

# CASG: Drive Text to IDs for Sequential Recommendation via Collaboration-Aware Spectral Guidance

Yichen Liu<sup>†</sup> Lianyue Zhang<sup>†</sup> Cong Xu<sup>§</sup> Zhangchi Zhu<sup>†</sup>  
Yanyu Chen<sup>†</sup> Yunhang He<sup>†</sup> Jun Wang<sup>†\*</sup> Wei Zhang<sup>†‡\*</sup>

<sup>†</sup>East China Normal University <sup>§</sup>Anhui Normal University <sup>‡</sup>Shanghai Innovation Institute

<sup>†</sup>{ycliu, 52285901015, zczhu, 51275901058}@stu.ecnu.edu.cn

<sup>†</sup>{yhhe2004, wongjun}@gmail.com <sup>§</sup>cong xu@eric@gmail.com <sup>†‡</sup>zhangwei.thu2011@gmail.com

## Abstract

Text embeddings incorporate additional rich semantic information; however, their performance in sequential recommendation still falls short of that of ID embeddings. Decomposing item embeddings along the spectrum of the collaborative item-item graph reveals a distinct discrepancy in their *spectral energy distribution*. Task-learned ID embeddings concentrate their energy heavily in the low- and mid-frequency directions of the collaborative graph topology, while textual representations spread more uniformly across the entire spectrum. Motivated by this insight, we propose **CASG**, a learnable polynomial graph filter that shifts text embeddings toward ID embeddings via *collaboration-aware spectral guidance*. Specifically, it leverages collaborative relationships to actively guide and filter the spectral energy distribution of frozen text embeddings, selectively preserving the frequency components aligned with the collaborative topology in a single offline pass. Across three datasets and two text encoders, CASG consistently achieves competitive or superior performance against learned ID embeddings, alignment-based, and fusion-based baselines at no online cost, suggesting that what semantic representations need is neither more alignment nor more fusion but *collaboration-aware spectral guidance*. Our source code is available at <https://github.com/blueColdSea/CASG>.

## 1 Introduction

Pre-trained text encoders (Devlin et al., 2019; Reimers and Gurevych, 2019; Gao et al., 2021) map item descriptions into high-dimensional vectors that package rich world knowledge for recommendation tasks (Hou et al., 2022; Li et al., 2023; Hou et al., 2023). In sequential recommendation,

however, a humble, interaction-trained ID embedding table—with a highly compact dimensionality—repeatedly matches or surpasses these knowledge-rich representations, and stronger encoders do not monotonically improve accuracy (Yuan et al., 2023; Hu et al., 2025). This naturally raises a question: **Why does the smaller, semantic-free embedding matrix win?**

To narrow the performance gap, existing solutions primarily focus on how to align or fuse knowledge from these two sources, spanning semantic initialization (Hou et al., 2022), projection-based alignment (Yuan et al., 2023; Sheng et al., 2025), and subspace fusion (Hu et al., 2025; Wang et al., 2026). While differing in execution, these approaches share a critical implicit assumption: that the entire pre-trained semantic space is task-relevant and its full energy should be preserved. None has questioned whether the intrinsic *spectral energy distribution* of these high-dimensional text embeddings aligns with the collaborative recommendation task, or if a significant portion of their unguided energy merely acts as high-frequency noise within the collaborative space.

We probe this question directly by investigating the *spectral energy distribution* of different embeddings along the collaborative item-item co-occurrence graph. The spectral energy at frequency  $\lambda_k$  is defined as  $E(\lambda_k) = \|\mathbf{u}_k^\top \mathbf{X}\|_2^2$ . A stark discrepancy emerges: while the raw BERT embedding exhibits a comparatively flat distribution across the entire spectrum, the learned ID embedding is far more concentrated, allocating the vast majority of its energy to specific frequency bands (Figure 1b). This indicates that the underperformance of pre-trained text encoders in sequential recommendation is not necessarily due to a lack of collaborative information, but rather a suboptimal distribution of

\*Corresponding authors.

their representation energy.

This insight motivates a fundamental shift in methodology: from conventional spatial alignment to **collaboration-aware spectral guidance**. Rather than passively adapting or projecting fixed representation spaces, we propose **CASG**, an offline polynomial graph filter that explicitly leverages collaborative relationships to guide and reshape the spectral energy distribution of text embeddings. By training a filter to dynamically modulate this energy spectrum—matching the concentrated profile of optimal learned IDs—we actively suppress uninformative frequency bands while amplifying task-relevant signals. Notably, introducing this collaboration-aware guidance directly at the spectral energy level ultimately yields larger performance gains than those achieved by traditional spatial alignment or embedding fusion methods, all while requiring no architectural changes to the language model or online computational overhead.

Extensive experiments on three datasets across two text encoders (BERT and stella-en-1.5B-v5 (Zhang et al., 2024a)) demonstrate that CASG consistently outperforms learned ID embeddings, alignment-based baselines, and fusion-based baselines. Controlled ablations further show that collaborative topology provides the primary guidance signal in CASG, while textual semantics offer a complementary and consistent improvement, with particularly clear benefits for interaction-sparse and long-tail items.

Our main contributions are threefold:

- **Diagnosis:** We spectrally decompose item embeddings along the collaborative graph Laplacian, demonstrating that the underperformance of text encoders stems from a fundamental discrepancy in their *spectral energy distribution*—specifically, spreading energy uniformly into high-frequency modes rather than concentrating it in task-relevant low- and mid-frequencies like learned IDs.
- **Methodology:** We introduce CASG, an alternative to alignment/fusion that provides *collaboration-aware spectral guidance*. It leverages collaborative graph relationships to actively guide and filter the spectral energy distribution of text embeddings, selectively preserving topology-aligned components at zero online inference cost.
- **Empirical Insights:** Through extensive eval-

uations and semantic ablations, we isolate the distinct roles of graph structure and text semantics, demonstrating robustness across text encoders and compatibility with standard sequential recommendation pipelines.

## 2 Related Work

**Text-Based Sequential Recommendation.** Sequential recommenders have evolved from optimizing learnable ID embeddings (Kang and McAuley, 2018; Sun et al., 2019; Zhou et al., 2020) to injecting semantic priors from frozen text encoders. Existing approaches typically bridge text and collaborative signals via semantic initialization (Hou et al., 2022; Qu et al., 2024; Li et al., 2023), adaptive projection (Yuan et al., 2023; Sheng et al., 2025; Hebert et al., 2025), or semantic reconstruction and fusion (Hu et al., 2025; Lichtenberg et al., 2025; Xu et al., 2024; Ren et al., 2024). While differing in execution, these methods implicitly assume the entire semantic space should be preserved, overlooking the fundamental mismatch between definitional semantics and user-behavior topology. CASG is complementary: it provides an offline, topology-aligned representation that drops into existing pipelines with zero online overhead.

**Anisotropy in Frozen Text Embeddings.** A persistent challenge with frozen language models is representation anisotropy, where embeddings occupy a narrow cone (Ethayarajh, 2019; Li et al., 2020; Cai et al., 2021; Rogers et al., 2020). Standard NLP solutions address this via contrastive fine-tuning (Reimers and Gurevych, 2019; Gao et al., 2021; Chuang et al., 2022; Jiang et al., 2022) or post-hoc whitening (Mu and Viswanath, 2018; Su et al., 2021). However, both optimize for general semantic uniformity and are agnostic to downstream graph structures. As we demonstrate empirically (§5), standard SVD whitening alone yields suboptimal recommendation performance. We argue that for recommendation tasks, isotropization must be explicitly guided by the collaborative task graph.

**Graph Spectral Filtering.** Polynomial graph filters efficiently approximate spectral convolutions with learnable band-pass shapes (Hammond et al., 2011; Defferrard et al., 2016; Bianchi et al., 2022; He et al., 2021). While recent theoretical work bridges contrastive objectives with spectral clustering on similarity graphs (HaoChen et al., 2021; Tan et al., 2024; Zhu et al., 2021), we are the

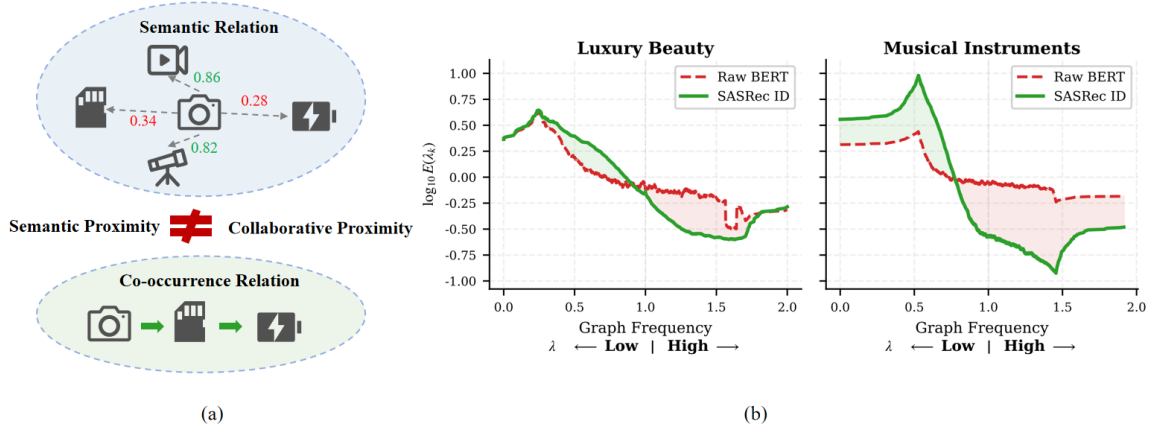


Figure 1: **Motivation of collaboration-aware spectral guidance.** (a) Semantic proximity from text does not always align with collaborative proximity from user interactions, indicating that raw text embeddings may contain task-irrelevant signals. (b) Raw BERT embeddings show a flatter spectral energy distribution, while learned SASRec ID embeddings concentrate energy in low- and mid-frequency bands of the collaborative graph. This mismatch motivates CASG to reshape text embeddings through collaboration-aware spectral guidance.

first to repurpose learnable polynomial filtering for *representation alignment*. Unlike standard graph smoothing (e.g., LightGCN (He et al., 2020)) that operates end-to-end, CASG reshapes frozen text embeddings in the spectral domain of an external co-occurrence graph offline, selectively preserving topology-aligned components without backpropagating through the language model.

