# INTEXPO: EXPLORING DIVERSE TARGET INTENTS FOR CROSS-DOMAIN RECOMMENDATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Cross-domain recommendation (CDR) leverages source-domain interactions to alleviate data sparsity in a target domain. Recent cross-domain sequence synthesis (CDSS) methods synthesize target-domain histories for source-only users to augment training data. However, a user's target-domain behavior depends on both that user's transferable preferences and target-specific intents. Source histories reveal transferable preferences that transfer across domains, but similar transferable preferences may lead to different target intents and behavior sequences. Existing CDSS methods often generate overly similar sequences for users with similar preferences, leaving other plausible intents underrepresented. We call this problem **Target-Intent Collapse**. To address it, we propose **IntExpo** to synthesize target-domain sequences from transferable preference profiles and diverse target-intent trajectories. Specifically, IntExpo comprises a *Transferable Preference Profiler*, an *Intent Generator*, and a *Behavior Generator*. Its Transferable Preference Profiler proposes discriminator-guided reinforcement learning (RL) to extract transferable profiles from source histories. The Intent Generator learns a conditional distribution over intent trajectories, using trajectories observed from users with similar profiles as additional supervision. Finally, the Behavior Generator produces multiple plausible target-domain sequences conditioned on a profile and diverse intent trajectories. Experiments across four cross-domain transfer settings show that IntExpo outperforms competitive baselines even under the same synthesis budget. Further analyses show that it generates diverse, plausible behaviors and alleviates the target-intent collapse problem. Our anonymous code is available at https://anonymous.4open.science/r/IntExpo-5672.

## 1 INTRODUCTION

Users on online service platforms often interact across multiple content or product domains. Their cross-domain histories provide complementary preference signals that help alleviate target-domain sparsity and cold-start problems (Liu et al., 2021; Yang et al., 2024). Conventional cross-domain recommendation methods exploit these signals by transferring shared preferences from source to target domains (Cao et al., 2022b; Liu et al., 2024b), but are constrained by the limited number of users observed in both domains. Recent cross-domain sequence synthesis (CDSS) (Xin et al., 2025; Chen et al., 2025; Lu et al., 2026) methods offer a data-centric alternative. They use LLMs or diffusion models to synthesize target-domain histories for source-only users, thereby converting them into pseudo cross-domain users and expanding the training data.

Despite their effectiveness, *existing CDSS methods still suffer from the* ***Target-Intent Collapse*** problem. In real-world scenarios, users' target-domain behaviors are jointly driven by transferable preferences and target-specific intents. Transferable preferences reflect what users generally like across domains, whereas target-specific intents represent what users seek to accomplish in the target domain and further guide which specific items they are likely to interact with. Source histories reveal users' transferable preferences but are insufficient to uniquely determine their target-domain intent; a single source history may be compatible with multiple plausible intents, leading to different target sequences. However, existing CDSS methods typically synthesize a single target sequence from each source-domain history, reducing multiple plausible intents to only one outcome. Therefore, they tend to generate overly similar target sequences for users with similar transferable preferences, leaving other plausible intent modes underrepresented and limiting recommendation performance.



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.

To further examine the target-intent collapse problem, we perform the following investigations.

- **Example Illustration**: Figure 1-(a) shows a Movie (source domain)→Book (target domain) example. Two users share similar transferable preferences for *science fiction and space exploration,* yet their real Book histories reveal distinct intents–learning about *Spaceflight Engineering* versus reading *Popular Space Fiction*.
- **Preference Visualization**: We first encode users' Movie-domain histories using a pretrained SASRec and cluster the resulting representations, retaining the three largest clusters. We then use LASSO (Chen et al., 2025) to synthesize one Book-domain sequence for each user. Finally, a pretrained Book-domain SASRec encodes both the real and synthetic target sequences. As shown in Figure 1-(b), within each cluster, users' source-domain preference representations are compact, whereas their real target-domain preference representations are dispersed. By contrast, their synthetic target-domain preference representations remain highly concentrated.

To address this issue, we propose **IntExpo**, which generates multiple plausible target-domain sequences by combining a shared transferable preference profile with diverse target-intent trajectories. IntExpo comprises three modules: (i) a Transferable Preference Profiler, (ii) an Intent Generator, and (iii) a Behavior Generator. The Transferable Preference Profiler optimizes an LLM through reinforcement learning with a domain-discriminator-based reward, enabling it to generate a transferable preference profile for user based on her source-domain behaviors. Conditioned on the resulting profile, the Intent Generator learns a conditional distribution over target-intent trajectories. Its Collaborative Intent Distribution (CID) loss leverages the observed trajectories of users with similar profiles as collaborative supervision, enabling the generator to sample multiple plausible and diverse intent trajectories. Finally, the Behavior Generator fine-tunes an LLM to generate multiple plausible target-domain sequences, each jointly conditioned on the user's transferable preference profile and a corresponding sampled target-intent trajectory. Extensive experiments across four cross-domain transfer settings on public datasets show that, under the same synthesis budget, IntExpo significantly outperforms the strongest baseline. Further analyses show that IntExpo effectively mitigates Target-Intent Collapse and generates diverse yet plausible target-domain behaviors.

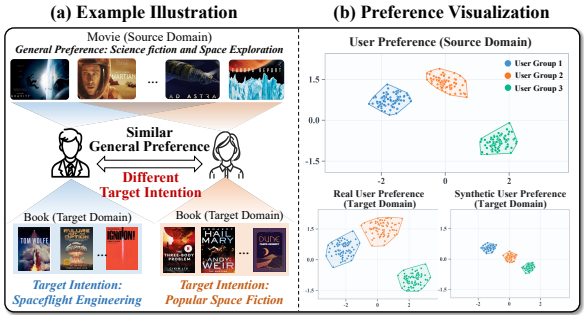


Figure 1: Example illustration and preference visualization of the target-intent collapse problem.

Overall, our main contributions are summarized as follows:

- **Target-Intent Collapse.** We identify and systematically investigate the Target-Intent Collapse problem in CDSS, whereby existing methods map similar source-domain preferences to overly concentrated target-domain behaviors, failing to capture the diversity and plausibility of target intents.
- **IntExpo Framework.** We propose IntExpo, a three-stage framework that generates target-domain sequences from transferable preference profiles and diverse target-intent trajectories. It introduces discriminator-guided reinforcement learning for transferable profile generation and a CID loss for learning diverse and plausible target-intent trajectories from users.
- **Empirical Results.** Extensive experiments across four cross-domain transfer settings demonstrate that, under the same synthesis budget, IntExpo significantly outperforms the strongest baseline. Further analyses confirm its effectiveness in mitigating the Target-Intent Collapse problem.

## 2 PRELIMINARY

**Concepts and Symbols.** Let $\mathcal{D}^s$ and $\mathcal{D}^t$ denote the source and target domains, with item sets $\mathcal{I}^s = \{v_i^s\}_{i=1}^{|\mathcal{I}^s|}$ and $\mathcal{I}^t = \{v_j^t\}_{j=1}^{|\mathcal{I}^t|}$. For each user $u$, let $S_u^d = [v_1^d, \ldots, v_{N^d}^d]$ denote their interaction sequence in domain $d \in \{s, t\}$. We use $\mathcal{U}^o$ to denote users observed in both domains and $\mathcal{U}^{s\setminus t}$ to denote users observed only in the source domain.

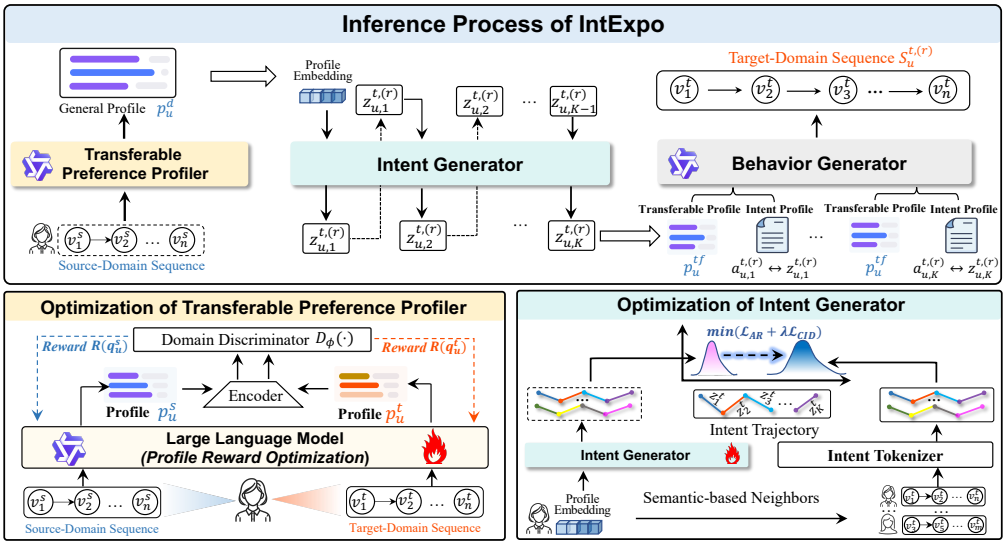


Figure 2: The framework of IntExpo, which consists of three core components, including a Transferable Preference Profiler, an Intent Generator and a Behavior Generator.

**Problem Formalization.** Cross-domain sequence synthesis (CDSS) synthesizes target-domain sequences for source-only users to augment the training data for target-domain recommenders. Using paired sequences from overlapping users $u \in \mathcal{U}^o$, conventional CDSS directly models the target sequence $S_u^t$ conditioned on the source sequence $S_u^s$. IntExpo instead conditions sequence generation on a transferable preference profile $p_u^{tf}$ and a target-intent trajectory $\mathbf{Z}_u^t = [z_1^t, \cdots, z_K^t]$. Their optimization objectives are formulated as follows:

$$\underbrace{\theta^* = \arg\min_{\theta} \left\{ -\sum_{u \in \mathcal{U}^o} \log p_\theta(S_u^t \mid S_u^s) \right\}}_{\text{Conventional CDSS}}, \quad \underbrace{\Theta^* = \arg\min_{\Theta} \left\{ -\sum_{u \in \mathcal{U}^o} \log p_\Theta(S_u^t \mid p_u^{tf}, \mathbf{Z}_u^t) \right\}}_{\text{IntExpo}}. \tag{1}$$

## 3 Methodology

As shown in Figure 2, IntExpo comprises three modules: a *Transferable Preference Profiler*, an *Intent Generator*, and a *Behavior Generator*. We introduce each of these modules in detail below.

