DuET: Dual-View Tensor-to-Topology Spectral Adapter for Enhancing Sparse Tensor Factorization

Jun-Gi Jang
jungi@dgist.ac.kr
Daegu Gyeongbuk Institute of Science and Technology
Daegu, Republic of Korea

Jingrui He
jingrui@illinois.edu
University of Illinois Urbana-Champaign
Champaign, USA

Andrew J. Margenot
margenot@illinois.edu
University of Illinois Urbana-Champaign
Champaign, USA

Hanghang Tong
htong@illinois.edu
University of Illinois Urbana-Champaign
Champaign, USA

Abstract

Many real-world datasets, ranging from web-interaction logs to biomedical networks, can be represented as sparse tensors where most entries are unobserved. Tensor factorization (TF) learns latent representations and a predictor to estimate unobserved entries and has been widely applied to higher-order recommendation, biomedical retrieval, and completion. However, under extreme sparsity, many entities participate in only a few interactions, yielding noisy and undertrained latent vectors with poor generalization. We propose DuET, a model-agnostic refinement framework that stabilizes TF embeddings by separating structural denoising from tensor factorization. DuET distills a dual-view structural prior from observed tuples and refines it via per-view low-rank spectral filtering and mode-wise gated residual refinement. We provide a theoretical guarantee that the resulting refinement is Frobenius-norm nonexpansive, ensuring stability under repeated refinement. DuET seamlessly integrates with existing TF models without modifying their scoring architectures. Experiments on nine real-world tensors demonstrate consistent improvements for both retrieval and completion, with gains of up to 52.1% in NDCG@10 and reductions of up to 13.8% in RMSE.

CCS Concepts

• **Information systems** → **Data mining**; • **Computing methodologies** → **Factorization methods**.

Keywords

Sparse tensor factorization, Dual-view topology, Spectral denoising, Higher-order interactions, Representation refinement

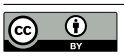
ACM Reference Format:
Jun-Gi Jang, Jingrui He, Andrew J. Margenot, and Hanghang Tong. 2026. DuET: Dual-View Tensor-to-Topology Spectral Adapter for Enhancing Sparse Tensor Factorization. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '26), August 09–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770855.3817884

1 Introduction

A tensor, a multi-dimensional array, naturally represents real-world datasets that capture higher-order relationships among entities. Many real-world datasets, including web-interaction logs (e.g., recommendation, check-ins, and social networks) and biomedical interaction networks, can be represented as sparse tensors where only a tiny fraction of entries are observed (e.g., over 99.99% entries are unobserved in the Gowalla dataset [34]). Tensor factorization (TF) has been a fundamental tool for analyzing such data. Multilinear tensor methods [6, 19, 27, 51] and their modern neural extensions [2, 11, 35] learn latent vectors of entities and a predictor to estimate missing entries. TF methods provide a general framework and have been successfully applied to diverse tasks, including tensor completion [1, 36, 48, 50], anomaly detection [22, 24], recommendation [7, 8], and knowledge graph completion [4, 31].

However, real-world tensors are often extremely sparse and imbalanced: most entities appear in only a handful of interactions out of an enormous space of possible index combinations. As a result, TF methods must learn latent representations from a limited and skewed subset of observed tuples, leading to high-variance noise that propagates through shared latent spaces and degrades generalization to unobserved entries. These challenges expose an implicit assumption in existing TF methods that latent representations learned from sparse supervision are sufficiently reliable, despite the absence of explicit mechanisms to assess or correct latent unreliability. This motivates the need for principled denoising strategies that exploit structural regularities inherent in observed tuples to stabilize representation learning. A key question then arises: how can we distill such structural priors from a sparse tensor and leverage them to explicitly refine latent representations before they are consumed by TF predictors?

In this paper, we propose DuET (**Du**al-View Tensor-to-Topology Spectral Adapter for **E**nhancing Sparse **T**ensor Factorization), a novel framework that decouples structural stabilization from the factorization process. DuET refines latent representations through a separate, seamless adapter, enhancing performance without modifying underlying TF objectives or model architectures. Specifically, DuET consists of three key components: (i) dual-view topology induction that captures complementary within-mode and cross-mode structural patterns via tensor unfoldings (marginal-view) and tensor-induced hypergraphs (incidence-view); (ii) per-view low-rank spectral denoising to distill informative structure from noisy inductions; and (iii) learnable mode-wise gated residual refinement for adaptive signal injection across modes and datasets. In such settings, DuET operates as a plug-and-play module that integrates seamlessly with existing TF models. By enriching scarce supervision with filtered structural evidence, DuET stabilizes factor learning under extreme sparsity and improves generalization to unobserved entries. Extensive experiments on nine real-world datasets



KDD '26, August 09–13, 2026, Jeju Island, Republic of Korea

ACM ISBN 979-8-4007-2259-2/2026/08
https://doi.org/10.1145/3770855.3817884

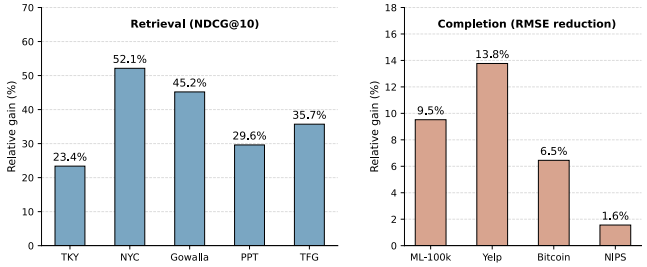


Figure 1: Average relative gain (%) of DuET over six TF backbones: NDCG@10 improvement on retrieval datasets (left) and RMSE reduction on completion datasets (right). DuET consistently improves performance across datasets and is compatible with 5–6/6 backbones in most cases (see Tables 2 and 3 for per-backbone results).

demonstrate consistent improvements across diverse TF backbones (See Figure 1), including CP [6, 19, 27], Tucker [51], CostCo [35], M^{2}DMTF [11], and NeAT [2]. These results show that DuET offers (1) improved accuracy via denoising, (2) seamless compatibility with diverse TF models, and (3) wide applicability across datasets and tasks.

