COMPOSITIONAL INDUCTIVE RECOMMENDERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern recommender systems must serve ever-growing catalogs, including items unseen during training. This challenges models based on ID embeddings, which work well for observed items but do not generalize to new, "cold" items. We advocate for modeling item cold start as a generalization problem and formalize the induction loss allowing recommenders to generalize on cold items. We show that, around a local optimum of the standard recommendation loss, the induction loss is controlled by the reconstruction error between each item's embedding and its reconstruction from its content-similars. Motivated by this analysis, we prove that the Softmax-Hull Embeddings (**SHE**), which represent each item as a content-weighted combination of neighboring item embeddings, provide the non-parametric construction that is optimal for generalization on cold items. We further show that the generalization capabilities of **SHE** are related to the alignment between the content representations and the item embeddings. We give empirical support to our work through recommendation experiments, demonstrating that **SHE** improves the overall accuracy of several ID-embedding models, simultaneously yielding the expected strong gains on cold items, and outperforming existing approaches for cold-start recommendation.

1 INTRODUCTION

Recommender Systems (RSs), which are ubiquitous in our online experiences, rely on representing catalog items as high-dimensional vectors, i.e., embeddings, to accurately recommend them. Many RSs face serious limitations in doing so while the item catalog is constantly expanding. This is the case for all ID-embedding models (Koren et al., 2009; Wang et al., 2019; He et al., 2020), including sequential ones (Kang & McAuley, 2018; Sun et al., 2019). These ID-embedding models assign each item a unique identifier that is mapped to a randomly initialized embedding, which is updated during training. As a result, the embeddings of cold items, such as those recently added to the catalog and therefore lacking interactions in the training set, are never updated during training. This leads to a lower overall model accuracy (Ricci et al., 2022). Accurately recommending cold items requires informative representations, which existing approaches to item cold start derive from content, e.g., representations of textual descriptions or images of the item (He & McAuley, 2016a; Ding et al., 2021; Harte et al., 2023; Boz et al., 2025; Hou et al., 2026). In the simplest formulation, content representations are obtained from pre-trained and frozen content encoding models, and used as item embeddings (Yuan et al., 2023), with optional fine-tuning or addition of a small trainable vector (Pembek et al., 2025). Other methods are trained to learn a joint representation space of content and behavioral (i.e., interaction) information in which embeddings of the same item are close (Zhang et al., 2025). This is done either with dropout- (Volkovs et al., 2017; Lichtenberg et al., 2025; Moscati et al., 2026), graph- (Li et al., 2024; Kim et al., 2024) or contrastive-learning (Zhou et al., 2023; Ganhör et al., 2024; 2025). More recent RSs address item cold start with Semantic IDs (SIDs), which convert content into discrete tokens which preserve semantics (Mei et al., 2025; Rajput et al., 2023; Penha et al., 2025; Zeghidour et al., 2021; Deng et al., 2025; Luo et al., 2024).

All these methods for cold start implicitly assume a connection between item behavioral and content representations, despite the evidence of the gap between generic content embeddings and behavioral ones (Cai et al., 2021; Sheng et al., 2025; Meehan & Pauwels, 2025). Instead, we advocate for modeling item cold start as part of the inductive (i.e., generalization) capabilities of a RS (Ding et al., 2026a;b). This implies addressing it by means of a loss term for better generalization, as commonly done in machine learning (Goodfellow et al., 2016). We follow this approach and propose the use

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.

of an induction loss term to complement the recommendation loss of any RS. In contrast with prior methods that assume a strict relation between behavioral and content spaces, we formally show that the induction loss is bounded by the alignment between these spaces, and by the error obtained when constructing behavioral embeddings from content.

This theoretical understanding allows us to identify a principled non-parametric constructor for behavioral embeddings of cold items, based on the composition of warm item embeddings. Our proposed Softmax-Hull Embeddings (**SHE**) corresponds to a kernel-regression estimate of the item embedding from its content, and we bound its reconstruction error in terms of the alignment between content and behavioral spaces. Although theoretically grounded, the induction loss can be computationally prohibitive. We therefore introduce a surrogate loss that approximates the induction loss with fewer forward passes, and we derive upper bounds on the additional error it introduces.

We empirically assess the effectiveness of our proposed approach on two recommendation domains: online shopping and music streaming, showing quantitatively that the induction loss and the embedding constructor lead to more accurate recommendations in both warm and cold scenarios.

The contributions of our paper are the following:

C1. We discuss the theory of RS generalization and derive local upper bounds on the induction loss in terms of the alignment between content and behavior spaces.

C2. We propose a way to improve the accuracy of RSs by complementing the recommendation loss with an induction loss term.

C3. We design a way to address item cold start based both on the induction loss and on a theoretically-grounded approach to construct cold item embeddings.

C4. We empirically show that induction loss leads to higher RS accuracies, and that our embedding construction effectively tackles item cold start.

2 METHODOLOGY

We formulate the recommendation problem as $\min_{\omega,E} \mathcal{L}(\omega, E)$, where $\mathcal{L}$ is the recommendation loss, $E \in \mathbb{R}^{N\times d}$ the d-dimensional embeddings of the N items in the catalog $\mathcal{I}$, and ω the remaining model parameters. This formulation separates the item embeddings E from the loss $\mathcal{L}$ and the model architecture ω. For each item i, we denote by $c_i \in \mathbb{R}^{d_c}$ its content representation obtained from a pre-trained model, and by n_i its number of occurrences in the training data.

2.1 INDUCTIVE RECOMMENDER

We call a recommender *inductive* (Mitchell, 1997) if it generalizes well to unseen cold items and is optimized on a loss J_λ that includes the induction loss as a term modeling cold items through a constructor function g: [1]

$$J_\lambda(\omega, E, g) := \mathcal{L}(\omega, E) + \lambda \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\omega, E_{i \leftarrow g(c_i, E_{\mathcal{I}\setminus i})}). \tag{1}$$

The first term $\mathcal{L}(\omega, E)$ denotes the *warm* loss, computed by representing each item i through its embedding $e_i \in E$. The second term, the induction loss, computes $\mathcal{L}$ while leaving out the ID embedding e_i of each item i and replacing it with the output of a constructor $g(c_i, E_{\mathcal{I}\setminus i}) : \mathbb{R}^{d_c} \times \mathbb{R}^{(N-1)\times d} \to \mathbb{R}^d$ that builds i's embedding based on its content c_i and the embeddings $E_{\mathcal{I}\setminus i}$ of other catalog items. Since cold items lack training interactions, this leave-one-out procedure *mimics* item cold start on warm items. Finally, the weight $\lambda \geq 0$ balances the warm and the induction loss.

Proposition 1. [2] *Assume a warm recommendation loss $\mathcal{L}(\omega, E)$ that, on a neighborhood $\mathcal{B}$ of a point of local minimum (ω^0, E^0), satisfies the interaction-weighted quadratic bound $\mathcal{L}(\omega^0, X) \leq \mathcal{L}(\omega^0, E^0) + \frac{\beta}{2} \sum_{i=1}^{N} n_i \|x_i - e_i^0\|^2$, and assume that every leave-one-out table $E^0_{i \leftarrow g(c_i, E^0_{\mathcal{I}\setminus i})}$ lies in*

[1] To simplify notation, we drop the explicit dependence of the loss components on the data $\mathcal{D}$.

[2] Proofs of all propositions are given in Appendix A.

$\mathcal{B}$. Then J_λ is locally controlled by the reconstruction error Q:

$$Q(E, g) = \sum_{i=1}^{N} n_i ||g(c_i, E_{\mathcal{I}\setminus i}) - e_i^0||^2 \tag{2}$$

$$J_\lambda(\omega^0, E^0, g) \leq (1+\lambda)\mathcal{L}(\omega^0, E^0) + \frac{\lambda\beta}{2N} Q(E^0, g). \tag{3}$$

It follows from Proposition 1 and the local minimality of (ω^0, E^0) that, at the warm solution, a constructor that perfectly reconstructs the embeddings E^0 attains the minimum of J_λ over constructors.

2.2 SOFTMAX-HULL-EMBEDDER (**SHE**) AS RECONSTRUCTION MINIMIZER

Proposition 1 motivates a constructor g that accurately reconstructs the learned embeddings, keeping the inductive loss low. Based on this insight, we derive the exact non-parametric minimizer of a local exponential-kernel regression objective for E from content. Constructors $g(c_i, E_{\mathcal{I}\setminus i})$ are designed such that content-similar items receive similar constructed embeddings.

Proposition 2. *For any content similarity s, fixed content neighborhood $\mathrm{kNN}_c(i) \subseteq \mathcal{I} \setminus \{i\}$, and temperature $\tau > 0$, the Softmax-Hull-Embedder (**SHE**)*

$$g(c_i, E_{\mathcal{I}\setminus i}) = \sum_{j \in \mathrm{kNN}_c(i)} \mathrm{softmax}_{j \in \mathrm{kNN}_c(i)}\left(\frac{s(c_i, c_j)}{\tau}\right) e_j \tag{4}$$

is the non-parametric kernel-regression estimate $g(c_i, E_{\mathcal{I}\setminus i})$ of the item embedding e_i from content, i.e., the minimizer of the content-weighted least-squares objective:

$$\arg\min_{x \in \mathbb{R}^d} \sum_{j \in \mathrm{kNN}_c(i)} K_\tau(c_i, c_j) \, \|x - e_j\|_2^2, \qquad K_\tau(c_i, c_j) = \exp\left(\frac{s(c_i, c_j)}{\tau}\right). \tag{5}$$

As the softmax weights are positive and sum to one, the **SHE** embedding lies in the convex hull of the neighbors' embeddings. Note that **SHE** is a specific instance of embeddings $e_i = \sum_m W_{im} \cdot A_m$, constructed as the weighted sum of M auxiliary vectors $A_m \in \mathbb{R}^d$. This formulation frames the **SHE** as an instance from a larger class of approaches to construct E, as discussed in the Appendix B.

