Geometric Residual Transfer with Similarity Gating for Cold-Start Sequential Recommendation

Alaa Miari
University of Haifa
Israel
alaamiari206@gmail.com

Aml Frehat
University of Haifa
Israel
2mal321@gmail.com

Abstract

Sequential recommendation models perform well when items have sufficient interaction history, but their performance degrades significantly for cold items that appear with limited training data. This issue is especially critical in dynamic real-world systems where new items are continuously introduced.

We study a practical cold-start enhancement for a SASRec-style sequential recommender trained with pretrained item embeddings and trainable residual (delta) embeddings. While pretrained vectors provide meaningful initialization, task-specific delta parameters for cold items remain undertrained.

To address this limitation, we introduce a KNN-based delta transfer mechanism that retrieves top-K warm neighbors in pretrained embedding space and transfers their learned delta representations. To mitigate negative transfer from weakly similar neighbors, we propose a similarity-based gating function that adaptively interpolates between transferred and original deltas.

We evaluate our approach on Beauty and Amazon_m2_fr across five random seeds (42–46), reporting NDCG@10 and Recall@10 with explicit cold/warm splits. On Beauty, cold NDCG@10 improves from 0.00572 ± 0.00058 to 0.00710 ± 0.00116, and cold Recall@10 improves from 0.01200 ± 0.00127 to 0.01536 ± 0.00132 at K=10, while warm performance remains stable. Similar trends are observed on Amazon_m2_fr.

Overall, similarity-gated KNN delta transfer provides a simple yet effective improvement for cold-start sequential recommendation with minimal architectural overhead.

Keywords

Sequential Recommendation, Cold-Start, kNN Retrieval, Gating, Delta Embeddings

1 Introduction

Sequential recommendation predicts the next item a user will interact with based on historical behavior. Transformer-based models such as SASRec achieve strong performance by modeling long-range dependencies through self-attention. Despite their effectiveness, item cold-start remains a fundamental challenge.

Cold items have limited interaction data during training, leading to unreliable task-specific parameters. Since item embeddings are used both within the sequence encoder and in final ranking, poorly estimated embeddings directly degrade recommendation quality.

A common practical solution is to leverage pretrained item embeddings derived from content or large-scale pretraining. These embeddings provide meaningful semantic structure even for infrequent items. However, pretrained vectors are not fully aligned with the downstream sequential objective.

Many systems therefore learn a residual adaptation, or *delta embedding*, on top of the pretrained representation. While this improves overall alignment, the delta component for cold items remains weak due to insufficient supervision.

We propose a lightweight enhancement that transfers learned delta information from warm items to cold items using KNN retrieval in pretrained embedding space. The intuition is that embedding proximity reflects semantic similarity; therefore, nearby warm items can provide useful adaptation signals. To prevent negative transfer from weak neighbors, we introduce a similarity-based gating mechanism that dynamically controls transfer strength.

Contributions.

- We introduce a KNN-based delta transfer mechanism for cold items.
- We design a similarity-based gate that adaptively blends transferred and original deltas.
- We conduct controlled experiments across five seeds with cold/warm evaluation splits.
- We demonstrate consistent cold-item improvements without degrading warm-item performance.

2 Background and Related Work

2.1 Sequential Recommendation

Sequential recommendation models predict future interactions from ordered user histories. Early methods relied on Markov chains or recurrent neural networks. Transformer-based architectures such as SASRec replace recurrence with self-attention, enabling improved modeling of long-range dependencies and efficient parallel training.

In these models, item embeddings are central components used both during sequence encoding and in final ranking. Their quality directly impacts performance.

2.2 Implicit Feedback Learning

Most recommendation settings rely on implicit feedback, where interactions indicate positive preference but non-interactions are ambiguous. Common objectives include sampled cross-entropy and pairwise ranking losses such as Bayesian Personalized Ranking (BPR). Under implicit supervision, accurate item representations are particularly important for effective ranking.

2.3 Cold-Start Recommendation

Item cold-start arises when new or infrequent items lack sufficient training interactions. Approaches to mitigate this problem include incorporating side information, hybrid collaborative–content models, and pretrained representations. While pretrained embeddings

improve initialization, they do not fully solve task-specific adaptation.

