WARMTH PROPAGATION: A LIGHTWEIGHT CONTENT-AWARE APPROACH TO COLD-START RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we propose Warmth Propagation, a novel approach to mitigating the cold-start problem in recommender systems. It uses content data to compute pairwise item similarity, then propagates collaborative information from warm items to similar cold ones, i.e., items with no click data. A key feature of the proposed approach is its efficiency: it computes pairwise similarity implicitly and can therefore handle large-scale item catalogs. Warmth Propagation is not a stand-alone approach and can be applied to pretrained recommenders that were not originally designed to handle cold items, requiring only that item embeddings be initialized from content features. We show experimentally that the proposed approach adds minimal computational overhead and outperforms various cold-start recommenders. Source code is available at `https://anonymous.4open.science/r/Warmth-Propagation/`

1 INTRODUCTION

The cold start problem, the inability of collaborative filtering models to score items with no interaction history, remains a major practical challenge in recommender system pipelines. The recent emergence of powerful pretrained encoders (language models, vision transformers, multi-modal models) offers a natural remedy: rich *content embeddings*, item representations derived from textual descriptions, images, or other side features, are available for all items and can inform predictions for newly added ones.

A direct approach, learning a parametric map from content embeddings to *collaborative embeddings* (item representations learned from user interaction data), is hampered by the inherent noise in both modalities. Collaborative embeddings are estimated from sparse, biased interaction logs: tail items accumulate only a small fraction of interactions, making their embeddings statistically unreliable. Content embeddings capture semantic similarity rather than collaborative utility and are limited by incomplete or inconsistent metadata: two semantically close items can occupy very different positions in collaborative space.

We therefore propose a *similarity-based* approach: rather than mapping a single cold item's content embedding directly to a collaborative embedding, we propagate collaborative signal from content-similar warm items. Aggregating collaborative signal from multiple content-similar warm items reduces the effect of noise in any individual embedding, and since cold-item predictions are formed from existing warm embeddings, they naturally remain in the inhabited region of collaborative space.

The proposed method operates as a plug-in on top of any recommender with content-initialized embeddings and admits a lightweight, exact analytical solution that scales linearly with catalog size, making it practical for large-scale deployments. While the method is general and applicable to any recommender that produces item embeddings, our experiments focus on *sequential recommenders*, which represent the current mainstream and pose a particularly challenging cold-start setting.

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.

2 Related Work

Transformer-based sequential recommenders (Kang & McAuley, 2018; Sun et al., 2019) dominate the field, yet cold-start solutions for this setting remain scarce. Recent work addresses this problem by modifying the base recommender's training: (Pembek et al., 2025) freezes content embeddings and learns a small trainable delta to adapt them to the recommendation task, while (Lichtenberg et al., 2025) learns a linear projection from the content embedding space into the ID embedding space. SimRec (Brody & Lagziel, 2024) and the language-model prior approach (Wang et al., 2024) are similarity-based like our method, exploiting content-derived item similarities via a training loss and a Bayesian regularizer, respectively, but neither provides a scalable solution. All four require modifying the underlying recommender, unlike our post-hoc plug-in approach.

Using content embeddings to enrich item representations has been explored beyond strict cold start: LLM-based initialization (Harte et al., 2023), cross-domain text grounding (Hou et al., 2022), and null-space fusion of language and collaborative signals (Hu et al., 2025) improve tail-item quality without targeting items with zero interactions. Classically, cold start was handled by mapping content to latent factors (Van den Oord et al., 2013), training with dropped collaborative inputs (Volkovs et al., 2017), or meta-learning (Lee et al., 2019; Dong et al., 2020); all require cold-start awareness during training.

3 Proposed approach

3.1 Notation

Let $\mathcal{I}$ denote the set of items, partitioned into disjoint sets of *warm* items $\mathcal{I}_w$ and *cold* items $\mathcal{I}_c$. Warm items have an accumulated interaction history; cold items do not. Each item $i \in \mathcal{I}$ is associated with a *content embedding* $\mathbf{c}_i \in \mathbb{R}^c$ generated by a fixed, pre-trained encoder $\phi\colon \mathcal{I} \to \mathbb{R}^c$ (e.g., a language model applied to the item description). We represent these as a content embedding matrix $\mathbf{C} \in \mathbb{R}^{c\times|\mathcal{I}|}$, with column-wise submatrices $\mathbf{C}_w \in \mathbb{R}^{c\times|\mathcal{I}_w|}$ and $\mathbf{C}_c \in \mathbb{R}^{c\times|\mathcal{I}_c|}$ for warm and cold items, respectively. A recommender trained on the interaction log provides collaborative embeddings for warm items, denoted by $\mathbf{E}_w \in \mathbb{R}^{l\times|\mathcal{I}_w|}$.

3.2 High-Level Overview

A natural family of approaches to the cold-start embedding problem relies on content similarity: predict the collaborative embedding of a cold item by aggregating the collaborative embeddings of warm items similar in content space. Representative instances include k-nearest-neighbor (k-NN) interpolation, which averages the embeddings of the k most content-similar warm items, and kernel regression, which weights all warm items by a kernel function of their content distance.

