Domain Adaptation for Cold-Start Users in Sequential Recommendation

Anonymous authors
Paper under double-blind review

Abstract

Sequential recommendation tracks users' preferences over time based on users' historical activities and makes prediction on their next most probable action. However, this approach faces limitations when dealing with cold-start users who possess minimal interaction data, leading to difficulty in learning their preferences. To address this challenge, by taking regular users with longer interaction histories and cold-start users as two domains, this paper introduces domain adaptation techniques to narrow the performance gap caused by knowledge shifts in domains. We propose a dual-transformer framework with separate models for long (source) and short (target) sequences, collaboratively trained with shared item embeddings. To enable effective knowledge transfer, we introduce an emulated target domain by sampling short sequences from the source, and apply contrastive learning to align their contextual representations. To further improve adaptation under complex knowledge shifts, we reduce item popularity bias and incorporate user similarity into the contrastive loss. Experiments on five public datasets show consistent improvements over strong baselines, demonstrating the robustness of our approach under both length shifts and compounded shifts involving item distribution changes. Our code can be found at https://anonymous.4open.science/r/DACSR-1.

1 Introduction

Recommender systems help users navigate overwhelming item choices by modeling preferences from historical interactions. Traditional methods such as collaborative filtering Wang et al. (2006); Hu et al. (2008) and generative models Liang et al. (2018) typically assume past interactions are equally important. In contrast, modern sequential recommendation methods model users' evolving interests using sequence models, particularly transformers Wang et al. (2019); Fang et al. (2020); Hidasi et al. (2016); Donkers et al. (2017); Hidasi & Karatzoglou (2018); Kang & McAuley (2018); Li et al. (2020); Sun et al. (2019); Xie et al. (2021). A major challenge arises in handling cold-start users, those with sparse history, compared to regular users who have rich behavioral data. Standard models, trained on all users, tend to be dominated by regular users, leading to biased learning and poor generalization to cold-start users.

We tackle the cold-start sequential recommendation problem through the lens of domain adaptation (DA), framing regular users as the source domain and cold-start users as the target domain. This perspective highlights two key types of domain shifts: length shift, where cold-start users have much shorter interaction histories, and complex shift, where item distributions also differ due to temporal item evolution, in addition to sequence length differences.

To address these challenges, we propose a contrastive DA framework built on dual transformer networks: one specialized for long sequences in the source domain, and the other for short sequences in the target domain. To bridge the gap between domains, we introduce an emulated target domain by sampling short subsequences from the source. We then apply contrastive learning to align the contextual representations of these sampled sequences with their original long counterparts, enabling effective and controllable knowledge transfer. To improve adaptation under complex shifts, we further reduce item popularity bias in user representations and incorporate user similarity into the contrastive objective. These enhancements progressively increase

robustness to distributional shifts and promote more personalized, domain-invariant knowledge alignment. Our contributions are as follows:

- We reformulate the cold-start problem in sequential recommendation as a domain adaptation task, identifying key knowledge shifts (length and compound shifts).
- We propose a contrastive learning framework with three model variants (DACSR, DACSR+, DACSR++) to progressively enable controllable knowledge transfer from regular to cold-start users.
- We introduce an emulated target domain and align representations through contrastive learning to mitigate biased learning and facilitate robust adaptation.
- Experiments on five public datasets demonstrate that the proposed models consistently outperform baselines under both length and complex shift settings, whereas standard transfer learning fails under complex shift.

2 Related Work

2.1 Sequential Recommendation

Traditional recommendation methods model collaborative signals using neighborhood-based approaches like item/user KNN Deshpande & Karypis (2004); Wang et al. (2006) or latent factorization techniques such as Matrix Factorization (MF) Hu et al. (2008); Rendle et al. (2009), with SLIM Ning & Karypis (2011) enhancing robustness via sparsity constraints. Deep learning extends these methods, including neural MF He et al. (2017), VAE-based collaborative filtering Liang et al. (2018), and attention-based models like DIN Zhou et al. (2018). Sequential recommenders capture temporal behavior for next-item prediction. Early approaches rely on Markov Chains and hybrids like FPMC and MFMC Shani et al. (2002); Rendle et al. (2010); He & McAuley (2016). Deep models improve this by learning complex dependencies: Caser and SimpleCNN use CNNs Tang & Wang (2018); Yuan et al. (2019a), while RNNs Hidasi et al. (2016); Donkers et al. (2017); Hidasi & Karatzoglou (2018) and transformer-based models like SASRec, BERT4Rec, and TiSASRec Kang & McAuley (2018); Sun et al. (2019); Li et al. (2020) have become dominant.

Beyond architecture, contrastive learning (CL) and self-supervision further enhance sequential models. CL4SRec Xie et al. (2021) applies a multitask contrastive objective. S3-Rec Zhou et al. (2020) aligns auxiliary attributes with interaction signals via contrastive loss. DuoRec Qiu et al. (2022) improves representation quality by selecting sequences with shared terminal items as positives. ICLRec Chen et al. (2022) clusters sequences into intent prototypes and enforces consistency between sequences and their intents. Inspired by these, we incorporate contrastive learning to reduce the representation gap between source and emulated target domains, enabling better transfer of user preferences across domains.

2.2 Cold-start Problem

Solutions to the cold-start problem generally fall into two categories: incorporating additional information and improving model generalization.

2.2.1 Incorporating Additional Information

A common approach is to leverage side information (e.g., user profiles or item content) in collaborative filtering Gantner et al. (2010); Li et al. (2019) and deep models Barkan et al. (2019); Zhu et al. (2020); Wei et al. (2021), aligning auxiliary features with collaborative embeddings to compensate for missing interactions. Contrastive learning has been used for feature alignment across datasets Wei et al. (2021), while domain adaptation Zhu et al. (2021); Khan et al. (2017) enables knowledge transfer from related tasks. In particular, cold-start scenarios where source and target domains share users have been studied in Chen et al. (2023); Zhao et al. (2020); Yuan et al. (2019b); Chen et al. (2021).

2.2.2 Model Generalization

Another strategy is to improve model robustness to unseen users/items. For instance, Du et al. (2020) addresses embedding discrepancies between training and test items, and DropoutNet Volkovs et al. (2017)

enhances generalization by randomly masking collaborative signals during training. Meta-learning methods Hospedales et al. (2022) such as Lee et al. (2019); Vartak et al. (2017); Wei et al. (2020) adopt MAML Finn et al. (2017) to learn models that adapt quickly to new users with few interactions. MetaTL Wang et al. (2021) extends this to dynamic, sequential settings, while Mecos Zheng et al. (2021) employs a prototypical network with a transformer backbone to tackle sequential cold-start item recommendation. Building on the generalization view, we treat cold-start and regular users as two domains with distribution shifts and propose a model that explicitly controls cross-domain knowledge transfer through domain adaptation.