2.3 INDUCTION LOSS AND SPACES ALIGNMENT

The quality of the **SHE** estimator intuitively depends on the alignment between content and embedding spaces: the better content similarity reflects embedding similarity, the more accurate the estimate. We formalize this intuition by bounding J_λ in terms of this alignment.

Proposition 3. *Under the assumptions of Proposition 1, let d be a distance in content space consistent with the similarity s, and let g be the **SHE** estimator. If the item embeddings are Λ-Lipschitz in content: $\|e_i^0 - e_j^0\| \leq \Lambda d(c_i, c_j)\ \forall i, j \in \mathcal{I}$, then:*

$$J_\lambda(\omega^0, E^0, g) \leq (1+\lambda)\mathcal{L}(\omega^0, E^0) + \frac{\lambda\beta}{2N}\Lambda^2 \sum_{i=1}^{N} n_i \bar{d}_i^2, \tag{6}$$

where $\bar{d}_i = \sum_{j \in \mathrm{kNN}_c(i)} \bar{w}_{ij} d(c_i, c_j)$ is the average content distance of i from its neighbors, weighted with the softmax weights $\bar{w}_{ij}$ of Proposition 2. , collected in the matrix $\bar{W}$ (with $\bar{w}_{ij} = 0$ for $j \notin \mathrm{kNN}_c(i)$) such that $g(c_i, E_{\mathcal{I}\setminus i}) = (\bar{W}E)_i$.

Proposition 3 bounds the cost of replacing the learned item embeddings with **SHE** embeddings. The bound is controlled by four factors: the curvature of the loss around the minimum (β), the misalignment between content and embedding spaces (Λ), the number of occurrences of each item i in the training set (n_i), and the average content distance of i from its k neighbors ($\bar{d}_i$). Catalogs with many near-duplicate items that share similar interaction patterns yield a stronger guarantee on the inductive objective, as close content neighbors reduce $\bar{d}_i^2$ without inflating Λ.

Proposition 4. *Under the assumptions of Proposition 3, minimizing $J_\lambda(\omega, E, \bar{W})$ reaches a lower loss than $J_\lambda(\omega^0, E^0, \bar{W})$, and the gap to the backbone minimum is controlled by the content space:*

$$\min_{\omega,E} J_\lambda(\omega, E, \bar{W}) \le J_\lambda(\omega^0, E^0, \bar{W}) \le (1+\lambda)\mathcal{L}(\omega^0, E^0) + \frac{\lambda\beta}{2N}\Lambda^2 \sum_{i=1}^{N} n_i \bar{d}_i^2. \tag{7}$$

Proposition 4 states that optimizing $J_\lambda(\omega, E, \bar{W})$ reaches a lower loss than the one of the backbone incremented by an error budget controlled by the misalignment between the embedding and the content space. Note that the assumption of Λ-Lipschitz-continuity implies that content neighbors have similar item embeddings, but not the opposite: items with similar embeddings may still be further apart in content space.

2.4 SINGLE-PASS **SHE** TRAINING

The induction loss is evaluated on a different embedding table $E_{i \leftarrow g(c_i, E_{\mathcal{I}\setminus i})}$ for every item, which can be computationally prohibitive. To reduce the training requirements of the **SHE**, we propose a surrogate loss allowing us to evaluate the warm loss *once* on the mixed table

$$W_q E = qE + (1-q)\bar{W}E, \qquad (W_q E)_i = qe_i + (1-q)g(c_i, E_{\mathcal{I}\setminus i}), \tag{8}$$

where the hull of item i also includes the embedding e_i of the item itself. The parameter $q := 1/(1+\lambda) \in (0,1)$ defines the amount of "self-share" in the hull, uniformly for all items ($q = 0$ corresponds to the limit $\lambda \to \infty$, i.e., to the pure hull).

Proposition 5. *Under the assumptions of Proposition 1, if the mixed table $W_q E^0$ also lies in $\mathcal{B}$, the warm loss $\mathcal{L}$ at the local minimum and at the mixed table $W_q E^0$ satisfies*

$$\mathcal{L}(\omega^0, E^0) \le \mathcal{L}(\omega^0, W_q E^0) \le \mathcal{L}(\omega^0, E^0) + \tfrac{\beta}{2}(1-q)^2 Q(E^0, \bar{W}); \tag{9}$$

and the discrepancy between the inductive loss J_λ and $\mathcal{L}(\omega^0, W_q E^0)$ is such that

$$|J_\lambda(\omega^0, E^0, \bar{W}) - (1+\lambda)\mathcal{L}(\omega^0, W_q E^0)| \le \frac{\lambda\beta}{2} Q(E^0, \bar{W}). \tag{10}$$

Proposition 5 sets an upper bound on the difference between the warm loss evaluated with the single embedding table $W_q E^0$, the loss at the local minimum, and the computationally expensive, inductive loss J_λ. Hence, their combination can be approximated with a single forward pass, and the approximation error has an upper bound depending on $\frac{\lambda\beta}{2} Q(E^0, \bar{W})$, i.e., the quadratic bound around warm loss minimum (β), the reconstruction error (Q), and the weight of the induction loss (λ). Proposition 5 justifies the surrogate locally, around a warm optimum (ω^0, E^0); in practice we use the single-pass objective as the training loss also outside this regime, initializing E at E^0 but re-initializing ω and training both jointly (Section 3).

3 EXPERIMENTS

We aim to answer the following research questions (**RQ**s) by means of recommendation experiments on two datasets from two different recommendation domains:

RQ1 Does the induction loss, when computed through its single-pass approximation, improve warm-target accuracy?

RQ2 After training with the induction loss, does treating cold items with the **SHE** improve accuracy on cold targets over existing cold-start methods, without hurting warm-target accuracy?

RQ3 Does the induction loss imply a trade-off between warm and cold target accuracy? Is the trade-off controlled by the self-share q, i.e., by the induction loss weight λ?

RQ4 Is the accuracy gain of the **SHE** attributable to initializing the embeddings from a pre-trained backbone?

RQ5 Is **SHE** more effective when content and behavioral spaces are better aligned?

3.1 EXPERIMENTAL SETUP

Datasets. We conduct our experiments on publicly available recommendation datasets that exhibit different characteristics, as shown in Table 1 and Appendix C.

Amazon (McAuley et al., 2015; He & McAuley, 2016b) contains user reviews of products from the Amazon shopping platform. Each review is converted into an implicit binarized interaction. We select the pre-filtered 5-core data (i.e. each user reviewed at least 5 items, each item was reviewed by at least 5 users) for the *Beauty*, *Sports and Outdoors*, and *Toys and Games* categories and merge them into a single catalog and interaction log. This is done to obtain a larger item catalog than with single categories, allowing us to sample a fraction for cold-start evaluation, and also yields a heterogeneous catalog where content neighbors and interaction logs can cross categories. For each item, we concatenate title, price, brand, and category, and embed the resulting text with a pre-trained instance of `multilingual-e5-small` (Wang et al., 2024) ($d_c = 384$). Following standard practices for recommendation (Kang & McAuley, 2018; Zhou et al., 2020), we sort each user's interactions by timestamp and apply a leave-one-out split: the last interaction is used for testing, the second-to-last for validation, and all remaining interactions for training. At test time, all interactions preceding the test one, including the validation interaction, are used as model input.

Onion (Santana et al., 2020; Moscati et al., 2022) is a music recommendation dataset that combines listening events from the LFM-2b dataset (Schedl et al., 2022) with precomputed content representations of the music tracks. Among these, we select the pre-embedded audio based on `musicnn` (Pons & Serra, 2019) ($d_c = 50$). Following prior work on the same dataset (Li et al., 2026), we restrict the listening events to 2018. We apply iterative 2-core filtering, dropping users and tracks with fewer than two interactions until convergence. Following best practices in sequential recommendation (Gusak et al., 2025a;b), we adopt a global temporal split strategy: after sorting interactions by timestamp, the earliest 80% form the training set. For each user's sequence in the 80% training set, the last interactions constitute the validation set, while the last interactions of sequences in the remaining 20% constitute the test, using all previous interactions as model input.

Table 1: Datasets after preprocessing. "Warm" items have at least one training interaction; "cold" items are 20% of the overall catalog (randomly sampled), and do not have training interactions.

		# Items				# Test Sequences	
Dataset	Users	Warm	Cold	Interact.	Item Content	Warm Target	Cold Target
Amazon	72,682	33,905	8,476	0.66M	text (*e5*) , d_c=384	58,184	14,498
Onion	20,456	42,702	10,676	17.3M	audio (*musicnn*), d_c=50	11,293	2,766

Baselines. We compare our approach with representative embedding-based models, described in more detail in Appendix C.2. For *ID-embedding* models, we include matrix factorization **MF** (Koren et al., 2009) and **SASRec** (Kang & McAuley, 2018) trained with sampled-negative binary cross-entropy. For *content-projection* models, we consider three methods: **MoRec** (Yuan et al., 2023) is a SASRec variant in which item embeddings are replaced by frozen content embeddings. **DenseRec** (Lichtenberg et al., 2025) maps content to item embeddings through a single linear projection. We also include a **Linear Imitation** model that fits a closed-form ridge regression from content to the item embeddings of a pre-trained SASRec and uses the regressed item embeddings without further training. For *Semantic IDs*, we rely on two content tokenizers: RQ-VAE (Lee et al., 2022) and RQ-Kmeans (Penha et al., 2025). After tokenization, we either train on the sequences of SIDs tokens, following the generative formulation of TIGER (Rajput et al., 2023) (**SID generative**), or train SASRec using as item embeddings the uniform average of the SIDs token embeddings (**SID two-tower**). Finally, we include **MostPop** and **Random** to anchor the scale of the metrics.