Our main contributions are summarized as follows:

- **Model-Agnostic Tensor-to-Topology Adapter.** We propose DuET, a plug-and-play method that stabilizes latent representations using dual-view topology induction, per-view spectral denoising, and mode-wise gated residual refinement.
- **Stability Guarantee.** We prove that the proposed two-stage gated refinement is non-expansive in Frobenius norm, ensuring stability under repeated refinement.
- **Comprehensive Evaluation.** Experiments on nine real-world tensors demonstrate consistent gains (See Figure 1) across diverse TF backbones and tasks.

Our code and datasets are available at **https://github.com/jungijang/DuET**.

2 Preliminaries

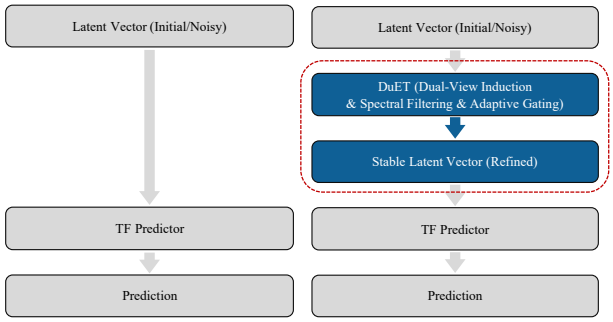
$\mathcal{X}$ denotes a tensor, and an index i_n indicates the i_n-th entity of the n-th dimension. $(i_1, ..., i_N)$ is an interaction between N entities $i_1, ..., i_N$, and I_n is the size of the n-th dimension. A set Ω_o of observed interactions and a set Ω_u of unobserved interactions are mutually disjoint sets whose union is Ω_a which is a set of all interactions in a sparse tensor. Table 9 in Appendix A presents symbols frequently used in this paper.

Tensor factorization has been widely used for analyzing real-world tensors. In the training phase, it learns decomposition results that minimize a loss function. Following [35], tensor factorization learns (1) factor matrices $\mathbf{U}^{(1)}, ..., \mathbf{U}^{(N)}$ corresponding to each dimension and (2) parameters θ.

$$\hat{\mathcal{X}}(i_1, ..., i_N) \leftarrow f(\mathbf{U}^{(1)}[i_1, :], ..., \mathbf{U}^{(N)}[i_N, :], \theta) \tag{1}$$

In the inference phase, given an interaction $(i_1, ..., i_N)$, it predicts the value of $(i_1, ..., i_N)$ by using the factor matrices and the parameters as shown in Equation (1).

Now, let us look into the TF scoring function f. As shown in Figure 2, existing tensor factorization methods can be viewed as



(a) Existing TF method (b) DuET with a TF method

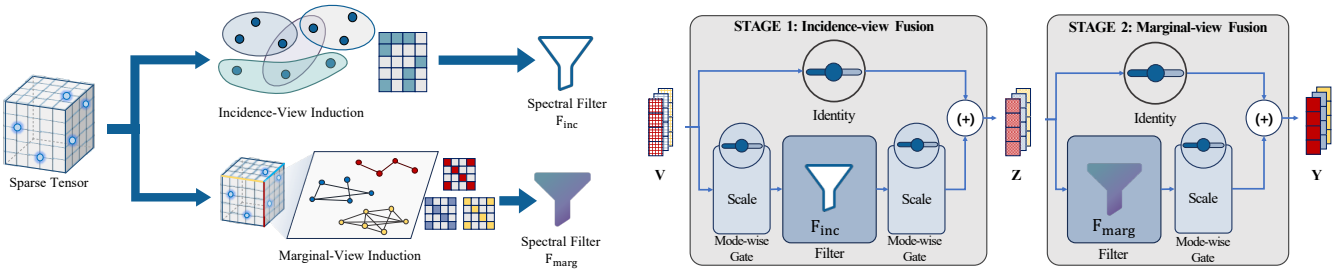
Figure 2: Comparison between conventional tensor factorization (TF) pipelines and the proposed DuET-based refinement framework. Conventional TF methods directly feed latent vectors into a TF predictor, whereas DuET stabilizes latent representations through dual-view topology induction, per-view spectral filtering, and adaptive gating, while keeping the TF predictor unchanged.

consisting of two modules: (1) an embedding lookup that retrieves latent vectors $\mathbf{U}^{(n)}[i_n, :]$ for entities $i_1, ..., i_N$ from factor matrices, and (2) a predictor that estimates the entry value using these vectors. Here, $\mathbf{U}^{(n)}[i_n, :]$ denotes the latent vector of the i_n-th entity in the n-th dimension. Prior work has largely focused on enhancing the predictor—ranging from multilinear scoring functions to expressive neural architectures—to better capture higher-order interactions. However, a critical imbalance remains: while the predictor is optimized using all observed tuples, each entity embedding is updated only when the entity appears in an observation. In highly sparse and long-tailed tensors, this imbalance produces noisy and undertrained embeddings for infrequent entities, which in turn harms generalization to unseen entries. These observations motivate representation enhancement as a complementary direction: rather than redesigning the predictor, we aim to refine entity embeddings to improve generalization under sparsity.