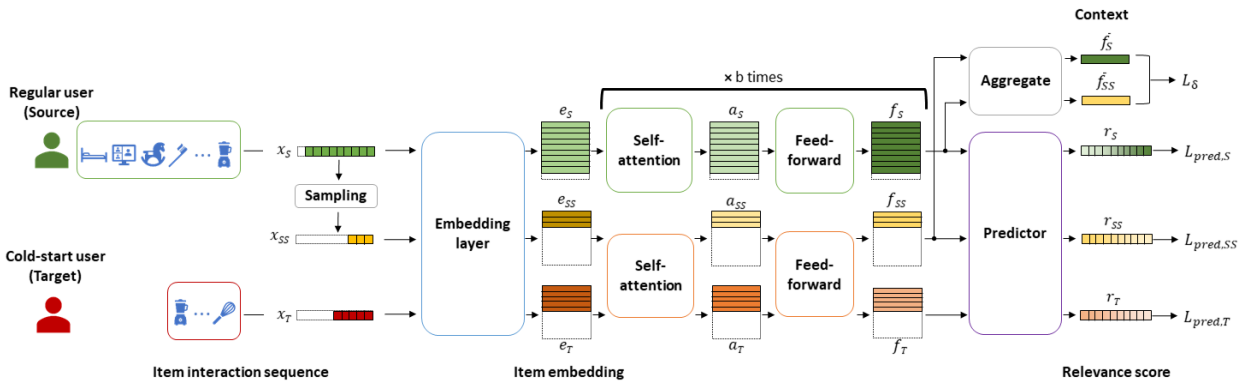

Figure 1: The DACSR framework consists of four main groups of components to predict the relevance of items at the next interaction: A subsampling operator ('Sampling') on the source (S domain) to generate an emulated cold-start target or sampled source (SS domain), embedding layer and predictor layer shared by all domains, b transformer blocks for the source and another b blocks shared by the emulated cold-start target and the target (T domain), and a measure of divergence δ between the contexts of the original source and the emulated target.

3 Methodology

We address the cold-start sequential recommendation problem, where users with limited interaction histories (cold-start users) coexist with users who have rich behavioral data (regular users). In our setting, sequences from regular users define the source domain, while those from cold-start users define the target domain. This domain shift is characterized by differences in sequence lengths and potentially item distributions over time.

3.1 Knowledge Shift and Transfer

3.1.1 Length Shift:

We propose a dual-network framework comprising two domain-specific networks for the source and target. These networks share item embeddings but maintain separate parameters, enabling each to specialize in its respective domain while allowing joint learning. To facilitate effective knowledge transfer from long source to short target sequences, we construct an emulated target domain by randomly sampling from source sequences with a controllable probability. This intermediate domain closely approximates the true target distribution, increasing target-side diversity and enhancing generalization.

3.1.2 Complex Shift with Length and Item Changes:

In more realistic settings where both sequence lengths and item distributions differ, we enhance the emulated domain with domain-invariant techniques to better regulate knowledge transfer. To address item distribution shifts, we propose to compute item popularity separately in each domain and generate popularity-weighted user representations in the domain alignment, reducing sensitivity to uneven item exposure. Furthermore, we introduce a user similarity-aware contrastive alignment to refine the adaptation process by promoting consistency between similar users across domains.

3.2 Dual-network Architecture

We propose DACSR (Domain Adaptation for Cold-Start Recommendation), a dual-transformer framework explicitly designed to transfer knowledge from regular to cold-start users. As illustrated in Figure 1, DACSR employs two domain-specific networks to model the distinct data distributions of the source and target domains along with an emulated target domain, formed by randomly sampling from source sequences with a controllable probability. The dual network of source and target share item embeddings with separate parameters, allowing each to specialize in its respective domain while training jointly. Both source and target adopt transformer architectures Kang & McAuley (2018); Zhang et al. (2019) due to their strength in modeling long-range dependencies.

Let U and V denote the sets of users and items, respectively. A user $i \in U$ has a chronologically ordered interaction sequence $x_i = (v_1, v_2, \ldots, v_L)$, where $v_l \in V$ is a one-hot vector denoting the item interacted at time l. The goal of sequential recommendation is to predict the next item a user is likely to interact with, using only historical sequences and without side information. Each item $v \in V$ is embedded into a d-dimensional vector using an embedding matrix $\mathbf{M} \in \mathbb{R}^{|V| \times d}$. Positional information is added using a trainable matrix $\mathbf{P} \in \mathbb{R}^{L \times d}$, resulting in the final sequence embedding:

$$\mathbf{E} = [\mathbf{m}_{v_1} + \mathbf{p}_1, \mathbf{m}_{v_2} + \mathbf{p}_2, \ldots, \mathbf{m}_{v_L} + \mathbf{p}_L], \tag{1}$$

where $\mathbf{m}_{v_l} \in \mathbf{M}$ and $\mathbf{p}_l \in \mathbf{P}$. Sequences are padded or truncated to a fixed length L.

The embedding $\mathbf{E}$ is passed through b stacked transformer blocks. Each block consists of a self-attention layer followed by a feed-forward layer. The attention is computed via:

$$\mathbf{S} = \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \tag{2}$$

where $\mathbf{Q} = \mathbf{E}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{E}\mathbf{W}^K$, $\mathbf{V} = \mathbf{E}\mathbf{W}^V$, and $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d}$ are learnable parameters. To preserve the autoregressive property of sequential recommendation, we apply a causal attention mask to prevent a position from attending to future tokens.

Each output $\mathbf{s}_l \in \mathbf{S}$ is passed through a feed-forward network:

$$\boldsymbol{f}_l = \text{ReLU}(\mathbf{s}_l\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{3}$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^d$ are shared across all positions. The output of the final transformer block at position l is used to compute the preference score for the next item using a dot product with the item embedding:

$$\boldsymbol{r}_{l,v} = \boldsymbol{f}_l \cdot \mathbf{m}_v, \tag{4}$$

where $\mathbf{m}_v \in \mathbf{M}$. We reuse the item embedding matrix in the output layer for efficiency and to reduce overfitting, following Kang & McAuley (2018). This dual-network structure, with shared embeddings and a common predictor, allows the model to capture domain-specific patterns while enabling shared learning across both domains. The following sections will describe how we align these domains using contrastive learning and explicitly regulate the transfer process.

3.3 Emulated Cold-Start Target Domain

To explicitly control knowledge transfer from long to short sequences, we introduce an emulated target domain by sampling short sequences from long user histories in the source domain. These sampled sequences, representing cold-start behaviors, are input into the target model, while their original long counterparts are processed by the source model. By aligning the contextual representations of similar sampled and long sequences, we aim to regulate the amount of knowledge transferred.

As illustrated in Figure 1, we apply a sampling operator to each long sequence from regular users using a uniform probability p, a hyperparameter that controls the average length of the emulated short sequence. The sampled interactions are ordered chronologically to form the sampled source (SS) sequence. For each

user, the original long sequence (S) is processed by the source model, while the sampled short sequence (SS) and the actual target sequence (T) are processed by the target model. The SS domain serves as an intermediate bridge between the source (S) and target (T) domains.

3.4 Domain Alignment

Let $\mathbf{F}^D = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_L]$ denote the sequence of contextual outputs from the final transformer layer in domain $D \in \{S, SS\}$. To minimize the representation gap between long and short sequences for similar users, we align $\mathbf{F}^S$ and $\mathbf{F}^{SS}$ using idea of the contrastive learning.

We compute a user-level representation vector by either:

- Simple average:

$$\bar{\mathbf{F}}^D = \mathbf{Avg}(\mathbf{F}^D) \tag{5}$$

- Popularity-adjusted average:

$$\bar{\mathbf{F}}_S^D = 1/L \sum_i \boldsymbol{f}_i / P_i \tag{6}$$

  where P_i is the popularity of item i, computed as the count of each item across all users and normalized.

The Popularity-adjusted average reduces bias from frequent items in the learned user representations. These one-dimensional representations are used in the contrastive learning to align the emulated and original contexts, facilitating robust knowledge transfer from source to target.

Two types of contrastive losses are employed:

- Traditional In-batch CL: positive pairs are defined as long-short sequences from the same user, and negative pairs are sequences from different users within a batch. The NT-Xent loss Chen et al. (2020) encourages alignment of positive pairs while separating negatives, capturing divergence between source and emulated target .
- Similarity-weighted CL: All user pairs are treated as positives but weighted by precomputed similarity scores adapted from He et al. (2022), with closer users pulled together more strongly for finer-grained alignment.

