF entries in these ablations repeat Tab. 2, and the BPR reference differs from the BPR-MF row of Tab. 2, which averages five seeds of the main search.

Diagnostic sweeps. Tabs. 9 and 10 vary T and d_h in fixed-budget diagnostic searches outside the main search. Their entries compare settings within each table, and Tab. 2 reports the main result.

Perturbations. Both perturbations change only the training interactions $\mathcal{O}_{\mathrm{train}} \subseteq \mathcal{O}$, so every setting evaluates the learned scorer on the same validation and test interactions and candidate sets, and every model keeps its clean-data configuration. Removal uses $\rho_s \in \{0, 0.2, 0.4, 0.6, 0.8\}$ on ML and $\rho_s \in \{0, 0.2, 0.4, 0.6\}$ on Food. It first keeps one stratified interaction per user $u \in \mathcal{U}$, so every user retains a training history, then draws $N_{\mathrm{rem}} = \lfloor |\mathcal{O}_{\mathrm{train}}|(1 - \rho_s) \rfloor - |\mathcal{U}|$ interactions uniformly without replacement. Replacement uses $\rho_n \in \{0, 0.1, \ldots, 0.5\}$ on both datasets. It samples $\mathcal{R} \subset \mathcal{O}_{\mathrm{train}}$ with $|\mathcal{R}| = \lfloor |\mathcal{O}_{\mathrm{train}}|\, \rho_n \rfloor$ and replaces the item of each $(u, i) \in \mathcal{R}$ with an item drawn uniformly from $\mathcal{I} \setminus \mathcal{I}_u$. The interaction count and density stay fixed while the observed associations change, and sampling among valid user-item edges keeps every perturbed set free of duplicate pairs.

C.2 Full Performance Statistics

Tabs. 5 and 6 report the five-seed runs of Tab. 2 with their standard deviations. PACF has the highest mean in every column, and its seed standard deviation stays below its margin over BPR-MF in every column. PACF and BPR-MF share the inner-product scorer of Eq. (1), so this margin comes entirely from the trained embeddings. The following sections trace how the pathwise losses produce it.

Table 5: Comparison of all methods on ML, Food, Dance, and Yelp, measured by Recall@10 and NDCG@10 in percent as five-seed mean ± std. Rules separate discriminative baselines, generative baselines, and PACF, and **bold** and underline mark the best and second-best mean per column. PACF has the highest mean in every column across the movie, short-video, and business domains.

Method	ML		Food		Dance		Yelp	
	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR-MF	5.59±1.34	2.66±0.65	6.91±0.22	3.36±0.08	6.62±0.24	3.29±0.08	2.77±0.26	1.37±0.13
NeuMF	6.18±0.52	3.11±0.15	7.38±0.32	3.55±0.13	6.66±0.24	3.31±0.07	3.11±0.13	1.54±0.04
NCF	6.39±0.44	3.24±0.11	7.57±0.30	3.63±0.11	6.72±0.19	3.37±0.11	3.19±0.11	1.60±0.05
LightGCN	6.52±0.24	3.16±0.11	7.44±0.34	3.60±0.15	6.19±0.30	3.09±0.13	3.26±0.06	1.62±0.04
SGL	6.72±0.18	3.35±0.06	9.08±0.03	4.48±0.02	9.23±0.14	4.83±0.04	4.23±0.07	2.10±0.02
SimGCL	6.13±1.05	2.98±0.56	9.46±0.35	4.71±0.16	10.17±0.15	5.17±0.06	4.40±0.09	2.19±0.02
LightCCF	7.38±0.62	3.56±0.28	9.56±0.13	4.68±0.06	10.21±0.07	5.18±0.04	4.46±0.05	2.21±0.01
MultiVAE	6.27±0.42	3.09±0.13	8.36±0.02	4.09±0.00	8.37±0.24	4.24±0.12	3.54±0.08	1.77±0.04
RecVAE	6.82±0.25	3.14±0.15	8.02±0.13	3.93±0.08	8.74±0.36	4.56±0.15	3.62±0.22	1.81±0.08
DiffRec	6.82±0.56	3.54±0.15	7.34±0.33	3.53±0.13	8.65±2.10	4.26±1.02	3.08±0.08	1.53±0.03
FlowCF	7.31±0.22	3.45±0.08	8.36±0.11	4.31±0.10	9.05±0.15	4.79±0.06	3.56±1.57	1.72±0.77
PACF	**7.48±0.41**	**3.73±0.03**	**9.75±0.08**	**4.79±0.05**	**10.32±0.12**	**5.25±0.05**	**4.49±0.08**	**2.23±0.05**

Table 6: Comparison of all methods on Amazon Baby, Sports, and Clothing, measured by Recall@10 and NDCG@10 in percent as five-seed mean ± std. Rules separate discriminative baselines, generative baselines, and PACF, and **bold** and underline mark the best and second-best mean per column. PACF has the highest mean in every column on the three sparsest datasets, all from e-commerce.

Method	Baby		Sports		Clothing	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR-MF	1.42±0.04	0.67±0.01	1.48±0.05	0.74±0.03	0.64±0.15	0.30±0.08
NeuMF	1.43±0.04	0.68±0.01	1.49±0.05	0.74±0.03	0.65±0.17	0.31±0.09
NCF	1.68±0.03	0.82±0.01	1.48±0.06	0.75±0.03	0.68±0.18	0.32±0.09
LightGCN	1.37±0.04	0.66±0.01	1.34±0.13	0.67±0.06	0.58±0.09	0.27±0.04
SGL	2.16±0.09	1.11±0.04	2.29±0.24	1.19±0.11	1.30±0.26	0.69±0.15
SimGCL	2.58±0.10	1.32±0.04	3.45±0.08	1.82±0.04	2.56±0.18	1.35±0.11
LightCCF	2.50±0.03	1.40±0.03	3.48±0.09	1.91±0.04	2.67±0.01	1.47±0.02
MultiVAE	1.67±0.04	0.80±0.01	1.36±0.12	0.69±0.06	1.49±0.05	0.77±0.04
RecVAE	1.61±0.16	0.80±0.08	1.60±0.56	0.81±0.29	1.47±0.11	0.79±0.06
DiffRec	1.57±0.02	0.72±0.00	1.58±0.05	0.85±0.00	0.65±0.03	0.30±0.01
FlowCF	1.79±0.02	0.91±0.02	1.98±0.01	1.08±0.00	1.22±0.02	0.66±0.00
PACF	**2.63±0.04**	**1.42±0.03**	**3.69±0.07**	**1.97±0.03**	**2.88±0.04**	**1.59±0.02**

C.3 How the Shared Field Updates the Embeddings

Gradient amortization needs two effects. The field must learn the BCF targets, and its residuals must move the embeddings that the scorer keeps. Fig. 7 examines the first effect and Tab. 7 the second.

Fig. 7 trains the field on two-dimensional synthetic data. With flow matching, positive predictions follow the displacement $\mathbf{q}_i - \mathbf{p}_u$ and negative predictions reverse it, as Eq. (4) requires. Without $\mathcal{L}_{\text{flow}}$, the trajectory term supervises only positive-label inputs, and the field points toward the item under both labels. Flow matching is the only loss shaping the field at negative-label inputs.

Tab. 7 measures how strongly this fit reaches the embedding tables at PACF checkpoints. It compares gradient norms on $(\mathbf{P}, \mathbf{Q})$ with the one-step displacement $\Delta_{\mathbf{P},\mathbf{Q}}(\mathcal{L})$, the ℓ_2 norm of the change in

Table 7: Gradient and embedding-drift ratios of $\mathcal{L}_{\text{flow}}$ to $\mathcal{L}_{\text{rec}}$ at PACF checkpoints on ML and Food. Flow matching moves PACF's embeddings by amounts comparable to the ranking loss, and its input route alone carries a nonzero gradient, so it directly reshapes the scorer used at inference.