2.4 Neighborhood-Based Transfer and Gating

Neighborhood-based methods share information among similar items. In modern systems, similarity is often computed directly in embedding space. KNN-based parameter transfer can be viewed as smoothing item-specific parameters according to embedding proximity.

However, naive transfer may cause negative effects when similarity is weak. Confidence-weighted blending mechanisms, such as gating functions, provide a principled way to control transfer strength. Our method adopts this idea by computing a gate based on maximum cosine similarity to warm neighbors.

3 Problem Setup

Let $\mathcal{U}$ denote the set of users and $\mathcal{I}$ the set of items. Each user $u \in \mathcal{U}$ is associated with an ordered interaction sequence

$$\mathbf{s}_u = (i_1, i_2, \ldots, i_T), \quad i_t \in \mathcal{I}.$$

The task of sequential recommendation is next-item prediction: given the prefix $(i_1, \ldots, i_{T-1})$, the model predicts the next item i_T.

A sequential recommender learns an embedding function

$$E : \mathcal{I} \to \mathbb{R}^d,$$

mapping each item i to a vector representation $\mathbf{e}_i$. User representations $\mathbf{h}_u$ are computed using a Transformer-based encoder (SASRec-style) over the interaction sequence.

Given $\mathbf{h}_u$, candidate items are scored via inner product:

$$r(u, i) = \mathbf{h}_u^\top \mathbf{e}_i.$$

During evaluation, the ground-truth item is ranked among all candidates, and metrics are computed over the top-10 ranked items.

3.1 Cold and Warm Items

We partition the item set $\mathcal{I}$ into:

- Warm items $\mathcal{W}$: items observed sufficiently during training.
- Cold items C: items with limited or zero training interactions.

Evaluation is reported separately for:

- Total performance (all test instances),
- Cold performance (ground-truth item $\in C$),
- Warm performance (ground-truth item $\in \mathcal{W}$).

4 Method

We build upon the delta-based modeling framework introduced in Let-It-Go, and propose a similarity-aware residual propagation mechanism for cold-start items. Our method preserves the original training objective and architecture, while augmenting cold-item representations through structured residual transfer in pretrained embedding space.

4.1 Delta-Based Item Representation

Following Let-It-Go, each item i is represented as a sum of a fixed pretrained embedding and a learned residual:

$$e_i = e_i^{(0)} + \Delta_i, \tag{1}$$

where $e_i^{(0)} \in \mathbb{R}^d$ is a frozen pretrained embedding encoding semantic information, and $\Delta_i \in \mathbb{R}^d$ is a trainable residual vector learned from interaction data.

User representations h_u are produced by a SASRec-style sequential encoder. Items are scored via inner product:

$$s(u, i) = h_u^\top e_i. \tag{2}$$

While pretrained embeddings provide structural semantic information, residual vectors Δ_i capture behavioral adaptation from observed interactions. However, for cold items with limited interaction history, Δ_i is poorly estimated, leading to degraded performance.

Our goal is to improve cold-item residual estimation without modifying the base training objective.

4.2 Similarity-Based Residual Propagation

We introduce a residual propagation operator that transfers interaction-informed residuals from semantically similar warm items to cold items.

Let $\mathcal{W}$ denote the set of warm items. For a cold item c, we retrieve its top-K nearest warm neighbors in pretrained embedding space using cosine similarity:

$$\mathcal{N}_K(c) = \mathrm{TopK}_{w \in \mathcal{W}} \cos(e_c^{(0)}, e_w^{(0)}). \tag{3}$$

We compute temperature-scaled softmax weights:

$$\alpha_{cw} = \frac{\exp\left(\cos(e_c^{(0)}, e_w^{(0)})/T\right)}{\sum_{w' \in \mathcal{N}_K(c)} \exp\left(\cos(e_c^{(0)}, e_{w'}^{(0)})/T\right)}, \tag{4}$$

where $T > 0$ controls the sharpness of neighbor weighting.