The two CL losses can be formed accordingly:

- *Traditional In-batch Contrastive Loss:*

$$L_\delta^{in} = -\sum_i \sum_j \log \frac{\exp(sim(\bar{\mathbf{F}}_i, \bar{\mathbf{F}}_j)/\tau)}{\sum_{k=1}^{2N} 1_{k \neq i} \exp(sim(\bar{\mathbf{F}}_i, \bar{\mathbf{F}}_k)/\tau)}, \tag{7}$$

  where $\bar{\mathbf{F}} = [\bar{\mathbf{F}}^S, \bar{\mathbf{F}}^{SS}]$ in a training batch with sizes $2N \times d$, where N is the number of users and $i \in \{1, \ldots, 2N\}$, (i, j) are the indices of positive pairs, sim denotes the cosine similarity with temperature parameter τ, and $1_{k \neq i}$ is an indicator function which is 1 when $k \neq i$.
- *Similarity-weighted Contrastive Loss:*

$$L_\delta^s = -\sum_i \sum_{j, j \neq i} S_{ij} \log \frac{\exp\big(sim(\bar{\mathbf{F}}_i^S, \bar{\mathbf{F}}_j^{SS})/\tau\big)}{\sum_{k=1}^{N} 1_{k \neq i} \exp\big(sim(\bar{\mathbf{F}}_i^S, \bar{\mathbf{F}}_k^{SS})/\tau\big)}, \tag{8}$$

  where S_{ij} is the similarity score between long sequences:

$$S_{ij} = \exp\big(sim(\bar{\mathbf{F}}_i^S, \bar{\mathbf{F}}_j^S)/\tau\big) / \sum_{k=1}^{N} \exp\big(sim(\bar{\mathbf{F}}_i^S, \bar{\mathbf{F}}_k^S)/\tau\big).$$

When using the similarity-weighted CL, two key considerations must be addressed. First, the similarity scores between users are computed using full (long) sequences in the source domain, as short sampled sequences tend to introduce high variance and yield unstable estimates. Second, since these similarity scores are learned by the model, they are unreliable during the early stages of training. To mitigate early-stage noise in the learned similarity scores, we employ an annealed interpolation between the standard in-batch contrastive loss L_δ^{in} and our similarity-weighted loss L_δ^{s} until the similarity estimates become sufficiently stable. Detailed training strategy can be found in the B.

3.5 Final Loss Function

We train the model with the set of users' interaction sequences $x_i = (v_1, v_2, \ldots, v_L)$. An input sequence from a user is $(v_1, v_2, \ldots, v_{L-1})$ without the last interaction, and the target sequence for prediction, $y_i = (v_2, v_3, \ldots, v_L)$, is the right shifted version of the input sequence. To assess the predictions of each domain, we employ the cross-entropy (CE) loss, which is defined as:

$$L_{pred,D} = \frac{-1}{N(L-1))} \sum_{i=1}^{N} \sum_{l=1}^{L-1} \sum_{v \in V} y_{i,l,v} \log \frac{\exp(r_{i,l,v})}{\sum_{j=1}^{|V|} \exp(r_{i,l,j})}, \tag{9}$$

where $D \in \{S, SS, T\}$, N is the batch size with $i \in \{1, \ldots, N\}$, L is the sequence length with $l \in \{1, \ldots, L-1\}$, V is the set of items from interactions, $y_{i,l,v}$ is the indicator and $r_{i,l,v}$ is the relevance score of item v for user i at time l given in Eq. (4).

The overall loss function to be minimized is then

$$L = L_{pred,S} + \beta_{SS} L_{pred,SS} + \beta_T L_{pred,T} + \lambda L_\delta, \tag{10}$$

where β_{SS}, β_T , λ are hyperparameters, $L_{pred,S}$, $L_{pred,SS}$ and $L_{pred,T}$ are the CE loss of source, emulated target and target, respectively, and L_δ is the contrastive loss in either Eq.(7) or Eq.(8). Discussions on the hyperparameters and their impacts are given in A.

3.6 Model Variants

Depending on whether popularity bias reduction is applied (via Eq. (5) vs. Eq. (6)) and whether user similarity is incorporated in the contrastive loss (Eq. (7) vs. Eq. (8)), we implement three main model variants:

- DACSR: Uses averaged user representation in Eq. (5) and traditional in-batch contrastive loss in Eq. (7).
- DACSR+: Applies popularity-weighted user representation in Eq. (6) to reduce item bias, while retaining the traditional contrastive loss in Eq. (7).
- DACSR++: Combines popularity-reduced user representation in Eq. (6) with similarity-weighted contrastive loss in Eq. (8) for more fine-grained and personalized knowledge transfer.

These variants progressively improve the model robustness to different knowledge shifts by incorporating domain-specific corrections and adaptive contrastive alignment. DACSR++ achieves the best performance and is used as the main model for evaluation and ablation comparisons.

4 Experimental Results

4.1 Datasets and Splits for Cold-start

We evaluate our models on five public Amazon product review datasets McAuley et al. (2015): Toys, Electronics, Beauty, Sports, and Kindle, treating reviews or ratings as implicit feedback and using timestamps to determine interaction sequences. These datasets were selected for their public availability and their widespread use as benchmark collections in recommender-system research, ensuring the reproducibility of our results and facilitating direct comparison with prior work such as SASRec Kang & McAuley (2018) and

Split	Setting	Electr	Kindle	Beauty	Toy	Sports
Random	-	0.6949	0.3986	0.5083	0.3986	0.6260
Time-based	Filtering	0.3423	0.2195	0.1905	0.1696	0.3513
	No filtering	–	0.1976	0.1299	0.1299	0.3245

Table 1: Jaccard indexes of the processed datasets. Lower values indicate greater item distribution shifts.

MetaTL Wang et al. (2021). Following Wang et al. (2021), we preprocess each dataset to ensure sufficient sequence lengths and data quality. Refer to C for detailed dataset statistics.

To simulate different domain shift scenarios, we apply two splitting strategies to divide users into cold-start (target) and regular (source) domains:

- Random Split (Length Shift): Users with fewer than 10 interactions are filtered out. We randomly select 20% of users as the target set (cold-start) and the rest as source (regular). Items not shared between domains are removed to minimize item shift. The maximum target sequence length is set by a parameter max_len, controlling the degree of length shift.
- Time-Based Split (Complex Shift): Users who interacted before 1/1/2014 are source users; others are targets. Items with fewer than 10 reviews are filtered, and user sequences shorter than 10 (source) or 4 (target) are discarded. This approach reflects real-world cold-start settings, where newer users face both shorter sequences and changing item distributions.

We define two item-distribution variants under the time-based split: (1) Filtering, where items in the source that are not present in target are filtered (considering shared item set), and (2) No filtering, where all target items are retained, including unseen items, introducing a larger item shift. Experiments for “No filtering" are conducted on all datasets except Electronics, which, per MetaTL’s public split, corresponds only to “Filtering".

To quantify item shift, we use the Jaccard index over the top-100 popular items from each domain:

$$Jaccard_{100} = \frac{|Set_S \cap Set_T|}{|Set_S \cup Set_T|}, \tag{11}$$

where Set_S and Set_S are the top-100 popular items in the source and target domains, respectively. Lower values indicate greater shifts in item popularity. We report Jaccard statistics in Table 1. All datasets use $max_len = 4$ for target sequences to ensure consistent length shift. As expected, time-based splits, especially "No filtering", exhibit significantly lower Jaccard values, reflecting increased item shift. Note that the Jaccard index captures only the popularity-based aspect of item shift, not the full distributional change.