### 3.1 Transferable Preference Profiler

Given the source-domain sequence $S_u^s$, the Transferable Preference Profiler uses an LLM $G_\theta(\cdot)$ (Team, 2026) to generate a transferable preference profile, i.e., $p_u^{tf} = G_\theta(S_u^s)$. The resulting profile is designed to capture the user's transferable, domain-agnostic preferences and serves as a conditioning signal for both target-intent and target-behavior generation.

However, prompting alone cannot guarantee that $p_u^{tf}$ is transferable. Source-domain-specific cues in $S_u^s$, such as item categories, content attributes, and interaction patterns, may persist in the generated profile, making its domain origin identifiable and limiting its cross-domain transferability.

To address this issue, we introduce a domain discriminator and use its feedback as a reward signal to fine-tune the LLM through reinforcement learning. Specifically, for each overlapping user $u \in \mathcal{U}^o$, we first apply the same prompt to the source sequence $S_u^s$ and target sequence $S_u^t$ separately, obtaining two preference profiles $p_u^s$ and $p_u^t$ (see Appendix B.3 for prompt details and examples):

$$p_u^s = G_\theta(S_u^s), \quad p_u^t = G_\theta(S_u^t). \tag{2}$$

Thereafter, we use Sentence-BERT (Reimers & Gurevych, 2019) as encoder $E(\cdot)$ to map the $p_u^s$ and $p_u^t$ to their semantic representations $\mathbf{h}_u^s \in \mathbb{R}^d$ and $\mathbf{h}_u^t \in \mathbb{R}^d$. We feed these representations into the

domain discriminator $D_\phi(\cdot)$:

$$q_u^d = D_\phi(\mathbf{h}_u^d),\ d \in \{s,t\}, \quad \mathbf{h}_u^s = E(p_u^s), \quad \mathbf{h}_u^t = E(p_u^t) \tag{3}$$

where $D_\phi(\cdot)$ is composed of MLPs (see Appendix B.2 for details) and $q_u^d \in (0,1)$ is the probability that the discriminator assigns the profile to the target domain. We label source- and target-derived profiles as 0 and 1, respectively, and optimize $D_\phi(\cdot)$ using the binary cross-entropy loss:

$$\mathcal{L}_D = -\frac{1}{2|\mathcal{U}^o|} \sum_{u \in \mathcal{U}^o} \left[\log(1 - q_u^s) + \log q_u^t\right]. \tag{4}$$

During discriminator training, we freeze Sentence-BERT and the LLM. By minimizing $\mathcal{L}_D$, the discriminator learns to identify domain-specific information retained in the generated profiles.

After optimizing $D_\phi(\cdot)$, we freeze its parameters and treat it as a domain-information detector. We then optimize the LLM with RL so that both $p_u^s$ and $p_u^t$ are difficult for the discriminator to identify to a specific domain. To this end, we define a profile reward $R_u$ as the normalized entropy of the discriminator outputs and use it to optimize the LLM via GRPO $f_{\text{GRPO}}(\cdot)$ (Shao et al., 2024):

$$\mathcal{J}_{\text{GRPO}}(\theta_G) = f_{\text{GRPO}}\big(\theta_G; \{R_u\}_{u \in \mathcal{U}^o}\big),\ R_u = -\frac{1}{2} \sum_{d \in \{s,t\}} \frac{q_u^d \log q_u^d + (1 - q_u^d)\log(1 - q_u^d)}{\log 2}, \tag{5}$$

where $\theta_G$ denotes the parameters of the LLM $G_{\theta_G}(\cdot)$. The $\log 2$ normalizes the reward to $[0,1]$, and the maximum is reached when $q_u^s = q_u^t = 0.5$. To avoid degenerate solutions, GRPO includes KL regularization to constrain deviations from the reference policy. With the optimized LLM, we can generate the transferable preference profile $p_u^{tf}$ given the source-domain sequence $S_u^s$, i.e., $p_u^{tf} = G_{\theta_G^*}(S_u^s)$.

### 3.2 Intent Generator

Given the transferable preference profiles $p_u^{tf} = G_{\theta_G^*}(S_u^s)$, the Intent Generator learns the conditional distribution of a user's target-domain intents and generates multiple plausible trajectories. It consists of two components: (i) *Intent Tokenizer* and (ii) *Intent-Path Distribution Learning*.

• **Intent Tokenizer.** The interaction data contain no explicit intent annotations. We therefore divide each target-domain sequence into multiple sessions and treat each session as a behavior segment with a relatively consistent intent. Specifically, for the target-domain sequence $S_u^t$ of user $u \in \mathcal{U}^o$, we start a new session whenever the time gap between two consecutive items exceeds a threshold $\Delta$, resulting in the session sequence $\mathcal{B}_u^t$:

$$\mathcal{B}_u^t = \text{Sessionize}(S_u^t; \Delta) = [B_{u,1}^t, \ldots, B_{u,T_u}^t], \tag{6}$$

where $B_{u,m}^t$ denotes the $m$-th target-domain session of user $u$, and $T_u$ is the total number of sessions.

For each session $B_{u,m}^t$, we feed the metadata of all its items, including their titles, categories, and descriptions, into a frozen LLM $F_{\text{int}}(\cdot)$ to obtain a concise textual intent (see Appendix B.3 for prompt details). Because the LLM generates free-form text, semantically equivalent intents may be phrased differently. To group semantically equivalent intents into the same intent ID, we encode textual intent $a_{u,m}$ into a semantic vector $\mathbf{x}_{u,m} \in \mathbb{R}^d$ using a frozen Sentence-BERT $E_I(\cdot)$:

$$\mathbf{x}_{u,m} = E_I(a_{u,m}), \qquad a_{u,m} = F_{\text{int}}\left(\text{Meta}(B_{u,m}^t); \mathcal{P}_{\text{int}}\right), \tag{7}$$

where $\mathcal{P}_{\text{int}}$ is a fixed prompt. **This preprocessing is performed only once offline, adding little runtime overhead.**

We then apply $K$-means to all session-level target-intent representations $\mathbf{x}_{u,m}$ in the training set, yielding $C$ clusters with centroids $\{\boldsymbol{\mu}_c\}_{c=1}^C$. Each session $B_{u,m}^t$ is assigned to its nearest centroid, and the corresponding cluster index serves as its discrete intent ID $z_{u,m}$:

$$z_{u,m} = \underset{c \in \{1,\ldots,C\}}{\arg\min} \|\mathbf{x}_{u,m} - \boldsymbol{\mu}_c\|_2^2. \tag{8}$$

For each cluster $c$, we select, among its assigned sessions, the textual intent whose semantic representation is closest to $\boldsymbol{\mu}_c$ as the canonical description $\bar{a}_c$. The resulting cluster ID, centroid, and canonical-description triplets define the shared target-intent vocabulary $\mathcal{V}_I = \{(c, \boldsymbol{\mu}_c, \bar{a}_c)\}_{c=1}^C$.

Following the temporal order of the sessions, we represent sequence $S_u^t$ as a discrete intent trajectory $Z_u^t$ and its textual counterpart $A_u^t$:

$$Z_u^t = [z_{u,1}, \ldots, z_{u,T_u}], \qquad A_u^t = [\bar{a}_{z_{u,1}}, \ldots, \bar{a}_{z_{u,T_u}}]. \tag{9}$$

The discrete trajectory $Z_u^t$ is used for subsequent intent-path distribution learning, while the textual trajectory $A_u^t$ provides semantic guidance for target-domain behavior generation.

• **Intent-Path Distribution Learning.** Given the transferable preference representation $\mathbf{h}_u^{tf} = E(p_u^{tf})$, this part learns the conditional distribution of their target-domain intent trajectory, i.e., $P_\psi(Z^t \mid \mathbf{h}_u^{tf})$.

Specifically, we project $\mathbf{h}_u^{tf}$ into a condition token and prepend it to the intent sequence. A causal Transformer then predicts the intent IDs autoregressively. For the target-domain intent trajectory $Z_u^t = [z_{u,1}, \ldots, z_{u,T_u}]$, its conditional probability is factorized as:

$$P_\psi(Z_u^t \mid \mathbf{h}_u^{tf}) = \prod_{m=1}^{T_u+1} P_\psi\left(z_{u,m} \mid z_{u,<m}, \mathbf{h}_u^{tf}\right), \tag{10}$$

where $z_{u,0} = \mathrm{BOS}$ and $z_{u,T_u+1} = \mathrm{EOS}$ mark the beginning and end of the trajectory, respectively, and $\psi$ denotes the trainable Transformer parameters. Using each overlapping user's observed intent trajectory as supervision, we define the autoregressive loss $\mathcal{L}_{AR}$:

$$\mathcal{L}_{\mathrm{AR}} = -\frac{1}{|\mathcal{U}^o|} \sum_{u \in \mathcal{U}^o} \log P_\psi\left(Z_u^t \mid \mathbf{h}_u^{tf}\right). \tag{11}$$

However, each $\mathbf{h}_u^{tf}$ is paired with only one observed trajectory in the training set. The model may therefore overlook other plausible target intents, causing the outputs for users with similar transferable preferences to concentrate on only a few intent trajectories at inference.