Our Method. We jointly apply the induction loss and the **SHE**, using the W_q approximation, to two backbones: MF (**MF + SHE**) and SASRec (**SASRec + SHE**). Each item's content neighborhood consists of $k-1 = 63$ items plus the self-share, and the softmax temperature τ is fitted to maximize the reconstruction of the warm item table E^0 through the softmax-hull. We initialize the embeddings E at E^0 and re-initialize the backbone weights ω. Both E and ω are then trained jointly. We sweep the self-share q (Appendix C.3) and select the smallest value whose validation HR@10 lies within

Table 2: Warm-target performance. Best in bold, second best underlined. The **SHE** rows use the self-share selected on validation: $q = 0.05$ (Amazon) and 0.1 (Onion) for SASRec, 0.1 (Amazon) and 0.3 (Onion) for MF. Confidence intervals and significance tests are in Appendix C.5.

	Amazon			Onion		
Method	HR@10	nDCG@10	HR@100	HR@10	nDCG@10	HR@100
Reference						
Random	.0001	.0000	.0026	.0002	.0001	.0029
MostPop	.0061	.0033	.0279	.0060	.0027	.0430
ID Embeddings						
MF	.0181	.0068	.1072	.1226	.0728	.2655
SASRec	<u>.0568</u>	.0251	.1954	<u>.3375</u>	<u>.2073</u>	<u>.4847</u>
Content Projections						
MoRec	.0515	.0242	.1711	.0015	.0008	.0106
Linear Imitation	.0103	.0048	.0673	.0020	.0007	.0223
DenseRec (linear)	.0564	<u>.0261</u>	<u>.2056</u>	.3030	.1904	.4546
Semantic IDs						
SID generative, RQ-VAE	.0417	.0224	.1481	.1129	.0766	.2042
SID generative, RQ-Kmeans	.0356	.0197	.1307	.1737	.1082	.2989
SID two-tower, RQ-VAE	.0233	.0107	.1238	.1504	.1165	.2281
SID two-tower, RQ-Kmeans	.0280	.0137	.1382	.1497	.1137	.2325
SHE (ours)						
MF + **SHE**	.0379	.0174	.1666	.1249	.0745	.2653
SASRec + **SHE**	**.0745**	**.0344**	**.2371**	**.3448**	**.2096**	**.4926**

the 95% band of the best one. This selects $q = 0.05$ (Amazon) and $q = 0.1$ (Onion) for SASRec, and $q = 0.1$ and $q = 0.3$ for MF. A cold item, instead, uses $q = 0$.

Evaluation. For every test sequence, the model ranks the *entire* candidate catalog against the single held-out target, as sampled metrics are inconsistent estimators of the full ranking (Krichene & Rendle, 2020). We report the HR@k and nDCG@k at $k = 10$ and $k = 100$. To simulate item cold start, we sample 20% of the catalog items uniformly at random before training and remove all of their interactions from the training set, while retaining those in the test set. Cold items can therefore still appear in the input sequences at test time, but are dropped for ID-embedding models, which cannot infer cold-item embeddings nor retrieve cold items, and therefore score zero on cold targets. For each trained model, we compute 95% confidence intervals over test users, and we compare the best two methods with a paired Wilcoxon signed-rank test (Wilcoxon, 1945) on per-user nDCG@10. Details are in Appendix C.5. Table 1 provides the summary of test interactions divided into sequences with *warm* or *cold* target items, respectively.

Implementation and Training All models are trained with Adam (Kingma & Ba, 2015) up to 300 epochs with early stop on HR@10, evaluated on warm target sequences. All methods relying on SASRec as backbone share the same architecture (2 blocks, 2 heads, dropout 0.5), with the embedding size set per dataset: 64 for Amazon and 256 for Onion. We report the hyperparameters of all methods in Appendix C.3. The "**+ SHE**" affects the item embeddings E while keeping the architectures identical to the backbone. Furthermore, we keep all optimization hyperparameters equal to ensure that the two are compared under identical training conditions. Since the training procedure and model architecture are equal, results when using "**+ SHE**" and those for the backbone differ only in how item embeddings are constructed, except for the self-share q that is selected on validation, and the backbone weights. Finally, each trained model is evaluated on the test set.

4 Results

4.1 Warm-Target Results

Table 2 reports results on sequences with a warm target item (**RQ1**).

We first compare the performance of the MF and SASRec in their standard embedding regime, i.e., using ID embeddings and training with the warm loss only, versus when combined with **SHE**.[3] We observe that SASRec + **SHE** achieves the highest score for both datasets and across all metrics. On Amazon, SASRec + **SHE** improves over SASRec by 31% in HR@10 (0.0745 vs. 0.0568) and by 37% in nDCG@10 (paired Wilcoxon test, $p = 2.5 \times 10^{-92}$). MF + **SHE** more than doubles the HR@10 (0.0379 vs. 0.0181) and nDCG@10 (0.0174 vs. 0.0068) of MF. On Onion, the improvement is less pronounced. SASRec + **SHE** reaches a higher HR@10 than SASRec (0.3448 vs. 0.3375), although the difference in nDCG@10 is not statistically significant ($p = 0.17$), and MF + **SHE** performs on par with MF (0.1249 vs. 0.1226 in HR@10). This is consistent with Section 2.3, where we show that the warm model is a feasible solution of the composed objective, on which **SHE** can improve up to the optimization error. Section 4.3 investigates why the improvement differs between the datasets, and in relation to the characteristics of the content space.

Among the *Content Projection* methods, DenseRec is the most competitive on warm targets, on par with SASRec on Amazon (.0564 vs. .0568) and close to it on Onion (.3030 vs. .3375), as it retains the collaborative signal. Purely content-based methods are less consistent: MoRec collapses on Onion (.0015) and Linear Imitation underperforms on both datasets. SASRec + **SHE** outperforms both, even if sharing the same SASRec backbone architecture of DenseRec and Linear Imitation.

Finally, *Semantic ID* methods perform worse than SASRec + **SHE** on warm targets for both datasets (at best .0417 vs. .0745 on Amazon and .1737 vs. .3448 on Onion), suggesting that representing items solely through a small set of shared token embeddings is too restrictive.

4.2 COLD-TARGET RESULTS

Table 3 reports the results on sequences with *cold* targets, i.e., the held-out items omitted from the training data. Since our aim is generalization, we analyze these results alongside the warm ones (**RQ2**): accuracy on cold targets should not come at the cost of lower accuracy on warm targets.

Table 3: Cold-target performance. Same layout as Table 2. The reference and ID-only rows are omitted, since without a representation for a cold target these models score zero by construction.

	Amazon			Onion		
Method	HR@10	nDCG@10	HR@100	HR@10	nDCG@10	HR@100
Content Projections						
MoRec	**.0370**	**.0166**	.1374	.0000	.0000	.0051
Linear Imitation	.0096	.0049	.0615	.0004	.0001	.0221
DenseRec (linear)	.0001	.0000	.0436	.0033	.0016	.0430
Semantic IDs						
SID generative, RQ-VAE	.0007	.0003	.0073	.0260	.0183	.0477
SID generative, RQ-Kmeans	.0057	.0020	.0351	.0611	.0381	.1095
SID two-tower, RQ-VAE	.0085	.0039	.0664	<u>.1095</u>	<u>.0955</u>	**.1457**
SID two-tower, RQ-Kmeans	.0141	.0059	.1032	**.1132**	**.0957**	**.1457**
SHE (ours)						
MF + **SHE**	.0280	.0119	<u>.1488</u>	.0441	.0255	.1012
SASRec + **SHE**	<u>.0335</u>	<u>.0161</u>	**.1517**	.0980	.0943	<u>.1114</u>

On cold targets, SASRec + **SHE** is just behind the best-performing method on each dataset: MoRec on Amazon (HR@10 .0335 vs. .0370) and the two-tower Semantic ID model on Onion (.0980 vs. .1132). In both cases, the difference in nDCG@10 is not statistically significant (.0161 vs. .0166, $p = 0.088$ on Amazon; $\Delta = .0014$, $p = 0.52$ on Onion). As seen in Section 4.1, however, neither baseline carries this advantage over to warm targets, where both fall well behind SASRec + **SHE** (.0515 vs. .0745 on Amazon; .1497 vs. .3448 on Onion). By achieving the highest warm-target accuracy, while remaining statistically on par with the best cold-target method, SASRec + **SHE** offers the best balance between the two settings. The same pattern holds with an MF backbone: MF + **SHE** is the third most accurate cold-target method on Amazon (.0280), while more than doubling the warm-target accuracy of MF (.0379 vs. .0181).

[3]Here and in the rest of the results discussion, **SHE** denotes models trained with an embedding constructor based on the softmax-hull (Section 2.2) and combined with the single-pass approximation (Section 2.4).