We now describe the proposed approach at a high level. Let $\mathbf{V} \in \mathbb{R}^{l\times|\mathcal{I}|}$ denote the matrix of *updated* collaborative embeddings for all items, with columns $\mathbf{v}_i$. We seek $\mathbf{V}$ that balances two competing objectives: (i) embeddings of content-similar items should be close to each other, and (ii) embeddings of warm items should remain close to their original collaborative embeddings $\mathbf{E}_w$. The updated embeddings can be viewed as prior embeddings enriched with content-based similarity information. Formally,

$$\mathbf{V}^* = \operatorname*{argmin}_{\mathbf{V}} \underbrace{\frac{1}{2}\sum_{i,j} \mathbf{W}_{ij}\,\|\mathbf{v}_i - \mathbf{v}_j\|_2^2}_{\text{smoothness}} + \underbrace{\|(\mathbf{V}-\mathbf{E})\mathbf{A}^{1/2}\|_F^2}_{\text{fidelity}}, \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ is a non-negative item-similarity matrix derived from content embeddings, $\mathbf{A}$ is a diagonal matrix of item-wise regularization coefficients, and $\mathbf{E} \in \mathbb{R}^{l\times|\mathcal{I}|}$ is a matrix of collaborative embeddings $\mathbf{E}_w \in \mathbb{R}^{l\times|\mathcal{I}_w|}$ augmented with zero columns corresponding to cold items. As a result, the fidelity term serves as an ℓ_2 regularizer for cold items; thus, it is reasonable for the regularization coefficients in $\mathbf{A}$ to differ between cold and warm items.

The smoothness term penalizes pairs of similar items, propagating collaborative signal from warm items to nearby cold items. Noting that $\frac{1}{2}\sum_{i,j}\mathbf{W}_{ij}\|\mathbf{v}_i-\mathbf{v}_j\|_2^2 = \operatorname{tr}(\mathbf{V}\mathbf{L}\mathbf{V}^\top)$, where $\mathbf{L} = \mathbf{\Delta}-\mathbf{W}$ is

the graph Laplacian matrix and $\mathbf{\Delta}_{ii} = \sum_j \mathbf{W}_{ij}$, the objective equation 1 can be written compactly as

$$\mathbf{V}^* = \underset{\mathbf{V}}{\operatorname{argmin}} \ \operatorname{tr}(\mathbf{V}\mathbf{L}\mathbf{V}^\top) + \|(\mathbf{V} - \mathbf{E})\mathbf{A}^{1/2}\|_F^2. \tag{2}$$

Equation equation 2 is an instance of *graph Laplacian regularization* (Belkin et al., 2006): the fidelity term anchors warm-item embeddings to their pretrained collaborative values, while the smoothness term propagates these signals along the content-similarity graph to cold items. This is directly analogous to *Label Propagation* (Zhu et al., 2003) with soft constraints (Zhou et al., 2003), a classical semi-supervised method that diffuses class labels from labeled to unlabeled points while keeping labeled outputs close to their original values; in our setting, collaborative embeddings play the role of labels and warm items play the role of labeled examples. Crucially, the method is *transductive*: it infers embeddings for all items jointly, so cold-item predictions are informed by one another through the graph, unlike k-NN or kernel regression, which predict each cold item independently. We name our method *Warmth Propagation* to reflect this analogy: collaborative information propagates from warm to cold items, like class labels from labeled to unlabeled ones in *Label Propagation*. However, we emphasize that our proposed method *is not* a special case of Label Propagation, as we show below.

Equation equation 2 raises two practical challenges. First, how should the similarity matrix $\mathbf{W}$ be constructed from the content embeddings $\mathbf{C}$? Second, how can we solve the optimization efficiently? With typical item catalogs of size 10^5–10^7, a naive approach that materializes the full $|\mathcal{I}| \times |\mathcal{I}|$ similarity matrix is infeasible. It is therefore essential to develop a method whose complexity scales linearly in $|\mathcal{I}|$. We address both questions by first considering an alternative objective based on Local Linear Embedding (LLE) and then relating it to Eq. equation 2.

3.3 Method

Our method draws inspiration from *Locally Linear Embedding* (LLE) (Roweis & Saul, 2000), a classic unsupervised dimensionality reduction technique. LLE operates in two stages: it first learns the local geometry of the high-dimensional input space by expressing each point as a linear combination of its neighbors, and then finds a low-dimensional embedding that preserves those learned reconstruction weights. We adopt the same two-stage philosophy, but apply it in a different direction: we learn geometry in the *content* embedding space and then update the *collaborative* embedding space to conform to it.

3.3.1 Step 1: Learning content geometry

We seek a matrix $\mathbf{B} \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{I}|}$ that describes the geometry of the content embedding space, i.e. expresses each item as a linear combination of other items in the content space. Formally,

$$\begin{aligned} \hat{\mathbf{B}} = \underset{\mathbf{B}}{\operatorname{argmin}} \ & \|\mathbf{C} - \mathbf{C}\mathbf{B}\|_F^2 + \lambda\|\mathbf{B}\|_F^2 \\ \text{s.t.} \quad & \operatorname{diag}\mathbf{B} = \mathbf{0}, \end{aligned} \tag{3}$$

where $\lambda > 0$ is a regularization coefficient and the diagonal constraint $\mathbf{B}_{ii} = 0$ prevents trivial self-reconstruction. The column $\mathbf{b}_i$ of $\hat{\mathbf{B}}$ gives the reconstruction weights for item i: item i's content embedding is approximated as $\mathbf{C}\mathbf{b}_i$, a weighted combination of all other items' content embeddings. One may recognize Eq. equation 3 as the EASE (Steck, 2019) objective applied to the content matrix $\mathbf{C}$ in place of interaction data.