Dataset	$\frac{\|\nabla_{\mathbf{P},\mathbf{Q}}\mathcal{L}_{\text{flow}}\|_2}{\|\nabla_{\mathbf{P},\mathbf{Q}}\mathcal{L}_{\text{rec}}\|_2}$	$\frac{\Delta_{\mathbf{P},\mathbf{Q}}(\mathcal{L}_{\text{flow}})}{\Delta_{\mathbf{P},\mathbf{Q}}(\mathcal{L}_{\text{rec}})}$	$\frac{\|\nabla_{\mathbf{P},\mathbf{Q}}\mathcal{L}_{\text{flow}}^{\text{detach}}\|_2}{\|\nabla_{\mathbf{P},\mathbf{Q}}\mathcal{L}_{\text{rec}}\|_2}$
ML	0.1375	0.9915	0.1095
Food	0.2399	1.4893	0.0716

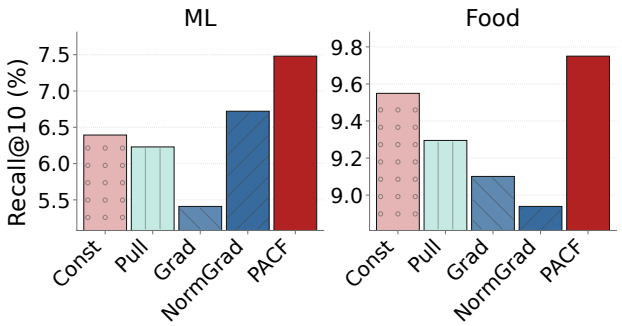


Figure 8: Recall@10 in percent for the target comparison of Fig. 5. Among the target controls, NormGrad leads on ML and Const on Food. PACF reports the separately tuned main result.

$(\mathbf{P}, \mathbf{Q})$ after one Adam step on a loss $\mathcal{L}$. The control $\mathcal{L}_{\text{flow}}^{\text{detach}}$ stops the gradient through the target $\mathbf{u}_t$ and keeps the error, so its gradient flows only through the state $\mathbf{h}_t$ and the item condition. The first column compares raw gradient magnitudes, the second measures the movement that Adam applies to the embeddings, and the third keeps only the part of the flow gradient that bypasses the target.

The flow gradient has a smaller norm than the ranking gradient, yet one Adam step on $\mathcal{L}_{\text{flow}}$ moves the embeddings about as far as a step on $\mathcal{L}_{\text{rec}}$ on ML and farther on Food. Adam divides each coordinate by its second-moment estimate, so the displacement follows how consistently a loss pushes each coordinate. The detached gradient is nonzero on both datasets, so by Eq. (30) flow residuals reach $\mathbf{p}_u$ through the input route $(1-t)\mathbf{J}_{\mathbf{h}}$ even with the target held fixed. Restoring the target route raises the gradient norm on both datasets. This route carries the item-end pull of each user toward the mean of its co-users. It acts on the same residual as the input route, so their relative directions set the size of the gradient. Flow matching therefore changes the embeddings at a rate comparable to the ranking loss, and the inner-product scorer keeps this change after the shared velocity field is discarded.

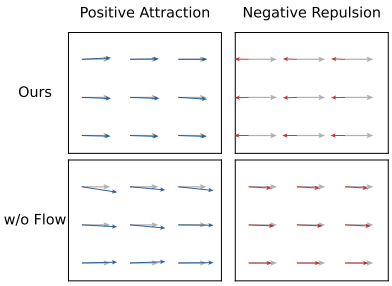


Figure 7: Field predictions on synthetic data, with one column per label and one row each for PACF with and without $\mathcal{L}_{\text{flow}}$. Gray arrows show $\mathbf{q}_i - \mathbf{p}_u$, and colored arrows show $\mathbf{v}_{\boldsymbol{\theta}}$. Flow matching makes PACF's field point toward the item for positive labels and away from it for negative labels.

C.4 Contributions of the Pathwise Components

The main text measures each choice by NDCG@10. This section reports these comparisons by Recall@10, adds loss controls and integration depth, and explains each effect through the regression.

Target form. Fig. 8 adds Recall to the NDCG comparison of Sec. 6.5.3. Grad is $y_{ui}\mathbf{q}_i$, the gradient of the signed inner-product score with respect to the user embedding, and NormGrad rescales it to unit norm. The four targets share one tuning protocol, and the PACF bar repeats the main result.

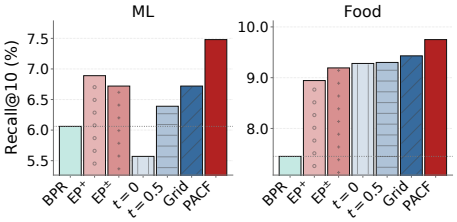


Figure 9: Sampling supports of t in $\mathcal{L}_{\text{flow}}$ on ML and Food, shown with BPR and the two endpoint penalties and measured by Recall@10 in percent. The dotted line marks the BPR level. PACF samples t uniformly over the whole segment.

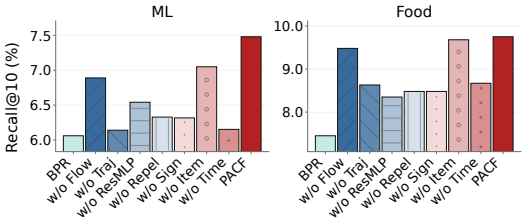


Figure 10: Removal of one PACF component at a time on ML and Food, measured by Recall@10 in percent against BPR. PACF feeds the complete input $(\mathbf{h}_t, \mathbf{q}_i, t, y_{ui})$ to one shared residual field and trains it with both of its pathwise loss terms.

NormGrad gives the highest control Recall on ML, while Const leads on Food. The different ML rankings under Recall and NDCG distinguish retrieval within the top ten from the higher positions that NDCG rewards. On Food, Const leads both metrics. Together, these comparisons evaluate how the target's user dependence, magnitude, and sign affect the static ranking learned through the shared field. Const retains the full user-to-item displacement and reverses it for sampled negatives, preserving both user differences and the distinction between the two labels throughout the path.

Sampling support. Fig. 9 repeats the sampling supports of Fig. 3 under Recall@10, and Recall rises from t=0 through t=0.5 to Grid on both datasets. At $t = 0$ every state equals its user embedding, so each user presents its own input and the field fits each displacement separately. Toward the item, the states of co-users approach one another by the factor $1 - t$ while their targets keep their difference, and the coefficient $(|\mu(y)| - L_{\mathbf{h}}(1 - t))_+$ in Eq. (11) grows. Uniform sampling covers the item-end interval where this coefficient is positive and also spreads the states over the whole segment, so its comparison with Grid changes the coverage of the segment and the proximity to the item together.

Field inputs. Fig. 10 repeats the component removals of Fig. 4 under Recall@10. The complete input determines the target through Eq. (29), which needs the state, the item, the time, and the label. Removing the item condition forces the field to infer $\mathbf{q}_i$ from a state that mixes user and item. Removing time leaves the remaining displacement $\mathbf{q}_i - \mathbf{h}_t$ without its scale $1/(1 - t)$, so one input fits a nearby user early on the segment and a distant user late on it. In both cases the residual records a missing input instead of a difference between users. Removing time lowers Recall more than removing the item condition on both datasets. The state still carries part of the item through $t\mathbf{q}_i$, while only t supplies the scale. The BCF target is linear in the embeddings up to the sign $\mu(y)$, so the residual skip of Eq. (6) carries the linear displacement and the nonlinear branch absorbs the sign and the conditional-mean correction. An equally sized plain MLP represents both through its nonlinear layers, which changes its fit and the state sensitivity $\mathbf{J}_{\mathbf{h}}$ that carries residuals into the embeddings.

Labels and negative targets. Zeroing the label in **w/o Sign** gives a positive and a negative pair near the same item similar inputs with opposite targets, so the field averages them and the residuals mix the two labels. **w/o Repel** removes negative regression and leaves negatives to BPR. Tab. 8 adds a finer control. Setting $\mu_- = 0$ keeps the negative flow terms and turns each into $\|\mathbf{v}_{\boldsymbol{\theta}}\|_2^2$, which removes the displacement target and the gradient through it, and every mean falls below PACF. At $\mu_- = 0$ the negative-label bound of Prop. 2 vanishes, so the signed negative targets of PACF add an item-end condition on the users who sample each item as a negative, with the weight $\mu_-^2 \leq 1$.