### 3 Preliminaries

#### 3.1 Problem Formulation

Let  $\mathcal{U}$  and  $\mathcal{I}$  denote the user and item sets, with  $|\mathcal{I}| = N$ . For each user  $u$ , sequential recommendation aims to predict the next item  $i_{T+1}$  from the interaction history  $H_u = (i_1, \dots, i_T)$ .

Instead of using learnable ID embeddings, each item  $i$  is represented by a textual embedding  $\mathbf{x}_i \in \mathbb{R}^d$ , extracted from its description using a frozen language encoder. All item embeddings form  $\mathbf{X} \in \mathbb{R}^{N \times d}$ .

A sequential encoder  $f_\phi$ , such as SASRec, takes  $\mathbf{X}$  as the item embedding table and encodes  $H_u$  into a user representation  $\mathbf{h}_u = f_\phi(H_u; \mathbf{X})$ . The score for candidate item  $j$  is computed by  $\hat{y}_{u,j} = \mathbf{h}_u^\top \mathbf{x}_j$ .

In CASG,  $\mathbf{X}$  can be replaced by the filtered textual embedding matrix without changing the downstream sequential model.

#### 3.2 Graph Spectral Filtering

Let  $G=(\mathcal{V}, \mathcal{E}, \mathbf{A})$  be an undirected weighted graph on  $N$  nodes with symmetric non-negative adjacency  $\mathbf{A} \in \mathbb{R}_+^{N \times N}$ . The normalized graph Lapla-

cian is

$$\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2} \quad (1)$$

where  $\mathbf{D}$  is the degree matrix of  $\mathbf{A}$ .  $\tilde{\mathbf{L}}$  is positive semi-definite with eigenvalues  $\lambda_i \in [0, 2]$ , and its eigendecomposition  $\tilde{\mathbf{L}} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top$  defines the *graph Fourier basis*  $\mathbf{U}$ . A graph signal  $\mathbf{X} \in \mathbb{R}^{N \times d}$  assigns a  $d$ -vector to each node; equivalently it is  $d$  scalar signals indexed by  $\mathcal{V}$ . A spectral filter is specified by a response function  $g : [0, 2] \rightarrow \mathbb{R}_+$  that re-weights each frequency component:

$$\tilde{\mathbf{X}} = \mathbf{U} g(\mathbf{\Lambda}) \mathbf{U}^\top \mathbf{X}. \quad (2)$$

Low frequencies (small  $\lambda$ ) correspond to signals smooth over  $G$ ; high frequencies (large  $\lambda$ ) capture node-local variation.

Computing Eq. (2) via explicit eigendecomposition costs  $\mathcal{O}(N^3)$ . *Polynomial graph filters* avoid this by parameterizing  $g_\theta(\lambda) = \sum_{k=0}^K \theta_k p_k(\lambda)$ , so the filtered signal evaluates as sparse matrix-matrix products in  $\mathcal{O}(K \cdot \text{nnz}(\tilde{\mathbf{L}}) \cdot d)$  time. Common bases include Chebyshev (Defferrard et al., 2016), ARMA (Bianchi et al., 2022), and Bernstein (He et al., 2021), plus fixed-form kernels such as the heat kernel  $g(\lambda) = e^{-t\lambda}$ ; CASG uses the Bernstein basis, motivated in §4.2.

### 4 Method: CASG

As revealed in Figure 1b, the fundamental bottleneck of text embeddings lies in their uniform spectral energy distribution, which starkly contrasts with the highly concentrated profile of optimal learned IDs. To explicitly bridge this spectral gap

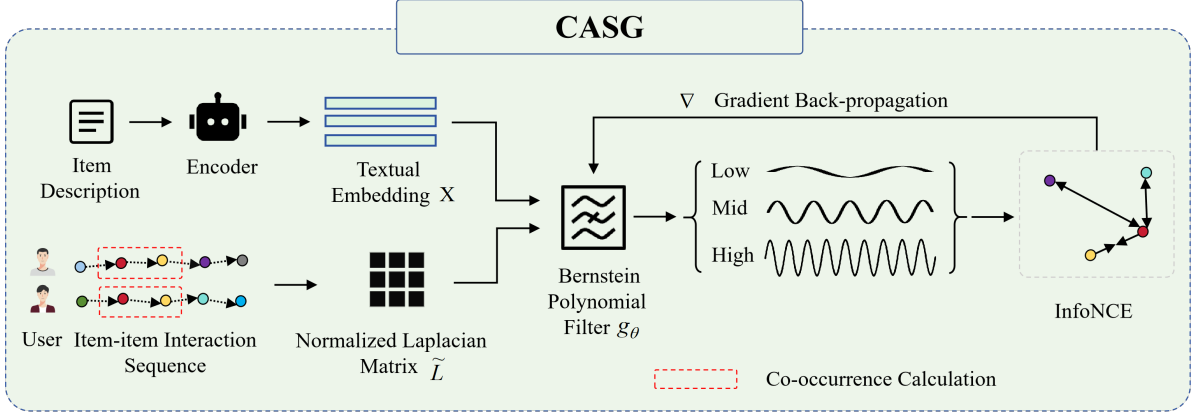


Figure 2: **Overall architecture of CASG.** CASG first constructs an item co-occurrence graph from user interaction sequences and derives the normalized graph Laplacian to encode collaborative topology. Meanwhile, item descriptions are fed into a frozen text encoder to obtain textual embeddings. A learnable Bernstein polynomial graph filter then takes both the textual embeddings and the graph Laplacian as inputs, reshaping the spectral energy of text embeddings across low-, mid-, and high-frequency bands. The filter is optimized with an InfoNCE objective over graph-neighbor item pairs, where gradients only update the filter coefficients. After offline training, the filtered embeddings are used as the item lookup table for sequential recommendation with no additional online inference cost.

and building on the mathematical framework established in §3, this section instantiates the spectral filter with concrete design choices: the item co-occurrence graph (§4.1), the Bernstein basis (§4.2), an InfoNCE training objective on graph neighborhoods (§4.3), and a zero-overhead deployment pipeline.

#### 4.1 Item Co-occurrence Graph

We construct an item–item co-occurrence graph from training user sequences (without leakage from validation or test data). Let  $\mathcal{I}$  be the item set with  $N = |\mathcal{I}|$ . For each pair of items  $(i, j)$ , we count co-occurrences within the same user sequence using a symmetric sliding window of half-width  $w=2$  (i.e., each item co-occurs with up to 4 neighbors), producing a symmetric weighted adjacency matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$ . To control graph density, we retain only the top-20 highest-weight neighbors per item (union of directed keep-sets), then form the symmetric normalized Laplacian  $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$  via Eq. (1). Full construction details and hyperparameter are provided in Appendix F. This graph captures collaborative transition structure—items that appear together in user histories are adjacent in the graph—providing an external structural prior that raw BERT embeddings lack.

#### 4.2 Bernstein Polynomial Filter

We instantiate the polynomial spectral filter of Eq. (2) with the Bernstein basis (He et al., 2021). Given the rescaled operators  $\mathbf{P} = \mathbf{I} - \frac{1}{2}\tilde{\mathbf{L}}$  and  $\mathbf{Q} = \frac{1}{2}\tilde{\mathbf{L}}$ , the  $k$ -th Bernstein basis filter is:

$$\mathbf{B}_k^{(K)}(\tilde{\mathbf{L}}) = \binom{K}{k} \mathbf{P}^k \mathbf{Q}^{K-k}, \quad k = 0, \dots, K. \quad (3)$$

For a graph frequency  $\lambda \in [0, 2]$ , the scalar response is  $b_k^{(K)}(\lambda) = \binom{K}{k} (1 - \lambda/2)^k (\lambda/2)^{K-k}$ . The complete filter is a non-negative linear combination:

$$\tilde{\mathbf{X}}_{\text{graph}} = \sum_{k=0}^K \theta_k \mathbf{B}_k^{(K)}(\tilde{\mathbf{L}}) \mathbf{X}, \quad (4)$$

where  $\theta_k = \text{softplus}(\rho_k) + \epsilon$  are learned coefficients constrained to be positive for numerical stability. Unlike the heat kernel (a fixed monotone low-pass), the Bernstein basis provides  $\mathcal{O}(K)$  degrees of freedom to shape arbitrary band-pass responses. This expressiveness matters when the spectral mismatch is band-dependent rather than a simple low-pass/high-pass imbalance.