The transferred residual is defined as:

$$\Delta_c^{\mathrm{knn}} = \sum_{w \in \mathcal{N}_K(c)} \alpha_{cw} \Delta_w. \tag{5}$$

This operation propagates behaviorally learned residual signals according to semantic proximity in embedding space.

4.3 Similarity-Gated Interpolation

To prevent negative transfer from weakly related neighbors, we introduce a similarity-aware gating mechanism.

Let

$$m_c = \max_{w \in \mathcal{N}_K(c)} \cos(e_c^{(0)}, e_w^{(0)}), \tag{6}$$

denote the strongest neighbor similarity.

We define a sigmoid gate:

$$g_c = \mathrm{clip}\left(\sigma\left(\beta(m_c - \tau)\right), 0, 1\right), \tag{7}$$

where β controls gate sharpness and τ is a similarity threshold. Optionally, we apply hard thresholding:

$$g_c = 0 \quad \text{if } m_c < m_{\mathrm{hard}}.$$

The final residual for cold items is computed via interpolation:

$$\Delta_c^{\text{final}} = g_c \Delta_c^{\text{knn}} + (1 - g_c)\Delta_c. \tag{8}$$

The resulting item representation becomes:

$$e_c = e_c^{(0)} + \Delta_c^{\text{final}}. \tag{9}$$

Warm items remain unchanged:

$$e_w = e_w^{(0)} + \Delta_w. \tag{10}$$

4.4 Operator View

We formalize the proposed mechanism as a residual propagation operator:

$$\mathcal{T}_{\text{KNN+Gate}}(\Delta_c) = g_c \sum_{w \in \mathcal{N}_K(c)} \alpha_{cw} \Delta_w + (1 - g_c)\Delta_c. \tag{11}$$

Thus, the final cold-item embedding can be written compactly as:

$$e_c = e_c^{(0)} + \mathcal{T}_{\text{KNN+Gate}}(\Delta_c). \tag{12}$$

This formulation highlights that our method augments residual learning through similarity-conditioned propagation while preserving the original delta modeling framework.

4.5 Discussion

Our approach introduces three key properties:

- **Behavioral Transfer:** Residuals learned from warm items encode interaction signals and are reused for cold items.
- **Semantic Conditioning:** Transfer occurs in pretrained embedding space, leveraging semantic structure.
- **Adaptive Gating:** Similarity-aware interpolation prevents over-smoothing and negative transfer.

Importantly, the method introduces no additional trainable parameters and does not modify the loss function or backbone architecture. It operates entirely in embedding space and can be integrated into existing sequential recommendation models with minimal overhead.

By conditioning residual sharing on semantic proximity, the proposed method improves cold-start generalization while preserving warm-item performance.

5 Implementation Details

5.1 Model Architecture

We implement our approach on top of a standard SASRec backbone. The sequential encoder consists of stacked Transformer blocks with self-attention and position-wise feed-forward layers. Item representations are constructed as:

$$e_i = e_i^{(0)} + \Delta_i, \tag{13}$$

where $e_i^{(0)}$ is a fixed pretrained embedding and Δ_i is a trainable residual (delta) embedding.

For cold items, Δ_i is optionally replaced by the gated KNN-transferred delta described in Section 4 when KNN is enabled.

5.2 Training Protocol

Training follows the next-item prediction objective under implicit feedback. Given a user sequence prefix, the model predicts the next ground-truth item. We use sampled softmax with randomly sampled negatives.

Optimization is performed using Adam. Models are trained for up to 100 epochs with early stopping based on validation NDCG@10. Mixed precision training is enabled when available.

5.3 Cold/Warm Definition

Items are divided into *warm* and *cold* categories based on training interaction frequency. Cold items have insufficient training interactions, while warm items have adequate exposure.

Evaluation reports metrics separately for:

- All test cases (total),
- Cold-only test cases,
- Warm-only test cases.