Flow and trajectory losses. Tab. 8 retunes each control over five seeds, matching the protocol of the main result. The control w/o $\mathcal{L}_{\text{traj}}$ sets $\beta = 0$, w/o $\mathcal{L}_{\text{flow}}$ sets $\lambda = 0$, and the Beta control sets $\beta = 0$ and samples $t \sim \text{Beta}(1, b)$ with $b > 1$, which moves the flow-matching states toward the user. The two losses constrain different quantities. Flow matching fits the field at reference states, and trajectory consistency checks the endpoint of the RK4 stages, where local errors add up or cancel. Removing either loss lowers both metrics on both datasets, and each drop exceeds the seed standard deviation on Food and for NDCG on ML. The Beta control recovers part of the NDCG decline on ML, stays at the level of w/o $\mathcal{L}_{\text{traj}}$ on Food, and remains below PACF on both datasets. The trajectory term therefore contributes beyond the placement of the flow states. Setting $\beta = 0$ removes the final RK4 stage at $t = 1$, which Remark 5 shows to couple all users of an item through one field output.

Table 8: Trajectory-term controls on ML and Food, measured by Recall@10 and NDCG@10 in percent as five-seed mean ± std. PACF exceeds every control in all four columns, including the Beta sampling control, so both pathwise terms and the signed negative target contribute to its ranking.

	ML		Food	
Setting	Recall	NDCG	Recall	NDCG
PACF	7.48 ± 0.41	3.73 ± 0.03	9.75 ± 0.08	4.79 ± 0.05
w/o $\mathcal{L}_{\text{traj}}$	7.05 ± 0.44	3.33 ± 0.13	9.32 ± 0.16	4.57 ± 0.06
$\beta = 0$, $t \sim \text{Beta}(1, b)$	7.25 ± 0.47	3.64 ± 0.27	9.21 ± 0.19	4.54 ± 0.09
w/o $\mathcal{L}_{\text{flow}}$	6.69 ± 0.76	3.22 ± 0.42	9.27 ± 0.08	4.52 ± 0.06
$\mu_- = 0$	7.08 ± 0.66	3.33 ± 0.35	9.17 ± 0.17	4.50 ± 0.12

Table 10: Hidden width d_h of the velocity network on ML and Food, with residual ratios and Recall@10 and NDCG@10 in percent as five-seed mean ± std. Endpoint fit tightens as the field widens while ranking changes non-monotonically, because PACF ranks with static embeddings.

	ML				Food			
d_h	ρ^+	ρ^-	Recall	NDCG	ρ^+	ρ^-	Recall	NDCG
64	0.5363 ± 0.0699	0.4833 ± 0.0706	6.36 ± 0.66	3.09 ± 0.29	0.3116 ± 0.0162	1.1969 ± 0.3042	8.99 ± 0.22	4.39 ± 0.12
128	0.2554 ± 0.0512	0.2474 ± 0.0471	6.10 ± 1.10	2.84 ± 0.47	0.2457 ± 0.0047	2.0545 ± 0.2827	9.16 ± 0.19	4.47 ± 0.08
256	0.1624 ± 0.0012	0.1883 ± 0.0069	6.56 ± 0.79	3.04 ± 0.37	0.2042 ± 0.0094	1.2037 ± 0.3624	9.35 ± 0.09	4.59 ± 0.06
512	0.1390 ± 0.0066	0.1755 ± 0.0109	6.59 ± 0.73	3.18 ± 0.32	0.1461 ± 0.0168	0.2491 ± 0.0318	9.19 ± 0.05	4.48 ± 0.04

Integration depth. Tab. 9 varies the number of RK4 steps T. ML stays nearly constant across T, and Food stays level up to $T = 8$ and declines at $T = 16$. Two properties of the solver make short integration sufficient. Each RK4 step evaluates the field at four generated states, so $T = 1$ already constrains the endpoint through composed predictions. The final-stage coupling of Remark 5 carries the weight $(6T)^{-2}$, so deeper integration dilutes the item-end constraint of the trajectory term, while training cost grows with the $4TB$ trajectory evaluations of the complexity analysis. Every listed dataset in Tab. 14 accordingly selects at most four RK4 steps.

Table 9: Number of RK4 steps T on ML and Food, measured by Recall@10 and NDCG@10 in percent. One RK4 step comes close to the deeper settings, so PACF can keep integration short.

Dataset	Metric	$T = 1$	$T = 4$	$T = 8$	$T = 16$
ML	Recall	6.39	6.39	6.39	6.39
	NDCG	3.16	3.05	3.21	3.11
Food	Recall	9.20	9.13	9.17	8.55
	NDCG	4.59	4.56	4.53	4.33

C.5 Field Capacity and Static Ranking

PACF uses the field as a regressor whose residuals train the embeddings, so fit quality and ranking quality can separate. Tab. 10 measures both as the width d_h grows. Writing $\Phi_1(\mathbf{p} \mid \mathbf{q}, y)$ for the RK4 endpoint from $\mathbf{p}$ under the condition $(\mathbf{q}, y)$, we normalize the endpoint error by the initial distance:

$$\rho_u^+ = \frac{\|\Phi_1(\mathbf{p}_u \mid \mathbf{q}_{i^+}, +1) - \mathbf{q}_{i^+}\|_2}{\|\mathbf{q}_{i^+} - \mathbf{p}_u\|_2}, \qquad \rho_u^- = \frac{\|\Phi_1(\mathbf{p}_u \mid \mathbf{q}_{i^-}, +1) - \mathbf{q}_{i^-}\|_2}{\|\mathbf{q}_{i^-} - \mathbf{p}_u\|_2}. \tag{16}$$

Both ratios integrate the positive-label trajectory. The ratio ρ_u^+ measures arrival at the observed item, and ρ_u^- measures how far the same trajectory carries the user toward a sampled negative item. Each entry averages the per-checkpoint median over all evaluated users across the five seeds.

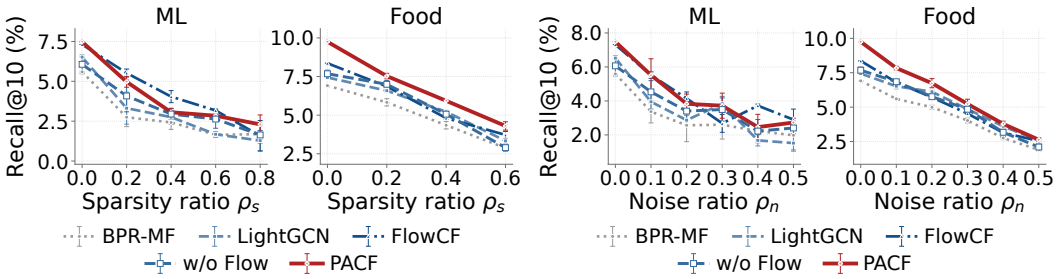
The endpoint fit improves monotonically with width on both datasets, while ranking changes non-monotonically, and on Food a narrower field ranks better than the widest one at a larger ρ^+. Eq. (30) explains this separation. Embedding updates scale with the flow residual, and Prop. 2 bounds the residuals of co-users by a coefficient that falls as the state sensitivity $L_{\mathbf{h}}$ grows. A more expressive field fits the displacements with smaller residuals and sends smaller updates into the embeddings, so a tighter endpoint fit can weaken the signal that reaches the scorer. The widest field also brings users close to sampled negative items under a supplied positive label, while ρ^- exceeds one on Food at narrower widths. An endpoint readout then finds every candidate close to its own endpoint, which

Table 11: Inner-product and cosine scoring under straight and spherical-geodesic reference paths on ML and Food, measured by Recall@10 and NDCG@10 in percent as five-seed mean ± std. The variants score closely, so the pathwise losses of PACF carry over to cosine scoring and geodesic paths.

Variant		ML		Food	
Score	Path	Recall	NDCG	Recall	NDCG
Inner product	Straight	7.48 ± 0.41	3.73 ± 0.03	9.75 ± 0.08	4.79 ± 0.05
Cosine	Straight	7.74 ± 0.96	3.71 ± 0.50	10.01 ± 0.09	4.93 ± 0.03
Cosine	Geodesic	7.18 ± 0.48	3.42 ± 0.23	9.93 ± 0.27	4.91 ± 0.08

Table 12: Learned item geometry under each scorer and reference path on ML and Food. With the inner product and straight paths, PACF keeps item norms widely spread and closely tied to popularity.