3.3.2 Step 2: Transferring geometry to collaborative space

Given the learned geometry $\hat{\mathbf{B}}$, we update the collaborative embeddings $\mathbf{V} \in \mathbb{R}^{l \times |\mathcal{I}|}$ to satisfy two requirements: (i) the reconstruction relationships encoded in $\hat{\mathbf{B}}$ should hold in collaborative space as well, and (ii) the embeddings of warm items should remain close to their original collaborative embeddings. Formally,

$$\hat{\mathbf{V}} = \underset{\mathbf{V}}{\operatorname{argmin}} \ \underbrace{\|\mathbf{V} - \mathbf{V}\hat{\mathbf{B}}\|_F^2}_{\text{geometry transfer}} + \underbrace{\|(\mathbf{V} - \mathbf{E})\mathbf{A}^{1/2}\|_F^2}_{\text{fidelity}}, \tag{4}$$

The geometry-transfer term penalizes any item i whose collaborative embedding $\mathbf{v}_i$ cannot be reconstructed from other items' embeddings using the same weights $\mathbf{b}_i$ learned in Step 1. By forcing the collaborative space to respect the linear geometry of the content space, warm collaborative embeddings are propagated to cold items along the content-similarity structure.

Our approach is related to *ImplicitSLIM* (Shenbin & Nikolenko, 2024), a scalable extension of LLE that learns item embeddings from user–item interaction matrices in time and memory linear in the number of users and items. Like our approach, *ImplicitSLIM* refines collaborative embeddings; however, it derives item–item similarities from the user-item interaction matrix instead of utilizing content embeddings.

ImplicitSLIM achieves scalability through a series of relaxations of the original LLE objective: it drops certain optimization constraints that are theoretically desirable but computationally prohibitive. We follow the same motivation: by adopting the same relaxations, we are able to avoid materializing $\hat{\mathbf{B}} \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$ entirely and instead work with quantities whose size is at most $O(c \times |\mathcal{I}|)$. In our setting, the problem structure is simpler than in ImplicitSLIM, allowing us to obtain a solution that is not only scalable but also exact. We therefore adopted the objective functions of ImplicitSLIM but, taking into account the specifics of our problem, modified them slightly for simplicity and naturalness. This minor modification enabled us to derive a closed-form solution in an entirely different way, without relying on any approximations. We derive this solution in the next section.

One can observe that this optimization problem closely resembles Eq. equation 1 and equation 2. Concretely, as shown in Section A.1, optimization problem from step 1 can be rewritten as

$$\begin{aligned}\mathbf{V}^* = \underset{\mathbf{V}}{\operatorname{argmin}} \frac{1}{2} \sum_{i,j} (-\mathbf{K}_{ij}) \, \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 + \\ + \sum_i d_i \|\mathbf{v}_i\|_2^2 + \\ + \|(\mathbf{V} - \mathbf{E})\mathbf{A}^{1/2}\|_F^2,\end{aligned} \tag{5}$$

here we defined

$$\mathbf{K} := (\mathbf{I} - \hat{\mathbf{B}})(\mathbf{I} - \hat{\mathbf{B}})^\top \tag{6}$$

and $d_i := \sum_j \mathbf{K}_{ij}$.

Since $\mathbf{K}_{ij}$ can be both positive and negative, in the proposed approach, the embeddings of items $\mathbf{v}_i$ and $\mathbf{v}_j$ are pulled toward each other, as in Label Propagation, when $\mathbf{K}_{ij} < 0$, whereas they are pushed away from each other when $\mathbf{K}_{ij} > 0$. We also highlight that the additional ℓ_2 regularization term with item-specific coefficients is another inconsistency with Label Propagation.

3.4 Scalable learning

In this section, we first provide closed-form solutions for both Steps 1 and 2, and then propose a way to combine them into a single scalable solution. The optimization task defined in equation 3 as *Step 1* was previously considered as an item-item collaborative filtering model in (Steck, 2019). In that paper, we derived a closed-form solution; we adapt it here to our notation.

Its closed-form solution is

$$\hat{\mathbf{B}} = \mathbf{I} - \mathbf{P}\mathbf{D}^{-1}. \tag{7}$$

Here, we define:

$$\begin{aligned}\mathbf{P} &:= \left(\mathbf{C}^\top \mathbf{C} + \lambda \mathbf{I}\right)^{-1} \\ \mathbf{D} &:= \operatorname{diagMat}\left(\operatorname{diag}\left(\mathbf{P}\right)\right),\end{aligned} \tag{8}$$

where $\operatorname{diagMat}(\mathbf{x})$ is a diagonal matrix such that its diagonal is $\mathbf{x}$. Since the direct evaluation of equation 7 requires inverting an $|\mathcal{I}| \times |\mathcal{I}|$ matrix, it is computationally prohibitive for large-scale datasets and should be bypassed in favor of a more scalable approach.

In Section A.2, we derive a closed-form solution for equation 4:

$$\hat{\mathbf{V}} = \mathbf{E}\mathbf{A}(\mathbf{K} + \mathbf{A})^{-1}. \tag{9}$$

Again, the solution requires inverting a matrix in $\mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$, presenting a similar computational challenge.

By leveraging these closed-form solutions, we can substitute equation 7 into equation 9. This lets us rearrange the resulting expression so the successive matrix inversions cancel, bypassing the need to invert the high-dimensional matrix.