DenseRec and Linear Imitation map content to the embedding space through a linear projection fitted on warm items, which does not transfer to cold items: on cold targets both remain far below SASRec + **SHE** and MF + **SHE** (DenseRec scores .0001 on Amazon and .0033 on Onion). Differently, a cold **SHE** embedding is instead a convex combination of warm embeddings, where the error is bounded by the content radius of its neighborhood, according to Proposition 3.

4.3 Ablations

Self-Share q (RQ3). We investigate the choice of q, which controls the weight of the induction loss through $\lambda = (1-q)/q$, and its effect on warm and cold HR@10 on both datasets. Figure 1 shows that, for the SASRec + **SHE**, accuracy on both warm and cold targets is highest at small values of q, with no trade-off between the two settings. At $q = 0$, SASRec + **SHE** outperforms the ID-based model on warm targets on Amazon (.0691 vs. .0568) and matches it on Onion (.3461 vs. .3375). Beyond $q = 0.1$, cold-target accuracy declines on both datasets, dropping to zero on Onion for $q \geq 0.3$. This is expected: as q grows, a cold item's embedding is increasingly dominated by its own untrained component, as in ID-embedding models. With an MF backbone, in contrast, we observe a trade-off, as no single value of q is optimal for both warm and cold targets: on Amazon, for instance, increasing q from 0.1 to 0.5 raises cold-target HR@10 from .0280 to .0366 while lowering warm-target HR@10 from .0379 to .0289.

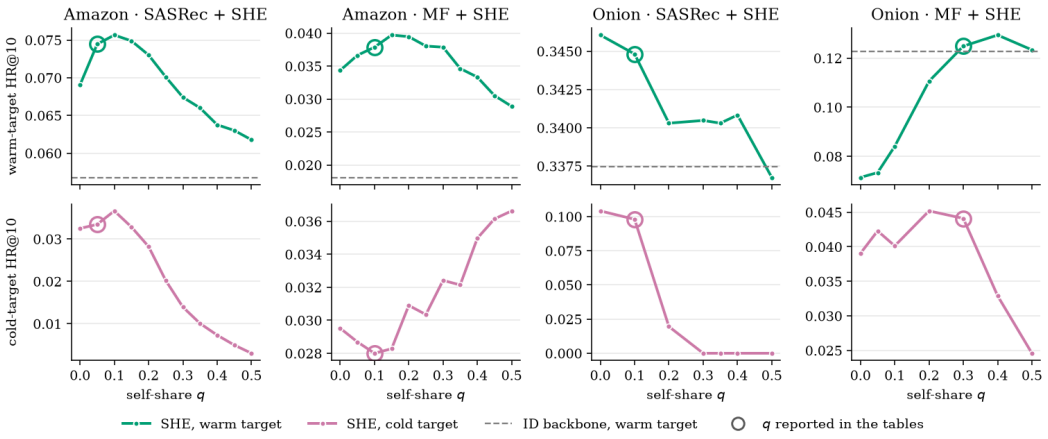


Figure 1: Warm-target (top) and cold-target (bottom) test HR@10 of the surrogate **SHE** as a function of q, for each dataset and backbone. The dashed line is the HR of the ID-embedding backbone on warm targets; there is no dashed line for cold targets since ID-embedding models are unable to recommend them (i.e. HR$= 0$). The open circle marks the value of q used for the results in Tables 2 and 3, based on our analysis during validation.

Initialization of E (RQ4). Table 4 compares the accuracy of SASRec + **SHE** when initializing E either randomly or from the E^0 of a pre-trained backbone, while the backbone itself is retrained from scratch. With random initialization, SASRec + **SHE** loses 4-6% of its warm-target HR@10 on both datasets (.0704 vs. .0745 on Amazon; .3301 vs. .3448 on Onion). On Onion, this places it slightly below its SASRec backbone (.3301 vs. .3375), whereas on Amazon it remains well above it (.0704 vs. .0568). On cold targets, random initialization leads to a 16% lower HR@10 on Amazon, whereas on Onion it yields a slightly higher HR@10 (.1063 vs. .0980), although the two Onion runs use different values of q (Table 4). As intended, the warm start keeps the embeddings in E close to E^0 via the constructor (Section 2.1); the initialization to E^0 matters for warm targets on both datasets and for cold targets on Amazon, the more aligned dataset.

Alignment (RQ5). Proposition 3 relates the cost of the **SHE** to the alignment between content and behavior. We measure alignment as the Pearson correlation ρ between distances using content representations and embeddings from a trained ID-embedding model (Appendix C.6). In Figure 2, we display the **SHE**'s gain, i.e., relative performance improvement over its backbone over warm targets (left plot, y axis), and the retention accuracy on cold targets, i.e., the cold-target accuracy

Table 4: Initialization of E in SASRec + **SHE**: the warm model's E^0 (Proposition 3) versus random initialization. Backbone weights ω are re-initialized in both cases. The random run uses $q = 0.05$ on both datasets; on Onion the warm-start grid has no $q = 0.05$ point, so its $q = 0.1$ row is shown.

			Warm Target			Cold Target		
Dataset	Init. of E	q	HR@10	nDCG@10	HR@100	HR@10	nDCG@10	HR@100
Amazon	Warm Model	0.05	.0745	.0344	.2371	.0335	.0161	.1517
Amazon	Random	0.05	.0704	.0320	.2319	.0281	.0127	.1351
Onion	Warm Model	0.1	.3448	.2096	.4926	.0980	.0943	.1114
Onion	Random	0.05	.3301	.2001	.4887	.1063	.0980	.1316

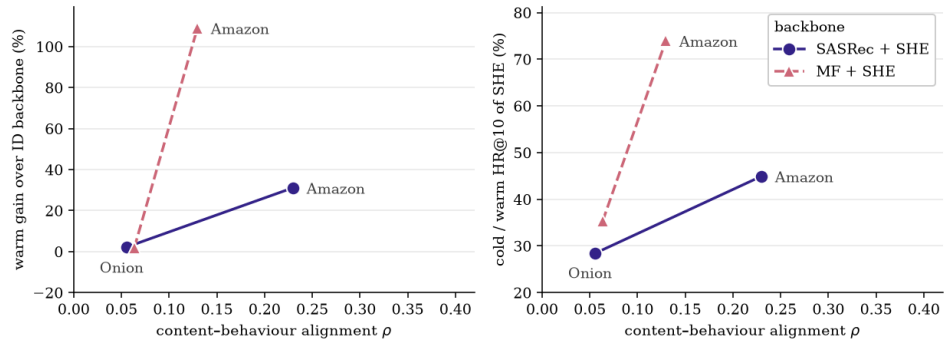


Figure 2: **SHE** accuracy gain and cold-target retention as a function of content–behavior alignment. Alignment is measured as Pearson correlation ρ of content and behavioral distances over random item pairs. Left: HR@10 gain of **SHE** over its ID-embedding backbone. Right: cold-target **SHE** retention, i.e., cold-target HR@10 as a fraction warm-target HR@10.

of the **SHE** as a fraction of its own warm-target accuracy. Both gain and retention are shown as a function of ρ (x axis). Content on Amazon is more aligned with the behavioral space ($\rho_{\text{SASRec}} = 0.23$, $\rho_{\text{MF}} = 0.13$) than content on Onion ($\rho = 0.06$ for both backbones). For both backbones, **SHE** achieves a larger warm-target gain over its backbone and retains a larger share of its warm accuracy on cold targets on the more aligned dataset (gain: +31% vs. +2% for SASRec, +109% vs. +2% for MF; retention: 45% vs. 28% and 74% vs. 35%), consistent with the qualitative implication of Proposition 3. We likewise attribute **SHE** only tying its backbone on Onion to this weaker alignment.

5 Conclusions and Future Work

In this paper, we formulated item cold start as a generalization problem of recommender systems and proposed an induction loss to address it. We provided theoretical bounds to the induction loss, relating them to the error obtained when constructing behavioral embeddings from content and to the alignment between the content and the behavioral spaces in which items are represented.

Based on these observations, we introduced a non-parametric constructor (**SHE**) for behavioral embeddings of cold items, noticing that the **SHE** falls into the broad category of methods that construct cold item embeddings from a combination of warm ones. Compared to previous methods, our embedding construction is the kernel-regression estimate of behavioral embeddings from content, with a reconstruction error bounded by the alignment between the two spaces. We introduced a surrogate to approximate the induction loss, while reducing the model forward passes to compute it, and derived theoretical upper bounds on the additional error due to this approximation. Finally, we empirically demonstrated that the generalization effects of induction loss and the embedding constructor lead to more accurate recommendations in both warm and cold scenarios.

Future work should combine induction loss and **SHE** within generative models for recommendation, and investigate **SHE** beyond unimodal content representations.

AI USE STATEMENT

In this work, we used generative AI tools to help formulate mathematical claims, assist in the writing of proofs, implement methods, support qualitative and thematic data analysis. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat dataset, interpret results. We have reviewed all AI-assisted work, ensuring the correctness of code, of proofs, and of edits suggested for the manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Artun Boz, Wouter Zorgdrager, Zoe Kotti, Jesse Harte, Panos Louridas, Vassilios Karakoidas, Dietmar Jannach, and Marios Fragkoulis. Improving Sequential Recommendations with LLMs. *ACM Transactions on Recommender Systems*, 4(2):19:1–19:35, 2025. doi: 10.1145/3711667. URL https://dl.acm.org/doi/10.1145/3711667.