Variant		ML			Food		
Score	Path	Norm std.	Correlation	Effective rank	Norm std.	Correlation	Effective rank
Inner product	Straight	0.1890	+0.6900	311.8	0.8892	+0.9967	371.6
Cosine	Straight	0.0780	+0.6342	111.9	0.0322	+0.9232	394.0
Cosine	Geodesic	0.0111	+0.2513	101.4	0.0153	−0.0151	400.7



(a) Data sparsity.

(b) Replacement noise.

Figure 11: Robustness of PACF and baselines to interaction removal in panel a and item replacement in panel b on ML and Food, measured by Recall@10 in percent. Error bars show seed standard deviations. PACF stays above w/o Flow at every level and leads all compared models on Food.

explains the weak endpoint readouts in Tab. 4. Endpoint residuals therefore diagnose the conditional regression, and ranking metrics evaluate the static inner-product scorer that PACF deploys.

C.6 Scoring and Path Geometry

PACF regresses along straight paths in $\mathbb{R}^d$ and ranks with inner products, so embedding norms enter both the pathwise targets and the score. Tab. 11 replaces the scorer with cosine similarity and then the path with a spherical geodesic. Tab. 12 measures item geometry by the standard deviation of item norms, their correlation with popularity, and the effective rank of the normalized spectrum.

Cosine scoring with straight paths raises the mean on Food and on ML Recall, and the inner product keeps the highest ML NDCG, so the pathwise losses carry over to a cosine scorer. Straight paths keep the embedding norms in their states and targets, so the regression uses radial information while a cosine scorer orders items by angle. Geodesics between normalized directions remove this radial information. Item norms collapse to nearly one value, their correlation with popularity weakens on ML and vanishes on Food, and cosine scoring with geodesics falls below cosine scoring with straight paths in every column. With the inner product and straight paths, item norms stay spread and track popularity, so the MF scorer keeps a popularity component that cosine scoring discards.

Table 13: Computational cost of nine methods on six datasets, with training time in seconds per epoch, inference latency in milliseconds per user, and peak GPU memory in GB. PACF keeps the latency of an inner-product scorer and trains fastest on average after BPR-MF on Sports and Clothing.

Dataset	Metric	BPR-MF	NCF	NeuMF	LightGCN	MultiVAE	RecVAE	DiffRec	FlowCF	PACF
ML	Training	0.96 ± 1.13	4.31 ± 0.30	3.98 ± 0.34	1.29 ± 1.16	1.04 ± 0.18	1.10 ± 0.21	1.09 ± 0.19	7.12 ± 1.21	2.33 ± 0.20
	Inference	0.02 ± 0.00	0.26 ± 0.01	0.38 ± 0.00	0.03 ± 0.00	0.13 ± 0.00	0.46 ± 0.07	25.36 ± 0.13	0.68 ± 0.05	0.07 ± 0.00
	GPU	$< 0.01 \pm < 0.01$	0.10 ± 0.00	0.10 ± 0.00	$< 0.01 \pm < 0.01$	0.30 ± 0.00	0.40 ± 0.00	0.40 ± 0.00	0.30 ± 0.00	1.10 ± 0.00
Food	Training	0.39 ± 0.46	1.37 ± 0.37	1.35 ± 0.37	0.46 ± 0.54	0.46 ± 0.69	0.51 ± 0.63	0.38 ± 0.43	1.50 ± 0.49	0.86 ± 0.67
	Inference	0.03 ± 0.00	0.23 ± 0.05	0.33 ± 0.01	0.03 ± 0.00	0.14 ± 0.01	0.45 ± 0.01	27.17 ± 0.12	0.68 ± 0.01	0.07 ± 0.00
	GPU	$< 0.01 \pm < 0.01$	0.10 ± 0.00	0.10 ± 0.00	$< 0.01 \pm < 0.01$	0.20 ± 0.00	0.30 ± 0.00	0.20 ± 0.00	0.10 ± 0.00	0.90 ± 0.00
Dance	Training	0.64 ± 0.74	2.48 ± 0.63	2.94 ± 0.72	0.91 ± 0.89	0.91 ± 0.85	0.81 ± 0.78	0.74 ± 0.74	4.97 ± 0.93	1.96 ± 0.90
	Inference	0.03 ± 0.00	0.23 ± 0.01	0.34 ± 0.03	0.03 ± 0.00	0.14 ± 0.00	0.45 ± 0.01	25.78 ± 0.11	0.68 ± 0.03	0.07 ± 0.00
	GPU	$< 0.01 \pm < 0.01$	0.10 ± 0.00	0.10 ± 0.00	0.10 ± 0.00	0.20 ± 0.00	0.30 ± 0.00	0.30 ± 0.00	0.20 ± 0.00	0.90 ± 0.00
Baby	Training	2.87 ± 4.31	7.40 ± 4.17	6.25 ± 3.56	2.69 ± 4.05	3.37 ± 4.29	3.88 ± 4.44	3.85 ± 4.31	8.70 ± 3.78	4.67 ± 4.19
	Inference	0.03 ± 0.00	0.30 ± 0.00	0.46 ± 0.00	0.03 ± 0.00	0.14 ± 0.00	0.46 ± 0.01	45.33 ± 0.11	0.66 ± 0.01	0.07 ± 0.00
	GPU	$< 0.01 \pm < 0.01$	0.10 ± 0.00	0.10 ± 0.00	0.10 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.10 ± 0.00	0.50 ± 0.00	0.90 ± 0.00
Sports	Training	10.20 ± 19.00	18.56 ± 18.59	19.84 ± 19.90	11.75 ± 19.55	13.58 ± 19.23	11.69 ± 15.66	13.19 ± 18.33	28.40 ± 19.68	11.13 ± 15.66
	Inference	0.03 ± 0.00	0.47 ± 0.01	0.74 ± 0.00	0.03 ± 0.00	0.28 ± 0.00	0.45 ± 0.01	81.37 ± 0.09	0.95 ± 0.07	0.07 ± 0.00
	GPU	0.10 ± 0.00	0.10 ± 0.00	0.20 ± 0.00	0.30 ± 0.00	3.60 ± 0.00	3.60 ± 0.00	4.10 ± 0.00	1.20 ± 0.00	1.10 ± 0.00
Clothing	Training	10.99 ± 22.05	20.81 ± 26.45	21.96 ± 28.02	14.28 ± 26.66	17.53 ± 28.40	14.72 ± 22.00	17.13 ± 26.60	31.11 ± 28.31	13.98 ± 22.75
	Inference	0.03 ± 0.00	0.59 ± 0.01	0.87 ± 0.00	0.04 ± 0.00	0.52 ± 0.00	0.45 ± 0.00	96.84 ± 0.11	0.82 ± 0.09	0.07 ± 0.00
	GPU	0.10 ± 0.00	0.10 ± 0.00	0.20 ± 0.00	0.30 ± 0.00	4.80 ± 0.00	4.80 ± 0.00	5.50 ± 0.00	1.60 ± 0.00	0.90 ± 0.00

C.7 Robustness to Sparsity and Noise

Fig. 11 repeats the perturbation study of Fig. 6 under Recall@10. On ML, FlowCF leads at intermediate removal levels, and under replacement LightGCN leads at one level while FlowCF leads or ties at the others. PACF stays above w/o Flow at every level of both perturbations on both datasets and has the highest Recall at every level on Food, the sparser dataset with fewer interactions per user.

Removal reduces the observed pairs of each user, and Eq. (2) constrains a user embedding only through these pairs. Every retained pair still enters the shared regression, so the item-end bound of Prop. 2 keeps acting on the users who share each remaining item, and the gap to w/o Flow measures this contribution. Replacement keeps the number of pairs and corrupts their associations, so part of the flow targets point to random items. PACF stays above w/o Flow at every replacement level, so the shared regression keeps its benefit when some targets are wrong. ML is the densest dataset, where each user keeps the most constraints of its own, and it also shows the smallest gain of PACF over BPR-MF in the main comparison, which matches its weaker lead under perturbation.

C.8 Computational Cost

Training evaluates the velocity network $1 + K_- + 4T$ times per positive pair, and inference scores with Eq. (1) at $O(d)$ per pair and $O(|\mathcal{I}|d)$ per user, the operations of MF. Tab. 13 measures every method with its selected configuration, with latency covering top-10 ranking over all items.