We follow Wang et al. (2021) to randomly divide the cold-start users in the target dataset into validation and test sets, with 30% of the cold-start users being allocated to the validation set. The last interaction of the sequence is used for performance evaluation and the remaining for training.

Following Wang et al. (2021), we randomly sample 100 negative items per ground truth item for evaluation. The model ranks the ground truth item among these negatives, and we assess performance using standard ranking metrics: Normalized Discounted Cumulative Gain (NDCG) and Hit Ratio (HR) at $k = 1, 5, 10$, and Mean Reciprocal Rank (MRR). We use NDCG@10 for model selection.

4.2 Baseline Models

To evaluate DACSR++, we compare it with several baselines including Matrix Factorization (MF), the transformer-based SASRec Kang & McAuley (2018), and Meta-TL Wang et al. (2021). MF and SASRec are trained on combined source and target data to implicitly leverage source knowledge, while SASRec_T, trained only on target data, serves as a lower bound using SASRec model. We also include standard

Dataset	Metric	SASRec_T	MF	SASRec	TL	MetaTL	SIGMA	LinRec	DACSR++	Impr.
Beauty	Hit@1	0.1290	0.2107	0.2236	0.2649	0.202	0.2126	0.2210	**0.2675**	+0.98%
	Hit@5	0.2965	0.3778	0.4023	0.4506	0.380	0.4299	0.4154	**0.4958**	+10.0%
	NDCG@5	0.2163	0.2991	0.3168	0.3624	0.296	0.3257	0.3223	**0.3914**	+8.00%
	Hit@10	0.3960	0.4405	0.4897	0.5322	0.464	0.5415	0.5210	**0.6329**	+16.9%
	NDCG@10	0.2487	0.3192	0.3450	0.3885	0.323	0.3616	0.3564	**0.4357**	+12.1%
	MRR	0.2189	0.2970	0.3179	0.3604	0.297	0.3224	0.3225	**0.3930**	+9.05%
Toys	Hit@1	0.1015	0.1917	0.2135	0.2120	0.201	0.1334	0.2035	**0.2663**	+24.7%
	Hit@5	0.2908	0.3543	0.3746	0.4122	0.396	0.3617	0.3817	**0.5302**	+28.6%
	NDCG@5	0.1976	0.2758	0.2967	0.3172	0.300	0.2488	0.2973	**0.4027**	+27.0%
	Hit@10	0.4225	0.4202	0.4762	0.5162	0.503	0.4892	0.4660	**0.6798**	+31.7%
	NDCG@10	0.2398	0.2970	0.3293	0.3509	0.335	0.2901	0.3246	**0.4510**	+28.5%
	MRR	0.2033	0.2716	0.3016	0.3155	0.301	0.2472	0.2989	**0.3952**	+25.3%
Sports	Hit@1	0.1276	0.1657	0.2308	0.2309	0.194	0.1862	0.2183	**0.2808**	+21.6%
	Hit@5	0.2964	0.3519	0.4683	0.4933	0.415	0.4408	0.4571	**0.5770**	+31.2%
	NDCG@5	0.2124	0.2617	0.3547	0.3690	0.310	0.3193	0.3452	**0.4364**	+18.3%
	Hit@10	0.3947	0.4383	0.5812	0.6134	0.516	0.5796	0.5701	**0.7100**	+15.7%
	NDCG@10	0.2440	0.2898	0.3912	0.4080	0.342	0.3642	0.3816	**0.4795**	+17.5%
	MRR	0.2141	0.2585	0.3478	0.3583	0.305	0.3131	0.3385	**0.4216**	+17.7%
Kindle	Hit@1	0.0510	0.3054	0.3757	0.3731	0.357	0.2476	0.3802	**0.4093**	+7.65%
	Hit@5	0.1414	0.5220	0.6259	0.6501	0.632	0.4217	0.6123	**0.6581**	+1.23%
	NDCG@5	0.0950	0.4194	0.5094	0.5217	0.504	0.3393	0.4949	**0.5413**	+3.76%
	Hit@10	0.2203	0.5918	0.7300	0.7383	0.742	0.5273	0.7070	**0.7631**	+2.84%
	NDCG@10	0.1201	0.4424	0.5430	0.5503	0.540	0.3730	0.5253	**0.5758**	+4.64%
	MRR	0.1122	0.4045	0.4951	0.5032	0.487	0.3418	0.4802	**0.5267**	+4.67%
Electronic	Hit@1	0.1458	0.1363	0.2163	0.2193	0.193	0.1913	0.2231	**0.2448**	+9.73%
	Hit@5	0.3326	0.3212	0.4627	0.4871	0.420	0.4240	0.4827	**0.5268**	+8.17%
	NDCG@5	0.2413	0.2337	0.3462	0.3622	0.310	0.3113	0.3586	**0.3931**	+8.53%
	Hit@10	0.4616	0.4040	0.5878	0.6233	0.553	0.5666	0.6009	**0.6818**	+9.39%
	NDCG@10	0.2828	0.2603	0.3871	0.4064	0.353	0.3574	0.3965	**0.4433**	+9.08%
	MRR	0.2435	0.2304	0.3406	0.3542	0.307	0.3109	0.3493	**0.3846**	+8.58%

Table 2: Performance comparison under *Random split for length shift* with $max_len = 4$. Impr. denotes the percentage improvement of DACSR++ (the best method) over the next best method.

transfer learning (pretrained on source, fine-tuned on target), LinRec Liu et al. (2023), a linear-complexity transformer variant, and SIGMA Liu et al. (2024), a recent Mamba-based model designed for capturing both long- and short-term preferences. For direct comparison, the chosen architecture of the backbone model remains the same as SASRec throughout the experiments: the latent dimension of the embedding is d = 64, the transformer model is single-head and b = 2 blocks are stacked. Detailed implementation and hyperperameter tuning can be found in D.

4.3 Overall Performance Comparison

Random Split (length shift): Table 2 shows model performance under the random split ($max_len = 4$). Comparing SASRec_T (trained only on target) with MF (trained on all data) reveals that even a transformer struggles without knowledge transfer, highlighting the value of leveraging source domain knowledge. While SASRec outperforms MF by modeling sequential patterns, MetaTL achieves similar results to SASRec in 3 out of 5 datasets but lags behind TL and our DACSR framework, likely due to its simple MLP backbone. The TL approach, which pre-trains on source and fine-tunes on target, consistently improves over SASRec, showing its effectiveness in addressing length shift. Our DACSR++ achieves the best performance, particularly on Electronics, Kindle, and Toys, benefiting from its fine-grained control over knowledge transfer via λ, β_T, and p.

Time-Based Split (compound shift): Table 3 shows results under the time-based split for the case of "filtering". Unlike random splits, TL offers only marginal or even negative gains over SASRec in some datasets due to significant item distribution shifts over time (see Jaccard index in Table 1), limiting the generalizability of pre-trained item embeddings. In contrast, DACSR++ consistently outperforms both SASRec and TL by better managing knowledge transfer and distribution mismatch, demonstrating its robustness under complex shifts.