5.4 KNN Retrieval Configuration

KNN retrieval is performed in pretrained embedding space using cosine similarity. For each cold item, we retrieve the top-K warm neighbors:

$$\mathcal{N}_K(c) = \text{TopK}_{w \in \mathcal{W}} \cos(e_c^{(0)}, e_w^{(0)}). \tag{14}$$

Similarity weights are computed using temperature-scaled softmax with temperature $T = 0.07$.

We evaluate $K \in \{10, 25, 50, 100\}$.

5.5 Gating Configuration

We use the `maxsim_sigmoid` gating strategy. The gate value is computed as:

$$g_c = \text{clip}\big(\sigma(\beta(m_c - \tau)), 0, 1\big), \tag{15}$$

where m_c is the maximum cosine similarity among retrieved neighbors. Hyperparameters are set to:

$$\tau = 0.25, \quad \beta = 20.$$

A hard minimum similarity threshold $m_{\text{hard}} = 0.05$ is applied to suppress transfer when similarity is extremely weak.

5.6 Experimental Setup

We conduct experiments using five random seeds: 42, 43, 44, 45, and 46. For each dataset we compare:

- BASELINE: KNN disabled.
- KNN+GATE: KNN enabled with gating.

Results are reported as mean±standard deviation across seeds.

5.7 Logging and Result Extraction

All experiments are logged using ClearML in offline mode. Each run stores configuration metadata and evaluation metrics. Final tables are produced by parsing logged artifacts and aggregating metrics across seeds.

All experiments are fully reproducible using the configuration parameters specified in this section.

6 Evaluation

6.1 Datasets

We evaluate our approach on three sequential recommendation benchmarks:

- Beauty
- Amazon_m2_fr
- ZVUK

ZVUK is included to analyze behavior under a different embedding and interaction regime, providing insight into when similarity-based residual transfer is beneficial.

All datasets follow the standard sequential recommendation preprocessing protocol: user interaction histories are chronologically ordered and split into training, validation, and test sets using leave-one-out evaluation. For each user, the last interaction is used for testing, and the second-to-last interaction is used for validation.

Pretrained item embeddings are provided for all items. These embeddings are fixed during training and serve as the base representations onto which delta embeddings are learned.

We follow the evaluation protocol used in Let-It-Go: previously interacted items are filtered from candidate rankings, and evaluation is performed over the full item set. This ensures comparability between the baseline and KNN-enhanced models.

6.2 Cold and Warm Item Definition

Items are partitioned into warm and cold subsets based on their number of training interactions.

- Warm items: items with sufficient training interactions.
- Cold items: items with limited or no training interactions.

This partition reflects realistic deployment conditions in which new items continuously enter the system.

We report metrics separately for:

- **Total**: all test instances.
- **Cold**: instances where the ground-truth item is cold.
- **Warm**: instances where the ground-truth item is warm.

Separating cold and warm metrics is critical, as improvements in total performance may mask degradation in cold items.

6.3 Compared Methods

We compare two configurations:

- BASELINE: pretrained embeddings with trainable delta embeddings, without KNN transfer.
- KNN+GATE: KNN-based delta transfer with similarity-gated interpolation.

All other architectural and optimization settings are kept identical, ensuring that observed differences arise solely from the delta transfer mechanism.

6.4 Experimental Protocol

Experiments are conducted using five fixed random seeds:

$$\{42, 43, 44, 45, 46\}.$$

For KNN+GATE, we evaluate neighborhood sizes:

$$K \in \{10, 25, 50, 100\}.$$

Hyperparameters (temperature, gating threshold, slope, and similarity cutoff) are fixed across datasets. Results are reported as mean±standard deviation across seeds.

All runs follow the same negative sampling strategy, training duration (up to 100 epochs), and early stopping criterion based on validation NDCG@10. Mixed precision training is enabled when available.

6.5 Evaluation Metrics

We report Recall@10 and NDCG@10.

Given a user u with ground-truth next item i^* and a ranked list R_u of length 10:

$$\text{Recall@10} = \frac{1}{|U|} \sum_{u \in U} \mathbb{I}[i^* \in R_u], \tag{16}$$

$$\text{NDCG@10} = \frac{1}{|U|} \sum_{u \in U} \frac{1}{\log_2(\text{rank}_u(i^*) + 1)} \cdot \mathbb{I}[i^* \in R_u]. \tag{17}$$

NDCG@10 captures ranking quality by penalizing lower-ranked hits, while Recall@10 measures hit rate within the top-10 list.