The costs follow from how each method scales with the catalog. The field of PACF has $O(d_h(d+d_h))$ parameters, independent of $|\mathcal{U}|$ and $|\mathcal{I}|$, and its cost per positive pair is fixed by K_- and T. PACF accordingly has the lowest mean epoch time among the eight methods other than BPR-MF on Sports and Clothing, the two largest datasets, and its peak training memory stays near one gigabyte on every dataset. MultiVAE, RecVAE, DiffRec, and FlowCF process an interaction vector of length $|\mathcal{I}|$ per user, so their memory grows with the catalog and exceeds that of PACF on Sports and Clothing. Epoch times on the Amazon datasets vary widely across epochs, so the comparison uses means. At inference, PACF keeps the latency of an inner-product scorer, like BPR-MF and LightGCN. NCF and NeuMF run an MLP on every user-item pair, the variational autoencoders encode and decode each interaction vector, and DiffRec and FlowCF generate scores iteratively, DiffRec most slowly.

C.9 Selected Hyperparameters

Tab. 14 lists the configuration that validation selects for each dataset, namely the embedding dimension d, hidden width d_h, RK4 steps T, learning rate η, flow weight λ, trajectory weight β, and negative-target scale μ_-. The search ranges of β and μ_- include zero, and every listed dataset selects positive values for both, so validation keeps the trajectory term and the signed negative regression. The flow weight has a positive lower bound by design. Through Eq. (23), μ_- sets both the size of the negative target and the weight μ_-^2 of the reparameterized negative regression, and its selected value varies

Table 14: Selected PACF hyperparameters on ML, Food, Dance, Baby, Sports, and Clothing. Every dataset selects positive flow, trajectory, and negative-target weights and at most four RK4 steps, so the validation search keeps all pathwise terms of PACF active with short RK4 integration.

Dataset	d	d_h	T	η	λ	β	μ_-
ML	256	128	1	4.83×10^{-2}	0.86	0.25	0.71
Food	512	64	4	2.26×10^{-2}	0.41	0.18	0.34
Dance	512	512	1	1.40×10^{-2}	0.97	0.17	0.23
Baby	512	64	4	1.18×10^{-2}	0.21	0.44	0.76
Sports	256	512	1	3.29×10^{-4}	0.98	0.30	0.99
Clothing	512	256	4	9.66×10^{-5}	0.69	0.11	0.05

widely, down to a small value on Clothing. Every dataset selects at most four RK4 steps, in line with the integration-depth analysis, and the two largest datasets select smaller learning rates.

D Theoretical Analysis

D.1 Straight Reference Path

The reference displacement of a pair (u, i) is $\mathbf{q}_i - \mathbf{p}_u$, and the segment in Eq. (3) traverses it at the constant velocity $\dot{\mathbf{h}}_t = \mathbf{q}_i - \mathbf{p}_u$ (Liu et al., 2023; Park et al., 2024). Among absolutely continuous paths $\mathbf{h} : [0, 1] \to \mathcal{M}$ with $\mathbf{h}_0 = \mathbf{p}_u$ and $\mathbf{h}_1 = \mathbf{q}_i$, this segment uniquely minimizes the kinetic energy $\mathcal{C}[\mathbf{h}] = \int_0^1 \|\dot{\mathbf{h}}_t\|_2^2 \, dt$ in Euclidean space (Villani, 2009; Finlay et al., 2020). Its Euclidean interpolation follows Rectified Flow (Liu et al., 2023) and flow matching (Lipman et al., 2023) in embedding space. The endpoints give $\mathbf{q}_i - \mathbf{p}_u = \int_0^1 \dot{\mathbf{h}}_t \, dt$ (Geng et al., 2025). Jensen's inequality for the strictly convex $\|\cdot\|_2^2$ then gives $\|\mathbf{q}_i - \mathbf{p}_u\|_2^2 \le \mathcal{C}[\mathbf{h}]$, with equality exactly when $\dot{\mathbf{h}}_t$ is constant. The boundary conditions fix this constant at $\mathbf{q}_i - \mathbf{p}_u$, and integration from $\mathbf{p}_u$ recovers Eq. (3).

For $t < 1$, Eq. (3) gives $\mathbf{p}_u = (\mathbf{h}_t - t\mathbf{q}_i)/(1 - t)$, so the signed target is the function $\mathbf{f}^\star$ of the complete network input in Eq. (29). Under the positive label, $\mathbf{f}^\star$ returns $\mathbf{q}_i - \mathbf{p}_u$ at every state of the segment, so Eq. (7) with $\mathbf{f}^\star$ in place of $\mathbf{v}_{\boldsymbol{\theta}}$ has the solution in Eq. (3) for $0 \le t < 1$, which converges to $\mathbf{q}_i$ as $t \to 1$. The negative label reverses this velocity and scales it by μ_-, so Eq. (4) keeps the reference displacement of every pair and lets the label set its direction and magnitude. The last RK4 stage evaluates the field at $t = 1$, and App. D.5 derives the constraint it places on the users of an item.

D.2 Positive-Score Monotonicity and Ranking Alignment

Along the positive path $\mathbf{h}_t^{(u,i^+)} = (1 - t)\mathbf{p}_u + t\mathbf{q}_{i^+}$, define the positive score $\psi(t) \triangleq \langle \mathbf{h}_t^{(u,i^+)}, \mathbf{q}_{i^+} \rangle$, which is the score $s(u, i^+)$ of Eq. (1) with the user embedding moved to the path state.

Proposition 3 (Pairwise Ranking Alignment)**.** *Fix $\mathbf{q}_{i^+}$ and $\mathbf{q}_{i^-}$, and let $\ell_{\mathrm{BPR}}(t) = -\log \sigma\big(\langle \mathbf{h}_t^{(u,i^+)}, \mathbf{q}_{i^+} - \mathbf{q}_{i^-} \rangle\big)$. Then ℓ_{BPR} is non-increasing in t exactly when this condition holds:*

$$\langle \mathbf{q}_{i^+} - \mathbf{p}_u, \, \mathbf{q}_{i^+} - \mathbf{q}_{i^-} \rangle \ge 0. \tag{17}$$

Proof. Eq. (3) makes ψ and the margin $\Delta(t) = \langle \mathbf{h}_t^{(u,i^+)}, \mathbf{q}_{i^+} - \mathbf{q}_{i^-} \rangle$ linear in t, with slopes:

$$\psi'(t) = \|\mathbf{q}_{i^+}\|_2^2 - \langle \mathbf{p}_u, \mathbf{q}_{i^+} \rangle, \qquad \Delta'(t) = \psi'(t) - \langle \mathbf{q}_{i^+} - \mathbf{p}_u, \mathbf{q}_{i^-} \rangle. \tag{18}$$

As $-\log \sigma$ strictly decreases, ℓ_{BPR} is non-increasing exactly when $\Delta' \ge 0$, which is Eq. (17). □

By Eq. (18), the positive score is non-decreasing in t if and only if the user embedding satisfies:

$$\langle \mathbf{p}_u, \mathbf{q}_{i^+} \rangle \le \|\mathbf{q}_{i^+}\|_2^2, \tag{19}$$

and strictly increasing under strict inequality. Equal norms $\|\mathbf{p}_u\|_2 = \|\mathbf{q}_{i^+}\|_2$ satisfy this condition because $\psi'(t)$ then equals $\frac{1}{2}\|\mathbf{q}_{i^+} - \mathbf{p}_u\|_2^2$. For a differentiable loss with $\partial \ell / \partial s(u, i^+) < 0$ and

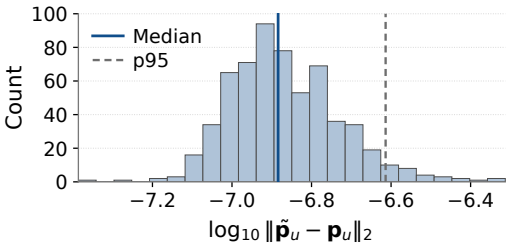


Figure 12: Distribution of RK4 round-trip residuals over ML users at one PACF checkpoint. The solid and dashed lines mark the median and the 95th percentile. The residuals stay below 10^{-6} for every user, so the RK4 map that PACF trains is nearly invertible for a fixed item and label.