#### 4.3 Filter Training

The filter coefficients  $\boldsymbol{\theta} = (\theta_0, \dots, \theta_K)$  are learned via an InfoNCE objective on graph neighborhoods. We sample positive item pairs from the co-occurrence graph: items  $(i, j)$  are positive if

they co-occur within a sliding window of size  $w$  in any user sequence. For a mini-batch of  $B$  anchor-positive pairs  $\{(a_i, p_i)\}_{i=1}^B$ ,

$$\mathcal{L} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(\text{sim}(\tilde{\mathbf{x}}_{a_i}, \tilde{\mathbf{x}}_{p_i})/\tau)}{\sum_{j=1}^B \exp(\text{sim}(\tilde{\mathbf{x}}_{a_i}, \tilde{\mathbf{x}}_{p_j})/\tau)}, \quad (5)$$

where  $\tilde{\mathbf{x}}_i$  is the filtered embedding of item  $i$ ,  $\text{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v} / (\|\mathbf{u}\| \|\mathbf{v}\|)$ , and  $\tau$  is a temperature. We optimize  $\theta$  with Adam; the Bernstein basis matrices  $\mathbf{B}_k^{(K)}(\tilde{\mathbf{L}})\mathbf{X}$  are precomputed via sparse matrix-matrix multiplication and cached, and gradients flow through  $\theta_k$  via Eq. (4).

**Why InfoNCE on graph neighbors?** Eq. (5) is not an arbitrary choice: with positive pairs drawn from the co-occurrence graph, the objective is a contrastive instantiation of Laplacian-based manifold regularization. Prior work shows that InfoNCE-style objectives are theoretically connected to spectral clustering and Laplacian Eigenmaps on the induced similarity graph (HaoChen et al., 2021; Tan et al., 2024; Zhu et al., 2021). Concretely, pulling co-occurring items together encourages graph smoothness, which can be characterized by

$$\begin{aligned} \text{Tr}(\tilde{\mathbf{X}}^\top \tilde{\mathbf{L}} \tilde{\mathbf{X}}) &= \frac{1}{2} \sum_{i,j} A_{ij} \left\| \frac{\tilde{\mathbf{x}}_i}{\sqrt{d_i}} - \frac{\tilde{\mathbf{x}}_j}{\sqrt{d_j}} \right\|_2^2 \\ &= \sum_k \lambda_k \|\hat{\mathbf{x}}_k\|_2^2. \end{aligned} \quad (6)$$

which assigns larger smoothness penalties to energy on high-eigenvalue (high-frequency) modes of  $\tilde{\mathbf{L}}$ , where  $\hat{\mathbf{x}}_k = \mathbf{u}_k^\top \tilde{\mathbf{X}}$  is the graph Fourier coefficient at the  $k$ -th eigenmode. Unlike pure Laplacian smoothing, the InfoNCE negatives prevent representational collapse toward the zero mode, allowing the learned Bernstein filter  $g_\theta$  to preserve task-discriminative bands rather than enforcing global smoothing. Figure 5 confirms this empirically:  $\tilde{\mathbf{X}}$  is realigned toward the learned ID reference across the spectrum, not flattened to a low-pass profile.

**Pipeline and cost.** Once filtered,  $\tilde{\mathbf{X}}$  is frozen and exported; a trainable linear projection feeds the aligned embeddings into a standard SASRec (Kang and McAuley, 2018) trained with binary cross-entropy. Downstream training and inference are identical to standard SASRec.

## 5 Experiments

### 5.1 Setup

**Datasets.** We evaluate our method on three benchmark datasets. Following standard practice, all datasets use leave-one-out evaluation, and co-occurrence graphs are built exclusively from training sequences to avoid information leakage. Detailed dataset statistics are summarized in Appendix A.

- **Amazon<sup>1</sup>:** We adopt two categories, namely *Luxury Beauty* (the densest) and *Musical Instruments* (sparser, featuring many niche items), to reflect different data densities.
- **Yelp<sup>2</sup>:** The largest dataset from business reviews, which is 10-core filtered following standard practice.

**Baselines.** We compare against MoRec (Yuan et al., 2023) (frozen BERT + learnable projection), which serves as the foundational architecture for our proposed CASG, UniSRec (Hou et al., 2022) (text-initialized SASRec, fine-tuned end-to-end), WhitenRec (Zhang et al., 2024b) (SVD-based representation whitening), AlphaFuse (Hu et al., 2025) (SVD-decomposed null-space injection), AlphaSASRec (our SASRec adaptation of AlphaRec (Sheng et al., 2025): same frozen encoder, projector, and LightGCN propagation as AlphaRec but with the CF head replaced by a SASRec backbone on the LightGCN-refined item embeddings). We additionally include three representative graph-based sequential/session-based recommendation models: MAERec (Ye et al., 2023), LESSR (Chen and Wong, 2020), and DMI-GNN (Lv et al., 2025).

**Downstream model, tuning, and compute.** To mitigate the impact of randomness, all main-table results are reported as the mean across three random seeds. Complete standard deviations are reported in Appendix C. We further conduct statistical significance tests using five aligned random seeds, with complete  $p$ -values provided in Appendix C.2. Hyperparameter configurations are reported in Appendix E.

### 5.2 Main Results

Table 1 groups methods by text encoder (BERT vs. stella). Three findings stand out.

<sup>1</sup>[https://cseweb.ucsd.edu/~jmcauley/datasets/amazon\\_v2/](https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/)

<sup>2</sup><https://www.yelp.com/dataset>

| Dataset          |                            | Luxury        |               |               |               | Instrument    |               |               |               | Yelp          |               |               |               |
|------------------|----------------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| Encoder/Type     | Method                     | HR@5          | NDCG@5        | HR@10         | NDCG@10       | HR@5          | NDCG@5        | HR@10         | NDCG@10       | HR@5          | NDCG@5        | HR@10         | NDCG@10       |
| <b>GNN-based</b> | MAERec                     | 0.5430        | 0.4692        | 0.6168        | 0.4912        | 0.3580        | 0.2751        | 0.4553        | 0.3065        | 0.5956        | 0.4184        | 0.7757        | 0.4803        |
|                  | LESSR                      | 0.5803        | 0.5100        | 0.6456        | 0.5300        | 0.4236        | 0.3280        | 0.5393        | 0.3675        | 0.6879        | 0.5104        | 0.8185        | 0.5436        |
|                  | DMI-GNN                    | 0.5715        | 0.5012        | 0.6424        | 0.5239        | 0.4270        | 0.3297        | 0.5357        | 0.3642        | 0.6878        | 0.5138        | 0.8373        | 0.5521        |
| <b>Reference</b> | Learned ID@64 <sup>†</sup> | 0.5890        | 0.5143        | 0.6590        | 0.5371        | <u>0.4405</u> | <u>0.3370</u> | <u>0.5610</u> | <u>0.3758</u> | 0.6822        | 0.4955        | 0.8341        | 0.5450        |
| <b>BERT</b>      | MoRec                      | 0.5786        | 0.4996        | 0.6586        | 0.5250        | 0.3630        | 0.2618        | 0.4947        | 0.3042        | 0.6577        | 0.4658        | 0.8263        | 0.5207        |
|                  | UniSRec                    | <u>0.6424</u> | <u>0.5586</u> | <u>0.7094</u> | <u>0.5799</u> | 0.3687        | 0.2784        | 0.4865        | 0.3164        | 0.6279        | 0.4431        | 0.8004        | 0.4990        |
|                  | WhitenRec                  | 0.5831        | 0.5086        | 0.6539        | 0.5312        | 0.3602        | 0.2646        | 0.4869        | 0.3054        | 0.6712        | 0.4792        | 0.8356        | 0.5327        |
|                  | AlphaFuse                  | 0.5557        | 0.4854        | 0.6348        | 0.5108        | 0.4318        | 0.3223        | 0.5580        | 0.3631        | 0.6470        | 0.4605        | 0.8165        | 0.5155        |
|                  | AlphaSASRec                | 0.6052        | 0.5174        | 0.6893        | 0.5446        | 0.4331        | 0.3207        | 0.5586        | 0.3612        | <u>0.7057</u> | <u>0.5155</u> | <u>0.8571</u> | <u>0.5649</u> |
|                  | CASG (ours)                | <b>0.6574</b> | <b>0.5663</b> | <b>0.7366</b> | <b>0.5920</b> | <b>0.4462</b> | <b>0.3396</b> | <b>0.5716</b> | <b>0.3800</b> | <b>0.7123</b> | <b>0.5202</b> | <b>0.8622</b> | <b>0.5690</b> |
| <b>stella</b>    | MoRec                      | 0.6517        | 0.5698        | 0.7213        | 0.5922        | 0.4322        | 0.3219        | 0.5668        | 0.3653        | 0.7027        | 0.5103        | 0.8555        | 0.5601        |
|                  | UniSRec                    | <b>0.6867</b> | <u>0.5945</u> | <u>0.7494</u> | <u>0.6147</u> | 0.4089        | 0.3099        | 0.5336        | 0.3501        | 0.6628        | 0.4794        | 0.8231        | 0.5315        |
|                  | WhitenRec                  | 0.6175        | 0.5415        | 0.6847        | 0.5631        | 0.4155        | 0.3089        | 0.5444        | 0.3505        | 0.7030        | 0.5093        | 0.8551        | 0.5588        |
|                  | AlphaFuse                  | 0.6136        | 0.5361        | 0.6907        | 0.5610        | 0.4612        | <u>0.3554</u> | 0.5838        | 0.3899        | 0.6835        | 0.4906        | 0.8426        | 0.5423        |
|                  | AlphaSASRec                | 0.6547        | 0.5650        | 0.7307        | 0.5896        | <u>0.4639</u> | <u>0.3519</u> | <u>0.5866</u> | <u>0.3915</u> | <u>0.7401</u> | <u>0.5514</u> | <u>0.8760</u> | <u>0.5957</u> |
|                  | CASG (ours)                | <u>0.6809</u> | <b>0.5947</b> | <b>0.7525</b> | <b>0.6179</b> | <b>0.4692</b> | <b>0.3585</b> | <b>0.5961</b> | <b>0.3996</b> | <b>0.7425</b> | <b>0.5535</b> | <b>0.8765</b> | <b>0.5972</b> |

Table 1: **Main results (SASRec backbone)**. Boldface: best; underline: second-best within each encoder setting. Learned ID@64 is included as a common encoder-independent reference in both comparisons. The GNN-based methods are included as additional encoder-independent references. Learned ID@64 (<sup>†</sup>) corresponds to the traditional SASRec model, which uses learnable ID embeddings without text encoders and serves as a reference for both text-encoder groups.