Dataset	Metric	SASRec	MetaTL	TL	SIGMA	LinRec	DACSR++	Impr.
Beauty	Hit@1	0.4190	0.347	0.3851	0.3882	0.3915	**0.4229**	+0.93%
	Hit@5	0.6444	0.534	0.6478	0.5766	0.6159	**0.6783**	+4.71%
	Hit@10	0.7342	0.638	0.7568	0.6722	0.7086	**0.7981**	+5.46%
	NDCG@5	0.5394	0.443	0.5228	0.4873	0.5114	**0.5581**	+3.47%
	NDCG@10	0.5687	0.477	0.5582	0.5182	0.5416	**0.5967**	+4.92%
	MRR	0.5276	0.441	0.5068	0.4828	0.5009	**0.5446**	+3.22%
Toy	Hit@1	0.3345	0.268	0.3245	0.2848	0.3486	**0.3807**	+9.21%
	Hit@5	0.5731	0.483	0.5863	0.4840	0.5700	**0.6696**	+14.2%
	Hit@10	0.6947	0.601	0.7013	0.6047	0.6710	**0.8073**	+15.1%
	NDCG@5	0.4575	0.379	0.4609	0.3871	0.4632	**0.5312**	+14.7%
	NDCG@10	0.4967	0.417	0.4981	0.4261	0.4955	**0.5763**	+15.7%
	MRR	0.4484	0.375	0.4478	0.3860	0.4542	**0.5144**	+13.3%
Sports	Hit@1	0.3026	0.257	0.3114	0.2573	0.3101	**0.3572**	+14.7%
	Hit@5	0.5753	0.517	0.6118	0.5139	0.5683	**0.6864**	+12.2%
	Hit@10	0.6916	0.634	0.7276	0.6328	0.6683	**0.8106**	+11.4%
	NDCG@5	0.4483	0.393	0.4701	0.3904	0.4445	**0.5307**	+12.9%
	NDCG@10	0.4859	0.431	0.5075	0.4287	0.4768	**0.5710**	+12.5%
	MRR	0.4346	0.382	0.4497	0.3800	0.4302	**0.5053**	+12.4%
Kindle	Hit@1	0.7223	0.678	0.8171	0.6795	0.7232	**0.8489**	+3.89%
	Hit@5	0.9719	0.932	0.9687	0.9286	0.9565	**0.9857**	+1.42%
	Hit@10	0.9891	0.969	0.9875	0.9730	0.9875	**0.9974**	+0.84%
	NDCG@5	0.8606	0.817	0.9064	0.8179	0.8477	**0.9281**	+2.39%
	NDCG@10	0.8660	0.829	0.9127	0.8322	0.8580	**0.9320**	+2.11%
	MRR	0.8256	0.785	0.8883	0.7879	0.8163	**0.9101**	+2.45%
Electr	Hit@1	0.2727	0.225	0.2697	0.2048	0.2558	**0.2966**	+8.76%
	Hit@5	0.5505	0.482	0.5509	0.4475	0.5154	**0.5837**	+5.95%
	Hit@10	0.6679	0.602	0.6680	0.5846	0.6402	**0.6991**	+4.66%
	NDCG@5	0.4186	0.359	0.4179	0.3297	0.3925	**0.4479**	+7.00%
	NDCG@10	0.4568	0.398	0.4557	0.3740	0.4331	**0.4854**	+6.26%
	MRR	0.4050	0.351	0.4041	0.3260	0.3838	**0.4323**	+6.74%

Table 3: Model performances under *Time-based split* with complex knowledge shift. Impr. denotes the percentage improvement of DACSR++ over the next best method.

Metrics	Model	Beauty	Impr.	Toy	Impr.	Sports	Impr.	Kindle	Impr.
Hit@1	SASRec	0.3443	+17.3%	0.3172	+11.2%	0.2784	+12.0%	0.6549	+10.2%
	DACSR++	**0.4039**		**0.3526**		**0.3118**		**0.7218**	
Hit@5	SASRec	0.5409	+11.0%	0.5172	+11.3%	0.5124	+9.1%	0.8927	+6.5%
	DACSR++	**0.6005**		**0.5754**		**0.5590**		**0.9508**	
Hit@10	SASRec	0.6200	+10.7%	0.6222	+9.0%	0.6120	+8.9%	0.9290	+5.4%
	DACSR++	**0.6863**		**0.6783**		**0.6663**		**0.9793**	
NDCG@5	SASRec	0.4485	+13.0%	0.4216	+11.1%	0.3987	+10.9%	0.7855	+8.4%
	DACSR++	**0.5067**		**0.4685**		**0.4424**		**0.8518**	
NDCG@10	SASRec	0.4742	+12.7%	0.4555	+10.1%	0.4312	+10.6%	0.7973	+8.0%
	DACSR++	**0.5343**		**0.5016**		**0.4769**		**0.8610**	
MRR	SASRec	0.4410	+13.0%	0.4163	+10.1%	0.3874	+10.8%	0.7570	+8.7%
	DACSR++	**0.4986**		**0.4583**		**0.4294**		**0.8228**	

Table 4: Performances of models under *Time-based split for both length and item shifts* when $max_len = 4$ for case of "No Filtering". Electronic data is from Wang et al. (2021) and all items in the target are observed in the source. Therefore, no result of Electronic are reported in this case.

Under the more challenging time-based split for case of "No Filtering", where entirely new items may appear in the target domain, DACSR++ yields pronounced relative gains over SASRec across every metric and dataset as shown in Table 4. These consistent uplifts show that augmenting popularity weighting with similarity-aware contrastive learning makes the model far more robust to simultaneous length and item-distribution shifts.

Dataset	Metric	Random split			Time split		
		DACSR	DACSR+	DACSR++	DACSR	DACSR+	DACSR++
Beauty	Hit@1	0.2548	0.2563	**0.2675**	**0.4355**	0.4329	0.4229
	Hit@5	0.4342	0.4374	**0.4958**	0.6605	0.6552	**0.6783**
	NDCG@5	0.3486	0.3507	**0.3914**	0.5560	0.5530	**0.5581**
	Hit@10	0.5317	0.5375	**0.6329**	0.7586	0.7599	**0.7981**
	NDCG@10	0.3802	0.3830	**0.4357**	0.5878	0.5866	**0.5967**
	MRR	0.3500	0.3529	**0.3930**	0.5444	0.5434	**0.5446**
Toys	Hit@1	0.2518	0.2611	**0.2663**	0.3748	0.3758	**0.3807**
	Hit@5	0.4270	0.4454	**0.5302**	0.6046	0.6261	**0.6696**
	NDCG@5	0.3440	0.3586	**0.4027**	0.4949	0.5100	**0.5312**
	Hit@10	0.5167	0.5354	**0.6798**	0.7133	0.7343	**0.8073**
	NDCG@10	0.3727	0.3877	**0.4510**	0.5298	0.5448	**0.5763**
	MRR	0.3449	0.3582	**0.3952**	0.4853	0.4977	**0.5144**
Sports	Hit@1	0.2516	0.2477	**0.2808**	0.3396	0.3512	**0.3572**
	Hit@5	0.4873	0.4727	**0.5770**	0.6100	0.6115	**0.6864**
	NDCG@5	0.3750	0.3666	**0.4364**	0.4815	0.4875	**0.5307**
	Hit@10	0.5890	0.5899	**0.7100**	0.7145	0.7289	**0.8106**
	NDCG@10	0.4082	0.4046	**0.4795**	0.5153	0.5258	**0.5710**
	MRR	0.3668	0.3626	**0.4216**	0.4646	0.4742	**0.5053**
Kindle	Hit@1	0.3981	**0.4093**	0.3647	0.8203	**0.8489**	0.8281
	Hit@5	0.6535	0.6581	**0.6621**	0.9797	**0.9857**	0.9828
	NDCG@5	0.5302	**0.5413**	0.5250	0.9114	**0.9281**	0.9160
	Hit@10	0.7535	**0.7631**	0.7519	0.9938	**0.9974**	0.9922
	NDCG@10	0.5627	**0.5758**	0.5540	0.9160	**0.9320**	0.9191
	MRR	0.5130	**0.5267**	0.5016	0.8905	**0.9101**	0.8951
Electr	Hit@1	0.2433	0.2439	**0.2448**	0.2900	**0.2983**	0.2966
	Hit@5	0.4969	0.5055	**0.5268**	0.5695	0.5716	**0.5837**
	NDCG@5	0.3755	0.3794	**0.3931**	0.4375	0.4421	**0.4479**
	Hit@10	0.6175	0.6183	**0.6818**	0.6850	0.6903	**0.6991**
	NDCG@10	0.4144	0.4159	**0.4433**	0.4748	0.4803	**0.4854**
	MRR	0.3654	0.3672	**0.3846**	0.4228	0.4284	**0.4323**