To address this issue, we use the observed trajectories of users with similar transferable preferences as additional plausible targets. Specifically, for each $u \in \mathcal{U}^o$, we retrieve $K$ neighbors $\mathcal{N}(u) \subseteq \mathcal{U}^o \setminus \{u\}$ based on the cosine similarity between their transferable preference representations. We then combine their observed intent trajectories into a weighted collaborative intent distribution:

$$P_u^{\mathrm{col}}(Z) = \sum_{v \in \mathcal{N}(u)} w_{u,v}\, \delta_{Z_v^t}(Z), \quad w_{u,v} = \frac{\exp\big(\mathrm{sim}(\mathbf{h}_u^{tf}, \mathbf{h}_v^{tf})/\tau\big)}{\sum_{v' \in \mathcal{N}(u)} \exp\Big(\mathrm{sim}(\mathbf{h}_u^{tf}, \mathbf{h}_{v'}^{tf})/\tau\Big)}, \tag{12}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity, and $\tau$ is a temperature parameter. Here, $\delta_{Z_v^t}(Z) = 1$ if $Z = Z_v^t$ and 0 otherwise. Thus, $P_u^{\mathrm{col}}$ is a weighted empirical distribution over the observed trajectories of user $u$'s neighbors.

To guide the generator toward this distribution, we define the Collaborative Intent Distribution (CID) loss $\mathcal{L}_{CID}$:

$$\mathcal{L}_{\mathrm{CID}} = -\frac{1}{|\mathcal{U}^o|} \sum_{u \in \mathcal{U}^o} \mathbb{E}_{Z \sim P_u^{\mathrm{col}}}\left[\log P_\psi(Z \mid \mathbf{h}_u^{tf})\right] = -\frac{1}{|\mathcal{U}^o|} \sum_{u \in \mathcal{U}^o} \sum_{v \in \mathcal{N}(u)} w_{u,v} \log P_\psi\big(Z_v^t \mid \mathbf{h}_u^{tf}\big). \tag{13}$$

Conditioned on $\mathbf{h}_u^{tf}$, this loss maximizes the weighted likelihood of the neighbors' observed trajectories. It therefore encourages the model to assign probability to multiple plausible target-intent patterns rather than concentrating on a single pattern.

Overall, the final training objective combines the two supervision signals:

$$\mathcal{L} = \mathcal{L}_{\mathrm{AR}} + \lambda \mathcal{L}_{\mathrm{CID}}, \tag{14}$$

where $\lambda$ controls their relative contributions. Appendix B.1 interprets the objective as fitting a mixture of the user's observed trajectory and the collaborative intent distribution, and derives an upper bound on the intent collision probability, explaining how CID mitigates the Target-Intent Collapse problem.

At inference, given only a source-only user's transferable preference representation $\mathbf{h}_u^{tf}$, we independently sample $R$ plausible target-intent trajectories from the conditional distribution learned by the optimized generator, i.e., $\widehat{Z}_u^{t,(r)} \overset{\text{i.i.d.}}{\sim} P_\psi(\cdot \mid \mathbf{h}_u^{tf})$, where $r = 1, \ldots, R$.

### 3.3 BEHAVIOR GENERATOR

The Behavior Generator fine-tunes an LLM to generate concrete target-domain sequences conditioned on a user's transferable preference profile and target intent trajectory.

Since the target domain contains a large number of items, generating a target-domain sequence from the full item set $\mathcal{I}^t$ creates a large search space for the LLM. We therefore construct a subset of item set $\mathcal{I}^t$ for each target-intent. As described in Sec. 3.2, each training session $B^t_{u,m} = [v^t_1, \ldots, v^t_{N_{u,m}}]$ is assigned an intent ID $z_{u,m}$. For a target-intent $z$, we collect all sessions satisfying $z_{u,m} = z$ and use their distinct items as candidates $\mathcal{I}^{\text{cand}}_z$.

Subsequently, we train a LoRA adapter while keeping the base LLM frozen. Formally, for each target intent $z_{u,i} \in Z^t_u$, the LLM receives three inputs: its textual description $\bar{a}_{z_{u,i}} \in A^t_u$, the user's transferable preference profile $p^{tf}_u$, and the IDs with metadata of the candidate items in $\mathcal{I}^{\text{cand}}_{z_{u,i}}$. The observed session $B^t_{u,i}$ serves as the supervised target:

$$\mathcal{L}_{\text{BG}} = -\frac{1}{|\mathcal{U}^o|} \sum_{u \in \mathcal{U}^o} \frac{1}{T_u} \sum_{i=1}^{T_u} \log P_{\theta_B} \left( B^t_{u,i} \mid p^{tf}_u, \bar{a}_{z_{u,i}}, \mathcal{I}^{\text{cand}}_{z_{u,i}} \right), \tag{15}$$

where $T_u$ is the number of sessions of target sequence $S^t_u$, and $\theta_B$ is the trainable LoRA parameters.

Given an optimized IntExpo, the inference process can be summarized as follows:

- (i) The Transferable Preference Profiler generates the user's transferable preference profile $p^{tf}_u$ from $S^s_u$.
- (ii) Conditioned on the representation $\mathbf{h}^{tf}_u$ of $p^{tf}_u$, the Intent Generator produces $R$ target intent trajectories, $\mathcal{Z} = \{Z^{t,(1)}_u, \cdots, Z^{t,(R)}_u\}$.
- (iii) The Behavior Generator generates the session $\widehat{B}^{t,(r)}_{u,i}$ for each intent $z^{t,(r)}_{u,i} \in Z^{t,(r)}_u$ in each trajectory, and the sessions from the same trajectory form the target-domain sequence $S^{t,(r)}_u$.

Overall, the inference process of IntExpo can be summarized as $S^s_u \rightarrow p^{tf}_u \rightarrow \{\widehat{Z}^{t,(r)}_u\}^R_{r=1} \rightarrow \{S^{t,(r)}_u\}^R_{r=1}$. IntExpo is optimized using overlapping users' source-domain histories and only the training portions of their target-domain sequences.

## 4 EXPERIMENTS

### 4.1 EXPERIMENTAL SETTINGS

**Datasets.** Following (Li et al., 2025a; Liu et al., 2026), we conduct experiments on the widely used Amazon dataset. It covers four domains (Movie, Book, Clothing, and Sports), which are paired into two cross-domain scenarios (Movie-Book and Clothing-Sports). Data preprocessing and detailed statistics are provided in Appendix C.1.

**Baselines.** We compare IntExpo with representative methods from three categories: (1) *Single-domain sequential recommendation methods*, including GRU4Rec (Hidasi et al., 2015), SASRec (Kang & McAuley, 2018), and BERT4Rec (Sun et al., 2019); (2) *Cross-domain recommendation methods*, including PTUPCDR (Zhu et al., 2022), DisenCDR (Cao et al., 2022b), D2C2R (Mao et al., 2024), and CDRNP (Li et al., 2024); and (3) *Cross-domain sequence synthesis methods*, including LASSO (Chen et al., 2025) and LGCD (Lu et al., 2026). Detailed descriptions of these baselines are provided in Appendix C.2.

**Evaluation Metrics**. We use Recall@$N$ and NDCG@$N$ for evaluation, where $N = 10$. They measure the ability to retrieve relevant items and rank them correctly, respectively.

### 4.2 OVERALL PERFORMANCE

Following the standard intra-domain CDR protocol (Li et al., 2025a; Kang et al., 2019; Zhu et al., 2022), we evaluate recommendation within the target domain while using source-domain information as auxiliary input. Under the chronological leave-one-out split, the penultimate and final target-domain interactions of each overlapping user are reserved for validation and testing, respectively,

Table 1: Overall performance under the intra-domain CDR protocol, averaged over five runs. Cells containing the best and second-best results are shaded yellow and gray, respectively; † indicates statistical significance under a paired $t$-test ($p < 0.05$).

| Methods | Movie → Book | | Book → Movie | | Clothing → Sports | | Sports → Clothing | |
|---|---|---|---|---|---|---|---|---|
| | R@10 | N@10 | R@10 | N@10 | R@10 | N@10 | R@10 | N@10 |
| SASRec(2018) | 0.0731 | 0.0448 | 0.1497 | 0.0861 | 0.0925 | 0.0507 | 0.0496 | 0.0289 |
| GRU4Rec(2015) | 0.0606 | 0.0361 | 0.1312 | 0.0741 | 0.1097 | 0.0610 | 0.0410 | 0.0221 |
| BERT4Rec(2019) | 0.0592 | 0.0337 | 0.1047 | 0.0576 | 0.0749 | 0.0418 | 0.0448 | 0.0220 |
| PTUPCDR(2022) | 0.0752 | 0.0463 | 0.1487 | 0.0857 | 0.0937 | 0.0515 | 0.0630 | 0.0361 |
| DisenCDR(2022) | 0.0749 | 0.0462 | 0.1490 | 0.0855 | 0.0896 | 0.0498 | 0.0541 | 0.0327 |
| D2C2R(2024) | 0.0594 | 0.0365 | 0.1418 | 0.0808 | 0.0854 | 0.0475 | 0.0569 | 0.0328 |
| CDRNP(2024) | 0.0773 | 0.0475 | 0.1516 | 0.0877 | 0.0925 | 0.0511 | 0.0589 | 0.0369 |
| LASSO(2025) | 0.0838 | 0.0515 | 0.1510 | 0.0870 | 0.1099 | 0.0649 | 0.0752 | 0.0440 |
| LGCD(2026) | 0.0812 | 0.0486 | 0.1497 | 0.0844 | 0.1043 | 0.0616 | 0.0703 | 0.0398 |
| **IntExpo** | **0.0853**† | **0.0534**† | **0.1570**† | **0.0915**† | **0.1155**† | **0.0685**† | **0.0854**† | **0.0476**† |
| $Imp(\%)$ | 1.80% | 3.69% | 3.56% | 4.33% | 5.10% | 5.55% | 13.56% | 8.18% |

while all preceding interactions form the original target-domain training set. For a fair comparison, each CDSS method generates exactly one target-domain sequence per source-only user. These synthetic sequences are added only to the target-domain training set, on which a common SASRec recommender is trained. Performance is evaluated on the held-out test items of overlapping users.