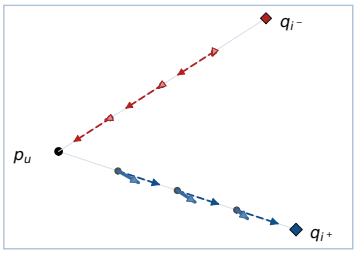


Figure 13: Learned field and BCF target for positive and negative ML pairs at a PACF checkpoint. Blue and red mark positive and negative pairs, dashed and solid arrows the target $\mathbf{u}_t$ and the field $\mathbf{v}_{\boldsymbol{\theta}}$, dots the sampled states, and diamonds the items. The field closely follows both signed targets, and the small remaining gaps are the residuals that update PACF's user embeddings.

fixed competing scores, the chain rule $d\ell/dt = (\partial\ell/\partial s(u, i^+))\,\psi'(t)$ makes ℓ non-increasing under Eq. (19), which covers BPR, binary cross-entropy on the positive pair, and sampled softmax. The margin slope in Eq. (18) shows that Eq. (19) implies Eq. (17) whenever $\langle \mathbf{q}_{i^+} - \mathbf{p}_u, \mathbf{q}_{i^-} \rangle \le 0$.

Measured condition. On retuned checkpoints, adding the pathwise losses reduces the violation rate $\Pr[\psi' < 0]$ of the positive-score slope in Eq. (18) from 0.044804 to 0.000074 on ML and from 0.002392 to 0.001233 on Food, so almost every measured pair satisfies Eq. (19). For these pairs, moving the user toward the positive item with competing scores fixed lowers every loss that decreases in the positive score, and the pairwise margin follows the alignment condition of Eq. (17).

D.3 Conditional Flow Regularity

The state space is $\mathcal{M} = \mathbb{R}^d$. The exact-flow results below assume that for every fixed item and label $(\mathbf{q}_i, y_{ui})$ the field $\mathbf{v}_{\boldsymbol{\theta}}$ is continuously differentiable in $\mathbf{h} \in \mathcal{M}$, continuous in $t \in [0, 1]$, and $L_{\mathbf{h}}$-Lipschitz in $\mathbf{h}$ with one state sensitivity $L_{\mathbf{h}}$ over all states, times, and item-label contexts.

We use $\varphi_{s\to t}$ for the exact map from time s to time t and write $\varphi_t = \varphi_{0\to t}$. The corresponding finite-step RK4 map is $\Phi_{s\to t}$, with $\Phi_1 = \Phi_{0\to 1}$ in the training objective. For a fixed item and label, the exact trajectory satisfies the conditional initial-value problem over the complete interval:

$$\dot{\widetilde{\mathbf{h}}}_t^{(u,i)} = \mathbf{v}_{\boldsymbol{\theta}}(\widetilde{\mathbf{h}}_t^{(u,i)}, t, \mathbf{q}_i, y_{ui}), \qquad \widetilde{\mathbf{h}}_0^{(u,i)} = \mathbf{p}_u. \tag{20}$$

The exact map from the initial state $\mathbf{p}_u$ to time t has the following equivalent integral form:

$$\varphi_t(\mathbf{p}_u, \mathbf{q}_i, y_{ui}) \triangleq \mathbf{p}_u + \int_0^t \mathbf{v}_{\boldsymbol{\theta}}\left(\widetilde{\mathbf{h}}_\tau^{(u,i)},\ \tau,\ \mathbf{q}_i,\ y_{ui}\right) d\tau. \tag{21}$$

Remark 4 (Exact Conditional Flow). Under this regularity condition, the Picard–Lindelöf theorem gives Eq. (20) a unique solution on $[0, 1]$ for every $\mathbf{p}_u \in \mathcal{M}$, and the same argument backward in time makes $\varphi_t(\cdot, \mathbf{q}_i, y_{ui})$ a bijection with inverse $\varphi_{t\to 0}$. Its Jacobian $\mathbf{J}(t) = \partial\varphi_t/\partial\mathbf{p}$ solves $\dot{\mathbf{J}}(t) = (\partial\mathbf{v}_{\boldsymbol{\theta}}/\partial\mathbf{h})\mathbf{J}(t)$ with $\mathbf{J}(0) = \mathbf{I}$, and Liouville's formula $\det \mathbf{J}(t) = \exp \int_0^t \operatorname{tr}(\partial\mathbf{v}_{\boldsymbol{\theta}}/\partial\mathbf{h})\, d\tau > 0$ keeps it invertible, so φ_t is a global C^1-diffeomorphism of the state space $\mathcal{M}$ for every $t \in [0, 1]$.

For two users with the same item and label, Grönwall's inequality applied to Eq. (21) gives $\|\varphi_1(\mathbf{p}_u, \mathbf{q}_i, y) - \varphi_1(\mathbf{p}_v, \mathbf{q}_i, y)\|_2 \le e^{L_{\mathbf{h}}} \|\mathbf{p}_u - \mathbf{p}_v\|_2$. A negative one-sided Lipschitz constant turns this bound into a contraction that pulls the exact endpoints of the two users together.

Fig. 12 integrates each ML user embedding forward and back with eight RK4 steps, $\tilde{\mathbf{p}}_u = (\Phi_{1\to 0} \circ \Phi_{0\to 1})(\mathbf{p}_u)$, and plots the distribution of $\log_{10} \|\tilde{\mathbf{p}}_u - \mathbf{p}_u\|_2$ over users at one PACF checkpoint. The round trip returns every user close to its starting point for a fixed item and label, so the finite-step map Φ_1 that PACF trains inherits the invertibility that Remark 4 establishes for the exact flow.

Because $\mathcal{L}_{\text{traj}}$ averages $\|\Phi_1(\mathbf{p}_u) - \mathbf{q}_{i^+}\|_2^2$ over training positives, Jensen's inequality gives $\mathbb{E}_{(u,i^+)}\|\Phi_1(\mathbf{p}_u) - \mathbf{q}_{i^+}\|_2 \le \sqrt{\mathcal{L}_{\text{traj}}}$ for the mean RK4 endpoint distance. Candidate ranking uses the static score in Eq. (1), which BPR trains directly and both pathwise losses shape.

D.4 Shared-Field Regression and Local Consistency

PACF fits the signed targets of all pairs with one network $\mathbf{v}_{\boldsymbol{\theta}}$, so an update driven by one pair changes the predictions and residuals at the inputs of other pairs, including other users of the same item.

Let $\mathcal{D}$ be the distribution of training tuples (u, i, y_{ui}, t) from Alg. 1, with $t \sim \text{Unif}[0,1]$ independent of (u, i, y_{ui}). Under $\mathcal{D}$, the state $\mathbf{h}_t = (1-t)\mathbf{p}_u + t\mathbf{q}_i$, target $\mathbf{u}_t = \mu(y_{ui})(\mathbf{q}_i - \mathbf{p}_u)$, and complete input $\mathbf{X} \triangleq (\mathbf{h}_t, t, \mathbf{q}_i, y_{ui})$ are random vectors. All conditional expectations use $\text{supp}(\mathbf{X})$. Letting the network range over all measurable fields $\mathbf{v} : \text{supp}(\mathbf{X}) \to \mathcal{M}$, Eq. (5) becomes the population risk:

$$\mathcal{L}_{\text{flow}}(\mathbf{v}) = \mathbb{E}_{\mathcal{D}}\Big[\, \big\|\mathbf{v}(\mathbf{X}) - \mathbf{u}_t\big\|_2^2 \,\Big]. \tag{22}$$

Over all measurable fields, Eq. (22) is minimized by the conditional mean $\mathbf{v}^*(\mathbf{x}) = \mathbb{E}_{\mathcal{D}}[\mathbf{u}_t \mid \mathbf{X} = \mathbf{x}] = \mu(y)\big(\mathbf{q} - \mathbb{E}_{\mathcal{D}}[\mathbf{p}_u \mid \mathbf{X} = \mathbf{x}]\big)$ for $\mathbf{X}$-almost every $\mathbf{x} = (\mathbf{h}, t, \mathbf{q}, y)$, and for $t < 1$ Eq. (3) pins $\mathbf{p}_u$ to $\mathbf{p} = (\mathbf{h} - t\mathbf{q})/(1-t)$, giving $\mathbf{v}^*(\mathbf{x}) = \mu(y)(\mathbf{q} - \mathbf{p})$. For $t < 1$, the complete input therefore fixes the target function in Eq. (29), and the population optimum matches each target pointwise.