First, **frozen text encoders (MoRec) struggle to match learned ID embeddings on sparse data, even with high semantic quality**. Under BERT, MoRec trails ID@64 across all three datasets. Upgrading to stella narrows the gap considerably, allowing MoRec to surpass ID@64 on the two denser datasets (Luxury and Yelp). However, on the sparsest dataset (Instrument), stella+MoRec still lags behind learned IDs. This confirms that while semantic quality is crucial, it cannot fully compensate for the lack of structural alignment when collaborative signals are scarce.

Second, **CASG consistently improves upon MoRec under both encoders, and stronger semantics coupled with spectral guidance yields the best mean performance among the compared methods**. The improvement holds in every evaluated (dataset, encoder) pair. Crucially, unlike earlier assumptions where simpler models might win on sparse graphs, these results demonstrate that stella+CASG achieves the best mean performance across *all* three datasets, including Instrument. These results suggest that semantic quality and structural guidance provide complementary benefits: providing a high-capacity encoder (stella) with collaboration-aware spectral guidance (CASG) consistently yields strong sequential recommendation performance. A controlled graph-sparsification experiment further shows that the effectiveness of such spectral guidance depends on

the quality of the available collaborative topology (Appendix B).

Third, **SVD-based baselines show inconsistent and often detrimental behavior**. WhitenRec yields only a marginal lift over MoRec under BERT, but actively degrades performance under the higher-capacity stella encoder across all datasets. AlphaFuse, which injects trainable IDs into the whitened null space, similarly struggles: it collapses below MoRec on the denser Luxury and Yelp datasets under both encoders. While AlphaFuse shows some recovery on the sparse Instrument dataset, it still falls short of CASG. This suggests that partitioning the space via hard SVD splits is a suboptimal and unstable strategy, whereas CASG’s approach of operating on the full embedding space safely avoids these failure modes. We further evaluate CASG with GRU4Rec (Hidasi et al., 2015), an RNN-based sequential recommender, and observe the same improvement trend across all three datasets; detailed results are provided in Appendix D.

### 5.3 Ablation: What Drives the Gain?

Table 2 evaluates the key components of CASG via an ablation study. First, standard graph embedding methods (e.g., DeepWalk, Node2Vec (Grover and Leskovec, 2016)) fail to match CASG, demonstrating that static or random-walk-based representations cannot substitute for our learned spectral coefficients. Second, component-wise analysis shows

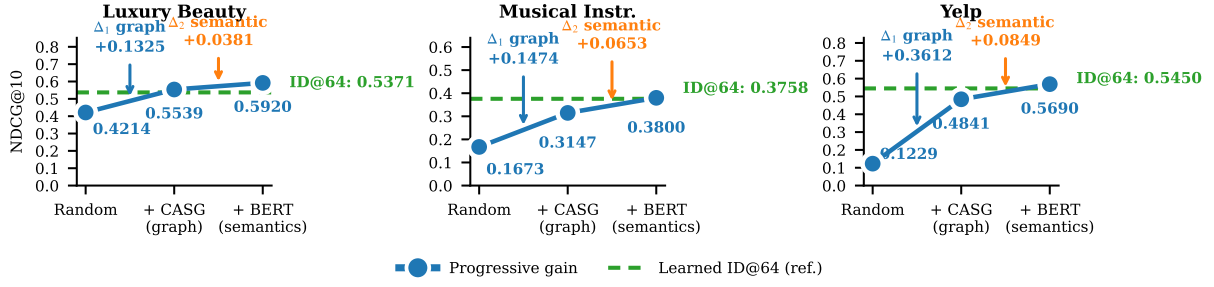


Figure 3: **Decomposition of CASG’s NDCG@10 gain.** Progressive ablation across three stages per dataset: random vectors + linear projection, then + CASG (graph structure,  $\Delta_1$ ), then + BERT (semantics,  $\Delta_2$ ). Dashed green line: Learned ID@64 reference. Along this graph-first insertion path, graph guidance yields the larger first-step gain, while BERT semantics provide a further consistent improvement.

| Representation                   | Luxury        |               | Instrument    |               |
|----------------------------------|---------------|---------------|---------------|---------------|
|                                  | HR@10         | NDCG@10       | HR@10         | NDCG@10       |
| <i>Graph-only baselines</i>      |               |               |               |               |
| Spectral SVD (Chen et al., 2021) | 0.4184        | 0.2943        | 0.2828        | 0.1701        |
| DeepWalk (Perozzi et al., 2014)  | 0.4399        | 0.3537        | 0.2713        | 0.1723        |
| Node2Vec                         | 0.4427        | 0.3609        | 0.2663        | 0.1659        |
| LINE (Tang et al., 2015)         | 0.4273        | 0.3084        | 0.2842        | 0.1778        |
| <i>Component ablation</i>        |               |               |               |               |
| Full CASG                        | <b>0.7366</b> | <b>0.5920</b> | <b>0.5716</b> | <b>0.3800</b> |
| w/o Semantics                    | 0.6793        | 0.5539        | 0.4583        | 0.3147        |
| w/o Semantics & Graph Filter     | 0.5121        | 0.4214        | 0.2582        | 0.1673        |
| <i>Reference</i>                 |               |               |               |               |
| Learned ID@64                    | 0.6590        | 0.5371        | 0.5610        | 0.3758        |

Table 2: **Representation source ablation.** “w/o Semantics”  $\rightarrow \mathcal{N}(0, 1)$  vectors; “w/o Semantics & Graph Filter”  $\rightarrow$  random vectors + no filter. The top block confirms standard graph embedding methods on the same I-I graph cannot substitute for CASG’s learned spectral coefficients.

that removing semantic features (“w/o Semantics”) leads to a clear performance drop, and further omitting the graph filter (“w/o Semantics & Graph Filter”) severely degrades the performance, dropping well below the Learned ID@64 reference. We therefore interpret Figure 3 as showing that collaborative topology is the primary guidance mechanism of CASG along this graph-first insertion path, while the relative marginal contributions of graph and semantics depend on the insertion order.

#### 5.4 Method Robustness

CASG is not sensitive to hyperparameter choices. Table 3 reports NDCG@10 as a function of filter order  $K$  ( $\alpha=0$ ). Across all four (encoder, dataset) pairs, NDCG@10 varies by at most 0.013 within  $K \in [2, 6]$ . We additionally examined an optional MSE anchor regularization term  $\mathcal{L}_{\text{anchor}} = \frac{1}{Nd} \sum_i \|\tilde{\mathbf{x}}_i - \mathbf{x}_i^{\text{raw}}\|_2^2$  added to Eq. (5) with weight  $\alpha$ , and observed that performance is largely insensi-

| $K$       | Luxury |        | Instrument |        |
|-----------|--------|--------|------------|--------|
|           | BERT   | stella | BERT       | stella |
| 2         | 0.5856 | 0.6181 | 0.3738     | 0.3985 |
| 3         | 0.5878 | 0.6147 | 0.3768     | 0.3967 |
| 4         | 0.5920 | 0.6179 | 0.3800     | 0.3971 |
| 5         | 0.5896 | 0.6054 | 0.3687     | 0.3899 |
| 6         | 0.5895 | 0.6081 | 0.3732     | 0.3996 |
| Max – min | 0.0064 | 0.0127 | 0.0113     | 0.0097 |

Table 3: **Robustness to filter order  $K$  (NDCG@10,  $\alpha=0$ ).** NDCG@10 varies by at most 0.013 across  $K \in [2, 6]$  for all four (encoder, dataset) pairs confirming that CASG is not sensitive to the filter order.

tive to  $\alpha$  across different  $K$ , with  $\alpha=0$  (no anchor) performing competitively, suggesting that the contrastive objective alone is sufficient for downstream recommendation. Full sensitivity tables, plus filter-training convergence checks and SASRec hyperparameter scans, are reported in Appendix E (Table 9).

## 6 Analysis and Discussion

### 6.1 Embedding Space Before and After Alignment

Figure 4 confirms that CASG effectively eliminates the raw BERT anisotropy, shifting the mean cosine similarity from roughly 0.86 to 0. Visually, the global pairwise similarity distribution of CASG mirrors that of PCA-whitening (evaluated as the WhitenRec baseline via SVD/PCA). However, despite this macroscopic resemblance, CASG outperforms WhitenRec downstream (Table 1). The underlying difference lies in their microscopic structures: while PCA-whitening applies a global linear transformation, CASG leverages graph topology to actively restructure the local semantic space, retaining only 13%–20% of the original nearest neigh-

bors.