3 Proposed Method

We propose DuET, a model-agnostic adapter that refines TF embeddings by distilling tuple-induced dual-view topologies into denoised spectral priors and injecting them in a stable, mode-adaptive manner. The core technical contributions are summarized below:

- **Dual-view Topology Induction.**
  - ⇒ **Challenge:** In real-world sparse tensors, only a small fraction of tuples is observed, and the observations are imbalanced; relying on a single induced graph or external side information is often unavailable or unreliable, and fails to capture both mode-specific proximity and joint interaction context across entities.
  - ⇒ **Idea:** Distill a tuple-induced dual-view structural prior from two complementary constructions: (1) a marginal view induced from unfoldings to capture mode-specific proximity, and (2) an incidence view induced from a tuple-incidence hypergraph to capture multi-way relations. These two views provide rich relational evidence.



(a) Dual-View Induction & Per-View Spectral Filtering

(b) Two-Stage Gated Fusion

Figure 3: Overview of DuET. (a) We construct incidence and marginal views from observed tensor tuples and derive denoised spectral filters (F_{inc} and F_{marg}). (b) Latent vectors are sequentially refined through mode-wise gated residual updates, which control the injection of structural signals and yield stable representations.

- **Per-View Spectral Denoising for Stable Refinement**
  - ⇒ **Challenge:** Topologies induced from a sparse tensor can be noisy; naive propagation can amplify spurious links and introduce unstable updates.
  - ⇒ **Idea:** Convert each induced view into a denoised spectral prior via per-view low-rank spectral filtering, attenuating high-frequency noise while preserving informative smooth components. This produces stable refinement signals before injection into TF embeddings.
- **Mode-wise Selective Fusion via Gated Residual Injection.**
  - ⇒ **Challenge:** The usefulness of marginal- and incidence-view structures is heterogeneous across modes and datasets; fixed fusion may inject misleading priors and hurt generalization.
  - ⇒ **Idea:** Learn mode-wise gates (τ_n, ρ_n) to control each view's contribution through residual mixing, enabling selective, non-intrusive injection that adopts a helpful structure while suppressing harmful transfer.

As shown in Figure 2(b), our framework refines TF embeddings by positioning a stabilizing refinement module between latent representations and the TF predictor. Figure 3 details the components of DuET, including dual-view topology induction, per-view spectral denoising, and mode-adaptive gated fusion.

3.1 Dual-View Topology Induction

Under extreme sparsity, many entities appear in only a handful of tuples, yielding noisy and poorly trained latent factors. While auxiliary relational cues (e.g., metadata) can help when available [17, 39, 49], we focus on the tuple-only setting and induce structural priors directly from observed tuples. Our key intuition is that the tensor itself already contains structural cues: by distilling self-induced topology from observed tuples, we can emulate the effect of external relations without relying on any auxiliary data. Concretely, we treat the observed tuples as the sole evidence for topology induction and extract structural priors to refine latent factors.

To capture tensor-implied structure under sparse regimes, we propose a dual-view topology induction strategy. Prior work [38, 43, 58] often induces mode-specific structural priors by deriving intra-mode affinities in the observation space, yielding relational structure within each mode. Such priors can be effective, but they do not explicitly preserve tuple-level, cross-mode dependencies and can thus discard the rich joint interaction context. Motivated by this, we characterize tensor-implied dependencies from two complementary perspectives, as illustrated in Figure 3(a): (1) a marginal view and (2) an incidence view. The marginal view leverages tensor unfoldings to capture mode-wise co-occurrence structure as a robust intra-mode prior. The incidence view introduces a hypergraph representation that treats each observed tuple as a hyperedge and constructs an incidence-induced spectral topology to explicitly model the joint interaction context. By reconciling the marginal and incidence views, DuET yields a structural prior that is simultaneously mode-wise informative and tuple-preserving, stabilizing representation learning even with limited observations.

Incidence-View Topology Induction. To capture tuple-level, multi-way dependencies implied by observed interactions, we model the tensor as a hypergraph in which each tuple forms a hyperedge. For compatibility with standard pairwise graph operators, we optionally project this hypergraph into a clique-expanded graph, yielding a pairwise representation. Given an N-th order sparse tensor of size $I_1 \times \cdots \times I_N$ with observed interaction set Ω_o, we define a hypergraph $G_H = (V, E)$ whose nodes correspond to all entities across modes and whose hyperedges correspond to observed tuples. Thus, $|V| = \sum_{n=1}^{N} I_n$ and $|E| = |\Omega_o|$. Let $\mathbf{B} \in \{0,1\}^{|V|\times|E|}$ be the incidence matrix, where $B_{v,e} = 1$ if node v participates in interaction e, and 0 otherwise. Constructing $\mathbf{B}$ requires $O(N|\Omega_o|)$ time and space. By construction, each hyperedge contains exactly one entity from each mode (i.e., G_H is N-uniform).