Signed regression. For $\mu_- > 0$ and either label, every field output $\mathbf{v}$ satisfies the identity:

$$\big\|\mathbf{v} - \mu(y)(\mathbf{q}_i - \mathbf{p}_u)\big\|_2^2 = \mu(y)^2 \big\|\mathbf{v}/\mu(y) - (\mathbf{q}_i - \mathbf{p}_u)\big\|_2^2. \tag{23}$$

Under the output reparameterization $\mathbf{g} = \mathbf{v}/\mu(y)$, each flow term regresses $\mathbf{g}$ onto the unsigned displacement with weight $\mu(y)^2$, which is 1 for positive and μ_-^2 for negative pairs. The label thus sets the target direction, and μ_- acts as a quadratic weight of this reparameterized regression.

Cross-pair coupling requires a bound on how much the shared prediction varies. If $\mathbf{v}$ is $L_{\mathbf{X}}$-Lipschitz in $(\mathbf{h}_t, t, \mathbf{q}_i)$ at each fixed label and $B_{y_0}(\mathbf{x}_0, r)$ is the set of label-y_0 inputs whose continuous components lie within distance r of those of $\mathbf{x}_0 = (\mathbf{h}_0, t_0, \mathbf{q}_0, y_0) \in \text{supp}(\mathbf{X})$, with $\Pr_{\mathcal{D}}(\mathbf{X} \in B_{y_0}(\mathbf{x}_0, r)) > 0$, then Jensen's inequality gives $\|\mathbf{v}(\mathbf{x}_0) - \mathbb{E}_{\mathcal{D}}[\mathbf{v}(\mathbf{X}) \mid \mathbf{X} \in B_{y_0}(\mathbf{x}_0, r)]\|_2 \le L_{\mathbf{X}} r$.

This bound relates training pairs whose states, times, and item conditions lie within r under one label, which is stricter than a crossing of two segments. Within such a neighborhood the shared field varies by at most $L_{\mathbf{X}} r$, while each pair compares the prediction with its own signed target, so their target difference enters their residuals and, through Eq. (30), their embedding gradients. Users of one item share its condition and label, and their states converge as $t \to 1$, which yields Prop. 2.

Fig. 13 projects one positive and one negative ML pair onto two principal components with a shared scale and aligned user endpoints. It shows the constant signed targets and the state-dependent predictions at three sampled states per segment, and the gaps between them are the fitting residuals that Eq. (30) maps through the input and target derivatives into the user embedding $\mathbf{p}_u$.

D.5 Proofs of the Item-End Results

The proofs use Eqs. (3) and (4). Prop. 2 uses a state-Lipschitz condition, implied by the regularity condition in App. D.3, and Remark 5 uses the RK4 step in Eq. (14) with step size $1/T$.

Proof of Proposition 2. **Step 1. Pointwise bound.** Write $\mathbf{h}_u = \mathbf{h}_t^{(u,i)}$, $\mathbf{h}_v = \mathbf{h}_t^{(v,i)}$, and $\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{h}) = \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{h}, t, \mathbf{q}_i, y)$. States differ by $(1-t)(\mathbf{p}_u - \mathbf{p}_v)$, and the targets differ by $-\mu(y)(\mathbf{p}_u - \mathbf{p}_v)$, so the target difference splits into the difference of field outputs and the difference of residuals:

$$-\mu(y)(\mathbf{p}_u - \mathbf{p}_v) = \big[\mathbf{v}_{\boldsymbol{\theta}}(\mathbf{h}_u) - \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{h}_v)\big] - \big[\mathbf{e}_u(t) - \mathbf{e}_v(t)\big]. \tag{24}$$

The triangle inequality and the Lipschitz condition on the segment between $\mathbf{h}_u$ and $\mathbf{h}_v$ give $|\mu(y)|\,\|\mathbf{p}_u - \mathbf{p}_v\|_2 \le L_{\mathbf{h}}(1-t)\,\|\mathbf{p}_u - \mathbf{p}_v\|_2 + \|\mathbf{e}_u(t)\|_2 + \|\mathbf{e}_v(t)\|_2$. Rearranging gives Eq. (11) when $|\mu(y)| > L_{\mathbf{h}}(1-t)$, and otherwise Eq. (11) holds trivially because its right side is zero.

Step 2. Integration over t. Since $(a+b)^2 \le 2(a^2+b^2)$, Eq. (11) gives the squared bound:

$$\|\mathbf{e}_u(t)\|_2^2 + \|\mathbf{e}_v(t)\|_2^2 \ge \tfrac{1}{2}\big(|\mu(y)| - L_{\mathbf{h}}(1-t)\big)_+^2\, \|\mathbf{p}_u - \mathbf{p}_v\|_2^2. \tag{25}$$

With $s = 1-t$, the positive part vanishes for $s \geq |\mu(y)|/L_{\mathbf{h}}$, which is at most 1 because $L_{\mathbf{h}} \geq |\mu(y)|$, so the expected squared coefficient over $t \sim \mathrm{Unif}[0,1]$ has the following closed form:

$$\mathbb{E}_{t\sim\mathrm{Unif}[0,1]}\big(|\mu(y)| - L_{\mathbf{h}}(1-t)\big)_+^2 = \int_0^{|\mu(y)|/L_{\mathbf{h}}} \big(|\mu(y)| - L_{\mathbf{h}} s\big)^2\, ds = \frac{|\mu(y)|^3}{3L_{\mathbf{h}}}. \tag{26}$$

Taking the expectation of the previous display therefore gives the following pairwise bound:

$$\mathbb{E}_t\big[\|\mathbf{e}_u(t)\|_2^2 + \|\mathbf{e}_v(t)\|_2^2\big] \geq \frac{|\mu(y)|^3}{6L_{\mathbf{h}}}\, \|\mathbf{p}_u - \mathbf{p}_v\|_2^2. \tag{27}$$

Step 3. Dispersion form. Let $n = |\mathcal{S}|$. For $n = 1$, Eq. (12) has a zero right side. For $n \geq 2$, sum Eq. (27) over unordered pairs. Each user appears $n-1$ times. The identity $\sum_{\{u,v\}\subseteq\mathcal{S}} \|\mathbf{p}_u - \mathbf{p}_v\|_2^2 = n\sum_{u\in\mathcal{S}} \|\mathbf{p}_u - \bar{\mathbf{p}}_{\mathcal{S}}\|_2^2$ converts pairwise distances to dispersion, giving the summed bound:

$$(n-1)\,\mathbb{E}_t \sum_{u\in\mathcal{S}} \|\mathbf{e}_u(t)\|_2^2 \geq \frac{|\mu(y)|^3}{6L_{\mathbf{h}}}\, n \sum_{u\in\mathcal{S}} \|\mathbf{p}_u - \bar{\mathbf{p}}_{\mathcal{S}}\|_2^2. \tag{28}$$

Dividing both sides by $n-1$ and using $n/(n-1) \geq 1$ yields Eq. (12) for every $n \geq 2$. □

Remark 5 (Shared Final RK4 Stage). Fix item i, a set $\mathcal{S}$ of users observed with it, and $\mathbf{r}_u = \Phi_1(\mathbf{p}_u \mid \mathbf{q}_i, +1) - \mathbf{q}_i$. If the positive field returns $\mathbf{q}_i - \mathbf{p}_u$ at the reference states of every $u \in \mathcal{S}$ at the stage times n/T and $(n+\frac{1}{2})/T$ for $n < T$, the last RK4 stage evaluates the same input $(\mathbf{q}_i, 1, \mathbf{q}_i, +1)$ for all these users. With $\mathbf{c}_i$ the output there, $\mathbf{r}_u = (\mathbf{c}_i - \mathbf{q}_i + \mathbf{p}_u)/(6T)$, so every output $\mathbf{c}_i$ gives $\sum_{u\in\mathcal{S}} \|\mathbf{r}_u\|_2^2 \geq (6T)^{-2} \sum_{u\in\mathcal{S}} \|\mathbf{p}_u - \bar{\mathbf{p}}_{\mathcal{S}}\|_2^2$, with equality if and only if $\mathbf{c}_i = \mathbf{q}_i - \bar{\mathbf{p}}_{\mathcal{S}}$.