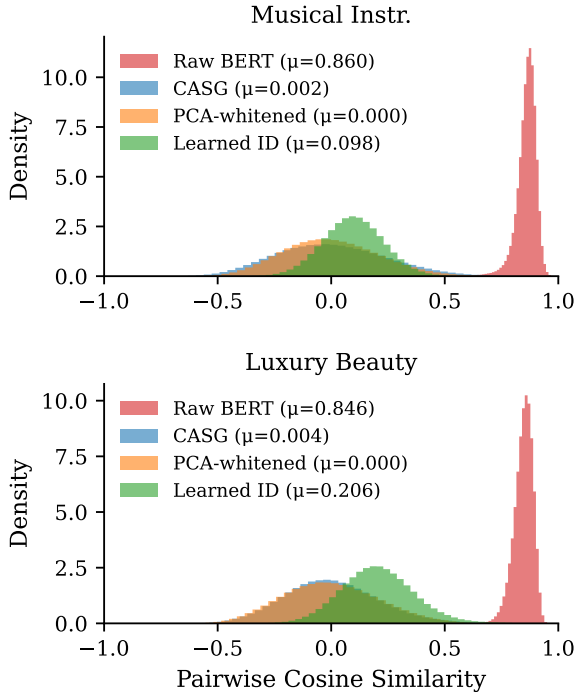


Figure 4: Cosine similarity before and after alignment. Raw BERT collapses (mean  $\approx 0.86$ ); CASG and PCA-whitening (corresponding to the WhitenRec baseline) both decorrelate (mean  $\approx 0$ ); task-learned ID falls in between (mean  $\approx 0.10$ – $0.21$ ).

Figure 5 then visually confirms this structural shift: while raw BERT exhibits a comparatively flatter spectral energy distribution across the spectrum, CASG successfully reshapes it to closely track the learned ID reference. Crucially, CASG preserves and amplifies the vital mid-frequency energy peaks, demonstrating the effectiveness of *collaboration-aware spectral guidance* in capturing collaboration signals rather than passively spreading energy.

## 6.2 Who Benefits Most? Long-Tail and Interaction-Sparse Analysis

If CASG *restructures* rather than *rotates* (§6), the strongest empirical signature should appear where collaborative signal is most needed: items with limited interaction history. We test this by partitioning items into three popularity buckets based on training interaction frequency: head (top 20%), mid (middle 60%), and tail (bottom 20%). Table 4 reports per-bucket NDCG@10 for three representations: learned ID embeddings, raw BERT, and CASG.

Three patterns emerge from Table 4. Learned ID embeddings exhibit severe tail collapse across

| Dataset | Method   | Head          | Mid           | Tail          | T/H                |
|---------|----------|---------------|---------------|---------------|--------------------|
| Luxury  | ID@64    | <b>0.8064</b> | 0.3762        | 0.1371        | 17.0%              |
|         | Raw BERT | 0.7163        | 0.2108        | 0.0728        | 10.2%              |
|         | CASG     | 0.8039        | <b>0.4484</b> | <b>0.2933</b> | <b>36.5%</b>       |
| Instr.  | ID@64    | <b>0.5538</b> | 0.1450        | 0.0456        | 8.2%               |
|         | Raw BERT | 0.0540        | 0.0441        | 0.0507        | 93.9% <sup>†</sup> |
|         | CASG     | 0.4929        | <b>0.2668</b> | <b>0.1956</b> | <b>39.7%</b>       |
| Yelp    | ID@64    | <b>0.6760</b> | 0.4745        | 0.3274        | 48.4%              |
|         | Raw BERT | 0.6416        | 0.4498        | 0.3521        | 54.9%              |
|         | CASG     | 0.6658        | <b>0.4951</b> | <b>0.4307</b> | <b>64.7%</b>       |

Table 4: **Per-bucket NDCG@10 by item popularity** (head/mid/tail = top 20% / middle 60% / bottom 20% by training frequency). CASG maintains competitive head performance while consistently improving the tail over ID@64: +114% on Luxury, and +329% on Instrument, +31.5% on Yelp. T/H = tail/head ratio. <sup>†</sup>On Instrument, Raw BERT is uniformly poor; the inflated ratio reflects a degraded head.

all three datasets, since tail items lack the per-item interaction signal IDs need. Raw BERT does not fix this—on Luxury it trails ID on every bucket, and on Instrument it is uniformly poor. CASG, in contrast, remains competitive with ID on the head while substantially lifting the tail; the mid bucket benefits as well. These results close the loop on our central claim: textual embeddings lack alignment, not information. Topology realignment transforms BERT from worse than ID on tail items to substantially better, without any change to the text encoder.

**Practical implications.** The gain pattern in Table 4 concentrates where it matters most for deployment: mid-frequency and tail items, where learned IDs lack the per-item interaction signal needed to converge. These are precisely the regimes that production systems must cover—long-tail catalogs in retrieval and early-ranking stages, and items whose interaction history is too sparse for learned IDs to discriminate end-to-end. CASG targets this gap as an offline, encoder-agnostic preprocessing step that integrates into existing pipelines without modifying the downstream architecture or introducing additional online inference overhead.

## 7 Conclusion

This work investigates the question of why compact ID embedding matrices repeatedly surpass semantic-rich pre-trained text encoders in sequential recommendation. We show that the underperformance of text encoders is caused by a sub-optimal distribution of their representation en-

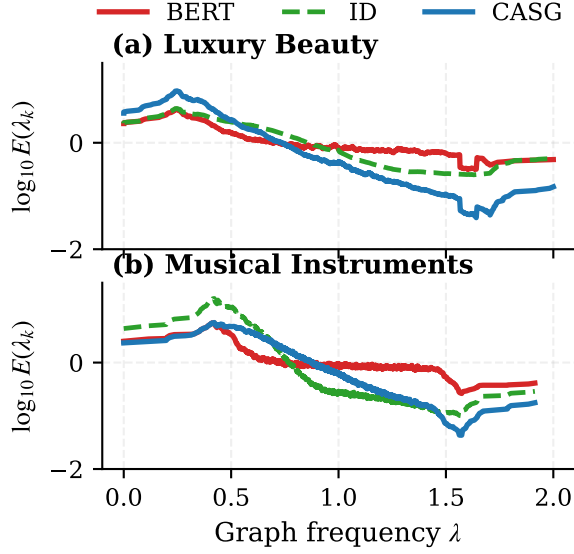


Figure 5: **Graph Fourier Transform (GFT) spectral analysis.** Log-energy per graph frequency on Luxury Beauty (a) and Musical Instruments (b). Raw BERT exhibits a flatter spectral profile, while learned ID embeddings concentrate energy more strongly in task-relevant frequency bands. CASG reshapes the BERT spectrum toward the learned ID reference.

ergy, which spreads uniformly into high-frequency modes rather than concentrating heavily in task-relevant low- and mid-frequencies like learned IDs. To bridge this gap, we present CASG, an offline polynomial graph filter that utilizes **collaboration-aware spectral guidance** to successfully **drive text to IDs**. By reshaping the spectral energy distribution, CASG suppresses uninformative frequency bands and amplifies task-relevant signals, consistently outperforming traditional projection and fusion baselines with zero online computational overhead. Finally, our evaluations confirm that while the graph structure serves as the primary engine for spectral shaping, text semantics offer an additive contribution that directly benefits interaction-sparse and long-tail items.

## Limitations

This work has several limitations that we document transparently. (1) **Limited LM coverage:** Experiments cover BERT-base-uncased and stellan-1.5B-v5. Other architectures (RoBERTa, T5, decoder-only LLMs) may encode item text differently and shift the degree of anisotropy-collapse and the benefit of realignment. (2) **Single task family:** Evaluation is limited to sequential recommendation. The core claim—that textual embeddings, used as IDs, lack alignment rather than

information—should hold for any structured task where items have both text and relational structure, but this generalization is not yet empirically tested. (3) **Unseen (“true cold”) items:** CASG requires an item to appear in the training co-occurrence graph in order to receive a topology-aligned embedding. New items added after training (e.g., newly released products) carry only raw text features and cannot benefit from offline graph filtering. Our long-tail analysis (§6.2) covers tail items present in training but not post-deployment items absent from the graph. A practical mitigation is to graft new items onto their nearest graph neighbors (e.g., k-NN to graph-filtered embeddings of training items with similar text), incurring no additional online cost; we leave systematic evaluation of this strategy to future work. (4) **Backbone scope:** Our main downstream evaluation uses SASRec as the representative sequential recommender. We additionally validate CASG with the RNN-based GRU4Rec in Appendix D, where the same improvement trend is observed. Nevertheless, broader evaluation across CNN-based, graph-based, and hybrid architectures remains future work. (5) **Inherited behavioral and semantic biases:** CASG learns its structural guidance from historical interaction topology and its semantic information from pretrained text encoders. Consequently, popularity or exposure biases in behavioral data, as well as social or cultural biases encoded by the language model, may be preserved or amplified after alignment. Our long-tail analysis evaluates performance disparity across popularity groups, but does not constitute a complete evaluation of item exposure or provider-level fairness.

## Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant (No. 62572198) and Key Laboratory of Advanced Theory and Application in Statistics and Data Science, Ministry of Education. Dr. Jun Wang is sponsored by CAAI-Ant Group Research Fund.

## Ethics Statement

This work uses publicly available benchmark datasets (Amazon Reviews and Yelp) and does not introduce additional user attributes, online tracking, or data collection beyond standard sequential recommendation pipelines. Nevertheless, CASG relies on historical interaction sequences and an

item co-occurrence graph constructed from them, and may therefore inherit existing popularity and exposure biases. In particular, frequently interacted items may receive stronger structural support than long-tail items or items from smaller providers. Our long-tail results show that CASG substantially improves tail-item ranking accuracy while approximately maintaining head-item performance, suggesting a reduced head–tail performance disparity. However, ranking accuracy does not directly imply fair exposure, catalog coverage, or provider-level fairness; evaluating these dimensions explicitly is an important direction for future work.

CASG also inherits the privacy risks of sequential recommendation because user interaction histories are used to construct aggregated co-occurrence statistics and train the downstream sequential model. Aggregation alone does not eliminate such risks. Practical deployments should therefore consider safeguards such as anonymization, minimum-count thresholds, data-retention controls, and privacy-preserving aggregation.