While G_H preserves multi-way associations, directly operating on hyperedges can be inefficient when $|E| \gg |V|$. Optionally, we construct a clique-expanded (pairwise) graph $G_C = (V, E_C)$ by projecting each hyperedge to pairwise co-occurrence edges.

$$\mathbf{A} = \mathbf{B}\mathbf{B}^\top - \mathrm{Diag}\big(\mathrm{diag}(\mathbf{B}\mathbf{B}^\top)\big), \tag{2}$$

where $\mathbf{A}_{ij}$ (for $i \neq j$) counts the number of observed tuples in which entities i and j co-occur, and $\mathbf{A}_{ii} = 0$. Constructing $\mathbf{A}$ requires $O(N^2|\Omega_o|)$ time and space in the worst case where N is typically small (e.g., $N \leq 4$). $\mathbf{A}$ links entities across modes while marginal-view operators are mode-wise. By projecting each hyperedge, we derive inter-mode co-occurrence links (e.g., user–item or item–time) from tuples, thereby reducing the loss of multi-way context incurred by marginal-view projections. This construction enables the model to exploit the heterogeneous topology of the tensor, where different

types of entities can mutually regularize each other through their shared participation in observed tuples.

Marginal-View Topology Induction. While incidence-view induction captures inter-dimensional dependencies, the marginal view focuses on structural correlations among entities within the same mode. The key intuition is that entities exhibiting similar interaction contexts across other modes should admit closer representations in the latent space, which is particularly helpful under extreme sparsity. For each mode $n \in \{1, \dots, N\}$, we unfold the observed tensor into a *binary* matrix $\mathbf{X}_{(n)} \in \{0,1\}^{I_n \times J_n}$, where $J_n = \prod_{k \neq n} I_k$. Each row of $\mathbf{X}_{(n)}$ represents the *existence* pattern of observed tuples involving a specific entity in mode n across the remaining $N-1$ modes, i.e., $\mathbf{X}_{(n)}[i_n, \mathbf{j}] = 1$ if the corresponding index combination $\mathbf{j}$ of the remaining modes co-occurs with entity i_n in at least one observed tuple and 0 otherwise.

Using these binary interaction profiles, we construct a marginal-view adjacency matrix $\mathbf{S}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ that captures marginal-view similarity via shared contexts:

$$\mathbf{S}^{(n)} = \mathbf{X}_{(n)}\mathbf{X}_{(n)}^\top - \mathrm{Diag}\big(\mathrm{diag}(\mathbf{X}_{(n)}\mathbf{X}_{(n)}^\top)\big). \tag{3}$$

For $i \neq j$, the entry $\mathbf{S}^{(n)}_{ij}$ counts the number of common interaction contexts that entities i and j share across the other $N-1$ modes. We set the diagonal to zero to focus on dependencies among distinct entities within the mode. Using sparse matrix operations, constructing $\mathbf{S}^{(n)}$ across all modes takes $O\Big(\sum_{n=1}^{N} |\Omega_o| \cdot d_n\Big)$ time, where d_n denotes the average number of observed interactions per entity in mode n. The space cost is $O\Big(\sum_{n=1}^{N} \mathrm{nnz}\big(\mathbf{S}^{(n)}\big)\Big)$, i.e., linear in the number of stored marginal-view edges. The resulting collection $\{\mathbf{S}^{(1)}, \dots, \mathbf{S}^{(N)}\}$ provides mode-specific structural guidance that complements the incidence-view topology induced from tuples.

3.2 Per-View Spectral Denoising

Next, we aim to efficiently leverage each induced topology while attenuating noise introduced by incidental co-occurrences. A straightforward approach is to operate on the induced graphs by explicitly materializing their adjacency matrices. However, as noted in Section 3.1, such materialization can be computationally prohibitive for large-scale tensors and does not explicitly denoise the induced connectivity. To address both issues, we perform *per-view spectral denoising*, which (i) avoids the overhead of adjacency materialization and (ii) suppresses high-frequency noise in each view before using it for latent factor refinement.

Concretely, we perform spectral denoising for each view by constructing a symmetric normalized operator and retaining only its dominant spectral components. Let $\mathbf{W} \in \{\mathbf{B}, \mathbf{X}_{(1)}, \dots, \mathbf{X}_{(N)}\}$ denote the sparse binary matrix of a view, hypergraph incidence or mode-wise matricization. We first define the corresponding diagonal-free co-occurrence adjacency

$$\mathbf{G}(\mathbf{W}) \;=\; \mathbf{W}\mathbf{W}^\top \;-\; \mathrm{Diag}\big(\mathrm{diag}(\mathbf{W}\mathbf{W}^\top)\big), \tag{4}$$

where $\mathrm{diag}(\cdot)$ denotes the operator that extracts the diagonal elements of a matrix into a vector, and $\mathrm{Diag}(\cdot)$ denotes the operator that creates a diagonal matrix from a vector. Although Eq. (4) is expressed in terms of a co-occurrence matrix, it does not require explicitly materializing the corresponding pairwise graph. In particular, for the incidence view where $\mathbf{W} = \mathbf{B}$, matrix–vector products with this operator are computed implicitly using sparse multiplications with $\mathbf{B}$ and $\mathbf{B}^\top$. Given its degree vector $\mathbf{d} = \mathbf{G}(\mathbf{W})\mathbf{1}$ with degree matrix $\mathbf{D} = \mathrm{Diag}(\mathbf{d})$, we then form the symmetric normalized operator

$$\mathbf{M}(\mathbf{W}) \;=\; \mathbf{D}^{-1/2}\,\mathbf{G}(\mathbf{W})\,\mathbf{D}^{-1/2}. \tag{5}$$

Instead of explicitly materializing $\mathbf{G}(\mathbf{W})$ or $\mathbf{M}(\mathbf{W})$, we compute only the top-q eigenpairs $\{(\lambda_i, \mathbf{u}_i)\}_{i=1}^{q}$ of $\mathbf{M}(\mathbf{W})$ using an iterative eigensolver (e.g., `eigsh` [33, 40, 53] and a GPU implementation [40]) with matrix–vector products implemented using $\mathbf{W}$ and $\mathbf{W}^\top$. This yields the truncated eigendecomposition $\mathbf{M}(\mathbf{W}) \approx \mathbf{U}_q\mathbf{\Lambda}_q\mathbf{U}_q^\top$.