Shared final stage. This paragraph proves Remark 5. Write $t_n = n/T$, and let step n apply Eq. (14) on $[t_n, t_{n+1}]$ with step size $1/T$. If $\Phi_{0\to t_n}(\mathbf{p}_u) = \mathbf{h}_{t_n}^{(u,i)}$, which holds at $n = 0$, the stages of step n evaluate reference states at t_n, at $t_n + \frac{1}{2T}$ twice, and at t_{n+1}. For $n < T-1$ all four return $\mathbf{q}_i - \mathbf{p}_u$, so $\Phi_{0\to t_{n+1}}(\mathbf{p}_u) = \mathbf{h}_{t_{n+1}}^{(u,i)}$, and induction reaches t_{T-1}. In the last step the first three stages return $\mathbf{q}_i - \mathbf{p}_u$, while the fourth evaluates $\mathbf{h}_{t_{T-1}}^{(u,i)} + \frac{1}{T}(\mathbf{q}_i - \mathbf{p}_u) = \mathbf{q}_i$ at $t = 1$ and returns $\mathbf{c}_i$. Hence $\Phi_1(\mathbf{p}_u) = \mathbf{q}_i + (\mathbf{c}_i - \mathbf{q}_i + \mathbf{p}_u)/(6T)$. Since $\sum_{u\in\mathcal{S}} \|\mathbf{p}_u - \mathbf{a}\|_2^2$ is smallest at $\mathbf{a} = \bar{\mathbf{p}}_{\mathcal{S}}$, the choice $\mathbf{a} = \mathbf{q}_i - \mathbf{c}_i$ gives the bound in Remark 5, with equality at $\mathbf{c}_i = \mathbf{q}_i - \bar{\mathbf{p}}_{\mathcal{S}}$. Under this fit one user reaches $\mathbf{q}_i$ exactly, while distinct users keep a positive total endpoint residual. Euler and midpoint steps evaluate the field only at $t < 1$, so this coupling comes from the final RK4 stage.

Item-end intuition. If the field gives nearly the same prediction at the states of the users in $\mathcal{S}$, the flow loss is smallest when this prediction equals their mean target $\mu(y)(\mathbf{q}_i - \bar{\mathbf{p}}_{\mathcal{S}})$, and the residual of user u is then $\mu(y)(\mathbf{p}_u - \bar{\mathbf{p}}_{\mathcal{S}})$. The target route of Eq. (30) then contributes $2\mu(y)^2(\mathbf{p}_u - \bar{\mathbf{p}}_{\mathcal{S}})$ to the gradient, so a descent step moves $\mathbf{p}_u$ toward the group mean, while the input route carries the factor $1-t$ and vanishes at the item end. For $y = +1$ the group holds the observed users of i, and for $y = -1$ it holds the users whose sampled negatives include i, with weight $\mu_-^2 \leq 1$.

State sensitivity. For fixed $(t, \mathbf{q}, y)$ with $t < 1$, $\mathbf{f}^\star$ in Eq. (29) has state Jacobian $-\mu(y)\mathbf{I}/(1-t)$ and state sensitivity $|\mu(y)|/(1-t)$, which exceeds a fixed $L_{\mathbf{h}}$ exactly when $1 - t < |\mu(y)|/L_{\mathbf{h}}$ and grows without bound as $t \to 1$ for $\mu(y) \neq 0$. A field matching this Jacobian near the segments requires $L_{\mathbf{h}} \geq |\mu(y)|$. If $0 < L_{\mathbf{h}} < |\mu(y)|$, integration instead gives the positive coefficient $\big(|\mu(y)|^3 - (|\mu(y)| - L_{\mathbf{h}})^3\big)/(6L_{\mathbf{h}})$ multiplying the user dispersion for that item and label.

D.6 Embedding Gradient Derivations

Proposition 6 (Residual-Dependent Flow Gradients)**.** *Fix a pair (u, i) with label $y = y_{ui}$ and a time $t \in [0, 1)$. Under Eq.* (3)*, the BCF target of Eq.* (4) *is the following function of the network input:*

$$\mathbf{f}^\star(\mathbf{h}, t, \mathbf{q}, y) = \mu(y)\, \frac{\mathbf{q} - \mathbf{h}}{1-t}. \tag{29}$$

Suppose the field is differentiable in its continuous inputs. Let $\mathbf{e} = \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{h}_t, t, \mathbf{q}_i, y) - \mu(y)(\mathbf{q}_i - \mathbf{p}_u)$ and $\mathbf{J}_{\mathbf{h}} = \partial\mathbf{v}_{\boldsymbol{\theta}}/\partial\mathbf{h}$ at this input. With $\boldsymbol{\theta}$ and $\mathbf{q}_i$ fixed, the squared residual has the user gradient:

$$\nabla_{\mathbf{p}_u} \|\mathbf{e}\|_2^2 = 2\big[(1-t)\mathbf{J}_{\mathbf{h}} + \mu(y)\mathbf{I}\big]^\top \mathbf{e}. \tag{30}$$

Both gradients are linear in the residual, so an exactly fitted pair contributes no flow gradient. The user gradient combines the state route $(1-t)\mathbf{J_h}$ with the target route $\mu(y)\mathbf{I}$, and the item gradient also differentiates the item condition of the network. The label sets the target direction, and after the output rescaling in Eq. (23), negative-pair regression carries weight μ_-^2 relative to positive pairs.

Proof of Proposition 6. Solving Eq. (3) gives $\mathbf{p}_u = (\mathbf{h}_t - t\mathbf{q}_i)/(1-t)$ and hence Eq. (29). The state derivative with respect to $\mathbf{p}_u$ is $(1-t)\mathbf{I}$, while the target derivative is $-\mu(y)\mathbf{I}$. Applying the chain rule to $\|\mathbf{e}\|_2^2$ gives Eq. (30). With $\mathbf{J_q} = \partial \mathbf{v}_{\boldsymbol{\theta}}/\partial \mathbf{q}$ at the same input, the item gradient is $\nabla_{\mathbf{q}_i}\|\mathbf{e}\|_2^2 = 2\big[t\mathbf{J_h} + \mathbf{J_q} - \mu(y)\mathbf{I}\big]^\top \mathbf{e}$, which adds the direct derivative through the item condition. □

Proof of Proposition 1. Eq. (1) gives $\Delta s = \langle \mathbf{p}_u, \mathbf{q}_{i^+} - \mathbf{q}_{i^-} \rangle$. For the loss $\ell_{\mathrm{rec}} = -\log \sigma(\Delta s)$, the chain rule starts from the derivative with respect to the margin Δs, which takes the form:

$$\frac{\partial \ell_{\mathrm{rec}}}{\partial \Delta s} = -\sigma(-\Delta s). \tag{31}$$

Since $\partial \Delta s/\partial \mathbf{p}_u = \mathbf{q}_{i^+} - \mathbf{q}_{i^-}$, the chain rule gives the BPR contribution to the user update:

$$-\nabla_{\mathbf{p}_u}\ell_{\mathrm{rec}} = \sigma(-\Delta s)(\mathbf{q}_{i^+} - \mathbf{q}_{i^-}). \tag{32}$$

The flow contribution is $-\lambda \nabla_{\mathbf{p}_u}\ell_{\mathrm{flow}}$, where $\nabla_{\mathbf{p}_u}\ell_{\mathrm{flow}}$ sums Eq. (30) over the flow terms of the triplet. For the trajectory contribution, use the endpoint residual $\mathbf{r}_u$ and integration Jacobian $\mathbf{J}_\Phi$ from Prop. 1. Differentiating $\ell_{\mathrm{traj}} = \|\mathbf{r}_u\|_2^2$ with $\mathbf{q}_{i^+}$ and $\boldsymbol{\theta}$ fixed gives the trajectory contribution:

$$-\beta \nabla_{\mathbf{p}_u}\ell_{\mathrm{traj}} = -2\beta \mathbf{J}_\Phi(\mathbf{p}_u)^\top \mathbf{r}_u. \tag{33}$$

Adding the three negative gradients gives Eq. (10), the decomposition stated in Prop. 1. □