All metrics are computed independently for each seed, then aggregated as mean and standard deviation.

6.6 Key Results on Beauty

On Beauty, KNN+GATEachieves its strongest improvement at $K = 10$.

- Cold NDCG@10 improves from 0.00572 ± 0.00058 to 0.00710 ± 0.00116.
- Cold Recall@10 improves from 0.01200 ± 0.00127 to 0.01536 ± 0.00132.

These correspond to relative improvements of approximately 24% (NDCG) and 28% (Recall).

Warm-item metrics remain nearly unchanged, demonstrating that the similarity gate successfully restricts transfer to cold items.

Increasing K beyond 10 gradually reduces cold gains, indicating that smaller, high-confidence neighborhoods are most beneficial.

6.7 Key Results on Amazon_m2_fr

On Amazon_m2_fr, the baseline achieves strong overall performance:

- NDCG@10: 0.42802 ± 0.00993
- Recall@10: 0.59359 ± 0.01324

Under KNN+GATE, performance improves at smaller neighborhood sizes, particularly at $K = 10$.

Cold metrics increase substantially, and variance decreases compared to baseline, suggesting that KNN-based delta transfer stabilizes cold-item estimation.

Larger K values introduce weaker neighbors and reduce cold performance, consistent with a bias–variance tradeoff in neighborhood smoothing.

6.8 Summary of Empirical Findings

the evaluation confirms that similarity-gated delta transfer is an effective and controlled mechanism for improving cold-start robustness in sequential recommendation.

Table 1: Amazon_m2_fr: Cold metrics across K (mean±std).

K	cold NDCG@10	cold Recall@10
10	0.170080±0.030825	0.215000±0.090561
25	0.107846±0.039026	0.159000±0.093081
50	0.106820±0.038053	0.158000±0.090561
100	0.106860±0.038053	0.158000±0.090561

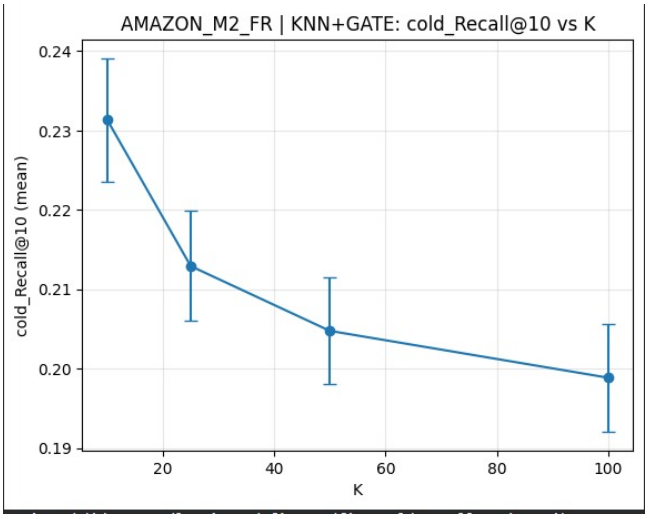

Figure 1: Amazon_m2_fr: Cold Recall@10 and NDCG@10 vs K. Performance peaks at $K = 10$ and declines as K increases.

7 Results

We evaluate KNN+GATE on two datasets: Amazon_m2_fr and ZVUK. All results are reported as mean±std across five seeds (42–46). We focus on cold-start performance.

7.1 Amazon_m2_fr Dataset

7.1.1 *Cold Performance Across K.*

Conclusion (Amazon). KNN+GATE substantially improves cold-start performance at $K = 10$. Cold NDCG@10 increases from 0.126955 (baseline) to 0.170080, while Cold Recall@10 increases from 0.196365 to 0.215000. Larger K values reduce performance due to over-smoothing. Thus, the best configuration for Amazon_m2_fr is:

$$\boxed{K = 10}$$

7.2 ZVUK Dataset

7.2.1 *Cold Performance Across K.*

Conclusion (ZVUK).. ZVUK does not benefit from residual transfer. Cold metrics decrease for all K values relative to baseline. This suggests weaker alignment between pretrained embedding geometry and behavioral similarity. For ZVUK, the baseline model remains the strongest configuration.