Renqin Cai, Jibang Wu, Aidan San, Chong Wang, and Hongning Wang. Category-aware Collaborative Sequential Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pp. 388–397, New York, NY, USA, July 2021. Association for Computing Machinery. ISBN 978-1-4503-8037-9. doi: 10.1145/3404835.3462832.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. OneRec: Unifying Retrieve and Rank with Generative Recommender and Iterative Preference Alignment. arXiv:2502.18965, February 2025. URL https://arxiv.org/abs/2502.18965.

Hao Ding, Yifei Ma, Anoop Deoras, Yuyang Wang, and Hao Wang. Zero-shot recommender systems. arXiv:2105.08318, 2021. URL https://arxiv.org/abs/2105.08318.

Yijie Ding, Zitian Guo, Jiacheng Li, Letian Peng, Shuai Shao, Wei Shao, Xiaoqiang Luo, Luke Simon, Jingbo Shang, Julian McAuley, and Yupeng Hou. On the memorization and generalization of generative recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '26, pp. 921–932, New York, NY, USA, 2026a. Association for Computing Machinery. ISBN 9798400722592. doi: 10.1145/3770855.3818148. URL https://doi.org/10.1145/3770855.3818148.

Yijie Ding, Zitian Guo, Jiacheng Li, Letian Peng, Shuai Shao, Wei Shao, Xiaoqiang Luo, Luke Simon, Jingbo Shang, Julian McAuley, and Yupeng Hou. How well does generative recommendation generalize?, 2026b. URL https://arxiv.org/abs/2603.19809.

Christian Ganhör, Marta Moscati, Anna Hausberger, Shah Nawaz, and Markus Schedl. A multimodal single-branch embedding network for recommendation in cold-start and missing modality scenarios. In *Proc. of ACM RecSys*, Bari, Italy, 2024.

Christian Ganhör, Marta Moscati, Anna Hausberger, Shah Nawaz, and Markus Schedl. Single-branch network architectures to close the modality gap in multimodal recommendation. *ACM Transactions on Recommender Systems*, 2025. doi: 10.1145/3769430. URL https://doi.org/10.1145/3769430.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Mihajlo Grbovic and Haibin Cheng. Real-time personalization using embeddings for search ranking at airbnb. In *Proc. of ACM SIGKDD*, London, United Kingdom, 2018.

Danil Gusak, Nikita Sukhorukov, and Evgeny Frolov. Recommendation is a dish better served warm. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys '25, pp. 1267–1272, New York, NY, USA, 2025a. Association for Computing Machinery. ISBN

9798400713644. doi: 10.1145/3705328.3759331. URL https://doi.org/10.1145/3705328.3759331.

Danil Gusak, Anna Volodkevich, Anton Klenitskiy, Alexey Vasilev, and Evgeny Frolov. Time to split: Exploring data splitting strategies for offline evaluation of sequential recommenders. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys '25, pp. 874–883, New York, NY, USA, 2025b. Association for Computing Machinery. ISBN 9798400713644. doi: 10.1145/3705328.3748164. URL https://doi.org/10.1145/3705328.3748164.

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, September 2023. doi: 10.1145/3604915.3610639.

Ruining He and Julian McAuley. VBPR: Visual bayesian personalized ranking from implicit feedback. In *AAAI Conference on Artificial Intelligence*, 2016a. URL https://api.semanticscholar.org/CorpusID:3099285.

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*, WWW '16, pp. 507–517, 2016b. doi: 10.1145/2872427.2883037.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pp. 639–648, New York, NY, USA, July 2020. Association for Computing Machinery. ISBN 978-1-4503-8016-4. doi: 10.1145/3397271.3401063.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking LLMs as semantic encoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3251–3265, 2026.

Clark Mingxuan Ju, Liam Collins, Leonardo Neves, Bhuvesh Kumar, Louis Yufeng Wang, Tong Zhao, and Neil Shah. Generative Recommendation with Semantic IDs: A Practitioner's Handbook. arXiv:2507.22224, July 2025. URL http://arxiv.org/abs/2507.22224.

Wang-Cheng Kang and Julian McAuley. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206. IEEE Computer Society, 2018. ISBN 978-1-5386-9159-5. doi: 10.1109/ICDM.2018.00035.

Jinri Kim, Eungi Kim, Kwangeun Yeo, Yujin Jeon, Chanwoo Kim, Sewon Lee, and Joonseok Lee. Content-based Graph Reconstruction for Cold-start Item Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pp. 1263–1273, New York, NY, USA, July 2024. Association for Computing Machinery. ISBN 979-8-4007-0431-4. doi: 10.1145/3626772.3657801.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263.

Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 1748–1757, 2020. doi: 10.1145/3394486.3403226.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11523–11532, 2022.

Guohui Li, Li Zou, Zhiying Deng, and Qi Chen. Neighborhood-Enhanced Multimodal Collaborative Filtering for Item Cold Start Recommendation. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7815–7819, April 2024. doi: 10.1109/ICASSP48485.2024.10446911.

Zihan Li, Gustavo Escobedo, Marta Moscati, Oleg Lesota, and Markus Schedl. Adaptive autoguidance for item-side fairness in diffusion recommender systems. In *Proc. of ACM SIGIR*, Melbourne, Australia, 2026.

Jan Malte Lichtenberg, Antonio De Candia, and Matteo Ruffini. DenseRec: Revisiting Dense Content Embeddings for Sequential Transformer-based Recommendation. arXiv:2508.18442, August 2025. URL https://arxiv.org/abs/2508.18442.

Xinchen Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, Changqing Qiu, Jiaqi Zhang, Xu Zhang, Zhiheng Yan, Jingming Zhang, Simin Zhang, Mingxing Wen, Zhaojie Liu, Kun Gai, and Guorui Zhou. QARM: Quantitative Alignment Multi-Modal Recommendation at Kuaishou. arXiv:2411.11739, November 2024. URL https://arxiv.org/abs/2411.11739.

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*, SIGIR '15, pp. 43–52, 2015. doi: 10.1145/2766462.2767755.

Gregor Meehan and Johan Pauwels. On Inherited Popularity Bias in Cold-Start Item Recommendation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys '25, pp. 649–654, New York, NY, USA, September 2025. Association for Computing Machinery. ISBN 979-8-4007-1364-4. doi: 10.1145/3705328.3748035.

M. Jeffrey Mei, Florian Henkel, Samuel E. Sandberg, Oliver Bembom, and Andreas F. Ehmann. Semantic IDs for Music Recommendation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 1070–1073, September 2025. doi: 10.1145/3705328.3748139.

Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., USA, 1 edition, 1997. ISBN 0070428077.

Marta Moscati, Emilia Parada-Cabaleiro, Yashar Deldjoo, Eva Zangerle, and Markus Schedl. Music4All-onion - A large-scale multi-faceted content-centric music recommendation dataset. In *Proc. of ACM CIKM*, pp. 4339–4343, Atlanta, GA, USA, 2022.

Marta Moscati, Jan Malte Lichtenberg, Davide Abbattista, Antonio De Candia, Laura Boggia, and Matteo Ruffini. SwapRec: Warming Up Cold Items Through Training-Time Swaps. arXiv:2609.00913, 2026. URL http://arxiv.org/abs/2609.00913.

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, Prague, Czech Republic, September 2025. ACM. ISBN 979-8-4007-1364-4. doi: 10.1145/3705328.3748038.

Gustavo Penha, Edoardo D'Amico, Marco De Nadai, Enrico Palumbo, Alexandre Tamborrino, Ali Vardasbi, Max Lefarov, Shawn Lin, Timothy Heath, Francesco Fabbri, and Hugues Bouchard. Semantic IDs for Joint Generative Search and Recommendation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys '25, pp. 1296–1301, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 979-8-4007-1364-4. doi: 10.1145/3705328.3759300.

Aleksandr V. Petrov and Craig Macdonald. RecJPQ: Training Large-Catalogue Sequential Recommenders. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, pp. 538–547, New York, NY, USA, 2024. Association for Computing Machinery. doi: 10.1145/3616855.3635821. URL https://dl.acm.org/doi/10.1145/3616855.3635821.

Jordi Pons and Xavier Serra. musicnn: Pre-trained convolutional neural networks for music audio tagging. In *Late-Breaking/Demo Session of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.48550/arXiv.2305.05065.

Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender systems: Techniques, applications, and challenges. In *Recommender Systems Handbook*, pp. 1–35. Springer US, New York, NY, 2022. ISBN 978-1-0716-2197-4. doi: 10.1007/978-1-0716-2197-4_1. URL https://doi.org/10.1007/978-1-0716-2197-4_1.

Igor André Pegoraro Santana, Fabio Pinhelli, Juliano Donini, Leonardo Catharin, Rafael Biazus Mangolin, Valéria Delisandra Feltrim, Marcos Aurélio Domingues, et al. Music4all: A new music database and its applications. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 399–404. IEEE, 2020.

Markus Schedl, Stefan Brandl, Oleg Lesota, Emilia Parada-Cabaleiro, David Penz, and Navid Rekabsaz. LFM-2b: A dataset of enriched music listening events for recommender systems research and fairness analysis. In *Proc. of ACM SIGIR CHIIR*, pp. 337–341, Regensburg, Germany, 2022.

Jinseok Seol, Minseok Gang, Sang-goo Lee, and Jaehui Park. Proxy-based item representation for attribute and context-aware recommendation. In *Proc. of ACM WSDM*, Merida, Mexico, 2024.

Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua. Language Representations Can be What Recommenders Need: Findings and Potentials. arXiv:2407.05441, April 2025. URL https://arxiv.org/abs/2407.05441.