Table 5: Performance of DACSR variants under two splits, with $max_len = 4$.

max_len	Model	Electr	Impr.	Kindle	Impr.	Beauty	Impr.	Toy	Impr.	Sports	Impr.
20	DACSR++	0.4824	+0.15%	0.6272	-0.06%	**0.4368**	+16.08%	**0.4214**	+11.54%	**0.4373**	+22.12%
	SASRec	**0.4817**		**0.6276**		0.3763		0.3778		0.3581	
10	DACSR++	**0.4811**	+6.67%	**0.6360**	+1.35%	**0.4566**	+22.74%	**0.4434**	+27.85%	**0.4717**	+25.99%
	SASRec	0.4510		0.6275		0.3720		0.3468		0.3744	
6	DACSR++	**0.4879**	+15.04%	**0.6173**	+6.28%	**0.4517**	+21.62%	**0.4314**	+33.31%	**0.4847**	+26.45%
	SASRec	0.4241		0.5808		0.3714		0.3236		0.3833	
4	DACSR++	**0.4433**	+14.52%	**0.5540**	+2.03%	**0.4357**	+26.29%	**0.4510**	+36.92%	**0.4795**	+25.10%
	SASRec	0.3871		0.5430		0.3450		0.3294		0.3911	

Table 6: Performances of models for *Random split* against different *max_len*. NDCG@10 values are reported.

4.4 Ablation Study

Comparisons for Model Variants: Table 5 shows the performance by DACSR, DACSR+ and DACSR++. All three variants consistently outperform the SASRec baseline. By weighting the loss by item popularity in DACSR+, this yields steady, if modest, uplifts over DACSR in most ranking metrics. Building on that, DACSR++ adds a similarity-weighted contrastive component that aligns full-sequence and partial-sequence embeddings more tightly; this additional signal produces the largest gains of all, especially under the random split which presents a smaller item shift relative to time split.

Performance against different *max_len*: The parameter *max_len* controls the maximum sequence length in the target data to simulate cold-start scenarios. To evaluate the impact of length shift, we vary $max_len \in \{4, 6, 10, 20\}$ under the *Random split* and compare DACSR++ with SASRec in Table 6, reporting DACSR++'s relative improvement. Results show that both models benefit from longer target sequences, confirming that longer interaction improves prediction. DACSR++ consistently outperforms SASRec, especially when *max_len* is small, highlighting DACSR++'s advantage under severe length shift. As *max_len* increases and the domain gap narrows, performance difference shrinks, with negligible gaps at $max_len = 20$

for some datasets, where knowledge shift is minimal.

Impact of the contrastive losses The user representations using T-SNE have been plotted in Figure 2,

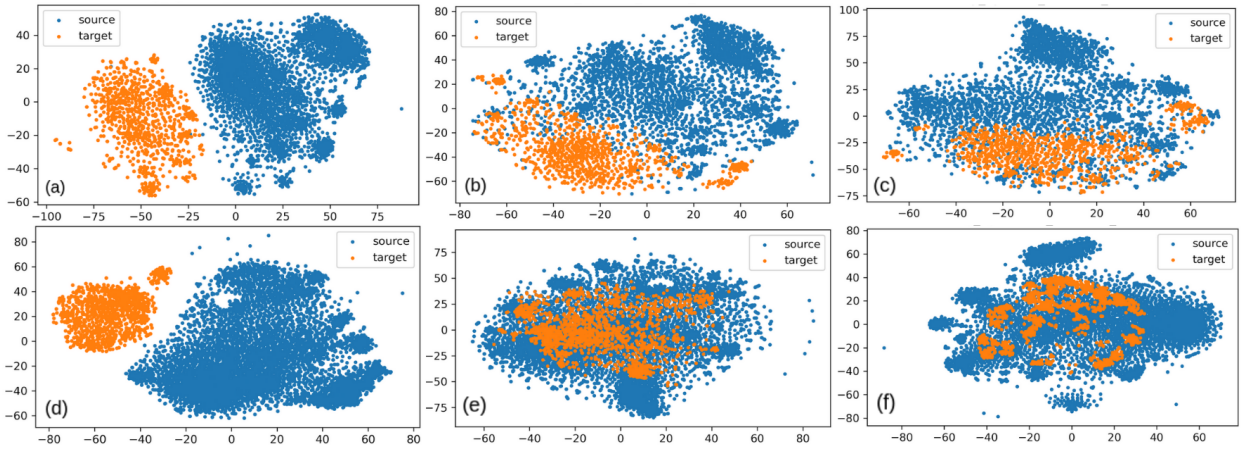

Figure 2: TSNE visualization of *Toys*'s *source* (blue) and *target* (orange) user representations under two splitting strategies: *time-split* (a, b, c) and *random-split* (d, e, f) for three models: *SASRec* (a, d), *DACSR+* (b, e), and *DACSR++* (c, f). Improved mixing of target and source clusters indicates more effective domain alignment.

where SASRec yields two nearly disjoint clusters of source and target embeddings under both time-based and random splits, reflecting a pronounced domain gap and limited representation alignment. In contrast, DACSR+ achieves greater overlap between source and target, and DACSR++ further enhances this alignment, producing the most coherent intermixing of domains. DACSR++ exhibits the tightest intra-target clustering, which can be attributed to its similarity-weighted contrastive loss: by pulling together semantically related user representations rather than treating all non-paired examples as strict negatives, it fosters more meaningful grouping of target users than the standard DACSR+ formulation. The TSNE plots for other datasets are provided in E, showing similar results.

Splits	Length shift	Item shift	Model	Electr	Kindle	Beauty	Toy	Sports
Time-based	✓	✓	DACSR w/o L_δ	0.4469	0.9104	0.5721	0.4966	0.5006
			DACSR	0.4748	0.9160	0.5878	0.5298	0.5153
			DACSR+	0.4803	**0.9320**	0.5866	0.5448	0.5258
			DACSR++	**0.4854**	0.9191	**0.5967**	**0.5763**	**0.5710**
Random	✓	✗	DACSR w/o L_δ	0.3522	0.4469	0.3359	0.2629	0.3675
			DACSR	0.4144	0.5627	0.3804	0.3727	0.4082
			DACSR+	0.4159	**0.5758**	0.3830	0.3877	0.4046
			DACSR++	**0.4433**	0.5540	**0.4357**	**0.4510**	**0.4795**

Table 7: Performances with or without contrastive loss when $max_len = 4$. NDCG@10 values are reported.