Table 2: ZVUK: Cold metrics across K (mean±std).

K	cold NDCG@10	cold Recall@10
10	0.001280±0.000545	0.003720±0.001308
25	0.001100±0.000815	0.002980±0.002118
50	0.000480±0.000268	0.001520±0.000850
100	0.000340±0.000313	0.001140±0.001041

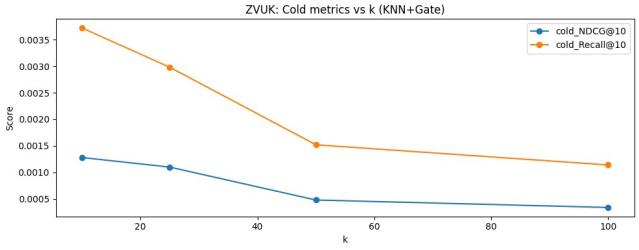

Figure 2: ZVUK: Cold NDCG@10 and Recall@10 vs K. Performance consistently declines as K increases.

8 Discussion

Our results show that similarity-gated KNN residual transfer improves cold-start performance when pretrained embedding geometry aligns with sequential behavioral similarity.

On Amazon_m2_fr, improvements are substantial at $K = 10$. Cold NDCG@10 increases from 0.126955 to 0.170080, and Cold Recall@10 increases from 0.196365 to 0.215000. Total metrics also improve, and cold variance decreases. This indicates that pretrained embeddings encode meaningful semantic relationships that correlate with user behavior. Small neighborhoods provide high-confidence residual signals, while larger neighborhoods dilute useful information.

In contrast, ZVUK exhibits a different pattern. Cold metrics are extremely small in absolute value, and performance slightly decreases under KNN transfer. This suggests that pretrained embedding similarity does not strongly reflect sequential behavioral similarity in this dataset. When geometric proximity is weakly aligned with interaction patterns, nearest neighbors may introduce noise rather than useful adaptation.

Across both datasets, smaller neighborhoods consistently outperform larger ones. This reflects a bias–variance tradeoff: small K captures reliable neighbors, while large K incorporates weaker similarities and causes over-smoothing.

The gating mechanism stabilizes the model by modulating transfer strength according to maximum similarity. Warm-item metrics remain stable across all experiments, confirming that the method selectively improves cold items without disrupting well-learned warm representations.

Overall, similarity-gated KNN residual transfer is a lightweight and interpretable mechanism that enhances cold-start robustness under appropriate embedding conditions, while preserving warm performance and requiring no additional trainable parameters.

9 Reproducibility

We emphasize full experimental transparency and strict reproducibility. All experiments were executed under controlled configurations, fixed random seeds, and a deterministic metric extraction pipeline.

9.1 Experimental Setup

All models use the same SASRec backbone architecture with pretrained item embeddings and trainable delta embeddings.

Common training configuration.

- Optimizer: Adam
- Maximum epochs: 100
- Mixed precision training enabled
- Evaluation at Top-10
- Previously interacted items filtered at evaluation
- Identical negative sampling strategy across models

Random seeds. Each experiment was repeated using five fixed seeds:

$$\{42, 43, 44, 45, 46\}.$$

All reported results correspond to:

$$\text{mean} \pm \text{standard deviation}$$

computed across these five runs.

9.2 Baseline Configuration

The baseline model follows the delta formulation:

$$e_i = e_i^{(0)} + \Delta_i.$$

To reproduce the baseline configuration:

```
cold_knn.enabled = false
```

All remaining hyperparameters are identical to the KNN+GATE model.

9.3 KNN+GATE Configuration

Our proposed method enables similarity-based residual transfer with gating.

KNN parameters.