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*, CIKM '19, pp. 1441–1450, New York, NY, USA, November 2019. Association for Computing Machinery. ISBN 978-1-4503-6976-3. doi: 10.1145/3357384.3357895.

Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. DropoutNet: Addressing Cold Start in Recommender Systems. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual E5 text embeddings: A technical report. arXiv:2402.05672, February 2024. URL https://arxiv.org/abs/2402.05672.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, pp. 165–174, New York, NY, USA, July 2019. Association for Computing Machinery. ISBN 978-1-4503-6172-9. doi: 10.1145/3331184.3331267.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. Where to Go Next for Recommender Systems? ID- vs. Modality-based Recommender Models Revisited. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pp. 2639–2649, New York, NY, USA, July 2023. Association for Computing Machinery. ISBN 978-1-4503-9408-6. doi: 10.1145/3539618.3591932.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An End-to-End Neural Audio Codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, November 2021. ISSN 2329-9290. doi: 10.1109/TASLP.2021.3129994.

Weizhi Zhang, Yuanchen Bei, Liangwei Yang, Henry Peng Zou, Peilin Zhou, Aiwei Liu, Yinghui Li, Hao Chen, Jianling Wang, Yu Wang, Feiran Huang, Sheng Zhou, Jiajun Bu, Allen Lin, James Caverlee, Fakhri Karray, Irwin King, and Philip S. Yu. Cold-Start Recommendation towards the Era of Large Language Models (LLMs): A Comprehensive Survey and Roadmap. arXiv:2501.01945, January 2025. URL https://arxiv.org/abs/2501.01945.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pp. 1893–1902, 2020. doi: 10.1145/3340531.3411954.

Zhihui Zhou, Lilin Zhang, and Ning Yang. Contrastive Collaborative Filtering for Cold-Start Item Recommendation. In *Proceedings of the ACM Web Conference 2023*, WWW '23, pp. 928–937, New York, NY, USA, April 2023. Association for Computing Machinery. ISBN 978-1-4503-9416-1. doi: 10.1145/3543507.3583286.

A PROOFS

A.1 INDUCTION LOSS UPPER BOUND

Since we assume content representations c_i to be obtained from a pre-trained and frozen model, we omit the dependence on c_i in the loss functions, since those parameters are not updated during training. Let (ω^0, E^0) be the local minimum of the parameters and ID-embeddings, i.e., the values for a trained ID-based backbone model. We assume the loss $\mathcal{L}$ to be quadratically bound in the embedding table at the local minimum (ω^0, E^0) and denote with β the multiplicative factor regulating the quadratic upper bound:

$$\mathcal{L}(\omega^0, X) \leq \mathcal{L}(\omega^0, E^0) + \frac{\beta}{2} \sum_{i=1}^{N} n_i ||x_i - e_i^0||^2, \qquad X = (x_1, \ldots, x_N). \tag{11}$$

This assumption is motivated by the second-order Taylor expansion of $\mathcal{L}(\omega^0, E)$ around the local minimum E^0: since $\nabla_E \mathcal{L}(\omega^0, E^0) = 0$, the first-order term in the expansion vanishes, leaving the second-order term as the leading-order one. The above inequality follows when the Hessian is locally upper bounded by $\beta \cdot \mathrm{diag}(n_1 I, \ldots, n_N I)$, where I is the identity matrix of dimension equal to the embedding dimension d. This Hessian bound is a structural assumption on the loss and not a consequence of the expansion: it requires the curvature with respect to e_i to scale with the number of training occurrences n_i of item i. Both the quadratic bound and the local optimality of E^0, i.e., $\mathcal{L}(\omega^0, E^0) \leq \mathcal{L}(\omega^0, X)$, are local statements: we denote by $\mathcal{B}$ a neighborhood of E^0 on which both hold, and we assume throughout that the tables at which they are evaluated ($E^0_{i \leftarrow g}$ in Appendix A.1 and $W_q E^0$ in Appendix A.4) belong to $\mathcal{B}$. Let g be a constructor function $g(c_i, E_{\mathcal{I} \setminus i})$ that reconstructs the item embedding of item i from content c_i and from the embeddings of other catalog items $\mathcal{I} \setminus \{i\}$. Let $Q(E, g)$ be the interaction-weighted squared reconstruction error for the constructor g, over all catalog items:

$$Q(E, g) = \sum_{i=1}^{N} n_i ||g(c_i, E_{\mathcal{I} \setminus i}) - e_i^0||^2. \tag{12}$$

Then:

$$\begin{aligned} J(\omega^0, E^0, g) :=& \mathcal{L}(\omega^0, E^0) + \lambda \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\omega^0, E^0_{i \leftarrow g_{-i}(c_i, E^0)}) \\ \leq& \mathcal{L}(\omega^0, E^0) + \lambda \frac{1}{N} \sum_{i=1}^{N} \Big[\mathcal{L}(\omega^0, E^0) + \frac{\beta}{2} n_i ||g_{-i}(c_i, E^0) - e_i^0||^2 \Big] \\ =& (1 + \lambda) \mathcal{L}(\omega^0, E^0) + \frac{\lambda \beta}{2N} Q(E^0, g), \end{aligned} \tag{13}$$

which shows that with a small squared reconstruction error $Q(E, g)$, the inductive loss with $g(c_i, E_{\mathcal{I} \setminus i})$ stays close to the local minimum.

A.2 SOFTMAX-HULL-EMBEDDER (**SHE**) AS RECONSTRUCTION MINIMIZER UNDER SPACES ALIGNMENT.

In this paragraph, we show that for any content similarity function $s(c_i, c_j)$, fixed content neighborhood $\mathrm{kNN}_c(i)$, and temperature $\tau > 0$, the non-parametric kernel-regression estimate $g(c_i, E_{\mathcal{I}\setminus i})$ of the item embedding e_i from $\mathrm{kNN}_c(i)$ is given by the softmax-weighted average of the neighbors embedding:

$$g_{-i}(c_i) = \arg\min_{x \in \mathbb{R}^d} \sum_{j \in \mathrm{kNN}_c(i)} K_\tau(c_i, c_j) \, \|x - e_j\|_2^2, \qquad K_\tau(c_i, c_j) = \exp\left(\frac{s(c_i, c_j)}{\tau}\right), \quad (14)$$

where $s(c_i, c_j)$ denotes the content similarity between items i, j and $\mathrm{kNN}_c(i)$ denotes the k items other than i most similar to c_i according to $s(\cdot, \cdot)$, and $\tau > 0$ is a temperature parameter.

Proof. The gradient with respect to x of the l_2 reconstruction is given by $\nabla_x L(x) = 2 \sum_{j \in \mathrm{kNN}_c(i)} K_\tau(c_i, c_j)(x - e_j)$. Requiring the gradient to vanish at the point of local minimum, we get

$$x = \frac{\sum_{j \in \mathrm{kNN}_c(i)} K_\tau(c_i, c_j) e_j}{\sum_{\ell \in \mathrm{kNN}_c(i)} K_\tau(c_i, c_\ell)}. \quad (15)$$

Substituting the exponential kernel, we get $g_{-i}(c_i) = \sum_{j \in \mathrm{kNN}_c(i)} \bar{w}_{ij} e_j$ with

$$\bar{w}_{ij} = \frac{\exp\left(s(c_i, c_j)/\tau\right)}{\sum_{\ell \in \mathrm{kNN}_c(i)} \exp\left(s(c_i, c_\ell)/\tau\right)}, \quad (16)$$

or, equivalently, $\bar{w}_{ij} = \mathrm{softmax}_{j \in \mathrm{kNN}_c(i)}\left(\frac{s(c_i, c_j)}{\tau}\right)$. Finally, $K_\tau(c_i, c_j) > 0$, the Hessian of the reconstruction loss being positive definite (i.e. the loss is strictly convex), implies the stationary point above is the point of global minimum. □

Thus, for a fixed content neighborhood and exponential kernel, the softmax-weighted convex combination of the neighboring item embeddings is the unique minimizer of the least-squares objective. Hence, the softmax hull provides the kernel-regression estimator of the item embedding induced by the content similarity.

A.3 INDUCTION LOSS AND SPACES ALIGNMENT

Although the softmax weights are the kernel-regression reconstruction of the item embeddings from embeddings of content-similar neighbors, the quality of the reconstructed embedding is limited by the alignment between embedding and content space. Since item i's embedding with **SHE** is a weighted average over embeddings of i's content neighbors, i's reconstructed embedding will be close to e_i if items that are close in content have close embeddings, i.e. if the two spaces are aligned.

We make this alignment an explicit hypothesis, assuming that item embeddings are Λ-Lipschitz in content: $\|e_i^0 - e_j^0\| \leq \Lambda \, d(c_i, c_j)$ for all i, j, where $d(c_i, c_j)$ denotes the distance between items i and j in content space. Notice that the assumption implies that content neighbors have similar item embeddings, but not the opposite; items with similar embeddings may still be further apart in content space.

We now prove that under this assumption, denoting the softmax-hull weights with $\bar{W}$, i.e., softmax over k neighbors and 0 otherwise, the loss of the inductive recommender satisfies:

$$J_\lambda(\omega^0, E^0, \bar{W}) \;\leq\; (1 + \lambda)\, \mathcal{L}(\omega^0, E^0) \;+\; \frac{\lambda\, \beta}{2N} \Lambda^2 \sum_{i=1}^{N} n_i \, \bar{d}_i^2, \quad (17)$$

where $\bar{d}_i = \sum_{j \in \mathrm{kNN}_c(i)} \bar{w}(i)_j \, d(c_i, c_j)$ is the softmax-weighted average distance between i's neighbors and i in content space.