Finally, we apply a monotone spectral filter $g(\cdot)$ to attenuate noisy components and obtain a denoised propagation operator

$$\mathbf{P}(\mathbf{W}) \;=\; \mathbf{U}_q\, g(\mathbf{\Lambda}_q)\, \mathbf{U}_q^\top, \tag{6}$$

which we use in the next section to refine latent factors. Since the eigenvalues satisfy $\lambda \in [-1, 1]$, we map them to $[0, 1]$ using $\tilde{\lambda} = \frac{1+\lambda}{2}$. This down-weights noisier (typically smaller) spectral components while preserving dominant, smooth signals.

A key advantage of our spectral denoising is that we never materialize $\mathbf{G}(\mathbf{W})$ or $\mathbf{M}(\mathbf{W})$. Instead, the eigensolver accesses $\mathbf{M}(\mathbf{W})$ only through matrix–vector products, which can be implemented using sparse multiplications with $\mathbf{W}$ and $\mathbf{W}^\top$. We summarize its complexity below:

Proposition 1 (Per-view spectral denoising complexity). *Let $\mathbf{W} \in \mathbb{R}^{N \times M}$ be a sparse view matrix with $\mathrm{nnz}(\mathbf{W})$ nonzeros. If an iterative eigensolver computes the top-q eigenpairs of $\mathbf{M}(\mathbf{W})$ using T matrix–vector products, then the time and space complexities are $O(T \cdot \mathrm{nnz}(\mathbf{W}))$ and $O(\mathrm{nnz}(\mathbf{W}) + Nq)$, respectively.*

The proof is provided in Appendix B.1. Note that storing the truncated eigenvectors requires $O\Big(q \sum_{n=1}^{N} I_n\Big)$ space across all views, which is linear in the number of entities.

3.3 Two-Stage Gated Refinement Operator

We aim to refine the mode-wise latent factor matrices using the denoised dual-view topologies in Sections 3.1 and 3.2. This setting differs from conventional feature propagation, where input node features are typically fixed. In sparse TF, the propagated representations are learnable latent factors updated from sparse supervision. Therefore, unrestricted propagation can repeatedly inject noisy co-occurrence signals into evolving embeddings. This motivates our controlled residual refinement design. While our framework is flexible enough to incorporate various GNN-based propagation functions proposed in the literature [16, 18, 29, 52, 54, 57, 60], two critical challenges call for a tailored refinement layer rather than directly adopting off-the-shelf operators. First, refinement requires stable, explicitly controlled signal injection to avoid over-smoothing and noise accumulation. Second, as modes exhibit different sparsity and co-occurrence statistics, a single global refinement strength is often inadequate. To address these challenges, we propose an adaptive gated refinement mechanism as shown in Figure 3(b): instead of uniform aggregation, it employs learnable mode-wise gates to control how much marginal- and incidence-view structural signals are injected into each mode.

Setup. Let $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times K}\}_{n=1}^{N}$ denote the factor matrices of a base TF method. We vertically concatenate the mode-wise factors

into a unified embedding matrix

$$\mathbf{V} = \left(\mathbf{U}^{(1)}; \cdots; \mathbf{U}^{(N)}\right) \in \mathbb{R}^{(\sum_{n=1}^{N} I_n) \times K}. \quad (7)$$

Let $\mathbf{F}_{\text{inc}} \in \mathbb{R}^{(\sum_n I_n) \times (\sum_n I_n)}$ denote the incidence-view denoised spectral filter induced from the hypergraph-based view. Let $\mathbf{F}_{\text{marg}}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ denote the denoised spectral filter for mode n in the marginal view. We then define the block-diagonal marginal-view operator as

$$\mathbf{F}_{\text{marg}} = \text{blkdiag}\left(\mathbf{F}_{\text{marg}}^{(1)}, \ldots, \mathbf{F}_{\text{marg}}^{(N)}\right), \quad (8)$$

which applies marginal-view propagation independently within each mode, while $\mathbf{F}_{\text{inc}}$ enables incidence-based propagation that couples entities across modes.

Implicit spectral implementation. We do not explicitly materialize $\mathbf{F}_{\text{inc}}$ or $\mathbf{F}_{\text{marg}}^{(n)}$. Instead, we apply them using the denoised spectral filtering operator $\mathbf{P}(\mathbf{W})$ described in Section 3.2. Accordingly, we set

$$\mathbf{F}_{\text{inc}} := \mathbf{P}(\mathbf{W}_{\text{inc}}), \text{ and } \mathbf{F}_{\text{marg}}^{(n)} := \mathbf{P}(\mathbf{W}_{\text{marg}}^{(n)}) \quad (n = 1, \ldots, N), \quad (9)$$

and apply $\mathbf{P}(\mathbf{W})$ to an embedding matrix $\mathbf{V}$ without forming $\mathbf{P}(\mathbf{W})$ explicitly, as $\mathbf{P}(\mathbf{W})\mathbf{V} = \mathbf{U}_q\left(g(\mathbf{\Lambda}_q)\,(\mathbf{U}_q^\top \mathbf{V})\right)$. This costs $O(qK\sum_n^N I_n)$ time and $O(q\sum_n^N I_n)$ memory, without forming $\mathbf{P}(\mathbf{W})$ explicitly.