We provide the detailed derivations in Section A.3. The final result is:

$$\begin{aligned}\hat{\mathbf{V}} &= \mathbf{E} - \mathbf{E}\hat{\mathbf{D}}^{-1}\mathbf{A}^{-1} + \\ &+ \mathbf{E}\left[\hat{\mathbf{D}}^{-1}\mathbf{U}^\top\left(\mathbf{Z}^{-1} + \mathbf{U}\hat{\mathbf{D}}^{-1}\mathbf{U}^\top\right)^{-1}\mathbf{U}\hat{\mathbf{D}}^{-1}\right]\mathbf{A}^{-1}.\end{aligned} \tag{10}$$

where

$$\hat{\mathbf{D}} = \mathbf{A}^{-1} + \lambda^2\mathbf{D}^2, \tag{11}$$

$$\mathbf{D} = \frac{1}{\lambda}\mathrm{diagMat}\left(\mathbf{1} - \left[(\lambda\mathbf{I} + \mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}\odot\mathbf{C}\right]^\top\mathbf{1}\right), \tag{12}$$

$$\mathbf{Z}^{-1} = \begin{bmatrix}\mathbf{0} & \frac{1}{\lambda}\mathbf{I} \\ \frac{1}{\lambda}\mathbf{I} & -\frac{1}{\lambda^2}\mathbf{C}\mathbf{D}^2\mathbf{C}^\top\end{bmatrix}, \tag{13}$$

$$\mathbf{U} = \begin{bmatrix}\mathbf{C} \\ \mathbf{C}\mathbf{D}^2\end{bmatrix}. \tag{14}$$

3.5 Complexity analysis.

All matrices to be inverted in the final solution have size $2c \times 2c$, independent of $|\mathcal{I}|$. The remaining operations are matrix-vector products, each linear in $|\mathcal{I}|$, giving overall time and memory $O\big((c^2 + lc)\,|\mathcal{I}|\big)$ with no $|\mathcal{I}| \times |\mathcal{I}|$ matrix ever formed. Since c and d are small fixed constants, the method adds negligible overhead to any pretrained recommender.

4 Experimental Study

4.1 Datasets

We evaluate on three datasets; statistics are reported in Table 1. Amazon-M2 FR (Jin et al., 2023) is the French-language split of the Amazon multilingual shopping session dataset. Beauty [1] is a widely used benchmark of Amazon product reviews in the Beauty category. Zvuk [2] is a music streaming dataset. We split all datasets by global time; for preprocessing details, see (Pembek et al., 2025).

Table 1: Dataset statistics after preprocessing. "Cold in GT" is the fraction of ground-truth items that are cold.

Dataset	# users	# items	# clicks	Cold in GT
Amazon-M2	129,983	44,049	566k	7%
Beauty	21,029	11,733	149k	25%
Zvuk	9,076	131,085	3,23m	13%

4.2 Baselines

We include baselines from (Pembek et al., 2025): *content initialization*, *content-based kNN*, and *delta embeddings* (referred to as *t.d.* in the original work), the latter of which adds a small learned delta to frozen content embeddings.

We additionally introduce three *mapper* baselines that learn a direct parametric map from content to collaborative embeddings: *linear regression*, *kernel regression*, and a *multi-layer perceptron* (MLP). The MLP is a 3-layer network with skip connections, layer normalization, and input

[1] https://amazon-reviews-2023.github.io/

[2] https://www.kaggle.com/datasets/alexxl/zvuk-dataset

Table 2: NDCG@10 on cold-item ground truth (Cold) and all items (Total). All methods use SASRec (Kang & McAuley, 2018) as the backbone recommender.

	Amazon-M2		Beauty		Zvuk	
Method	Cold	Total	Cold	Total	Cold	Total
Baseline	0.000	0.407	0.000	0.033	0.000	0.055
Content-based kNN	0.312	0.238	0.022	0.024	0.006	0.001
Content initialization	0.297	0.438	0.018	0.044	0.014	0.052
Delta embeddings	0.359	**0.440**	0.022	0.046	0.021	0.055
Linear regression mapper	0.349	0.307	0.021	0.041	0.016	0.045
Approximate kernel regression mapper	0.356	0.309	0.023	0.043	0.017	0.052
MLP mapper	0.349	0.307	0.023	0.042	0.016	0.049
Our approach	**0.380**	**0.440**	**0.058**	**0.055**	**0.023**	**0.056**

dropout. As a scalable similarity-based baseline, we include *Falkon* [3], a kernel regression method using the Nyström approximation that scales to large item catalogs. In subsection 5.3 we also use the `sklearn` [4] implementation of exact, non-scalable kernel regression. See subsection B.1 for a more detailed description of both the exact and approximate versions of kernel regression.

4.3 Implementation Details

We use SASRec (Kang & McAuley, 2018), the de facto industry standard for sequential recommendation, and BERT4Rec (Sun et al., 2019), a widely adopted bidirectional alternative, as backbone recommenders, and initialize their item embeddings with content embeddings in all experiments.

The diagonal matrix $\mathbf{A} = \mathrm{diagMat}\left(\left[a_1, \ldots, a_{|\mathcal{I}|}\right]\right)$ contains individual regularization coefficients, where each a_i is defined as:

$$a_i = \begin{cases} a_{\mathrm{cold}}, & i \in \mathcal{I}_c, \\ a_{\mathrm{warm}}, & i \in \mathcal{I}_w. \end{cases}$$

In total, *Warmth Propagation* has three hyperparameters: λ, a_{cold}, and a_{warm}. To tune these parameters, we used the *hyperopt* library (Bergstra et al., 2013). We maximize the following target value during hyperparameter optimization:

$$\text{target} = \text{cold_NDCG} \times \min\left(1, \left(\frac{\text{total_NDCG}}{\text{threshold}}\right)^{10}\right),$$

where *threshold* is the minimum acceptable total NDCG score, i.e., the score of the model we aim to outperform. In our case, this is the strong *delta embeddings* baseline. This function prioritizes cold-item performance while penalizing solutions that degrade overall performance below a certain threshold. The soft constraint in the second term is intended to provide a smoother landscape for the optimizer.

We tuned the hyperparameters on the validation set and evaluated the best-performing model on the test set, searching each hyperparameter in the range $[10^{-3}, 10^{3}]$.

While *Warmth Propagation* is a standalone approach applicable to various pre-trained models, it requires that the model's embedding matrices be initialized with content embeddings. Without this initialization, cold-item performance remains near zero for both our method and the baselines.