Proof. Using $\sum_{j\in \mathrm{kNN}_c(i)} \bar{w}_{ij} = 1$, the triangle inequality, and the alignment assumption, we have

$$\begin{aligned}
\|e_i^0 - (\bar{W}E^0)_i\| &= \| \sum_{j\in \mathrm{kNN}_c(i)} \bar{w}_{ij} e_i^0 - \sum_{j\in \mathrm{kNN}_c(i)} \bar{w}_{ij} e_j^0 \| \\
&= \| \sum_{j\in \mathrm{kNN}_c(i)} \bar{w}_{ij} (e_i^0 - e_j^0) \| \\
&\leq \sum_{j\in \mathrm{kNN}_c(i)} \bar{w}_{ij} \|e_i^0 - e_j^0\| \\
&\leq \Lambda \sum_{j\in \mathrm{kNN}_c(i)} \bar{w}_{ij}\, d(c_i, c_j) \\
&\leq \Lambda\, \bar{d}_i.
\end{aligned} \tag{18}$$

Squaring Equation (18) and summing over all items i, we get $Q(E^0, \bar{W}) \leq \Lambda^2 \sum_i n_i \bar{d}_i^2$. Finally, substituting into Equation (13) gives Equation (17). □

A.4 Single-pass training

To reduce the training requirements of **SHE**, we propose to evaluate the warm loss *once* on the mixed table

$$W_q E \;=\; q\,E + (1-q)\,\bar{W}E, \qquad (W_q E)_i = q\,e_i + (1-q)\,g(c_i, E_{\mathcal{I}\setminus i}), \tag{19}$$

where the parameter $q \in (0,1)$ depends on λ and allows for the hull of item i to include the embedding e_i of the item itself. While this constitutes an approximation to the actual inductive loss in Equation (1), in the next paragraph, we quantify the discrepancy between the inductive loss and the warm loss computed with the mixed embedding E, and around the local minimum of the warm loss.

Substituting Equation (19) in the assumption on the local quadratic upper bound for the loss (Equation (11)) and around a minimizer E^0 of the warm loss, we find

$$\mathcal{L}(\omega^0, E^0) \leq \mathcal{L}(\omega^0, W_q E^0) \leq \mathcal{L}(\omega^0, E^0) + \tfrac{\beta}{2}(1-q)^2 Q(E^0, \bar{W}). \tag{20}$$

Similarly, for the inductive recommender loss and $q = 1/(1+\lambda)$, we get

$$\frac{J_\lambda}{1+\lambda} = q\,\mathcal{L}(\omega^0, E^0) + (1-q)\,\frac{1}{N}\sum_i \mathcal{L}(\omega^0, E^0_{i\leftarrow(\bar{W}E^0)_i}), \tag{21}$$

$$\mathcal{L}(\omega^0, E^0) \leq \frac{J_\lambda}{1+\lambda} \leq \mathcal{L}(\omega^0, E^0) + (1-q)\,\tfrac{\beta}{2N} Q(E^0, \bar{W}), \tag{22}$$

where the lower bound originates from E^0 being a local minimizer of $\mathcal{L}$ on $\mathcal{B}$, after summing over items i, while the upper bound originates from the limit on J_λ (Equation (13)) and from $\lambda \geq 0$.

Since by Equations (20) and (22) both $\mathcal{L}(\omega^0, W_q E^0)$ and $J_\lambda/(1+\lambda)$ lie in intervals with common left endpoint $\mathcal{L}(\omega^0, E^0)$, they differ by at most the larger interval width, $\frac{\beta}{2}(1-q)\max\{1-q, 1/N\}\, Q(E^0, \bar{W}) \leq \frac{\beta}{2}(1-q)\, Q(E^0, \bar{W})$; multiplying by $1+\lambda = 1/q$ and using $(1-q)/q = \lambda$ gives:

$$|J_\lambda(\omega^0, E^0, \bar{W}) - (1+\lambda)\mathcal{L}(\omega^0, W_q E^0)| \leq \frac{\lambda\beta}{2} Q(E^0, \bar{W}). \tag{23}$$

This sets an upper bound on the difference between the warm loss evaluated with the single embedding table $W_q E^0$ and the loss at the local minimum (Equation (20)), and the computationally expensive inductive recommendation loss.

B Existing Embedding Methods as Weighted-Sum Reconstructions

Several item embedding construction methods can be regarded as a weighted average $E = W \cdot A$ of auxiliary vectors; here, $E \in \mathbb{R}^{N\times d}$ is the final item embedding table for N catalog items,

with embedding dimension d, $A \in \mathbb{R}^{M \times d}$ is the table of d-dimensional auxiliary vectors, and $W \in \mathbb{R}^{N \times N}$ is the table of weights, determining the linear decomposition of each item i on the atomic vectors. Notice that the number of atomic vectors and catalog items might differ, $M \neq N$.

In this section we show how several embedding construction methods fall into this more general formalization, including our **SHE**.

ID-Embeddings. ID-embedding-based models Koren et al. (2009); Kang & McAuley (2018); Sun et al. (2019) associate catalog items to randomly initialized embedding e_i. This limit case is reached when weights is the identity matrix, $W_{im} = \delta_{im}$, and the auxiliary matrix A is the model's embedding table itself, E.

Since ID-embeddings represent each item through distinct embeddings that are updated during training, cold items will stay at their initial, randomly-initialized values, leading to worse recommendations.

Several approaches to tackle this situation disentangle *which* embeddings an item is made of (decided by W), from *what* those embeddings encode (decided by A). Below, we discuss several instances of such approaches to item cold start, within the $E = W \cdot A$ formalism.

ID-Embedding Reconstruction from Content.

Linear Maps from Content. One family of methods for item cold start Volkovs et al. (2017); Lichtenberg et al. (2025) embeds items by constructing a map f_θ from the content representations, $e_i = f_\theta(c_i)$. For a same f_θ, models of this family can be differentiated according to the loss function and training procedure. If the encoder is a linear function, $e_i = f_\theta(c_i) = Wc_i$.

A limiting case Yuan et al. (2023) of this class of models, with $d = d_c$ and W the identity matrix, uses a frozen ID embedding table corresponding to frozen item content representations $e_i = c_i$. During training, only the model's parameter ω are optimized.

Embeddings from Content-Based Swaps. For heuristic-based swap approaches Grbovic & Cheng (2018); Seol et al. (2024); Moscati et al. (2026), the auxiliary embeddings are those of warm items, $A = E_{\text{warm}} \in \mathbb{R}^{N_{\text{warm}} \times d}$. The weight matrix W is then the identity for warm items, and it is a sparse binary weights matrix $W_{\text{cold}} \in \mathbb{R}^{N_{\text{cold}} \times N_{\text{warm}}}$ for cold items, for which item embeddings are replaced by an embedding of the warm ones.

Embeddings from Semantic IDs. The embedding decomposition $E = W \times A$ can be generalized to a tensor form as

$$e_i = \sum_m W_{ilm} \cdot A_{lm}, \qquad A \in \mathbb{R}^{L \times M \times d}, \quad W \in \mathbb{R}^{N \times L \times M} \tag{24}$$

where L is the number of sources and M is the number of auxiliaries per source. When the weights W in Equation 24 capture semantic information on catalog items, Equation (24) can be seen as a formulation of semantic IDs:

$$W_{ilm} = \delta_{mm^*}, \quad m^* = \arg\min_m \text{sim}(r_{i,l-1}, A_{lm}), \quad r_{i,l} = r_{i,l-1} - A_{lm^*} \tag{25}$$

where r_{i0} is a representation of the content of item i, optionally down-projected with an encoder architecture as the one of RQ-VAE, and A_{lm} are the learnable embeddings for token m in semantic layer l.

C Experimental details

C.1 Preliminary Analysis of the Content Space

Figure 3 shows t-SNE projections of a random sample of content vectors for both datasets. The Amazon embeddings exhibit a more pronounced cluster structure than those of Onion. To quantify the structure of the space, we also compute the median cosine similarity between randomly sampled pairs of vectors, obtaining 0.7986 for Amazon and 0.5612 for Onion.



(a) Amazon (b) Onion

Figure 3: t-SNE plot of content embeddings.

C.2 Baselines

MF Koren et al., 2009 and **SASRec** (Kang & McAuley, 2018) are the representative of the ID-embedding model class. Both models are trained with binary cross-entropy with negative samples. We use cosine similarity with temperature as scoring function as preliminary experiments showed it outperformed the original dot-product. Other transformer-based models use the same architecture and training strategy of SASRec's.

MoRec Yuan et al. (2023) uses the same architecture and training strategy of SASRec, and differs only in the item embedding table, which is replaced with the frozen table representing item embeddings extracted from the content.

Both **DenseRec** (Lichtenberg et al., 2025) and **Linear Imitation** use a single linear projection from the content to the item embedding, leaving the rest of the architecture the same as for SASRec. DenseRec also uses the same training strategy, with the content-projection and ID-embedding paths alternating at random during training. Linear Imitation substitutes the ID embeddings of a pre-trained SASRec with their closed-form ridge regression from content.

Both SID approaches are tested with two state-of-the-art tokenizers: RQ-VAE (Lee et al., 2022) and RQ-Kmeans Penha et al. (2025). **SID generative** Rajput et al. (2023) relies on an encoder–decoder transformer trained over the token sequences and uses constrained beam search over the trie of catalog items with a beam five times the largest cutoff.