Table 1 reports the performance of all methods under four cross-domain transfer settings. The experimental results show that IntExpo achieves the best results on all eight metrics. Compared with the strongest baseline, IntExpo improves Recall@10 and NDCG@10 by 6.01% and 5.44% on average, respectively, with consistent gains across different transfer directions. These results indicate that explicitly modeling diverse target-domain intents alleviates the Target-Intent Collapse problem and improves downstream recommendation performance.

Table 2: Ablation Study of IntExpo. Results are averaged over five runs.

| Variant Models | Movie → Book | | Book → Movie | | Clothing → Sports | | Sports → Clothing | |
|---|---|---|---|---|---|---|---|---|
| | R@10 | N@10 | R@10 | N@10 | R@10 | N@10 | R@10 | N@10 |
| w/o DD | 0.0784 | 0.0454 | 0.1548 | 0.0897 | 0.1139 | 0.0660 | 0.0836 | 0.0462 |
| w/o IG | 0.0822 | 0.0508 | 0.1547 | 0.0904 | 0.1059 | 0.0625 | 0.0793 | 0.0451 |
| w/o CID | 0.0824 | 0.0509 | 0.1538 | 0.0906 | 0.1106 | 0.0638 | 0.0809 | 0.0455 |
| w/o BGS | 0.0831 | 0.0523 | 0.1522 | 0.0899 | 0.1001 | 0.0616 | 0.0822 | 0.0464 |
| **IntExpo** | **0.0853**† | **0.0534**† | **0.1570**† | **0.0915**† | **0.1155**† | **0.0685**† | **0.0854**† | **0.0476**† |

### 4.3 Ablation Study

To evaluate the contribution of each component in IntExpo, we consider the following variants:

- **w/o Domain Discriminator (DD)**, which removes the domain discriminator and directly generates user preference profiles with the base LLM.
- **w/o Intent Generator (IG)**, which removes explicit target-domain intent modeling and conditions the Behavior Generator only on the transferable preference profile.
- **w/o CID Loss (CID)**, which removes the Collaborative Intent Distribution loss $\mathcal{L}_{CID}$ and trains the Intent Generator using only the autoregressive loss $\mathcal{L}_{AR}$.
- **w/o Behavior Generator SFT (BGS)**, which removes supervised fine-tuning (SFT) of the Behavior Generator and directly uses the frozen base LLM to generate behavior sequences.

The results are reported in Table 2. Removing any component leads to a performance drop, confirming the contribution of each design. Removing the Intent Generator or the CID loss, i.e.,*w/o IG* and *w/o CID*, causes the significant decline. These show that explicitly modeling target-domain intents and learning the intent distribution of users with similar preferences are crucial for alleviating target-intent collapse. Removing the domain discriminator (*w/o DD*) also degrades performance, demonstrating the importance of transferable preference profiles for cross-domain transfer. Finally, removing SFT of the Behavior Generator (*w/o BGS*) reduces performance, indicating that task-specific adaptation is necessary for generating coherent target-domain sequences.

Table 3: Performance comparison on cross-domain cold-start recommendation tasks. All results are averaged over five runs.

| Methods | Movie → Book | | Book → Movie | | Clothing → Sports | | Sports → Clothing | |
|---|---|---|---|---|---|---|---|---|
| | R@10 | N@10 | R@10 | N@10 | R@10 | N@10 | R@10 | N@10 |
| PTUPCDR | 0.0505 | 0.0271 | 0.0320 | 0.0158 | 0.0486 | 0.0252 | 0.0394 | 0.0254 |
| CDRNP | 0.0459 | 0.0262 | 0.0281 | 0.0138 | 0.0307 | 0.0118 | 0.0726 | 0.0349 |
| LASSO | 0.0432 | 0.0244 | 0.0334 | 0.0170 | 0.0921 | 0.0495 | 0.0665 | 0.0408 |
| LGCD | 0.0349 | 0.0180 | 0.0362 | 0.0187 | 0.0358 | 0.0191 | 0.0435 | 0.0207 |
| **IntExpo** | **0.0519**$^{\dagger}$ | **0.0286**$^{\dagger}$ | **0.0387**$^{\dagger}$ | **0.0194**$^{\dagger}$ | **0.0972**$^{\dagger}$ | **0.0512**$^{\dagger}$ | **0.0844**$^{\dagger}$ | **0.0426**$^{\dagger}$ |

### 4.4 Cold-Start Recommendation

We evaluate IntExpo in the cross-domain cold-start setting, where test users have only source-domain histories. Users are split into training, validation, and test sets at a ratio of 8:1:1. For each test user, all target-domain interactions are hidden, and the first target-domain item is used as ground truth. Since single-domain recommenders require target-domain histories, we compare against two conventional CDR methods (PTUPCDR and CDRNP) and two CDSS methods (LASSO and LGCD). Conventional CDR methods rank target-domain items from the source history. For a fair comparison, each CDSS method generates one sequence per test user and returns the 10 highest-probability items from its first-item distribution. IntExpo first samples one target-intent trajectory, then uses it and the user's transferable preference profile to predict the first-item distribution.

Table 3 reports the results across four cross-domain tasks. IntExpo achieves the best performance across all tasks. Conventional CDR methods learn a direct source-to-target mapping, while LASSO and LGCD synthesize target behaviors without explicitly modeling the distribution of plausible target intents. By modeling this distribution before behavior generation, IntExpo better captures the one-to-many relationship between source preferences and target intents, thereby mitigating the Target-Intent Collapse problem in cold-start recommendation.

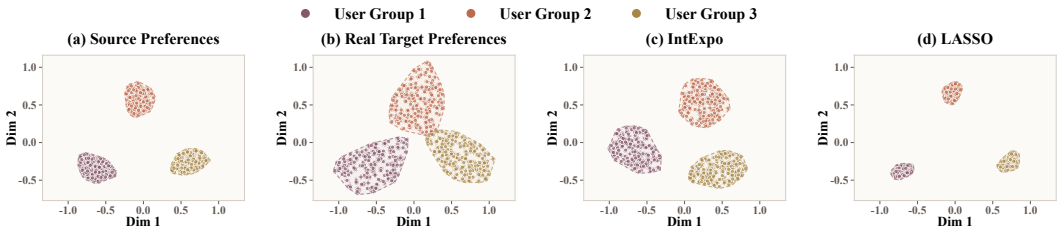


Figure 3: Target-intent collapse analysis: (a) source representations, (b) real target representations, and target representations synthesized by (c) IntExpo and (d) LASSO.

### 4.5 Intent Collapse Investigation

We investigate target-intent collapse, using Movie→Book as an illustrative example. We randomly select three held-out overlapping users as seeds and retrieve their 200 nearest neighbors by cosine similarity of SASRec-encoded source sequences. Figure 3-(a) visualizes these source representations using t-SNE. IntExpo and LASSO each generate one target sequence per user using only source histories. Real target histories are withheld from synthesis and serve as references. We encode all real and generated target sequences with the same frozen target-domain SASRec and jointly project them using t-SNE. Figure 3-(b) shows substantial within-group dispersion in the real target representations. IntExpo retains much of this dispersion (Figure 3-(c)), whereas LASSO produces much tighter clusters (Figure 3-(d)). This suggests that IntExpo alleviates target-intent collapse by learning diverse intent trajectories from similar users.

### 4.6 Effect of Synthesis Scale

We vary the number of generated sequences per user ($R \in \{1, 3, 5, 10\}$) and compare IntExpo with LASSO and IntExpo w/o CID, whose intent generator is trained solely with $\mathcal{L}_{\mathrm{AR}}$. As shown in

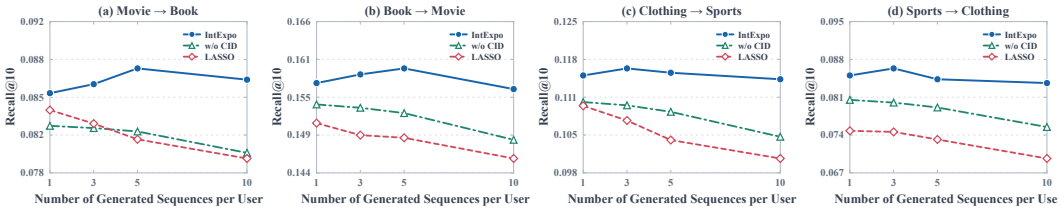


Figure 4: Effect of synthesis scale on IntExpo, w/o CID, and LASSO.

Figure 4, IntExpo performs best at $R = 3$ or $R = 5$, whereas both LASSO and w/o CID decline as $R$ increases. These results suggest that CID improves the utility of multiple generated sequences beyond sampling from an AR-only intent model. Nevertheless, excessive generation may introduce redundancy or noise, limiting the gains from further increasing $R$.

### 4.7 Case Study

We select a test user from the Clothing→Sports setting to illustrate the generation process of IntExpo using only the source-domain sequence. As shown in Figure 5, the user mainly interacts with outdoor products such as thermal clothing and hiking shoes, indicating transferable preferences for functionality and comfort. IntExpo summarizes these preferences into a transferable profile emphasizing durability, practicality, comfort, and outdoor preparedness.

Based on this profile, we sample three intent trajectories from the Intent Generator, corresponding to overnight camping, outdoor emergency preparation, and fitness and recovery. The intent descriptions shown in the figure use the intent name specified in Appendix B.3. The Behavior Generator then generates corresponding sequences of camping equipment, emergency tools, and fitness and recovery products. This case illustrates how IntExpo generates diverse, plausible, and interpretable target-domain behaviors consistent with the user's transferable preferences.