Ablation studies comparing DACSR variants with and without CL are provided, where we evaluate the impact of contrastive loss under both split settings with $max_len = 4$. Setting $\lambda = 0$ removes the contrastive loss and the emulated target domain, reducing DACSR to two independently trained transformers with shared item embeddings and balanced by β_T. As shown in Table 7, the contrastive loss significantly improves performance across both split methods. The gains are more pronounced under the *Random split*, where length shift dominates, allowing more effective knowledge transfer via the emulated domain. Under the *Time-based split*, the added complexity from item distribution shift limits the extent of information flow, but improvements are still evident.

5 Conclusion

We tackle cold-start sequential recommendation via domain adaptation by addressing two key shifts: length shift and combined length-item distribution shift. Our main contribution is a dual-transformer framework with separate models for long (source) and short (target) sequences, using balanced joint training to mitigate bias and enable adaptive knowledge transfer. We emulate a cold-start target domain by sampling short sequences from source data, enriching the target data diversity under length shift. For the compound shift, we reduce item bias with popularity-weighted user representations and enhance transfer via contrastive learning based on user similarity. Extensive experiments demonstrate that our approach consistently outperforms state-of-the-art methods across both shift scenarios.

References

Oren Barkan, Noam Koenigstein, Eylon Yogev, and Ori Katz. Cb2cf: A neural multiview content-to-collaborative filtering model for completely cold item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 228–236, 2019.

Lei Chen, Fajie Yuan, Jiaxi Yang, Xiangnan He, Chengming Li, and Min Yang. User-specific adaptive fine-tuning for cross-domain recommendations. *CoRR*, abs/2106.07864, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1597–1607, 2020.

Xu Chen, Ya Zhang, Ivor W. Tsang, Yuangang Pan, and Jingchao Su. Toward equivalent transformation of user preferences in cross domain recommendation. *ACM Transactions on Information Systems*, 41(1): 1–31, 2023.

Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pp. 2172–2182, 2022.

Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.

Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys)*, pp. 152–160, 2017.

Xiaoyu Du, Xiang Wang, Xiangnan He, Zechao Li, Jinhui Tang, and Tat-Seng Chua. How to learn item representation for cold-start multimedia recommendation? In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 3469–3477, 2020.

Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems*, 39(1):1–42, 2020.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1126–1135, 2017.

Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *2010 IEEE International Conference on Data Mining*, pp. 176–185, 2010.

Ruining He and Julian McAuley. Fusing similarity models with markov chains for sparse sequential recommendation. In *Proceedings of the 2016 IEEE 16th International Conference on Data Mining*, pp. 191–200, 2016.

Wanwei He, Yinpei Dai, Binyuan Hui, Min Yang, Zheng Cao, Jianbo Dong, Fei Huang, Luo Si, and Yongbin Li. SPACE-2: Tree-structured semi-supervised contrastive pre-training for task-oriented dialog understanding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 553–569, 2022.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pp. 173–182, 2017.

Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 843–852, 2018.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016.

T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2022.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pp. 263–272, 2008.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, 2018.

Muhammad Murad Khan, Roliana Ibrahim, and Imran Ghani. Cross domain recommender systems: A systematic literature review. *ACM Computing Surveys*, 50(3):1–34, 2017.

Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. Melu: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 1073–1082, 2019.

Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, pp. 322–330, 2020.

Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. From zero-shot learning to cold-start recommendation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 4189–4196, 2019.

Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, pp. 689–698, 2018.

Lingxi Liu, Lin Cai, Chenyang Zhang, Xiangyu Zhao, Jing Gao, Weinan Wang, Yanchi Lv, Wen Fan, Yujing Wang, and Meihong He. Linrec: Linear attention mechanism for long-term sequential recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 289–299, 2023.

Ziwei Liu, Qidong Liu, Yejing Wang, Wanyu Wang, Pengyue Jia, Maolin Wang, Zitao Liu, Yi Chang, and Xiangyu Zhao. Sigma: Selective gated mamba for sequential recommendation. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence*, 2024.

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

Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, pp. 497–506, 2011.

Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*, pp. 813–823, 2022.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pp. 452–461, 2009.

Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pp. 811–820, 2010.

Guy Shani, Ronen I. Brafman, and David Heckerman. An mdp-based recommender system. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pp. 453–460, 2002.

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)*, pp. 1441–1450, 2019.

Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 565–573, 2018.

Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A meta-learning perspective on cold-start recommendations for items. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6907–6917, 2017.

Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. Dropoutnet: Addressing cold start in recommender systems. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4964–4973, 2017.

Jianling Wang, Kaize Ding, and James Caverlee. Sequential recommendation for cold-start users with meta transitional learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1783–1787, 2021.

Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 501–508, 2006.

Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet Orgun. Sequential recommender systems: Challenges, progress and prospects. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 6332–6338, 2019.

Tianxin Wei, Ziwei Wu, Ruirui Li, Ziniu Hu, Fuli Feng, Xiangnan He, Yizhou Sun, and Wei Wang. Fast adaptation for cold-start collaborative filtering with meta-learning. In *2020 IEEE International Conference on Data Mining*, pp. 661–670, 2020.

Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 5382–5390, 2021.

Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. Contrastive learning for sequential recommendation. *arXiv preprint arXiv:2010.14395*, 2021.

Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. A simple convolutional generative network for next item recommendation. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, pp. 582–590, 2019a.

Feng Yuan, Lina Yao, and Boualem Benatallah. Darec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 4227–4233, 2019b.

Shuai Zhang, Yi Tay, Lina Yao, Aixin Sun, and Jake An. Next item recommendation with self-attentive metric learning. In *Proceedings of the AAAI Workshop on Recommender Systems and Natural Language Processing (RecNLP)*, 2019.

Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. Catn: Cross-domain recommendation for cold-start users via aspect transfer network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 229–238, 2020.

Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. Cold-start sequential recommendation via meta learner. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4706–4713, 2021.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1059–1068, 2018.

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 and Knowledge Management*, pp. 1893–1902, 2020.

Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. Cross-domain recommendation: Challenges, progress, and prospects. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pp. 4721–4728, 2021.

Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. Recommendation for new users and new items via randomized training and mixture-of-experts transformation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1121–1130, 2020.

A Hyperparameters and Their Impacts

The overall loss function to be minimized is then

$$L = L_{pred,S} + \beta_{SS} L_{pred,SS} + \beta_T L_{pred,T} + \lambda L_\delta, \tag{12}$$

where β_{SS}, β_T , λ are hyperparameters, $L_{pred,S}$, $L_{pred,SS}$ and $L_{pred,T}$ are the CE loss of source, emulated target and target, respectively, and L_δ is the contrastive loss.

The elements outlined above, together with the sampling probability p for the augmentation operator, collectively constitute the hyperparameters of our framework. These hyperparameters affect the characteristics of the emulated target domain, as well as balance the learning focus placed on each domain. Depending on the specific configurations, the framework is tailored to address particular aspects of the knowledge transfer process.

For example, if p is close to 1, the target network will take short and long sequences, and we therefore try to align the latter, at context level, with similar sequences from the source model. By contrast, if p is relatively low, the lengths of the emulated target and target sequences may be similar. The probability p assumes a role in regulating the extent of information needed to be aligned between the emulated target context and the source model's context.

In our approach, we establish a particular setup where β_{SS} is set to 0, which enables the target model to prioritize on minimizing the prediction error for the target data instead of the sampled source. This helps the target model to focus specifically on accuracy of cold-start data. The alignment of contexts between the source and emulated target (and target) domains is obtained by the contrastive loss of L_δ. Importantly, the hyperparameter λ is utilized to regulate the extent to which the alignment between the two domains should

be. Hyperparameter λ together with sampling probability p control the alignment process. Moderating p and λ allows us to adjust the extent of knowledge shared and transferred through source to target.