Table 3: Additional results (NDCG@10) with the non-scalable version of kernel regression.

	Beauty	
Method	Cold	Total
Content-based kNN	0.022	0.024
Approximate kernel regression mapper	0.023	0.043
Kernel regression mapper (cold only)	0.032	0.052
Kernel regression mapper (cold & warm)	0.057	0.051
Warmth Propagation (cold only)	0.015	0.046
Warmth Propagation (cold & warm)	**0.058**	**0.055**

5 RESULTS

5.1 GENERAL RESULTS

Tables 2 and 4 reports the NDCG@10 results for the SASRec and BERT4Rec backbones, respectively. The proposed approach outperforms all baselines on cold-item performance while requiring no changes to the backbone's training procedure. Mapper baselines improve cold-item performance but at the cost of warm-item quality, yielding poor total NDCG@10. Kernel regression's relatively low performance may be attributed to the approximations used. The proposed approach shows its strongest results on the Beauty dataset; interestingly, content-based kNN performs unexpectedly well on this dataset, which could point to the dominance of similarity-based models in this domain.

5.2 TRADE-OFF BETWEEN TOTAL AND COLD-ITEM PERFORMANCE

Figure 1 illustrates the cold–total performance trade-off. The proposed method forms the Pareto front on the total NDCG@10 vs. cold NDCG@10 plot: it achieves substantially better cold-item recommendations with only a small drop in total performance.

5.3 EMBEDDING CONSISTENCY

We conduct additional experiments to identify the critical factors for effective content-to-collaborative mappers and to understand why *Warmth Propagation* succeeds in this role while standard approaches, such as linear regression, approximate kernel regression, and multi-layer perceptrons, struggle.

We introduce an additional baseline: exact kernel regression (as opposed to the approximate variant). Exact kernel regression is generally not scalable, as training requires inverting a $|\mathcal{I}| \times |\mathcal{I}|$ matrix. Nevertheless, we can evaluate it on the Beauty dataset, which has the smallest number of items in our experimental setup.

Table 3 presents these additional results. We conduct experiments in a *cold-only* setting, where we update the collaborative embeddings exclusively for cold items, and in the *cold & warm* setting, where we update both cold and warm collaborative embeddings.

Two mappers, Warmth Propagation and Kernel Regression, perform significantly better than the others. These two mappers are also the only ones in our experiments that rely on collaborative embeddings during inference, though in different ways. Indeed, the resulting embeddings in Warmth Propagation are the sum of the prior embeddings and their transformed versions, as shown in Equation 30. For Kernel Regression, we discuss this in subsection B.1. This feature seems to be important for content-to-collaborative mappers since, as we noted earlier, it is important for cold and warm items by the same model, and relying on training data during inference makes it possible to

[3] https://github.com/falkonml/falkon

[4] https://scikit-learn.org/stable/modules/generated/sklearn.kernel_ridge.KernelRidge.html

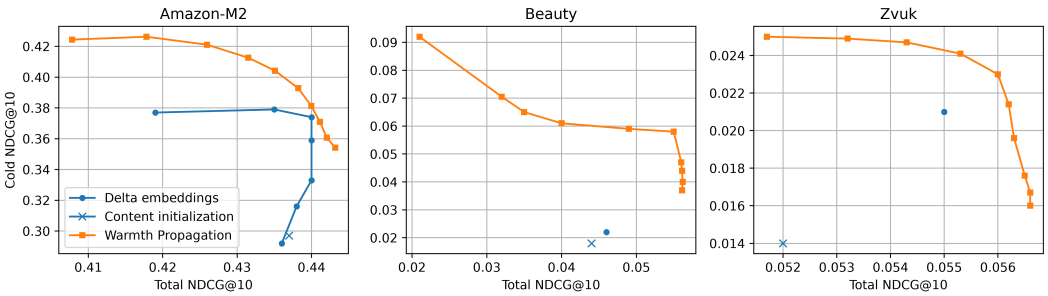


Figure 1: Total NDCG@10 vs. Cold NDCG@10 trade-off. Each point corresponds to a different value of the controlling parameter: δ for delta embeddings (Pembek et al., 2025) (available only for Amazon-M2 in the original work), and the warm-item diagonal entries of $\mathbf{A}$ for the proposed method.

obtain updated collaborative embeddings of warm items that are close to prior embeddings and have the same bias caused by the mapper as cold items.

We observe that two mappers, Warmth Propagation and Kernel Regression, significantly outperform the others. Furthermore, these are the only methods in our evaluation that leverage collaborative embeddings during inference, albeit through different mechanisms. This property appears crucial for content-to-collaborative mapping. As noted earlier, the downstream model must process both cold and warm items seamlessly; by relying on training data at inference, the mapper yields updated warm embeddings that remain close to their prior embeddings while sharing the exact same mapper-induced bias as the cold items.

Overall, both Warmth Propagation and exact Kernel Regression are highly effective as cold-start mappers, but Warmth Propagation is also computationally efficient. Meanwhile, utilizing an efficient approximation of Kernel Regression leads to significantly worse performance.

5.4 Negative Results

5.4.1 Popularity-aware Reweighting

Intuitively, items that have only a few interactions in their logs have noisier collaborative embeddings. Following this intuition, we performed an experiment with a more complex setting for the regularization coefficients in $\mathbf{A}$. Denoting $n(i)$ as the number of interactions for the i-th item, we proposed the following rule: $a_i > a_j$ if $n(i) > n(j)$. Specifically, we redefined a_i for $i \in \mathcal{I}_w$ as follows:

$$a_i = a_{\text{warm}} n(i)^{\alpha},$$

where α is a hyperparameter. Experiments show that the best results are achieved when α is close to zero, meaning popularity-aware reweighting does not improve the proposed method's performance.