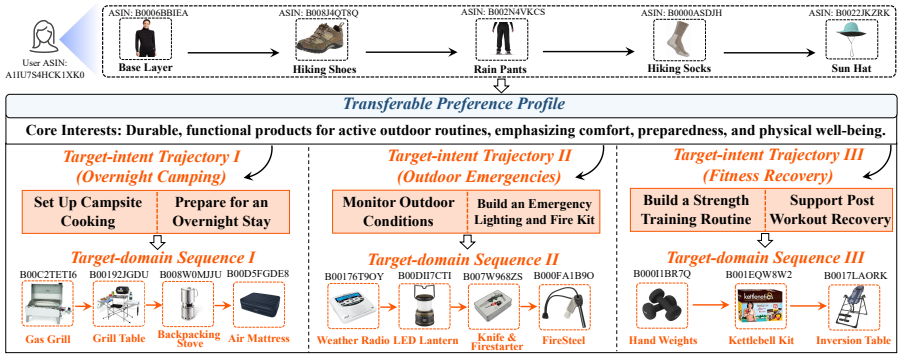


Figure 5: Case study with a random user.

### 4.8 Additional Experiments

We further conduct **Hyperparameter Sensitivity** and **Model Complexity Analyses**, which examine the impact of key hyperparameters and the computational cost of IntExpo, respectively. Detailed results are provided in Appendix C.3 and Appendix C.4.

## 5 Conclusion

We study the target-intent collapse problem in cross-domain sequence synthesis, where similar source preferences lead to overly similar target behaviors. We propose IntExpo, a three-stage framework that models general preferences, generates diverse target intents, and synthesizes intent-aligned target sequences. Experiments across multiple cross-domain tasks show that IntExpo improves recommendation performance and intent diversity.

## LLM USAGE DISCLOSURE

In preparing this manuscript, we used GPT-5.6 solely for grammar checking and language polishing. All suggested revisions were carefully reviewed and approved by the authors, who take full responsibility for the manuscript.

## ETHICS STATEMENT

This work uses publicly available Amazon Reviews benchmark datasets and processes only anonymized user–item interaction sequences and item metadata. We do not collect private user data or attempt to identify individual users, and all results are reported in aggregate. Since IntExpo synthesizes target-domain sequences for source-only users, these generated sequences should be treated as model-produced training signals rather than factual records of user behavior, and they may contain inaccuracies, hallucinations, or biases inherited from the data, item metadata, or LLM backbones. Real-world deployment should therefore include appropriate privacy safeguards, validation across user groups and domains, clear marking of synthetic data, and monitoring of potential feedback loops in downstream recommendations.

## REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide an anonymous repository at https://anonymous.4open.science/r/IntExpo-5672 containing the implementation, data preprocessing pipeline, sessionization and intent-tokenization code, collaborative neighbor retrieval, CID loss, GRPO-based profile training, LoRA-based behavior-generator training, evaluation scripts, prompts, and configuration files. The main paper and appendix report the dataset statistics, preprocessing and splitting protocol, baseline descriptions, model architecture, training objectives, hyperparameter settings, hardware configuration, evaluation metrics, and repeated-run protocol. The Amazon Reviews datasets and pretrained models used in our experiments are publicly available from their original sources, and these resources together support end-to-end reproduction of IntExpo.

## REFERENCES

Jiangxia Cao, Xin Cong, Jiawei Sheng, Tingwen Liu, and Bin Wang. Contrastive cross-domain sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 138–147, 2022a.

Jiangxia Cao, Xixun Lin, Xin Cong, Jing Ya, Tingwen Liu, and Bin Wang. Disencdr: Learning disentangled representations for cross-domain recommendation. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pp. 267–277, 2022b.

Shu Chen, Zitao Xu, Weike Pan, Qiang Yang, and Zhong Ming. A survey on cross-domain sequential recommendation. *arXiv preprint arXiv:2401.04971*, 2024.

Yue Chen, Susen Yang, Tong Zhang, Chao Wang, Mingyue Cheng, Chenyi Lei, and Han Li. Lasso: Large language model-based user simulator for cross-domain recommendation. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, pp. 207–216, 2025.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

Guangneng Hu, Yu Zhang, and Qiang Yang. Conet: Collaborative cross networks for cross-domain recommendation. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pp. 667–676, 2018.

Clark Mingxuan Ju, Leonardo Neves, Bhuvesh Kumar, Liam Collins, Tong Zhao, Yuwei Qiu, Qing Dou, Sohail Nizam, Sen Yang, and Neil Shah. Revisiting self-attention for cross-domain sequential recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1094–1105, 2025.

SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. Semi-supervised learning for cross-domain recommendation to cold-start users. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1563–1572, 2019.

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

Hanyu Li, Jiayu Li, Weizhi Ma, Peijie Sun, Haiyang Wu, Jingwen Wang, Yuekui Yang, Min Zhang, and Shaoping Ma. Cd-cdr: Conditional diffusion-based item generation for cross-domain recommendation. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*, pp. 1789–1798, 2025a.

Hourun Li, Yifan Wang, Zhiping Xiao, Jia Yang, Changling Zhou, Ming Zhang, and Wei Ju. Disco: graph-based disentangled contrastive learning for cold-start cross-domain recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 12049–12057, 2025b.

Pan Li and Alexander Tuzhilin. Ddtcdr: Deep dual transfer cross domain recommendation. In *Proceedings of the 13th international conference on web search and data mining*, pp. 331–339, 2020.

Xiaodong Li, Jiawei Sheng, Jiangxia Cao, Wenyuan Zhang, Quangang Li, and Tingwen Liu. Cdrnp: Cross-domain recommendation to cold-start users via neural process. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 378–386, 2024.

Guanyu Lin, Chen Gao, Yu Zheng, Jianxin Chang, Yanan Niu, Yang Song, Kun Gai, Zhiheng Li, Depeng Jin, Yong Li, et al. Mixed attention network for cross-domain sequential recommendation. In *Proceedings of the 17th ACM international conference on web search and data mining*, pp. 405–413, 2024.

Jing Liu, Lele Sun, Weizhi Nie, Peiguang Jing, and Yuting Su. Graph disentangled contrastive learning with personalized transfer for cross-domain recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 8769–8777, 2024a.

Qidong Liu, Xiangyu Zhao, Yejing Wang, Zijian Zhang, Howard Zhong, Chong Chen, Xiang Li, Wei Huang, and Feng Tian. Bridge the domains: Large language models enhanced cross-domain sequential recommendation. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*, pp. 1582–1592, 2025.

Weiming Liu, Jiajie Su, Chaochao Chen, and Xiaolin Zheng. Leveraging distribution alignment via stein path for cross-domain cold-start recommendation. *Advances in Neural Information Processing Systems*, 34:19223–19234, 2021.

Weiming Liu, Xiaolin Zheng, Chaochao Chen, Jiahe Xu, Xinting Liao, Fan Wang, Yanchao Tan, and Yew-Soon Ong. Reducing item discrepancy via differentially private robust embedding alignment for privacy-preserving cross domain recommendation. In *Forty-first International Conference on Machine Learning*, 2024b.

Ziwei Liu, Qidong Liu, Wanyu Wang, Yejing Wang, Pengyue Jia, Tong Xu, Wei Huang, Chong Chen, and Xiangyu Zhao. Llm-edt: Large language models enhanced cross-domain sequential recommendation with dual-phase training. In *SIGIR '26 - Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1186–1197. Association for Computing Machinery, 2026.

Ziang Lu, Lei Sang, Lin Mu, and Yiwen Zhang. From clues to generation: Language-guided conditional diffusion for cross-domain recommendation. *arXiv preprint arXiv:2604.05365*, 2026.

Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Jie Zhou. Triple sequence learning for cross-domain recommendation. *ACM Transactions on Information Systems*, 42(4): 1–29, 2024.

Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. Cross-domain recommendation: An embedding and mapping approach. In *Ijcai*, volume 17, pp. 2464–2470, 2017.

Qingyang Mao, Qi Liu, Zhi Li, Likang Wu, Bing Lv, and Zheng Zhang. Cross-reconstructed augmentation for dual-target cross-domain recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2352–2356, 2024.

Chung Park, Taesan Kim, Hyungjun Yoon, Junui Hong, Yelim Yu, Mincheol Cho, Minsung Choi, and Jaegul Choo. Pacer and runner: Cooperative learning framework between single-and cross-domain sequential recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2071–2080, 2024.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

Qwen Team. Qwen3. 5-omni technical report. *arXiv preprint arXiv:2604.15804*, 2026.

Alexandre B Tsybakov and Vladimir Zaiats. *Introduction to nonparametric estimation*, volume 11. Springer, 2009.

Haoran Xin, Ying Sun, Chao Wang, and Hui Xiong. Llmcdsr: Enhancing cross-domain sequential recommendation with large language models. *ACM Transactions on Information Systems*, 43(5): 1–33, 2025.

Wujiang Xu, Xuying Ning, Wenfang Lin, Mingming Ha, Qiongxu Ma, Qianqiao Liang, Xuewen Tao, Linxun Chen, Bing Han, and Minnan Luo. Towards open-world cross-domain sequential recommendation: A model-agnostic contrastive denoising approach. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 161–179, 2024a.

Wujiang Xu, Qitian Wu, Runzhong Wang, Mingming Ha, Qiongxu Ma, Linxun Chen, Bing Han, and Junchi Yan. Rethinking cross-domain sequential recommendation under open-world assumptions. In *Proceedings of the ACM Web Conference 2024*, pp. 3173–3184, 2024b.

Ziqi Yang, Zhaopeng Peng, Zihui Wang, Jianzhong Qi, Chaochao Chen, Weike Pan, Chenglu Wen, Cheng Wang, and Xiaoliang Fan. Federated graph learning for cross-domain recommendation. *Advances in neural information processing systems*, 37:64865–64888, 2024.

Hongyu Zhang, Dongyi Zheng, Xu Yang, Jiyuan Feng, and Qing Liao. Feddcsr: Federated cross-domain sequential recommendation via disentangled representation learning. *arXiv preprint arXiv:2309.08420*, 2023.

Jiaqing Zhang, Mingjia Yin, Hao Wang, Yuxin Tian, Yuyang Ye, Yawen Li, Wei Guo, Yong Liu, and Enhong Chen. Generative data transformation: From mixed to unified data. In *Proceedings of the ACM Web Conference 2026*, pp. 5764–5773, 2026.

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.

Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. Personalized transfer of user preferences for cross-domain recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pp. 1507–1515, 2022.

# Appendix


- **A. Related Work** ..... 13
- **B. Theoretical Analysis and Implementation Details** ..... 14
  - B.1 *Bounding Target-Intent Collapse with CID Loss* ..... 14
  - B.2 *Parameter Setting and Algorithm* ..... 16
  - B.3 *Details of LLM Prompts* ..... 17
- **C. Supplementary Experiments** ..... 18
  - C.1 *Dataset Description* ..... 18
  - C.2 *Baseline Introduction* ..... 18
  - C.3 *Hyperparameter Analysis* ..... 19
  - C.4 *Complexity and Inference Efficiency* ..... 19


## A Related Work

**Cross-Domain Recommendation (CDR).** CDR aims to leverage interactions from a data-rich source domain to alleviate data sparsity in the target domain. Early methods mainly connect domains through representation sharing or latent-space mapping (Man et al., 2017; Hu et al., 2018; Li & Tuzhilin, 2020). For example, EMCDR (Man et al., 2017) learns domain-specific user representations and uses overlapping users to learn a nonlinear mapping between them. CoNet (Hu et al., 2018) adds cross-connections between domain-specific networks to enable bidirectional transfer. Subsequent studies explore finer-grained and more personalized transfer (Cao et al., 2022b; Zhao et al., 2020; Zhu et al., 2022). CATN (Zhao et al., 2020) transfers aspect-level preferences, while PTUPCDR (Zhu et al., 2022) learns a personalized mapping for each user. DisenCDR (Cao et al., 2022b) improves transfer through disentangled learning. More recently, GDCCDR (Liu et al., 2024a) combines item-aware disentanglement with personalized transfer, while DisCo (Li et al., 2025b) models diverse user intents and aligns them across domains.

Beyond static preference transfer, cross-domain sequential recommendation models temporal dependencies across domains (Chen et al., 2024). $C^2$DSR (Cao et al., 2022a) jointly models temporal dependencies within sequences and collaborative relationships across sequences, while Tri-CDR (Ma et al., 2024) encodes source-domain, target-domain, and mixed-domain sequences to capture their connections. MAN (Lin et al., 2024) combines local and global attention, while AMID (Xu et al., 2024b) transfers multi-interest information among overlapping and non-overlapping users under open-world assumptions. LLM4CDSR (Liu et al., 2025) further enhances cross-domain representations through LLM-based item semantics and hierarchical user profiles.

A complementary line of work addresses transfer reliability and practical constraints. DisAlign (Liu et al., 2021) uses Stein-path distribution alignment for cross-domain cold-start recommendation. MACD (Xu et al., 2024a) denoises auxiliary behavior sequences, while SyNCRec (Park et al., 2024) and AutoCDSR (Ju et al., 2025) mitigate negative transfer through cooperative learning and attention-level optimization, respectively. For privacy-preserving transfer, RidCDR (Liu et al., 2024b) performs robust embedding alignment without shared users or items, and FedGCDR (Yang et al., 2024) uses federated graph learning to transfer useful knowledge while filtering harmful signals. FedDCSR (Zhang et al., 2023) extends federated transfer to sequential recommendation by disentangling domain-shared and domain-exclusive features.

Despite these advances, these methods primarily improve representations or prediction, rather than directly synthesize reusable target-domain training sequences for source-only users.

**Cross-Domain Sequence Synthesis (CDSS).**

Unlike model-centric representation transfer, CDSS takes a data-centric approach by synthesizing potential target-domain interactions for source-only users. LLMCDSR (Xin et al., 2025) uses an LLM to generate pseudo target-domain items and maps them to real items through collaborative-textual alignment and meta-recall. LASSO (Chen et al., 2025) treats an LLM as a cross-domain user simulator, using personalized candidate pools and confidence-guided inference to generate target-domain interactions. LGCD (Lu et al., 2026) combines LLM-based pseudo-interaction generation with conditional diffusion for cross-domain recommendation. Related approaches include

LLM-EDT (Liu et al., 2026), which generates and filters cross-domain items to augment behavior sequences, and TAESAR (Zhang et al., 2026), which reconstructs mixed-domain behaviors into target-aligned sequences through contrastive decoding.

However, improvements in sequence-level quality or diversity do not necessarily prevent synthetic behaviors from concentrating on a limited set of target-intent patterns. IntExpo addresses this complementary aspect by explicitly modeling a distribution over alternative target-intent trajectories compatible with the same source-derived preference before generating concrete target-domain behaviors.

# B THEORETICAL ANALYSIS AND IMPLEMENTATION DETAILS

## B.1 BOUNDING TARGET-INTENT COLLAPSE WITH CID LOSS

In this subsection, we analyze the CID loss $\mathcal{L}_{\text{CID}}$ (Eq. (13)) from a distributional perspective. We first show that the joint objective combining the AR and CID losses is equivalent to fitting a mixture of the user's observed trajectory and the neighbors' intent distribution. We then derive an upper bound on the conditional intent collision probability, showing how the CID loss mitigates Target-Intent Collapse at the distributional level.

For user $u$, let the conditional distribution produced by the Intent Generator be:

$$p_u(Z) = P_\psi(Z \mid \mathbf{h}_u^{tf}), \tag{16}$$

where $\mathbf{h}_u^{tf}$ is the user's transferable preference representation and $Z$ is a generated target-intent trajectory. Next, we independently sample two trajectories from this distribution $Z^{(1)}, Z^{(2)} \overset{\text{i.i.d.}}{\sim} p_u(Z)$. We define the degree of intent collapse for user $u$ as:

$$\mathcal{C}_u(p_u) = \Pr\left(Z^{(1)} = Z^{(2)}\right) = \sum_Z p_u(Z)^2. \tag{17}$$

We call $\mathcal{C}_u$ the intent collision probability, which measures how likely two independent generations are to produce the same intent trajectory. When $\mathcal{C}_u = 1$, all generations are identical. A smaller $\mathcal{C}_u$ indicates a less concentrated distribution, while $1 - \mathcal{C}_u$ is the probability of generating two different intent trajectories.

Therefore, our theoretical goal is to derive an upper bound on $\mathcal{C}_u$ and show when it is strictly smaller than 1.

For user $u$, let $Z_u^t$ denote the observed target-intent trajectory and $P_u^{\text{col}}$ the collaborative intent distribution formed by the user's neighbors. The per-user training objective is

$$\mathcal{L}_u = -\log p_u(Z_u^t) - \lambda \mathbb{E}_{Z \sim P_u^{\text{col}}} \log p_u(Z). \tag{18}$$

Based on it, we define the mixture target distribution as $Q_u$:

$$Q_u = \frac{1}{1+\lambda}\delta_{Z_u^t} + \frac{\lambda}{1+\lambda}P_u^{\text{col}}, \tag{19}$$

where $\delta_{Z_u^t}$ is a point-mass distribution concentrated on $Z_u^t$.

The cross-entropy between $Q_u$ and the model distribution $p_u$ is $H(Q_u, p_u) = -\mathbb{E}_{Z \sim Q_u} \log p_u(Z)$. Substituting the definition of $Q_u$ gives:

$$H(Q_u, p_u) = -\frac{1}{1+\lambda}\log p_u(Z_u^t) - \frac{\lambda}{1+\lambda}\mathbb{E}_{Z \sim P_u^{\text{col}}} \log p_u(Z). \tag{20}$$

Therefore, we have:

$$\mathcal{L}_u = (1+\lambda) H(Q_u, p_u). \tag{21}$$

Since $1 + \lambda$ is a positive constant, minimizing $\mathcal{L}_u$ is equivalent to fitting the model distribution $p_u$ to the mixture target $Q_u$. Thus, the joint objective no longer treats only $Z_u^t$ as the target, but also assigns probability to intent trajectories supported by similar users.

In addition, we define the probability assigned by $Q_u$ to intent trajectories different from $Z_u^t$ as $q_u$:

$$q_u = Q_u(Z \neq Z_u^t) = \frac{\lambda}{1+\lambda} P_u^{\text{col}}(Z \neq Z_u^t). \tag{22}$$

Accordingly, $Q_u$ assigns probability $1 - q_u$ to the user's observed trajectory and total probability $q_u$ to all alternative intent trajectories. Since the total probability assigned to all alternative intent trajectories is $q_u$, we have

$$\sum_{Z \neq Z_u^t} Q_u(Z)^2 \leq \left( \sum_{Z \neq Z_u^t} Q_u(Z) \right)^2 = q_u^2. \tag{23}$$

Therefore, the collision probability of the mixture target satisfies

$$\mathcal{C}_u(Q_u) = Q_u(Z_u^t)^2 + \sum_{Z \neq Z_u^t} Q_u(Z)^2 \leq (1-q_u)^2 + q_u^2 = 1 - 2q_u(1-q_u). \tag{24}$$

Based on these, we have:

$$\mathcal{C}_u(Q_u) \leq 1 - 2q_u(1-q_u). \tag{25}$$

When $0 < q_u < 1$, the collision probability of the CID target is strictly smaller than 1.

We next consider the practical case where the model cannot perfectly fit $Q_u$. We define the theoretical minimum loss as

$$\mathcal{L}_u^* = (1+\lambda) H(Q_u), \tag{26}$$

where $H(Q_u)$ is the entropy of $Q_u$. The gap between the current loss and this theoretical minimum is defined as

$$\eta_u = \mathcal{L}_u - \mathcal{L}_u^* = (1+\lambda)[H(Q_u, p_u) - H(Q_u)] = (1+\lambda) KL(Q_u || p_u). \tag{27}$$

We can drive $\text{KL}(Q_u \| p_u) = \frac{\eta_u}{1+\lambda}$. Applying Pinsker's inequality (Tsybakov & Zaiats, 2009) to the above KL divergence, we obtain

$$\text{TV}(Q_u, p_u) \leq \sqrt{\frac{1}{2} \text{KL}(Q_u \| p_u)} = \sqrt{\frac{\eta_u}{2(1+\lambda)}}, \tag{28}$$

where TV denotes the total variation distance between two distributions.