Stage 1: Incidence-view fusion with symmetric gating. To propagate information via tuple-incidence structure without amplifying embedding magnitudes, we apply a symmetric gated incidence-view update:

$$\mathbf{Z} = (\mathbf{I} - \mathbf{\Phi}_\tau)\mathbf{V} + \mathbf{\Phi}_\tau^{1/2}\,\mathbf{F}_{\text{inc}}\left(\mathbf{\Phi}_\tau^{1/2}\mathbf{V}\right), \quad (10)$$

where $\mathbf{\Phi}_\tau = \text{blkdiag}(\tau_1\mathbf{I}_{I_1}, \ldots, \tau_N\mathbf{I}_{I_N})$ and $\tau_n \in (0, 1)$ are learnable mode-wise gates. Intuitively, the incidence-view filter mixes information through tuple co-occurrence. We use the symmetric form $\mathbf{\Phi}_\tau^{1/2}(\cdot)\mathbf{\Phi}_\tau^{1/2}$ for two reasons: (i) it yields a balanced residual update, where the incidence-view structural prior is gated for both the contributing and receiving entities, and (ii) it preserves non-expansiveness when $\mathbf{F}_{\text{inc}}$ is non-expansive (see Proposition 3). Accordingly, the update provides a stable interpolation between the identity mapping ($\tau_n \to 0$) and full incidence-view filtering ($\tau_n \to 1$).

Stage 2: Marginal-view denoising with residual mixing. We then denoise each mode using a residual mixer:

$$\mathbf{Y} = \mathbf{Z} + \mathbf{\Phi}_\rho\left(\mathbf{F}_{\text{marg}}\mathbf{Z} - \mathbf{Z}\right) = (\mathbf{I} - \mathbf{\Phi}_\rho)\mathbf{Z} + \mathbf{\Phi}_\rho\,\mathbf{F}_{\text{marg}}\mathbf{Z}, \quad (11)$$

where $\mathbf{\Phi}_\rho = \text{blkdiag}(\rho_1\mathbf{I}_{I_1}, \ldots, \rho_N\mathbf{I}_{I_N})$ and $\rho_n \in (0, 1)$ are learnable mode-wise gates. This stage enforces mode-specific smoothness based on shared contexts in the unfolding view, while preserving fidelity to the current embeddings via the residual form.

Iterated refinement and usage. We apply the two stages in Eqs. (10) and (11) for L steps, yielding refined embeddings $\mathbf{Y}^{[L]}$. In practice, we parameterize the gates as $\rho_n = \sigma(\rho_n^{\text{raw}})$ and $\tau_n = \sigma(\tau_n^{\text{raw}})$ to keep them in $(0, 1)$. The incidence-view stage captures inter-dimensional dependencies implied by tuples, whereas the marginal-view stage captures intra-mode similarity induced by shared contexts. The mode-wise gates provide fine-grained control that prevents over-smoothing on sparse or noisy structures.

Proposition 2 (Computational Complexity). *Let $V_{total} = \sum_{n=1}^{N} I_n$, K the embedding dimension, q the truncated spectral rank, and L the number of propagation steps. One propagation step incurs $O(V_{total} \cdot q \cdot K)$ time. Thus, the overall complexity is $O(L \cdot V_{total} \cdot q \cdot K)$.*

Table 1: Description of real-world tensor datasets. For each dataset, we split all observed interactions into train, valid, and test sets with the ratio 7:1:2.

Dataset	Tensor Size	# observed interactions	Density
Retrieval Task			
FS-TKY[a]	$2,293 \times 15,177 \times 168$	373,391	0.0063%
FS-NYC[a]	$1,083 \times 9,989 \times 168$	140,955	0.0077%
Gowalla[b]	$6,342 \times 8,322 \times 168$	524,108	0.0059%
PPT-Ohmnet[c]	$96 \times 3,955 \times 4,134$	2,364,452	0.1506%
TFG-Ohmnet[d]	$48 \times 420 \times 2,144$	20,619	0.0477%
Completion Task			
ML-100k[e]	$943 \times 1,682 \times 31 \times 8$	100,000	0.0254%
Yelp[f]	$70,818 \times 15,580 \times 109$	333,481	0.00027%
Bitcoin[g]	$4,814 \times 5,858 \times 273$	35,592	0.0004%
NIPS[h]	$2,482 \times 2,862 \times 14,036 \times 17$	3,101,609	0.0053%

[a] https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[b] https://snap.stanford.edu/data/loc-gowalla.html
[c] https://snap.stanford.edu/biodata/datasets/10013/10013-PPT-Ohmnet.html
[d] https://snap.stanford.edu/biodata/datasets/10014/10014-TFG-Ohmnet.html
[e] https://grouplens.org/datasets/movielens/
[f] https://www.yelp.com/dataset
[g] https://snap.stanford.edu/data/soc-sign-bitcoin-otc.html
[h] http://frostt.io/tensors/nips/

This complexity depends on the number of entities, the number of refinement steps, the retained spectral rank, and the embedding dimension, rather than the full tensor space, making the refinement cost controllable mainly through q and L.