5.4.2 Content-aware Embedding Updates During Training

Warmth Propagation does not take user-item interactions into account, unlike many other approaches for cold-start mitigation. A simple way to address this is to update the embeddings periodically (e.g., every few epochs) while the backbone model trains. This technique was successfully used in (Shenbin & Nikolenko, 2024). In this case, item embeddings are alternately updated with content and collaborative information. Nevertheless, we did not observe any performance improvement using this more complex approach.

6 Conclusion

We present *Warmth Propagation*, a post-hoc plug-in for cold-start recommendation that leaves the backbone model unchanged and requires only that item embeddings be initialized from content

features. The method propagates collaborative embeddings from warm items to cold ones along a content-similarity graph, similar to Label Propagation. We derive an exact closed-form solution whose time and memory complexity scale linearly in catalog size, with no $|\mathcal{I}| \times |\mathcal{I}|$ matrix ever formed. Experiments on three datasets show that Warmth Propagation outperforms all cold-start baselines, including approaches that require editing the backbone model, while preserving warm-item performance and adding negligible runtime overhead.

Several directions remain open, such as incorporating user-item interactions and learning an end-to-end differentiable version that jointly optimizes the content encoder and the propagation objective.

AI USE STATEMENT

Large language models were used for early drafting, text polishing, grammar correction, and checking derivations.

REFERENCES

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7 (11), 2006.

James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pp. 115–123. PMLR, 2013.

Shaked Brody and Shoval Lagziel. Simrec: Mitigating the cold-start problem in sequential recommendation by integrating item similarity. *arXiv preprint arXiv:2410.22136*, 2024.

Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming Zhu. Mamo: Memory-augmented meta-optimization for cold-start recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 688–697, 2020.

Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. Leveraging large language models for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 1096–1102, 2023.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 585–593, 2022.

Guoqing Hu, An Zhang, Shuo Liu, Zhibo Cai, Xun Yang, and Xiang Wang. Alphafuse: Learn id embeddings for sequential recommendation in null space of language embeddings. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1614–1623, 2025.

Wei Jin, Haitao Mao, Zheng Li, Haoming Jiang, Chen Luo, Hongzhi Wen, Haoyu Han, Hanqing Lu, Zhengyang Wang, Ruirui Li, Zhen Li, Monica Xiao Cheng, Rahul Goutam, Haiyang Zhang, Karthik Subbian, Suhang Wang, Yizhou Sun, Jiliang Tang, Bing Yin, and Xianfeng Tang. Amazon-m2: A multilingual multi-locale shopping session dataset for recommendation and text generation, 2023. URL https://arxiv.org/abs/2307.09688.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1073–1082, 2019.

Jan Malte Lichtenberg, Antonio De Candia, and Matteo Ruffini. Denserec: Revisiting dense content embeddings for sequential transformer-based recommendation. *arXiv preprint arXiv:2508.18442*, 2025.

Anton Pembek, Artem Fatkulin, Anton Klenitskiy, and Alexey Vasilev. Let it go? not quite: Addressing item cold start in sequential recommendations with content-based initialization. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 626–631, 2025.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

Ilya Shenbin and Sergey Nikolenko. ImplicitSLIM and how it improves embedding-based collaborative filtering. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=6vF0ZJGor4.

Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, pp. 3251–3257, 2019.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441–1450, 2019.

Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. *Advances in neural information processing systems*, 26, 2013.

Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. Dropoutnet: Addressing cold start in recommender systems. *Advances in neural information processing systems*, 30, 2017.

Shiyu Wang, Hao Ding, Yupeng Gu, Sergul Aydore, Kousha Kalantari, and Branislav Kveton. Language-model prior overcomes cold-start items. *arXiv preprint arXiv:2411.09065*, 2024.

Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.

Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pp. 912–919, 2003.

A Derivations

A.1 Decomposition of the Geometry-Transfer Term

The geometry-transfer term rewrites as

$$\|\mathbf{V} - \mathbf{V}\hat{\mathbf{B}}\|_F^2 = \|\mathbf{V}(\mathbf{I} - \hat{\mathbf{B}})\|_F^2 = \operatorname{tr}\left(\mathbf{V}\mathbf{K}\mathbf{V}^\top\right), \tag{15}$$

where we denote $\mathbf{K} := (\mathbf{I} - \hat{\mathbf{B}})(\mathbf{I} - \hat{\mathbf{B}})^\top \succeq 0$. Using $\mathbf{v}_i^\top \mathbf{v}_j = \frac{1}{2}(\|\mathbf{v}_i\|^2 + \|\mathbf{v}_j\|^2 - \|\mathbf{v}_i - \mathbf{v}_j\|^2)$:

$$\begin{aligned}
&\operatorname{tr}\left(\mathbf{V}\mathbf{K}\mathbf{V}^\top\right) = \qquad (16)\\
&= \sum_{i,j} \mathbf{K}_{ij}\, \mathbf{v}_i^\top \mathbf{v}_j \\
&= \frac{1}{2}\sum_{i,j} \mathbf{K}_{ij}\big(\|\mathbf{v}_i\|^2 + \|\mathbf{v}_j\|^2\big) - \frac{1}{2}\sum_{i,j} \mathbf{K}_{ij}\|\mathbf{v}_i - \mathbf{v}_j\|^2 \\
&= \sum_i d_i \|\mathbf{v}_i\|^2 + \frac{1}{2}\sum_{i \neq j} (-\mathbf{K}_{ij})\|\mathbf{v}_i - \mathbf{v}_j\|^2, \qquad (17)
\end{aligned}$$

where $d_i = \sum_j \mathbf{K}_{ij}$.