Since the collision probability is the probability that two independent samples are identical, its difference under $p_u$ and $Q_u$ is bounded by

$$|\mathcal{C}_u(p_u) - \mathcal{C}_u(Q_u)| \leq 2\,\text{TV}(p_u, Q_u). \tag{29}$$

Combining the above results gives

$$\mathcal{C}_u(p_u) \leq \mathcal{C}_u(Q_u) + \sqrt{\frac{2\eta_u}{1+\lambda}}. \tag{30}$$

Substituting the upper bound on $\mathcal{C}_u(Q_u)$(Eq. (25)), we obtain

$$\mathcal{C}_u(p_u) \leq 1 - 2q_u(1-q_u) + \sqrt{\frac{2\eta_u}{1+\lambda}}. \tag{31}$$

This result provides a quantitative upper bound on the conditional intent collision probability of the learned model. The bound is informative when $\sqrt{\frac{2\eta_u}{1+\lambda}} < 2q_u(1-q_u)$. Under this condition, $\mathcal{C}_u(p_u) < 1$, and equivalently,

$$\Pr_{Z^{(1)}, Z^{(2)} \overset{\text{i.i.d.}}{\sim} p_u} \left( Z^{(1)} \neq Z^{(2)} \right) \geq 2q_u(1-q_u) - \sqrt{\frac{2\eta_u}{1+\lambda}} > 0. \tag{32}$$

Therefore, when the collaborative distribution contains alternative intents and the model sufficiently fits the mixture target, CID loss rules out collapse of the conditional intent distribution.

**Algorithm 1** Training and Inference Procedure of IntExpo

**Data:** $\mathcal{D}^o = \{(S_u^s, S_u^t)\}_{u \in \mathcal{U}^o}$, $\mathcal{D}^{s\setminus t} = \{S_u^s\}_{u \in \mathcal{U}^{s\setminus t}}$, and target-item metadata, where $S_u^t$ contains only training interactions;
**Models:** Profiler $G_{\theta_G}(\cdot)$; auxiliary discriminator $D_\phi(\cdot)$; intent generator $P_\psi(\cdot)$; behavior generator $P_{\theta_B}(\cdot)$; frozen text encoders $E(\cdot), E_I(\cdot)$ and frozen LLM $F_{\text{int}}(\cdot)$;
**Output:** Optimized parameters $\Theta^\star = \{\theta_G^\star, \psi^\star, \theta_B^\star\}$ and synthetic sequence set $\widehat{\mathcal{D}}^t$;
**Procedure:**
For each $u \in \mathcal{U}^o$, compute $p_u^s, p_u^t$ and $\mathbf{h}_u^s, \mathbf{h}_u^t$ via Eqs. (2)–(3);
**Stage I: Transferable Preference Profiling**
  **for each discriminator-training epoch do**
    Freeze $G_{\theta_G}$ and $E$; recompute $q_u^d = D_\phi(\mathbf{h}_u^d)$ for $d \in \{s, t\}$ and update $D_\phi$ by minimizing $\mathcal{L}_D$ (Eq. (4));
  **end for**
  **for each profiler-training epoch do**
    Freeze $D_{\phi^\star}$ and $E$; sample profiles from $S_u^s, S_u^t$ and recompute their rewards (Eq. (5));
    update $G_{\theta_G}$ using GRPO with KL regularization;
  **end for**
For each $u \in \mathcal{U}^o$, obtain $p_u^{tf} = G_{\theta_G^\star}(S_u^s)$ and $\mathbf{h}_u^{tf} = E(p_u^{tf})$;
Sessionize $\{S_u^t\}_{u \in \mathcal{U}^o}$ and construct $\mathcal{V}_I$ and $\{(Z_u^t, A_u^t)\}_{u \in \mathcal{U}^o}$ via Eqs. (6)–(9);
For each $u \in \mathcal{U}^o$, retrieve $K$ profile-similar users from $\mathcal{U}^o \setminus \{u\}$ and construct $P_u^{\text{col}}$ via Eq. (12);
**Stage II: Intent Generator Training**
  **for each intent-generator epoch do**
    Update $P_\psi(\cdot \mid \mathbf{h}_u^{tf})$ by minimizing $\mathcal{L}_{\text{AR}} + \lambda \mathcal{L}_{\text{CID}}$ (Eq. (14));
  **end for**
Construct $\mathcal{I}_z^{\text{cand}}$ for every intent $z$ from the target-domain training sessions assigned to $z$;
**Stage III: Behavior Generator Training**
  **for each behavior-generator epoch do**
    Freeze the base LLM; update the LoRA parameters $\theta_B$ of $P_{\theta_B}$ by minimizing $\mathcal{L}_{\text{BG}}$ (Eq. (15));
  **end for**
**Inference: Sequence Synthesis**
  **for each source-only user** $u \in \mathcal{U}^{s\setminus t}$ **do**
    Obtain $p_u^{tf} = G_{\theta_G^\star}(S_u^s)$ and $\mathbf{h}_u^{tf} = E(p_u^{tf})$; independently sample $\widehat{Z}_u^{t,(r)} \sim P_{\psi^\star}(\cdot \mid \mathbf{h}_u^{tf})$ for $r = 1, \ldots, R$;
    For each $r$, decode every intent in $\widehat{Z}_u^{t,(r)}$ with $P_{\theta_B^\star}$, conditioned on $p_u^{tf}$, its canonical description, and the corresponding candidate items; concatenate the generated sessions in trajectory order into $\widehat{S}_u^{t,(r)}$;
  **end for**
**return** $\Theta^\star$ and $\widehat{\mathcal{D}}^t = \{\widehat{S}_u^{t,(r)} \mid u \in \mathcal{U}^{s\setminus t},\ r = 1, \ldots, R\}$;

### B.2 Parameter Setting and Algorithm

IntExpo is implemented in PyTorch. The Transferable Preference Profiler optimizes the LLM with GRPO, using group size 4, KL coefficient 0.005, learning rate $1e^{-5}$, and batch size 16. The Intent Generator uses a 4-layer Transformer with 8 attention heads, hidden dimension 256, dropout 0.1, K=16 nearest neighbors, and temperature $\tau = 0.5$. The Behavior Generator fine-tunes the base LLM with LoRA. We use the default LoRA settings from the original paper: rank 16, alpha 32, and 16-bit quantization. All experiments are conducted on NVIDIA A100 * 8. In Eq. (14), we set $\lambda = 1.0$. The number of intent clusters is $C = 512$, the session threshold is $\Delta = 120$ days. The detailed training settings are listed in Table 4.

Algorithm 1 summarizes the training and inference procedures of IntExpo. Here, $\theta_G$, $\psi$, and $\theta_B$ denote the trainable parameters of the Transferable Preference Profiler, Intent Generator, and Behavior Generator's LoRA adapter, respectively, while $D_\phi$ serves as an auxiliary discriminator for profiler

optimization. During training, IntExpo learns transferable preference profiles, constructs the target-intent vocabulary and collaborative intent distributions, and then optimizes the intent and behavior generators. At inference, IntExpo derives a transferable preference profile from a source-only user's source-domain sequence, samples $R$ target-intent trajectories conditioned on its representation, and decodes each trajectory into a synthetic target-domain sequence.

Table 4: The training settings of IntExpo.

| **Transferable Preference Profiler** | | **Intent Generator** | | **Behavior Generator** | |
|---|---|---|---|---|---|
| Parameter Name | Parameter Setting | Parameter Name | Parameter Setting | Parameter Name | Parameter Setting |
| Base LLM | Qwen3-4B | Session Threshold | 120 days | Base LLM | Qwen3.5-2B-Base |
| Sentence-BERT | BGE-large-en-v1.5 | Intent LLM $F_{int}(\cdot)$ | Qwen3.6-27B | LoRA Rank | 16 |
| Domain Discriminator | MLPs(1024/256/1) | Intent Clusters Size $C$ | 512 | LoRA Alpha | 32 |
| Group Size(GRPO) | 4 | Transformer Layers | 4 | Quantization Bits | BF16 |
| KL Coefficient(GRPO) | 0.005 | Attention Heads | 8 | Max Prompt Length | 8192 |
| Global Batch Size | 16 | Number of Neighbors $K$ | 16 | Global Batch Size | 32 |
| Grad Accumulation | 1/4 | Temperature $\tau$ | 0.5 | Grad Accumulation | 1/8 |
| Learning Rate | $1e^{-5}$ | Learning Rate | $1e^{-4}$ | Learning Rate | $1e^{-4}$ |

### B.3 Details of LLM Prompts

As shown in Figure 6, IntExpo uses three types of LLM prompt templates. First, the Transferable Preference Profiler ($G_{\theta_G}(\cdot)$) takes a user's chronologically ordered interactions and extracts transferable core interests and preferred attributes, producing a structured JSON profile. Second, the frozen teacher LLM ($F_{\text{int}}(\cdot)$) extracts an immediate intent from each target-domain training session, represented by a concise intent name and description. These textual intents are then semantically encoded and clustered using K-means to obtain discrete intent IDs; the clustering itself does not involve an LLM. Finally, the Behavior Generator ($P_{\theta_B}(\cdot)$) conditions on the transferable preference profile, the canonical description of a sampled target intent, and its candidate item set to generate an intent-consistent target-domain session. The same prompt structure is used across domains, with only the domain name, input fields, and item-token prefixes adapted accordingly. For clarity, repeated constraints and some candidate items are omitted from the figure.