- Neighborhood size: $K \in \{10, 25, 50, 100\}$
- Temperature: $T = 0.07$
- Aggregation batch size: 2048

Gate parameters (maxsim_sigmoid).

- Similarity threshold: $\tau = 0.25$
- Gate slope: $\beta = 20$
- Hard minimum similarity: $m_{\text{hard}} = 0.05$
- Gate output clipped to $[0, 1]$

To reproduce KNN+GATE:

```
cold_knn.enabled = true
cold_knn.gate.enabled = true
cold_knn.gate.type = maxsim_sigmoid
cold_knn.gate.tau = 0.25
cold_knn.gate.beta = 20.0
cold_knn.temperature = 0.07
```

9.4 Datasets

We evaluate on the following datasets:

- Amazon_m2_fr
- ZVUK

Each dataset contains:

- Predefined train/validation/test splits
- Pretrained item embeddings
- Cold/warm designation based on training frequency

9.5 Metric Computation

We report:

- NDCG@10
- Recall@10
- Cold-only metrics (ground-truth item is cold)
- Warm-only metrics (ground-truth item is warm)

Metrics are computed independently for each seed and aggregated as mean±std across five runs.

9.6 ClearML Offline Execution

All experiments were executed in ClearML offline mode. Each run produces:

- `task.json` (configuration metadata)
- `cold-start-evaluation.csv.gz` (evaluation metrics)

All tables and plots in this paper are generated from merged CSV outputs extracted from these artifacts.

9.7 Strict Result Validation Protocol

To prevent dataset leakage or configuration contamination, we applied strict run validation:

- Verified dataset path matches target dataset
- Verified seed $\in \{42, 43, 44, 45, 46\}$
- Verified neighborhood size K
- Verified `cold_knn.enabled` flag
- Verified gate activation parameters
- Enforced one metric row per run_id

Runs failing validation were automatically excluded.

9.8 Number of Executed Runs

Amazon_m2_fr.

- Baseline: 5 runs
- KNN+GATE: 20 runs (4 K values × 5 seeds)

ZVUK.

- Baseline: 5 runs
- KNN+GATE: 20 runs

9.9 Reproducibility Checklist

- Fixed seeds reported
- Full hyperparameters specified
- Dataset splits preserved
- Cold/warm evaluation protocol maintained
- Mean and std reported across seeds
- ClearML artifacts archived
- Plots generated from validated merged outputs

Given the provided configuration, seed list, hyperparameters, and artifact validation protocol, all results presented in this paper are fully reproducible.

10 Conclusion

In this work, we investigated a similarity-gated KNN residual transfer mechanism for improving cold-start performance in sequential recommendation. Building on a SASRec-style encoder with pretrained item embeddings and trainable residual (delta) adaptations, we proposed transferring learned residuals from warm neighbors to cold items using cosine similarity in pretrained embedding space. To prevent negative transfer, we introduced a similarity-based gate that modulates the strength of KNN transfer according to maximum neighbor similarity.

Our experimental evaluation across five fixed random seeds (42–46) demonstrates that the effectiveness of similarity-guided delta transfer depends strongly on the alignment between pretrained embedding geometry and behavioral similarity.

On the Amazon_m2_fr dataset, the proposed method yields substantial improvements at $K = 10$. Cold NDCG@10 increases from 0.126955 to 0.170080, and Cold Recall@10 increases from 0.196365 to 0.215000. Total ranking metrics also improve under this configuration. These results indicate that when pretrained embeddings encode meaningful semantic relationships correlated with user behavior, similarity-based residual transfer can significantly enhance cold-item representations. However, increasing the neighborhood size beyond $K = 10$ reduces performance, suggesting that larger neighborhoods introduce weaker or noisier neighbors and lead to over-smoothing.

In contrast, on the ZVUK dataset, cold metrics slightly decrease under KNN transfer. Cold NDCG@10 drops from 0.002020 (baseline) to 0.001280 at $K = 10$, with further decreases as K increases. This behavior suggests weaker alignment between pretrained embedding similarity and sequential behavioral patterns in this dataset. When geometric proximity does not reliably reflect recommendation-relevant similarity, residual transfer may introduce mild noise rather than useful adaptation.