Finally, CASG does not remove social or cultural biases encoded in pretrained text representations such as BERT or stella, nor biases reflected in historical user behavior. Because spectral filtering aligns textual representations with collaborative interaction topology, biases from either source may be retained or amplified. Combining CASG with embedding-debiasing and fairness-aware recommendation techniques is therefore an important avenue for future work.

## References

- Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. 2022. Graph neural networks with convolutional ARMA filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3496–3510.
- Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the contextual embedding space: Clusters and manifolds. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1172–1180.
- Yuxin Chen, Yuejie Chi, Jianqing Fan, and Cong Ma. 2021. Spectral methods for data science: A statistical perspective. *Foundations and Trends in Machine Learning*, 14(5):566–806.
- Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljacic, Shang-Wen Li, Wen tau Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based contrastive learning for sentence embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4207–4218. Association for Computational Linguistics.
- Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3844–3852.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186. Association for Computational Linguistics.
- Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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)*, pages 55–65. Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910. Association for Computational Linguistics.
- Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 855–864. ACM.
- David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. 2011. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150.
- Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. 2021. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in neural information processing systems*, 34:5000–5011.
- Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. 2021. BernNet: Learning arbitrary graph spectral filters via Bernstein approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 14239–14251.

- Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 639–648.
- Liam Hebert, Marialena Kyriakidi, Hubert Pham, Krishna Sayana, James Pine, Sukhdeep Sodhi, and Ambarish Jash. 2025. Flare: Fusing language models and collaborative architectures for recommender enhancement. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 2235–2244.
- Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. [Session-based recommendations with recurrent neural networks](#). *CoRR*, abs/1511.06939.
- Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2023. [Learning vector-quantized item representation for transferable sequential recommenders](#). In *Proceedings of the ACM Web Conference (WWW)*, pages 1260–1271. ACM.
- Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. [Towards universal sequence representation learning for recommender systems](#). In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 585–593. ACM.
- Guoqing Hu, An Zhang, Shuo Liu, Zhibo Cai, Xun Yang, and Xiang Wang. 2025. Alphafuse: Learn id embeddings for sequential recommendation in null space of language embeddings. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1614–1623.
- Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. [PromptBERT: Improving BERT sentence embeddings with prompts](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8826–8837. Association for Computational Linguistics.
- Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining (ICDM)*, pages 197–206.
- Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. [On the sentence embeddings from pre-trained language models](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130. Association for Computational Linguistics.
- Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. [Text is all you need: Learning language representations for sequential recommendation](#). In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1258–1267. ACM.
- Jan Malte Lichtenberg, Antonio De Candia, and Matteo Ruffini. 2025. [DenseRec: Revisiting dense content embeddings for sequential transformer-based recommendation](#). In *RecSys 2025 Workshop on Embedding and Representation Learning (EARL)*.
- Mingyang Lv, Xiangfeng Liu, and Yuanbo Xu. 2025. [Dynamic multi-interest graph neural network for session-based recommendation](#). In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI’25/IAAI’25/EAAI’25*. AAAI Press.
- Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. [DeepWalk: Online learning of social representations](#). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 701–710. ACM.
- Zekai Qu, Ruobing Xie, Chaojun Xiao, Yuan Yao, Zhiyuan Liu, Fengzong Lian, Zhanhui Kang, and Jie Zhou. 2024. [Thoroughly modeling multi-domain pre-trained recommendation as language](#). *ACM Transactions on Information Systems*.
- Nils Reimers and Iryna Gurevych. 2019. [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)*, pages 3982–3992. Association for Computational Linguistics.
- Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. [Representation learning with large language models for recommendation](#). In *Proceedings of the ACM Web Conference (WWW)*. ACM.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics (TACL)*, 8:842–866.
- Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua. 2025. Language representations can be what recommenders need: Findings and potentials. In *international Conference on Learning Representations*, volume 2025, pages 91632–91658.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. **BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer**. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1441–1450. ACM.

Zhiqian Tan, Yifan Zhang, Jingqin Yang, and Yang Yuan. 2024. Contrastive learning is spectral clustering on similarity graph. In *International Conference on Learning Representations*, volume 2024, pages 19019–19035.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. **LINE: Large-scale information network embedding**. In *Proceedings of the 24th International Conference on World Wide Web (WWW)*, pages 1067–1077. ACM.

Maolin Wang, Dongze Wu, Jianing Zhou, Hongyu Chen, Beining Bao, Yu Jiang, Chenbin Zhang, Chang Wang, Jian Liu, and Lei Sha. 2026. Rethinking semantic collaborative integration: Why alignment is not enough. *arXiv preprint arXiv:2604.22195*.

Lanling Xu, Zhen Tian, Bingqian Li, Junjie Zhang, Daoyuan Wang, Hongyu Wang, Jinpeng Wang, Sheng Chen, and Wayne Xin Zhao. 2024. **Sequence-level semantic representation fusion for recommender systems**. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*. ACM.

Yaowen Ye, Lianghao Xia, and Chao Huang. 2023. **Graph masked autoencoder for sequential recommendation**. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. **Where to go next for recommender systems? id- vs. modality-based recommender models revisited**. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 2639–2649. ACM.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fu-Yun Wang. 2024a. **Jasper and stella: distillation of sota embedding models**. *ArXiv*, abs/2412.19048.

Lingzi Zhang, Xin Zhou, Zhiwei Zeng, and Zhiqi Shen. 2024b. **Are id embeddings necessary? whitening pre-trained text embeddings for effective sequential recommendation**. *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 530–543.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and

Ji-Rong Wen. 2020. **S<sup>3</sup>-Rec: Self-supervised learning for sequential recommendation with mutual information maximization**. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1893–1902. ACM.

Hao Zhu, Ke Sun, and Peter Koniusz. 2021. Contrastive laplacian eigenmaps. *Advances in neural information processing systems*, 34:5682–5695.

## A Dataset Statistics

The details of our evaluation datasets are reported as follows.

| Dataset             | Users  | Items  | Interactions | Train   | Density |
|---------------------|--------|--------|--------------|---------|---------|
| Luxury Beauty       | 9,086  | 2,606  | 57,931       | 39,759  | 0.245%  |
| Musical Instruments | 27,403 | 10,449 | 230,358      | 175,552 | 0.080%  |
| Yelp                | 41,801 | 26,512 | 1,022,604    | 939,002 | 0.092%  |

Table 5: **Dataset statistics.** Density = Interactions/(Users×Items).

## B Effect of Collaborative Graph Sparsity

The results on Musical Instruments suggest that the effectiveness of CASG depends not only on the quality of the textual representation, but also on the amount of reliable collaborative topology available for learning the graph filter. Musical Instruments is substantially sparser than Luxury Beauty (interaction density: 0.080% vs. 0.245%; see Table 5), and therefore provides a weaker collaborative signal for spectral guidance.

To isolate the effect of graph quality from other dataset-specific factors, we conduct a controlled sparsification experiment on the denser Luxury Beauty dataset. Starting from its original item–item co-occurrence graph, we randomly retain an increasing fraction of graph edges while keeping all other components, including the BERT features, filter architecture, training objective, and downstream recommender, unchanged. We evaluate both (i) the cosine similarity between the resulting spectral energy distribution and that of Learned ID@64, and (ii) the NDCG@10 improvement of CASG over raw BERT.

As shown in Table 6, progressively weakening the collaborative graph causes the spectral profile obtained by CASG to continuously diverge from the Learned ID@64 reference: the cosine similarity decreases from 0.981 with the full graph to 0.508 when only 20% of the edges are retained. At the same time, the NDCG@10 improvement over raw

| Edge Keep Ratio | Cosine Sim.<br>to ID@64 | $\Delta$ NDCG@10<br>vs. Raw BERT |
|-----------------|-------------------------|----------------------------------|
| 1.0             | 0.981                   | +0.0670                          |
| 0.7             | 0.926                   | +0.0564                          |
| 0.5             | 0.833                   | +0.0492                          |
| 0.3             | 0.658                   | +0.0383                          |
| 0.2             | 0.508                   | +0.0383                          |

Table 6: Effect of graph sparsification on spectral alignment and recommendation performance on Luxury Beauty with BERT. As the collaborative graph becomes sparser, CASG becomes less effective at reshaping the textual representation toward the learned-ID spectral profile, accompanied by a reduction in downstream recommendation gain.

BERT decreases from +0.0670 to +0.0383, corresponding to a reduction of approximately 43%.

This controlled experiment demonstrates that reliable collaborative topology is an important condition for effective graph-guided spectral reshaping. It also helps explain the smaller BERT-based improvement observed on Musical Instruments: compared with Luxury Beauty, its substantially lower interaction density provides less collaborative evidence from which CASG can learn the spectral filter. Importantly, this result does not imply that CASG universally eliminates the gap between textual and ID representations. Rather, CASG consistently improves frozen textual representations by exploiting the collaborative topology available in the interaction graph, while stronger text encoders and stronger collaborative graphs provide complementary sources of information.

## C Full Results and Statistical Analysis

### C.1 Full Results with Standard Deviations

Table 11 reports the complete results corresponding to the main experiments. Each result is reported as mean  $\pm$  standard deviation over the same three random seeds used to obtain Table 1.