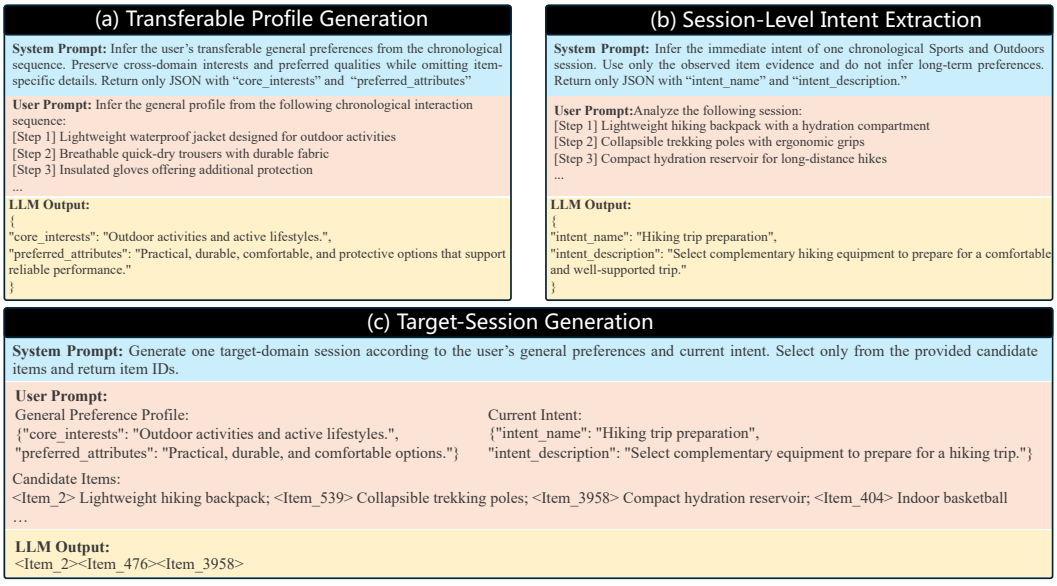


Figure 6: Examples of the three LLM prompt templates used in IntExpo. (a) Extracting transferable preferences from source-domain interaction sequences. (b) Identifying immediate intents from real target-domain sessions. (c) Generating ordered target-domain item sequences conditioned on transferable preferences, target intents, and candidate item sets.

# C SUPPLEMENTARY EXPERIMENTS

## C.1 DATASET DESCRIPTION

To evaluate IntExpo across different cross-domain scenarios, we use the Amazon Reviews dataset, covering four domains: Movie, Book, Clothing, and Sports. We consider both transfer directions within the Movie–Book and Clothing–Sports pairs, yielding four cross-domain transfer settings. User interactions are ordered chronologically, and users are categorized as overlapping or single-domain users. Overlapping users provide paired histories for learning cross-domain preference and behavior relationships. For each transfer direction, source-only users provide source-domain histories for synthesizing target-domain sequences. Under the intra-domain evaluation protocol, these synthetic sequences augment target-domain training, and recommendation performance is evaluated on held-out target-domain interactions of overlapping users. The cold-start evaluation follows the user-level split described in Section 4.4. Detailed dataset statistics are reported in Table 5.

Table 5: Statistics of the datasets used in the four cross-domain sequence synthesis tasks.

| Transfer Settings | Dataset | Users | Overlapping Users | Items | Interactions |
|---|---|---|---|---|---|
| Book → Movie | Movie | 86,572 | 37,388 | 50,052 | 1,697,533 |
| Movie → Book | Book | 566,280 | | 367,982 | 8,898,041 |
| Sports → Clothing | Clothing | 35,479 | 3,908 | 23,033 | 278,677 |
| Clothing → Sports | Sports | 31,690 | | 18,357 | 296,337 |

## C.2 BASELINE INTRODUCTION

To evaluate IntExpo on cross-domain sequence synthesis, we compare it with three categories of representative baselines: single-domain sequential recommendation methods, cross-domain recommendation methods, and cross-domain sequence synthesis methods.

**Single-Domain Sequential Recommendation Methods.**

- **GRU4Rec** (Hidasi et al., 2015) uses gated recurrent units to model users' sequential interaction patterns and predict the next item.
- **SASRec** (Kang & McAuley, 2018) employs unidirectional self-attention to capture both long- and short-term dependencies in user interaction sequences.
- **BERT4Rec** (Sun et al., 2019) adopts a bidirectional Transformer with masked-item prediction to learn contextualized sequential representations.

**Cross-Domain Recommendation Methods.**

- **PTUPCDR** (Zhu et al., 2022) learns personalized preference-transfer functions through meta-learning to transfer user preferences across domains.
- **DisenCDR** (Cao et al., 2022b) disentangles domain-shared and domain-specific user representations to facilitate cross-domain preference transfer.
- **D2C2R** (Mao et al., 2024) performs cross-reconstructed augmentation to improve recommendation in both source and target domains.
- **CDRNP** (Li et al., 2024) adopts a neural-process framework to model the distribution of user preferences and transfer knowledge to cold-start users across domains.

**Cross-Domain Sequence Synthesis Methods.**

- **LASSO** (Chen et al., 2025) uses an LLM as a cross-domain user simulator and generates target-domain interactions through personalized candidate construction and confidence-guided inference.
- **LGCD** (Lu et al., 2026) combines LLM-based pseudo-interaction generation with conditional diffusion for cross-domain recommendation. In our experiments, we use only its pseudo-interaction generation module to construct target-domain sequences.

For a fair comparison, all methods use the same data splits and evaluation protocol. For cross-domain sequence synthesis methods, we generate one target-domain sequence for each source-only

user. The synthesized sequences are combined with the original target-domain training sequences of overlapping users to train SASRec. Evaluation is conducted on the held-out target-domain items of overlapping users following the leave-one-out protocol.

### C.3 Hyperparameter Analysis

We analyze the effects of the CID loss weight $\lambda$ and the number of collaborative neighbors $K$ on IntExpo using Recall@10 across four cross-domain settings. As shown in Figure 7, performance varies non-monotonically with $\lambda$, and the best-performing value differs across settings. An appropriate CID weight improves performance over $\lambda = 0$, while larger weights do not consistently yield further gains. These results suggest that balancing individual and collaborative supervision is important. We use $\lambda = 1$ as the default across all settings, although it is not optimal in every setting. The effect of $K$ also varies across settings. Expanding the neighborhood can provide useful collaborative supervision, but including more neighbors does not always improve performance, potentially because less relevant neighbors introduce noise. Performance peaks at $K = 16$ in three settings, while Clothing→Sports benefits from a smaller neighborhood. We therefore adopt $K = 16$ as a shared default across settings.

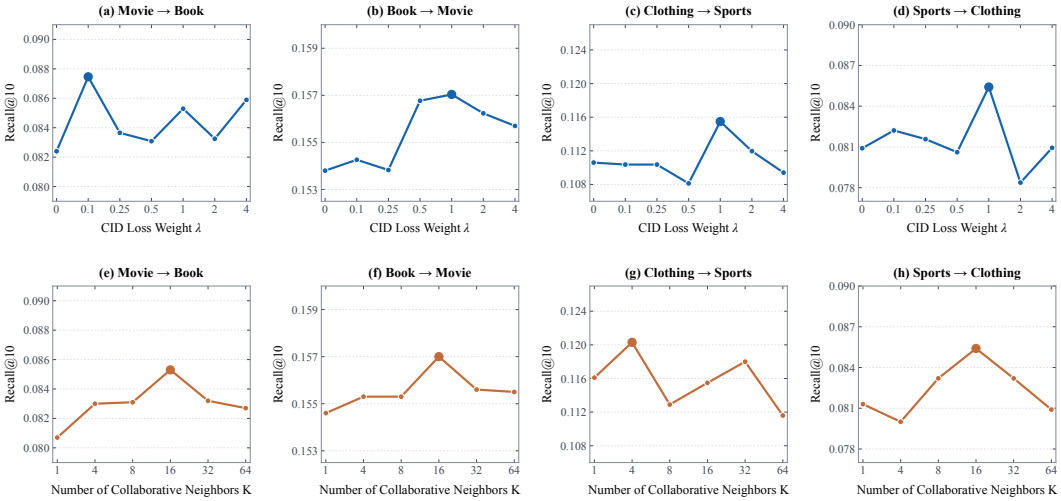


Figure 7: Hyperparameter sensitivity of IntExpo to the CID loss weight $\lambda$ and the number of collaborative neighbors $K$ across four cross-domain settings.

Table 6: Computational cost and inference efficiency of different LLM scales on the Sports dataset.

| Component | LLM Size | Training | | Inference | |
|---|---|---|---|---|---|
| | | Time/Epoch (h) | GPU Memory (GB) | Throughput (tokens/s) | GPU Memory (GB) |
| GRPO | Qwen3.5-0.8B | 6.425 | 186.495 | 552.264 | 46.755 |
| | Qwen3.5-2B | 6.731 | 197.218 | 384.804 | 49.866 |
| | Qwen3.5-4B | 9.285 | 217.191 | 297.499 | 54.362 |
| | Qwen3.5-9B | 10.252 | 317.462 | 269.022 | 66.487 |
| SFT | Qwen3.5-0.8B | 0.046 | 25.899 | 70.213 | 5.279 |
| | Qwen3.5-2B | 0.071 | 40.603 | 61.949 | 7.622 |
| | Qwen3.5-4B | 0.153 | 64.737 | 37.978 | 12.503 |
| | Qwen3.5-9B | 0.224 | 109.864 | 31.589 | 21.833 |

### C.4 Complexity and Inference Efficiency

To examine how LLM size affects the training and inference efficiency of IntExpo's two components, we benchmark the GRPO-based Transferable Preference Profiler and SFT-based Behavior Generator using Qwen3.5 backbones of different sizes. We compare their training time, inference throughput, and GPU memory usage.

As shown in Table 6, increasing the LLM size consistently leads to higher computational costs for both components. For the GRPO-based Transferable Preference Profiler, larger backbones require longer training time and substantially more GPU memory, while inference throughput gradually decreases as the model scale grows. A similar trend is observed for the SFT-based Behavior Generator, where both training and inference become increasingly resource-intensive with larger models. Compared with SFT, GRPO introduces considerably heavier computational overhead, particularly in training, due to the additional optimization and sampling procedures involved in reinforcement learning. Overall, these results highlight a clear trade-off between model scale and efficiency, suggesting that smaller backbones provide a more computationally efficient choice for both training and deployment.