In our experiments, we tune the other hyperparameters according to the model's performance on validation data. The hyperparameter β_T when tuned plays a key role in regulating the information balance within the framework between the source and target domain.

B Training Strategy

For DACSR++, we introduce an epoch-dependent weight $\alpha(e)$ defined as

$$\alpha(e) = \begin{cases} 1.0, & e < E_{\text{warmup}} = 15, \\ 1 - \dfrac{\min\big(e,\, E_{\text{decay}}\big)}{E_{\text{decay}}}, & e \geq E_{\text{warmup}}, \end{cases}$$

where e is the epoch index, E_{decay} is a tunable hyperparameter and E_{warmup} denotes the number of initial training epochs, during which the model uses the standard in-batch contrastive loss. This yields the overall objective:

$$L_\delta(e) \;=\; \alpha(e)\, L_\delta^{in} \;+\; \big(1 - \alpha(e)\big)\, L_\delta^{s}.$$

By adjusting E_{decay}, we control the rate at which the model transitions from the robust in-batch loss to the finer-grained similarity-weighted variant as the learned similarity estimates become reliable.

C Statistics of Datasets

The statistics of the datasets after preprocessing using the two split methods are summarized in Tables 8 and 9.

The *Random split for length shift* maintains a substantial similarity in item distribution between the target and source datasets after the preprocessing. As a result, the item shift between data in source and target in Table 8 is minimal.

	Source							Target						
Dataset	#users	#items	Avg	min	25%	50%	75%	#users	#items	Avg	min	25%	50%	75%
Kindle	6448	21220	21.5	10	12	15	23	1612	13770	20.7	6	11	15	23
Electronics	8651	41822	16.2	10	11	13	17	2153	16935	13.1	4	9	10	14
Beauty	6423	47432	17.4	10	11	13	18	1532	17607	12.9	4	7	10	14
Sports	10965	73399	16.2	10	11	13	27	2625	27109	12.2	4	8	10	13
Toys	8038	73114	18.0	10	11	13	18	1855	26470	12.6	4	7	10	14

Table 8: Statistics of different datasets after preprocessing using *Random split for length shift*.

	Source							Target						
Dataset	#users	#items	Avg	min	25%	50%	75%	#users	#items	Avg	min	25%	50%	75%
Kindle	16465	21220	25.2	10	12	16	27	983	871	10.2	4	5	7	12
Electronics	20865	20712	17.3	10	11	13	18	5963	9374	5.7	4	4	5	6
Beauty	3653	14841	17.3	10	11	13	18	2172	17607	6.1	4	4	5	6
Sports	6493	22995	15.6	10	11	13	17	2010	6716	5.1	4	4	4	6
Toys	3984	18052	17.2	10	11	13	18	1829	4665	4.8	4	4	4	5

Table 9: Statistics of different datasets after pre-processing using *Time-based split for both length and item shifts*.

D Implementation Details

The chosen architecture of the backbone model remains the same throughout the experiments: the latent dimension of the embedding is $d = 64$, the transformer model is single-head and $b = 2$ blocks are stacked. For training, we set the batch size to 128, and use the Adam optimizer with a learning rate of 0.001. Network weights of all the models are initialized with a normal-distribution with a standard deviation of 0.01. We use the validation performance in the target domain for early stopping when there is no performance improvement over 20 training epochs. Based on the dataset statistics in Table 8 and Table 9, we set the sequence length L of both backbone models to be 30 for Kindle dataset and 20 for the other four datasets. The values considered for other hyperparameters of the backbone model and the DACSR++ framework are summarized in Table 10. These hyperparameters are set through the following empirical grid search approach:

- First, we train a transformer model on the combined source and target data. Note that this model also serves as a baseline in our evaluation. The optimal model and embedding dropout rates are searched over their respective ranges. The resulting model, along with the determined dropout rate, is used as the backbone for both source and target network in DACSR++.
- With the backbone models established, we set p to a fixed value of 0.5. Using this fixed value, we search for the optimal values of the domain information balance control parameter, i.e., β_T and the contrastive loss control parameter, i.e., λ within their respective ranges.
- Finally, with optimal β_T and λ values determined, we subsequently tune the sampling probability p.

hyperparameter	Searching Values
Model dropout rate	0.1, 0.2, 0.3, 0.4, 0.5
Embedding dropout rate	0.1, 0.2, 0.3, 0.4, 0.5
β_T	0, 0.01, 0.03, 0.04, 0.05, 0.07, 0.1, 0.2, 0.3
λ	0, 0.01, 0.03, 0.04, 0.05, 0.07, 0.1, 0.2, 0.3
p	0.3, 0.4, 0.5, 0.6, 0.7, 0.8

Table 10: Hyperparameters and their sets of values.

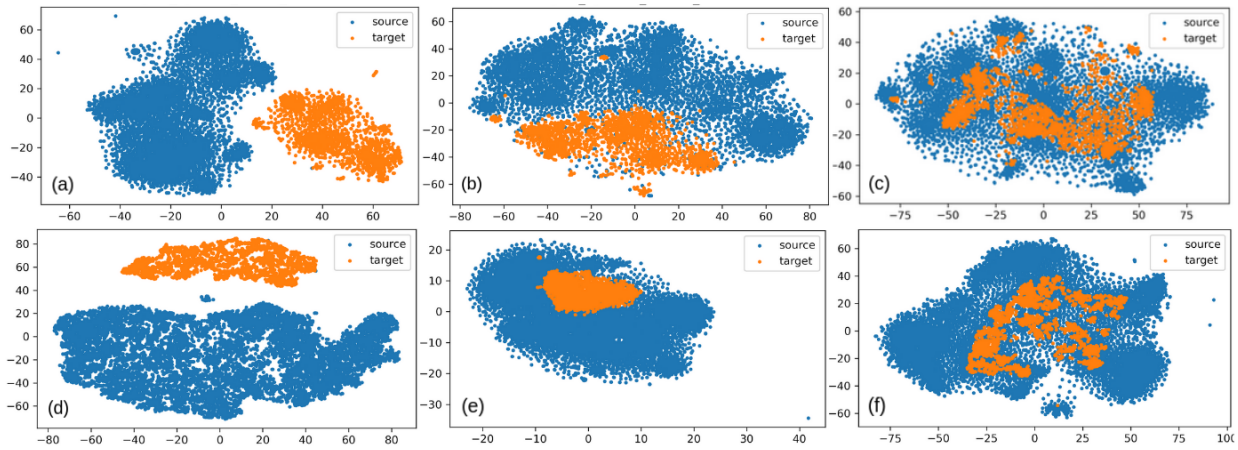

Figure 3: TSNE visualization of *Sports* dataset's *source* (blue) and *target* (orange) user embeddings under two data-splitting strategies — *by timestamp* (a, b, c) and *by random* (d, e, f) — for three models: *SASRec* (a, d), *DACSR+* (b, e), and *DACSR++* (c, f). Improved separation of target and source clusters indicates more effective domain adaptation.

E User Representations of Model Variants

The T-SNE visualization of user representations on the Sports dataset is shown in Figure 3, supporting conclusions similar to those from Figure 2. SASRec produces two nearly disjoint clusters of source and

target embeddings under both time-based and random splits, indicating a pronounced domain gap and limited representation alignment. In contrast, DACSR+ increases the overlap between source and target, while DACSR++ further improves alignment, achieving the most coherent intermixing of domains. Notably, DACSR++ also demonstrates the tightest intra-target clustering, which can be attributed to its similarity-weighted contrastive loss: by pulling together semantically related user representations instead of treating all non-paired examples as strict negatives, it fosters more meaningful groupings of target users compared to the standard DACSR+ formulation..