Integration of DuET with TF Methods. We describe how to integrate DuET with existing TF methods. DuET is model-agnostic: it refines the latent factors while keeping the original TF predictor unchanged. Let $\mathbf{Y}^{[L]} = (\mathbf{U}^{(1)[L]}; \cdots; \mathbf{U}^{(N)[L]})$ denote the refined embeddings after L refinement steps, split into mode-wise factors $\mathbf{U}^{(n)[L]} \in \mathbb{R}^{I_n \times K}$, where $\mathbf{U}^{(n)[0]} = \mathbf{U}^{(n)}$. Given an interaction $(i_1, \ldots, i_N)$, the integrated prediction is

$$\hat{\mathcal{X}}(i_1, \ldots, i_N) = f\left(\mathbf{U}^{(1)[L]}[i_1, :], \ldots, \mathbf{U}^{(N)[L]}[i_N, :]; \theta\right), \quad (12)$$

which differs from Eq. (1) only by replacing $\mathbf{U}^{(n)}$ with $\mathbf{U}^{(n)[L]}$. Thus, a wide range of multilinear and nonlinear TF models can be seamlessly integrated with DuET (see Section 4.1). We note that all parameters, including factor matrices $\mathbf{U}^{(n)}$, the base TF parameters θ, and the gates $\{\rho_n, \tau_n\}_{n=1}^N$, are learned end-to-end.

Stability and controllability. The proposed two-stage gated refinement offers both structural flexibility and numerical stability. We show that the resulting operator is non-expansive, providing a formal stability guarantee even under repeated refinement.

Proposition 3 (Non-expansive two-stage gated refinement). *Consider the two-stage refinement operator defined in Eqs. (10)–(11). Assume that the spectral filters are symmetric with eigenvalues in $[0, 1]$, and the mode-wise gates satisfy $0 \le \rho_n, \tau_n \le 1$ (equivalently, $0 \preceq \mathbf{\Phi}_\rho, \mathbf{\Phi}_\tau \preceq \mathbf{I}$). Then:*

1. ***Incidence-view stage.*** *The incidence-view update is non-expansive: $\|\mathbf{Z}\|_F \le \|\mathbf{V}\|_F$ in Eq. (10).*
2. ***Marginal-view stage.*** *The marginal-view mixer is non-expansive: $\|(\mathbf{I} - \mathbf{\Phi}_\rho) + \mathbf{\Phi}_\rho\mathbf{F}_{\text{marg}}\|_2 \le 1$.*

As a consequence, the composite mapping $T : \mathbf{V} \mapsto \mathbf{Y}$ is non-expansive in Frobenius norm, i.e., $\|\mathbf{Y}\|_F \le \|\mathbf{V}\|_F$. In particular, every Frobenius ball $\Omega_C = \{\mathbf{V} : \|\mathbf{V}\|_F \le C\}$ is invariant under repeated refinement.

Proposition 3 ensures that neither stage amplifies embedding norms, preventing instability even under repeated refinement. Meanwhile,

Table 2: Compatibility of DuET for the retrieval task. The triangles ▲ and ▼ indicate performance improvement and degradation, respectively. A bold text indicates the best performance, and an underline indicates the second best. The last column shows the average improvement % of DuET over six baselines, computed as ((avg. DuET+6 TF methods/avg. 6 TF methods) − 1) ∗ 100, and the number of improved baselines (# of ▲/6).

Setting		CP		Tucker		CostCo		NCFT		M^2DMTF		NeAT		Improvement
Dataset	Metric	w/o DuET	w/ DuET	w/o DuET	w/ DuET	w/o DuET	w/ DuET	w/o DuET	w/ DuET	w/o DuET	w/ DuET	w/o DuET	w/ DuET	xx.xx% (▲/6)
FS-TKY	HR@10	0.1246	0.1416 ▲	0.1197	0.1334 ▲	0.1076	**0.1519** ▲	0.1035	0.1234 ▲	0.1000	0.0918 ▼	0.0839	0.1281 ▲	20.45% (5/6)
	MAP@10	0.0054	0.0061 ▲	0.0054	0.0062 ▲	0.0047	**0.0067** ▲	0.0046	0.0055 ▲	0.0041	0.0043 ▲	0.0035	0.0057 ▲	24.53% (6/6)
	NDCG@10	0.0159	0.0182 ▲	0.0158	0.0180 ▲	0.0140	**0.0197** ▲	0.0136	0.0161 ▲	0.0123	0.0123	0.0103	0.0168 ▲	23.38% (5/6)
FS-NYC	HR@10	0.0947	0.0849 ▼	0.0303	0.0308 ▲	0.0709	**0.1579** ▲	0.0199	0.0356 ▲	0.0161	0.0240 ▲	0.0480	0.0626 ▲	41.42% (5/6)
	MAP@10	0.0042	0.0038 ▼	0.0011	0.0012 ▲	0.0025	**0.0076** ▲	0.0007	0.0013 ▲	0.0005	0.0009 ▲	0.0020	0.0029 ▲	59.51% (5/6)
	NDCG@10	0.0124	0.0111 ▼	0.0035	0.0037 ▲	0.0080	**0.0216** ▲	0.0023	0.0042 ▲	0.0017	0.0027 ▲	0.0060	0.0083 ▲	52.13% (5/6)
Gowalla	HR@10	0.0602	0.0667 ▲	0.0606	0.0588 ▼	0.1142	**0.2156** ▲	0.0480	0.0543 ▲	0.0187	0.0244 ▲	0.1071	0.1170 ▲	31.34% (5/6)
	MAP@10	0.0026	0.0030 ▲	0.0025	0.0027 ▲	0.0059	**0.0139** ▲	0.0021	0.0023 ▲	0.0007	0.0009 ▲	0.0054	0.0069 ▲	54.29% (6/6)
	NDCG@10	0.0077	0.0087 ▲	0.0076	0.0077 ▲	0.0164	**0.0356** ▲	0.0062	0.0069 ▲	0.0021	0.0029 ▲	0.0150	0.0181 ▲	45.19% (6/6)
PPT-Ohmnet	HR@10	0.3684	0.4653 ▲	0.3621	0.4695 ▲	0.4421	0.5074 ▲	0.3368	**0.5347** ▲	0.2842	0.3263 ▲	0.4505	0.4968 ▲	24.76% (6/6)
	MAP@10	0.0167	0.0216 ▲	0.0191	0.0230 ▲	0.0220	0.0271 ▲	0.0143	**0.0299** ▲	0.0132	0.0144 ▲	0.0219	0.0250 ▲	31.45% (6/6)
	NDCG@10	0.0478	0.0623 ▲	0.0520	0.0649 ▲	0.0627	0.0752 ▲	0.0428	**0.0810** ▲	0.0376	0.0418 ▲	0.0622	0.0704 ▲	29.61% (6/6)
TFG-Ohmnet	HR@10	0.1750	0.2333 ▲	0.1500	0.1833 ▲	0.1375	**0.2958** ▲	0.1208	0.1625 ▲	0.0583	0.1042 ▲	0.1250	0.1125 ▼	42.39% (5/6)
	MAP@10	0.0113	**0.0147** ▲	0.0079	0.0092 ▲	0.0081	0.0135 ▲	0.0094	0.0101 ▲	0.0037	0.0053 ▲	0.0053	0.0075 ▲	32.16% (6/6)
	NDCG@10	0.0293	0.0385 ▲	0.0229	0.0261 ▲	0.0211	**0.0404** ▲	0.0225	0.0257 ▲	0.0095	0.0151 ▲	0.0158	0.0185 ▲	35.71% (6/6)