SID two-tower Petrov & Macdonald (2024) constructs for each layer of the SIDs an embedding table and constructs item embeddings as the average of the embeddings of its tokens; the SID embeddings are optimized during training together with the backbone transformer, using the same architecture and training of SASRec.

C.3 Hyperparameters

Table 5 lists the hyperparameter for every model–dataset pair, selected according to the values reported in the reference paper. All models (aside from the closed-form Linear Imitation, which requires no training) use the Adam optimizer with the listed learning rate and batch size, a maximum of 300 epochs and early stopping on validation HR@10 with best-weight restoration. The early-stopping patience is set to 20 with evaluation every epoch for the baselines; for models with **SHE** the patience is set to 4 with evaluation every 5 epochs, to reduce **SHE** evaluation time.

All SASRec-based rows of a dataset share the same architecture, which is the one reported in the first row.

Table 5: Hyperparameters of every model. d is the item embedding dimension, "negs" the number of uniformly sampled negatives per positive, T the temperature applied to the cosine logits during training. Content embeddings are ℓ_2-normalized before quantization in the semantic-ID rows.

Model	Hyperparameter	Amazon	Onion
SASRec (shared by all SASRec-based rows Kang & McAuley (2018))	d / FFN width	64 / 256	256 / 256
	blocks / heads / dropout	2 / 2 / 0.5	2 / 2 / 0.5
	max. sequence length	50	40
	negs / head / T	64 / cosine / 0.05	128 / cosine / 0.05
	batch / learning rate	256 / 10^{-3}	512 / 10^{-3}
	weight decay	0	0
	tied item tables / learned positions	yes / yes	yes / yes
MF Koren et al. (2009)	d / negs / head / T	64 / 64 / cosine / 0.05	256 / 128 / cosine / 0.05
	batch / learning rate / weight decay	16,384 / 10^{-2} / 10^{-4}	16,384 / 10^{-2} / 10^{-4}
	epochs (ceiling)	300	40
MoRec Yuan et al. (2023)	item vector	frozen e5, $d = 384$	frozen musicnn, $d = 50$
	hidden width / batch	64 / 256	50 / 4,096
DenseRec (linear) Lichtenberg et al. (2025)	projection / dense-path prob.	linear / 0.5	linear / 0.5
Linear Imitation	ridge λ / target table	0.01 / SASRec	0.01 / SASRec
SID generative (RQ-VAE) Rajput et al. (2023)	levels $\times$ codebook	3×256	3×256
	latent / encoder widths	64 / 512-256-128	32 / 256-128-64
	β / epochs / lr / STE	0.25 / 400 / 10^{-3} / yes	0.25 / 40 / 10^{-3} / yes
	dead-code re-init	every 5 epochs	every 5 epochs
	seq2seq	d_{model} 128, 4+4 layers, FFN 1024, dropout 0.1, beam factor 5	
SID generative (RQ-Kmeans) Ju et al. (2025)	levels $\times$ codebook / k-means iters	3×256 / 3,000	3×256 / 3,000
	seq2seq	as above	
SID seq2seq training Rajput et al. (2023)	heads / item max. length	6 / 20	8 / 40
	batch / lr / weight decay	8,192 / 10^{-3} / 10^{-4}	256 / 10^{-3} / 10^{-4}
SID two-tower Petrov & Macdonald (2024)	tokens / weights / init. of E	same quantizers / $1/L$ / random	same / $1/L$ / random
SHE (ours)	neighbors $k-1$ / similarity	63 / content cosine	63 / content cosine
	τ, fitted by leave-one-ID-out reconstruction of E^0 (SASRec / MF)	0.0275 / 0.0343	0.0715 / 0.1052
	q (SASRec / MF), selected on validation	0.05 / 0.1	0.1 / 0.3
	init. of E / weight decay on E (MF)	E^0 / 10^{-4}	E^0 / 10^{-4}

C.4 Selection of the self-share on validation

Table 6 gives the validation HR@10 of every trained **SHE** on the q grid. The curves are nearly flat: over the whole grid the SASRec + **SHE** moves within 0.010 on Amazon and 0.007 on Onion, and the MF+ **SHE** within 0.007 on Amazon; only the MF+ **SHE** on Onion has a clear trend, rising with q. Since the validation HR is almost flat in q, we select the optimal q as the smallest value with a validation HR within 1.96 standard errors of the best value. This criterion was set before the final evaluation on the test set and without using cold items. Notice that selecting the q with the best test performance would result in other values, instead (See Figure 1). For instance on Amazon and for SASRec+ **SHE**, the best test HR@10 is reached for $q = 0.1$ (0.0757), whereas our criterion selects $q = 0.05$ (0.0745).

Table 6: Validation HR@10 of the fixed-share **SHE** along the q grid. Best value per row underlined; the selected operating point in bold.

Dataset	backbone	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Amazon	SASRec	.0675	**.0712**	<u>.0731</u>	.0718	.0696	.0691	.0685	.0667	.0654	.0639	.0632
Amazon	MF	.0328	.0347	**.0368**	.0374	.0373	<u>.0381</u>	.0371	.0353	.0336	.0322	.0312
Onion	SASRec	.4126	–	**.4155**	–	.4192	–	.4190	<u>.4198</u>	.4195	–	.4168
Onion	MF	.0776	.0781	.0897	–	.1223	–	**.1415**	–	.1440	–	<u>.1448</u>

C.5 Statistical tests

We report final results as mean and 95% confidence interval over test users in Table 7 and the results of the paired one-sided Wilcoxon signed-rank test between the best and second-best models in Table 8.

On the warm targets, the gap between SASRec + **SHE** and SASRec is statistically significant (Table 8) on Amazon ($p = 2.5 \times 10^{-92}$) and not statistically significant on Onion ($p = 0.17$). On the cold targets, the best-performing method on each dataset (MoRec on Amazon, SID two-tower with RQ-Kmeans on Onion) is not statistically distinguishable from SASRec + **SHE** ($p = 0.088$ and $p = 0.52$, respectively).

Table 7: HR@10 with its 95% confidence interval over test users, for every method of Tables 2 and 3.

	Amazon		Onion	
Method	warm target	cold target	warm target	cold target
users n	58,184	14,498	11,293	2,766
Reference				
Random	.0001 ± .0001	.0000 ± .0000	.0002 ± .0002	.0000 ± .0000
MostPop	.0061 ± .0006	.0000 ± .0000	.0060 ± .0014	.0000 ± .0000
ID embeddings				
MF	.0181 ± .0011	.0000 ± .0000	.1226 ± .0061	.0000 ± .0000
SASRec	.0568 ± .0019	.0000 ± .0000	.3375 ± .0087	.0000 ± .0000
Content projections				
MoRec	.0515 ± .0018	.0370 ± .0031	.0015 ± .0007	.0000 ± .0000
Linear Imitation	.0103 ± .0008	.0096 ± .0016	.0020 ± .0008	.0004 ± .0007
DenseRec (linear)	.0564 ± .0019	.0001 ± .0001	.3030 ± .0085	.0033 ± .0021
Semantic IDs				
SID generative, RQ-VAE	.0417 ± .0016	.0007 ± .0004	.1129 ± .0058	.0260 ± .0059
SID generative, RQ-Kmeans	.0356 ± .0015	.0057 ± .0012	.1737 ± .0070	.0611 ± .0089
SID two-tower, RQ-VAE	.0233 ± .0012	.0085 ± .0015	.1504 ± .0066	.1095 ± .0116
SID two-tower, RQ-Kmeans	.0280 ± .0013	.0141 ± .0019	.1497 ± .0066	.1132 ± .0118
SHE (ours)				
MF + **SHE**	.0379 ± .0016	.0280 ± .0027	.1249 ± .0061	.0441 ± .0077
SASRec + **SHE**	.0745 ± .0021	.0335 ± .0029	.3448 ± .0088	.0980 ± .0111

Table 8: Paired one-sided Wilcoxon signed-rank tests on per-user nDCG@10 between the leading pair of each aggregate (p is one-sided, alternative: first $>$ second; Δ is the difference of per-user means).

Dataset	target	first	second	ΔnDCG@10	n	p
Amazon	warm	SASRec + **SHE**	SASRec	+0.0093	58,184	2.5×10^{-92}
Amazon	cold	MoRec	SASRec + **SHE**	+0.0006	14,498	0.088
Onion	warm	SASRec + **SHE**	SASRec	+0.0023	11,293	0.17
Onion	cold	SID two-tower (RQ-Kmeans)	SASRec + **SHE**	+0.0014	2,766	0.52

C.6 Alignment measure

We aim to provide a measurement of the alignment between the content and the behavioral spaces, using the item embeddings after each model training to measure distance in behavioral space. For each trained ID backbone we draw random pairs of warm items and compute the cosine distance between their content embeddings and the cosine distance between their trained behavioral embeddings E^0. We measure spaces alignment as ρ Pearson correlation over the sampled pairs. Amazon text embeddings and the behavioral embeddings E^0 are the most aligned pair ($\rho = 0.23$ for SASRec; $\rho = 0.13$ for MF); the audio embeddings of Onion are nearly uncorrelated with either backbone's geometry ($\rho = 0.06$ for SASRec). We use ρ as a descriptive proxy for the alignment that Proposition 3 quantifies through Λ; it is not an estimate of Λ, which is a worst-case constant over all item pairs.