A.2 Analytical Solution for $\hat{\mathbf{V}}$

The objective for the matrix $\mathbf{V}$ is:

$$\mathcal{L}_{\text{step 2}}(\mathbf{V}) = \|\mathbf{V} - \mathbf{V}\hat{\mathbf{B}}\|_F^2 + \|(\mathbf{V} - \mathbf{E})\mathbf{A}^{1/2}\|_F^2 \tag{18}$$

The optimality condition $\nabla_{\mathbf{V}}\mathcal{L}_{\text{step 2}} = \mathbf{0}$ yields:

$$\begin{aligned} \mathbf{V}(\mathbf{K} + \mathbf{A}) = \mathbf{E}\mathbf{A} \implies \\ \implies \hat{\mathbf{V}} = \mathbf{E}\mathbf{A}(\mathbf{K} + \mathbf{A})^{-1}. \end{aligned} \tag{19}$$

A.3 Derivation of scalable solution

We begin with the closed-form solution from Step 2. By rewriting using the Woodbury matrix identity, we obtain:

$$(\mathbf{K} + \mathbf{A})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}(\mathbf{A}^{-1} + \mathbf{K}^{-1})^{-1}\mathbf{A}^{-1} \tag{20}$$

Substituting this back into the closed-form solution for Step 2, we have:

$$\hat{\mathbf{V}} = \mathbf{E} - \mathbf{E}(\mathbf{A}^{-1} + \mathbf{K}^{-1})^{-1}\mathbf{A}^{-1}. \tag{21}$$

Because $\mathbf{A}$ is diagonal, $\mathbf{A}^{-1}$ is computed trivially. We then apply the Woodbury matrix identity to $(\mathbf{A}^{-1} + \mathbf{K}^{-1})^{-1}$; to facilitate this, we decompose $\mathbf{A}^{-1} + \mathbf{K}^{-1}$ into the form $\hat{\mathbf{D}} + \mathbf{U}^\top\mathbf{Z}\mathbf{U}$, where $\hat{\mathbf{D}}$ and $\mathbf{Z}$ are constructed to be easily invertible.

A.3.1 Low-Rank Decomposition of $\mathbf{K}^{-1}$

We begin by substituting the closed-form solution from Step 1 into the expression $\mathbf{K} = (\mathbf{I} - \hat{\mathbf{B}})(\mathbf{I} - \hat{\mathbf{B}})^\top$:

$$\mathbf{K} = (\mathbf{I} - \hat{\mathbf{B}})(\mathbf{I} - \hat{\mathbf{B}})^\top = \mathbf{P}\mathbf{D}^{-2}\mathbf{P}. \tag{22}$$

Using $\mathbf{P}^{-1} = \mathbf{C}^\top\mathbf{C} + \lambda\mathbf{I}$, we expand $\mathbf{K}^{-1}$:

$$\begin{aligned} &\mathbf{K}^{-1} = \\ =&\mathbf{P}^{-1}\mathbf{D}^2\mathbf{P}^{-1} = \\ =&\lambda^2\mathbf{D}^2 + \mathbf{C}^\top(\mathbf{C}\mathbf{D}^2\mathbf{C}^\top)\mathbf{C} + \lambda\mathbf{C}^\top(\mathbf{C}\mathbf{D}^2) + \lambda(\mathbf{D}^2\mathbf{C}^\top)\mathbf{C}. \end{aligned} \tag{23}$$

We then express $\mathbf{K}^{-1}$ as a diagonal matrix plus a product of block matrices:

$$\mathbf{K}^{-1} = \lambda^2\mathbf{D}^2 + \underbrace{\begin{bmatrix}\mathbf{C}^\top & \mathbf{D}^2\mathbf{C}^\top\end{bmatrix}}_{\mathbf{U}^\top \in \mathbb{R}^{|\mathcal{I}| \times 2c}} \underbrace{\begin{bmatrix}\mathbf{C}\mathbf{D}^2\mathbf{C}^\top & \lambda\mathbf{I} \\ \lambda\mathbf{I} & \mathbf{0}\end{bmatrix}}_{\mathbf{Z} \in \mathbb{R}^{2c \times 2c}} \underbrace{\begin{bmatrix}\mathbf{C} \\ \mathbf{C}\mathbf{D}^2\end{bmatrix}}_{\mathbf{U} \in \mathbb{R}^{2c \times |\mathcal{I}|}} \tag{24}$$

A.3.2 Efficient computation of $\mathbf{D}$

The only missing component for the efficient computation of $\mathbf{K}^{-1}$ is $\mathbf{D}$. Since $\mathbf{C} \in \mathbb{R}^{c \times |\mathcal{I}|}$, we can compute $\mathbf{P}$ efficiently using the Woodbury matrix identity:

$$\mathbf{P} = \frac{1}{\lambda}\left(\mathbf{I} - \mathbf{C}^\top(\lambda\mathbf{I} + \mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C}\right). \tag{25}$$

This allows us to compute $\mathbf{D} = \text{diagMat}\,(\mathbf{P})$ efficiently without explicitly materializing $\mathbf{P}$:

$$\text{diag}\,(\mathbf{P}) = \frac{1}{\lambda}\left(\mathbf{1} - \left[(\lambda\mathbf{I} + \mathbf{C}\mathbf{C}^\top)^{-1}\mathbf{C} \odot \mathbf{C}\right]^\top \mathbf{1}\right). \tag{26}$$