### C.2 Statistical Significance Tests

For a more reliable significance analysis, we additionally run two random seeds for CASG and the strongest competing method in each setting, resulting in five aligned seeds in total. We conduct a two-sided paired  $t$ -test between CASG and the strongest competing method for each dataset–encoder–metric combination. Table 7 reports the complete  $p$ -values.

| Encoder | Dataset    | HR@5     | NDCG@5   | HR@10    | NDCG@10  |
|---------|------------|----------|----------|----------|----------|
| BERT    | Luxury     | 0.003*   | 0.051    | < 0.001* | 0.003*   |
|         | Instrument | 0.042*   | 0.159    | < 0.001* | 0.016*   |
|         | Yelp       | < 0.001* | < 0.001* | < 0.001* | < 0.001* |
| stella  | Luxury     | —        | 0.269    | 0.009*   | 0.017*   |
|         | Instrument | 0.736    | 0.018*   | 0.020*   | 0.008*   |
|         | Yelp       | 0.112    | 0.011*   | 0.095    | 0.010*   |

Table 7: **Statistical significance tests.**  $p$ -values from two-sided paired  $t$ -tests between CASG and the strongest non-CASG baseline for each metric and (encoder, dataset) setting, using five aligned random seeds. \* indicates  $p < 0.05$ . — indicates no mean improvement; significance is not reported.

## D Evaluation with GRU4Rec

We additionally evaluate CASG with GRU4Rec, an RNN-based sequential recommender, to examine whether the observed gains are specific to the Transformer-based SASRec backbone used in our main experiments. As shown in Table 8, CASG consistently improves both BERT and stella representations across all three datasets.

## E Training Hyperparameters

This appendix reports sensitivity analyses referenced from §5.4.

**Filter training convergence.** We trained the Bernstein filter on Luxury (stella,  $K=2$ ) for up to 300 epochs across two learning rates ( $1e-2$ ,  $5e-3$ ) and two InfoNCE temperatures (0.07, 0.10). The contrastive loss decreased from 4.67 at epoch 100 to 4.54 at epoch 300, yet the resulting downstream embeddings are nearly identical (mean cosine similarity  $> 0.999$  across all 300+ epoch checkpoints). Downstream SASRec NDCG@10 lies within  $\pm 0.001$  across all converged checkpoints.

**SASRec hyperparameters.** On Luxury with stella+Bern ( $K=2$ ), a  $2 \times 2$  grid over learning rate  $\{5e-4, 1e-3\}$  and dropout  $\{0.3, 0.5\}$ . We report the default configuration in Table 1 for a fair comparison with baselines, which use the same SASRec hyperparameters.

**AlphaSASRec adaptation and tuning.** AlphaRec (Sheng et al., 2025) was originally designed for collaborative filtering and does not contain a sequential encoder, making direct evaluation in the next-item prediction setting infeasible. We therefore construct **AlphaSASRec** by preserving AlphaRec’s representation-learning components—the frozen text encoder, learnable projection layer,

Table 8: Performance with GRU4Rec as the downstream sequential recommender. CASG consistently improves both BERT and stella representations across all three datasets.

| Condition     | Luxury Beauty |               | Musical Instruments |               | Yelp          |               |
|---------------|---------------|---------------|---------------------|---------------|---------------|---------------|
|               | HR@10         | NDCG@10       | HR@10               | NDCG@10       | HR@10         | NDCG@10       |
| Pure ID       | 0.6099        | 0.4692        | 0.4822              | 0.3043        | 0.8085        | 0.5135        |
| BERT raw      | 0.6271        | 0.4853        | 0.4790              | 0.2902        | 0.8248        | 0.5205        |
| BERT + CASG   | <b>0.6471</b> | <b>0.5167</b> | <b>0.5476</b>       | <b>0.3552</b> | <b>0.8520</b> | <b>0.5625</b> |
| stella raw    | 0.6884        | 0.5466        | 0.5438              | 0.3376        | 0.8407        | 0.5386        |
| stella + CASG | <b>0.7341</b> | <b>0.5713</b> | <b>0.5832</b>       | <b>0.3827</b> | <b>0.8679</b> | <b>0.5852</b> |

| (a) Filter order $K$ ( $\alpha=0$ ) |                     |                       | (b) Preserve weight $\alpha$ |                   |                     |
|-------------------------------------|---------------------|-----------------------|------------------------------|-------------------|---------------------|
| $K$                                 | BERT<br>( $K=4^*$ ) | stella<br>( $K=4^*$ ) | $\alpha$                     | BERT<br>( $K=4$ ) | stella<br>( $K=2$ ) |
| 2                                   | 0.5856              | <b>0.6181</b>         | 0                            | 0.5920            | 0.6181              |
| 3                                   | 0.5878              | 0.6147                | 0.001                        | 0.5920            | 0.6187              |
| 4                                   | <b>0.5920</b>       | 0.6179                | 0.01                         | 0.5921            | 0.6157              |
| 5                                   | 0.5896              | 0.6054                | 0.1                          | 0.5920            | 0.6180              |
| 6                                   | 0.5895              | 0.6081                | 1                            | 0.5919            | 0.6189              |

Table 9: **Method robustness on Luxury (NDCG@10).** (a) Bernstein order  $K$ : both encoders show a broad plateau.  $*$  denotes the validation-selected  $K$  used in Table 1 for Luxury. (b) MSE preserve weight  $\alpha$ : evaluated with  $K=4$  for BERT and  $K=2$  for stella; BERT remains highly stable across  $\alpha \in [0, 1]$ , while stella varies by at most 0.0032.

LightGCN propagation, and graph-enhanced item representations—and replacing only its original collaborative-filtering prediction head with the same SASRec backbone used in our sequential recommendation experiments. The LightGCN-refined item representations are used as the item embeddings for SASRec, while the remaining AlphaRec representation pipeline is unchanged.

We independently tune AlphaSASRec using the same validation protocol as CASG. Specifically, we search the LightGCN propagation depth over 1, 2, 3, projector learning rate over  $5e-4, 1e-3$ , and dropout over 0.2, 0.3, 0.5. This yields 18 adaptation-specific configurations, which is comparable to, and in practice larger than, the tuning budget used for CASG. All model selection is performed on the validation set, and test results are reported only for the selected configuration.

## F Graph construction Reproducibility: Hyperparameters

**Graph construction procedure.** Given training sequences  $\{(s_1, s_2, \dots, s_{|S|})\}$  over item set  $\mathcal{I}$ , we build the item-item co-occurrence graph as follows.

1. **Co-occurrence counting.** For each sequence,

| Group      | Hyperparameter                | Value                |
|------------|-------------------------------|----------------------|
| Graph      | co-occurrence window $w$      | 2                    |
|            | top- $k$ neighbor truncation  | 20                   |
| Filter     | Bernstein order $K$ (BERT)    | 2-6                  |
|            | Bernstein order $K$ (stella)  | 2-6                  |
| Training   | optimizer                     | Adam                 |
|            | learning rate                 | $1e-2$               |
|            | weight decay                  | 0                    |
|            | epochs                        | 100                  |
|            | InfoNCE batch size            | 512                  |
|            | InfoNCE temperature $\tau$    | 0.1                  |
| Encoder    | embedding dim $d$             | 768                  |
| Downstream | projection $d \rightarrow d'$ | 768 $\rightarrow$ 64 |
|            | SASRec dim / blocks / heads   | 64 / 2 / 1           |

Table 10: **Offline filter-training hyperparameters.** All values are shared across the three datasets unless explicitly noted; the Bernstein order  $K$ , is selected on the validation set for each encoder-dataset setting, based on the sensitivity analysis in Table 9.

we slide a symmetric window of half-width  $w=2$  centered on every item  $i$ . Every distinct pair  $(i, j)$  within the window contributes one undirected edge; the edge weight  $c_{ij}$  is the raw co-occurrence count aggregated across all sequences and all positions (no distance-based decay). Self-loops are excluded.

2. **Top- $k$  sparsification.** For each item  $i$ , we sort its neighbors by descending  $c_{ij}$  (ties broken lexicographically) and retain the top  $k=20$  neighbors. An undirected edge  $(i, j)$  is kept if *either* endpoint retains the other; this union strategy preserves asymmetry in locally dense neighborhoods.
3. **Symmetric normalization.** The retained edges form the adjacency matrix  $\mathbf{A}$ . We compute the degree matrix  $\mathbf{D} = \text{diag}(\mathbf{A}\mathbf{1})$  and the symmetric normalized Laplacian  $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ , with  $D_{ii}^{-1/2} = 0$  for isolated nodes.