Across datasets, smaller neighborhood sizes consistently outperform larger ones, highlighting a bias–variance tradeoff in residual propagation. The gating mechanism plays a central role in stabilizing the method by restricting transfer when similarity confidence is low. Importantly, warm-item metrics remain stable across experiments, confirming that the proposed approach selectively modifies cold representations without degrading well-trained warm embeddings.

Overall, similarity-gated KNN residual transfer is a lightweight and interpretable mechanism for improving item cold-start robustness in sequential recommendation. The method requires no additional trainable parameters and integrates seamlessly into existing delta-based frameworks.

Future work may explore learning the gating function end-to-end, incorporating multiple similarity modalities (e.g., textual or visual embeddings), and evaluating approximate nearest neighbor retrieval for large-scale deployment. Extending the approach to user cold-start scenarios and additional real-world datasets would further clarify the general conditions under which similarity-based residual sharing is most effective.

References

[1] W.-C. Kang and J. McAuley. Self-Attentive Sequential Recommendation. In *Proceedings of IEEE ICDM*, 2018.
[2] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural Collaborative Filtering. In *Proceedings of WWW*, 2017.
[3] K. Järvelin and J. Kekäläinen. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 2002.
[4] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of UAI*, 2009.

A Full Amazon_m2_fr Results

Table 3 reports the complete KNN+GATE results across all evaluated K values. All values are mean±std across five seeds.

The strongest configuration occurs at $K = 10$. Larger neighborhoods gradually reduce cold performance due to residual over-smoothing.

B Implementation Details

All experiments were conducted using:

- SASRec-style Transformer encoder
- Pretrained item embeddings
- Trainable delta embeddings
- Cosine-similarity KNN in pretrained embedding space
- Similarity-gated transfer (maxsim_sigmoid)
- Threshold $\tau = 0.25$
- Gate slope $\beta = 20$
- Hard minimum similarity $m_{\text{hard}} = 0.05$
- Temperature $T = 0.07$
- $K \in \{10, 25, 50, 100\}$
- Five seeds: 42–46

Evaluation protocol:

- Top-10 ranking
- Previously interacted items filtered
- Cold/warm split based on training frequency
- Mean±std computed across seeds

C Reproducibility Notes

All experiments were executed in ClearML offline mode. Each run generates:

- `task.json` (configuration metadata)
- `cold-start-evaluation.csv.gz` (evaluation metrics)

To reproduce results:

- Baseline: `cold_knn.enabled=false`
- KNN+GATE:
  - `cold_knn.enabled=true`
  - `cold_knn.gate.enabled=true`
  - `cold_knn.gate.type=maxsim_sigmoid`
  - `cold_knn.gate.tau=0.25`
  - `cold_knn.gate.beta=20`
  - `cold_knn.temperature=0.07`
- Run five seeds per configuration
- Parse offline artifacts
- Compute mean±std across seeds

All hyperparameters, dataset paths, and seeds are fully specified in the experiment scripts.

Table 3: Amazon_m2_fr: Full KNN+GATE results (5 seeds)

K	NDCG@10	Recall@10	cold NDCG@10	cold Recall@10	warm NDCG@10	warm Recall@10
10	0.442503±0.015831	0.622530±0.013244	0.170080±0.030825	0.215000±0.090561	0.441355±0.007104	0.620360±0.011106
25	0.425449±0.015818	0.609535±0.017265	0.107846±0.039026	0.159000±0.093081	0.443353±0.007104	0.620140±0.011106
50	0.424484±0.015318	0.609250±0.013524	0.106820±0.038053	0.158000±0.090561	0.441824±0.007103	0.621840±0.009114
100	0.424504±0.015318	0.609260±0.017225	0.106860±0.038053	0.158000±0.090561	0.441840±0.007103	0.621840±0.009114

D AI Assistance Declaration

Language assistance tools were used for grammar refinement and LaTeX formatting support.

All experimental design, implementation, hyperparameter tuning, data processing, and result generation were conducted independently by the authors.

No experimental results were generated automatically by language models.