A.3.3 Final steps

As a result of the preceding derivations, we can write:

$$\mathbf{A}^{-1} + \mathbf{K}^{-1} = (\mathbf{A}^{-1} + \lambda^2\mathbf{D}^2) + \mathbf{U}^\top\mathbf{Z}\mathbf{U}, \tag{27}$$

where

$$\hat{\mathbf{D}} = \mathbf{A}^{-1} + \lambda^2\mathbf{D}^2, \tag{28}$$

is a diagonal matrix. Applying the Woodbury identity to the term $(\mathbf{A}^{-1} + \mathbf{K}^{-1})^{-1}$ yields:

$$\begin{aligned}&(\mathbf{A}^{-1} + \mathbf{K}^{-1})^{-1} = \\ =&(\hat{\mathbf{D}} + \mathbf{U}^\top \mathbf{Z}\mathbf{U})^{-1} = \\ =&\hat{\mathbf{D}}^{-1} - \hat{\mathbf{D}}^{-1}\mathbf{U}^\top \left(\mathbf{Z}^{-1} + \mathbf{U}\hat{\mathbf{D}}^{-1}\mathbf{U}^\top\right)^{-1} \mathbf{U}\hat{\mathbf{D}}^{-1}.\end{aligned} \tag{29}$$

Here, the inversion of $\mathbf{Z}$ is computed analytically using the block-matrix inversion formula:

$$\mathbf{Z}^{-1} = \begin{bmatrix} \mathbf{0} & \frac{1}{\lambda}\mathbf{I} \\ \frac{1}{\lambda}\mathbf{I} & -\frac{1}{\lambda^2}\mathbf{C}\mathbf{D}^2\mathbf{C}^\top \end{bmatrix}. \tag{30}$$

Substituting this result into equation 21, we have:

$$\hat{\mathbf{V}} = \mathbf{E} - \mathbf{E}\left[\hat{\mathbf{D}}^{-1} - \hat{\mathbf{D}}^{-1}\mathbf{U}^\top \left(\mathbf{Z}^{-1} + \mathbf{U}\hat{\mathbf{D}}^{-1}\mathbf{U}^\top\right)^{-1} \mathbf{U}\hat{\mathbf{D}}^{-1}\right]\mathbf{A}^{-1}. \tag{31}$$

Expanding the terms, we avoid the materialization of large matrices and obtain the final solution:

$$\begin{aligned}\hat{\mathbf{V}} = &\mathbf{E} - \mathbf{E}\hat{\mathbf{D}}^{-1}\mathbf{A}^{-1} + \\ &+ \mathbf{E}\left[\hat{\mathbf{D}}^{-1}\mathbf{U}^\top \left(\mathbf{Z}^{-1} + \mathbf{U}\hat{\mathbf{D}}^{-1}\mathbf{U}^\top\right)^{-1} \mathbf{U}\hat{\mathbf{D}}^{-1}\right]\mathbf{A}^{-1}.\end{aligned} \tag{32}$$

B Additional Details on Baselines

B.1 Kernel Regression

Here we consider one of our mapper baselines, kernel regression, and highlight its properties that we consider important in the context of the cold start problem. Below, $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are training data samples, $y_1, \ldots, y_n$ are the corresponding labels, $K(\cdot, \cdot)$ is some kernel function, $\mathbf{K}$ is the kernel matrix whose (i, j)-th element describes the similarity between the i-th and j-th data samples, $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{x}_*$ is the data sample we want to make a prediction for. The decision function of kernel regression is as follows

$$\hat{y}_* = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_*, \mathbf{x}_i), \tag{33}$$

and the closed-form solution for the weights $\boldsymbol{\alpha}$ is

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}, \tag{34}$$

where λ is the regularization coefficient and $\mathbf{y}$ is the vector of data labels. To avoid the inversion of an $n \times n$ matrix, we use the Nyström approximation in our experiments, where $\mathbf{K}$ is approximated in a low-rank form as follows:

$$\hat{\mathbf{K}} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{\dagger}\mathbf{K}_{nm}^\top, \tag{35}$$

where $\mathbf{K}_{mm}$ is the kernel matrix built from m inducing samples, such that $m << n$, $\mathbf{K}_{mm}^{\dagger}$ is the pseudoinverse of $\mathbf{K}_{mm}$, and $\mathbf{K}_{nm}$ is the kernel matrix between the training data samples and the inducing samples.

Now we consider kernel regression prediction for the r-th training data sample. First, we note that the vector $[K(\mathbf{x}_r, \mathbf{x}_1), \ldots, K(\mathbf{x}_r, \mathbf{x}_n)]$ is the r-th row of $\mathbf{K}$; as a result,

$$\hat{y}_r = \mathbf{e}_r^\top \mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{y}, \tag{36}$$

where $\mathbf{e}_r$ is the vector of zeros with a one at the r-th coordinate. If $\lambda = 0$ and $\mathbf{K}$ is non-singular, then $\hat{y}_r = y_r$. As a result, kernel regression in its pure form is able to reproduce labels from the training data, unlike the version with the Nyström approximation, where $\hat{\mathbf{K}}$ is low-rank by design.

C ADDITIONAL RESULTS

Table 4: Additional results (NDCG@10) with BERT4Rec as the backbone.

	Amazon-M2		Beauty		Zvuk	
Method	Cold	Total	Cold	Total	Cold	Total
Baseline	0.000	0.358	0.000	0.031	0.000	0.040
Content initialization	0.287	0.379	0.016	0.041	0.008	0.042
Delta embeddings	0.333	0.380	0.018	0.043	0.012	0.043
Linear regression mapper	0.334	0.284	0.020	0.035	0.012	0.039
Our approach	**0.350**	**0.385**	**0.023**	**0.045**	**0.014**	**0.044**