Table 10 lists the offline filter-training hyperpa-

Table 11: Performance comparison on three datasets.

| Method      | Encoder | Dataset    | HR@5                                  | NDCG@5                                | HR@10                                 | NDCG@10                               |
|-------------|---------|------------|---------------------------------------|---------------------------------------|---------------------------------------|---------------------------------------|
| MAERec      | –       | Luxury     | $0.5430 \pm 0.0035$                   | $0.4692 \pm 0.0032$                   | $0.6168 \pm 0.0033$                   | $0.4912 \pm 0.0020$                   |
|             |         | Instrument | $0.3580 \pm 0.0009$                   | $0.2751 \pm 0.0008$                   | $0.4553 \pm 0.0003$                   | $0.3065 \pm 0.0005$                   |
|             |         | Yelp       | $0.5956 \pm 0.0038$                   | $0.4184 \pm 0.0033$                   | $0.7757 \pm 0.0076$                   | $0.4803 \pm 0.0045$                   |
| LESSR       | –       | Luxury     | $0.5803 \pm 0.0028$                   | $0.5100 \pm 0.0030$                   | $0.6456 \pm 0.0051$                   | $0.5300 \pm 0.0034$                   |
|             |         | Instrument | $0.4236 \pm 0.0014$                   | $0.3280 \pm 0.0019$                   | $0.5393 \pm 0.0021$                   | $0.3675 \pm 0.0022$                   |
|             |         | Yelp       | $0.6879 \pm 0.0014$                   | $0.5104 \pm 0.0013$                   | $0.8185 \pm 0.0007$                   | $0.5436 \pm 0.0009$                   |
| DMI-GNN     | –       | Luxury     | $0.5715 \pm 0.0009$                   | $0.5012 \pm 0.0012$                   | $0.6424 \pm 0.0035$                   | $0.5239 \pm 0.0022$                   |
|             |         | Instrument | $0.4270 \pm 0.0009$                   | $0.3297 \pm 0.0010$                   | $0.5357 \pm 0.0009$                   | $0.3642 \pm 0.0010$                   |
|             |         | Yelp       | $0.6878 \pm 0.0207$                   | $0.5138 \pm 0.0203$                   | $0.8373 \pm 0.0142$                   | $0.5521 \pm 0.0181$                   |
| Pure ID@64  | –       | Luxury     | $0.5890 \pm 0.0034$                   | $0.5143 \pm 0.0019$                   | $0.6590 \pm 0.0038$                   | $0.5371 \pm 0.0020$                   |
|             |         | Instrument | $0.4405 \pm 0.0025$                   | $0.3370 \pm 0.0013$                   | $0.5610 \pm 0.0009$                   | $0.3758 \pm 0.0007$                   |
|             |         | Yelp       | $0.6822 \pm 0.0038$                   | $0.4955 \pm 0.0081$                   | $0.8341 \pm 0.0072$                   | $0.5450 \pm 0.0092$                   |
| MoRec       | BERT    | Luxury     | $0.5786 \pm 0.0016$                   | $0.4996 \pm 0.0003$                   | $0.6586 \pm 0.0025$                   | $0.5250 \pm 0.0012$                   |
|             |         | Instrument | $0.3630 \pm 0.0041$                   | $0.2618 \pm 0.0020$                   | $0.4947 \pm 0.0034$                   | $0.3042 \pm 0.0020$                   |
|             |         | Yelp       | $0.6577 \pm 0.0012$                   | $0.4658 \pm 0.0021$                   | $0.8263 \pm 0.0023$                   | $0.5207 \pm 0.0018$                   |
|             | stella  | Luxury     | $0.6517 \pm 0.0051$                   | $0.5698 \pm 0.0026$                   | $0.7213 \pm 0.0030$                   | $0.5922 \pm 0.0019$                   |
|             |         | Instrument | $0.4322 \pm 0.0015$                   | $0.3219 \pm 0.0018$                   | $0.5668 \pm 0.0022$                   | $0.3653 \pm 0.0019$                   |
|             |         | Yelp       | $0.7027 \pm 0.0008$                   | $0.5103 \pm 0.0015$                   | $0.8555 \pm 0.0010$                   | $0.5601 \pm 0.0016$                   |
| UniSRec     | BERT    | Luxury     | $0.6424 \pm 0.0043$                   | $0.5586 \pm 0.0045$                   | $0.7094 \pm 0.0016$                   | $0.5799 \pm 0.0035$                   |
|             |         | Instrument | $0.3687 \pm 0.0012$                   | $0.2784 \pm 0.0002$                   | $0.4865 \pm 0.0003$                   | $0.3164 \pm 0.0002$                   |
|             |         | Yelp       | $0.6279 \pm 0.0010$                   | $0.4431 \pm 0.0007$                   | $0.8004 \pm 0.0013$                   | $0.4990 \pm 0.0008$                   |
|             | stella  | Luxury     | <b><math>0.6867 \pm 0.0006</math></b> | <u><math>0.5945 \pm 0.0019</math></u> | <u><math>0.7494 \pm 0.0021</math></u> | <u><math>0.6147 \pm 0.0026</math></u> |
|             |         | Instrument | $0.4089 \pm 0.0042$                   | $0.3099 \pm 0.0041$                   | $0.5336 \pm 0.0044$                   | $0.3501 \pm 0.0040$                   |
|             |         | Yelp       | $0.6628 \pm 0.0032$                   | $0.4794 \pm 0.0036$                   | $0.8231 \pm 0.0034$                   | $0.5315 \pm 0.0037$                   |
| WhitenRec   | BERT    | Luxury     | $0.5831 \pm 0.0035$                   | $0.5086 \pm 0.0033$                   | $0.6539 \pm 0.0034$                   | $0.5312 \pm 0.0018$                   |
|             |         | Instrument | $0.3602 \pm 0.0009$                   | $0.2646 \pm 0.0005$                   | $0.4869 \pm 0.0018$                   | $0.3054 \pm 0.0007$                   |
|             |         | Yelp       | $0.6712 \pm 0.0010$                   | $0.4792 \pm 0.0008$                   | $0.8356 \pm 0.0004$                   | $0.5327 \pm 0.0004$                   |
|             | stella  | Luxury     | $0.6175 \pm 0.0024$                   | $0.5415 \pm 0.0025$                   | $0.6847 \pm 0.0021$                   | $0.5631 \pm 0.0014$                   |
|             |         | Instrument | $0.4155 \pm 0.0025$                   | $0.3089 \pm 0.0022$                   | $0.5444 \pm 0.0033$                   | $0.3505 \pm 0.0022$                   |
|             |         | Yelp       | $0.7030 \pm 0.0005$                   | $0.5093 \pm 0.0006$                   | $0.8551 \pm 0.0014$                   | $0.5588 \pm 0.0010$                   |
| AlphaFuse   | BERT    | Luxury     | $0.5557 \pm 0.0038$                   | $0.4854 \pm 0.0012$                   | $0.6348 \pm 0.0040$                   | $0.5108 \pm 0.0008$                   |
|             |         | Instrument | $0.4318 \pm 0.0003$                   | $0.3223 \pm 0.0002$                   | $0.5580 \pm 0.0006$                   | $0.3631 \pm 0.0002$                   |
|             |         | Yelp       | $0.6470 \pm 0.0008$                   | $0.4605 \pm 0.0008$                   | $0.8165 \pm 0.0019$                   | $0.5155 \pm 0.0011$                   |
|             | stella  | Luxury     | $0.6136 \pm 0.0027$                   | $0.5361 \pm 0.0023$                   | $0.6907 \pm 0.0056$                   | $0.5610 \pm 0.0032$                   |
|             |         | Instrument | $0.4612 \pm 0.0023$                   | <u><math>0.3554 \pm 0.0007</math></u> | $0.5838 \pm 0.0007$                   | $0.3899 \pm 0.0006$                   |
|             |         | Yelp       | $0.6835 \pm 0.0026$                   | $0.4906 \pm 0.0015$                   | $0.8426 \pm 0.0005$                   | $0.5423 \pm 0.0008$                   |
| AlphaSASRec | BERT    | Luxury     | $0.6052 \pm 0.0039$                   | $0.5174 \pm 0.0026$                   | $0.6893 \pm 0.0054$                   | $0.5446 \pm 0.0028$                   |
|             |         | Instrument | $0.4331 \pm 0.0019$                   | $0.3207 \pm 0.0019$                   | $0.5586 \pm 0.0013$                   | $0.3612 \pm 0.0016$                   |
|             |         | Yelp       | $0.7057 \pm 0.0017$                   | $0.5155 \pm 0.0004$                   | $0.8571 \pm 0.0017$                   | $0.5649 \pm 0.0005$                   |
|             | stella  | Luxury     | $0.6547 \pm 0.0020$                   | $0.5650 \pm 0.0038$                   | $0.7307 \pm 0.0020$                   | $0.5896 \pm 0.0035$                   |
|             |         | Instrument | <u><math>0.4639 \pm 0.0015</math></u> | $0.3519 \pm 0.0005$                   | <u><math>0.5866 \pm 0.0014</math></u> | <u><math>0.3915 \pm 0.0002</math></u> |
|             |         | Yelp       | <u><math>0.7401 \pm 0.0014</math></u> | <u><math>0.5514 \pm 0.0017</math></u> | <u><math>0.8760 \pm 0.0003</math></u> | <u><math>0.5957 \pm 0.0013</math></u> |
| CASG        | BERT    | Luxury     | $0.6574 \pm 0.0022$                   | $0.5663 \pm 0.0004$                   | $0.7366 \pm 0.0034$                   | $0.5920 \pm 0.0006$                   |
|             |         | Instrument | $0.4462 \pm 0.0017$                   | $0.3396 \pm 0.0008$                   | $0.5716 \pm 0.0024$                   | $0.3800 \pm 0.0011$                   |
|             |         | Yelp       | $0.7123 \pm 0.0083$                   | $0.5202 \pm 0.0073$                   | $0.8622 \pm 0.0050$                   | $0.5690 \pm 0.0063$                   |
|             | stella  | Luxury     | <u><math>0.6809 \pm 0.0050</math></u> | <b><math>0.5947 \pm 0.0038</math></b> | <b><math>0.7525 \pm 0.0034</math></b> | <b><math>0.6179 \pm 0.0032</math></b> |
|             |         | Instrument | <b><math>0.4692 \pm 0.0005</math></b> | <b><math>0.3585 \pm 0.0018</math></b> | <b><math>0.5961 \pm 0.0032</math></b> | <b><math>0.3996 \pm 0.0009</math></b> |
|             |         | Yelp       | <b><math>0.7425 \pm 0.0006</math></b> | <b><math>0.5535 \pm 0.0013</math></b> | <b><math>0.8765 \pm 0.0004</math></b> | <b><math>0.5972 \pm 0.0012</math></b> |

rameters used to produce every CASG row in the main paper. Downstream SASRec hyperparameters follow the per-method grid search described

in §5 (Setup). Filter training is deterministic given the seed; we run three seeds (42, 2026, 3407) per (encoder, dataset).