the mode-wise gates $\{\rho_n\}$ and $\{\tau_n\}$ act as continuous knobs controlling structural influence: setting $\rho_n, \tau_n = 0$ recovers the identity mapping (no refinement), while $\rho_n, \tau_n = 1$ corresponds to applying full spectral filtering in each view. This design allows the model to adaptively balance fidelity to the original TF factors with the induced topology-driven priors. In particular, the mode-wise gates allow the model to learn an appropriate smoothing strength for each mode. All lemmas and proofs are deferred to Appendix B.3.

4 Experiments

In this section, we evaluate the impact of latent representation stabilization across diverse tensor factorization models and tasks. Our experiments are designed to assess the generality and benefits of DuET by answering the following research questions:

- Q1 **Compatibility of DuET (Section 4.1).** Can DuET consistently improve the performance of various TF methods on retrieval and completion tasks?
- Q2 **Comparative Study of Propagation and Regularization (Section 4.2).** How does DuET compare to GNN-based propagation, graph-regularization, and HNN-based propagation baselines when exploiting the induced dual-view topology?
- Q3 **Ablation Study (Section 4.3).** Does each component of DuET contribute meaningfully to its performance?
- Q4 **Gate Analysis (Section 4.4).** How does adaptive gating control the strength of topology-induced propagation across datasets and tasks?

To further understand how DuET behaves under different sparsity regimes, we additionally analyze performance by partitioning entities into buckets according to their interaction frequency. We also analyze the hyperparameter sensitivity and computational cost of DuET in Appendices C.3 and C.4.

Datasets. We use nine real-world datasets summarized in Table 1. FS-TKY, FS-NYC, and Gowalla are check-in datasets of the form (user, venue, time slot), where the time slot corresponds to one of the 168 hours in a week, obtained by mapping day of week × hour into a single cyclic index (0–167). PPT-Ohmnet represents protein–protein interaction networks across human tissues, whereas TFG-Ohmnet represents protein–function association networks across human tissues. ML-100k is a movie recommendation dataset of the form (user, movie, day, month). Yelp is a recommendation dataset of the form (user, business, time). Bitcoin is a trust network of the form (user, user, time), where the values denote the degree of trust. NIPS is a publication dataset of the form (paper, author, word, year), where the values correspond to word counts.

Tasks and Training Objectives. We evaluate DuET on two representative sparse-tensor tasks: retrieval and completion. For retrieval, we train with a ranking loss (BPR [45]); for completion, we train with a regression loss (MSE).

Metrics. For the retrieval task, we use three ranking metrics for evaluation: HR@k (Hit Rate at k), MAP@k (Mean Average Precision at k), and NDCG@k (Normalized Discounted Cumulative Gain at k). For the completion task, we use standard error-based metrics: RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error).

Baselines. We compare DuET with existing tensor factorization methods: **CP [27]**, **Tucker [51]**, **NCF [20] extension for tensor (NCFT)** which extends a neural network-based collaborative filtering method to a tensor, **CostCo [35]**, **M^2DMTF [11]**, and **NeAT [2]**. We implement all the methods using PyTorch. We additionally compare DuET with representative GNN-based propagation including GAT [52], ChebNet [9], APPNP [15], and LEConv [44], and graph regularization [5] approaches to examine alternative ways of injecting structural information into tensor factorization. After Section 4.1, we adopt DuET + CostCo for subsequent experiments, as it generally achieves better performance than DuET combined with other TF methods (See Tables 2 and 3).

Evaluation Protocol. Observed interactions are randomly split into train/validation/test with a 7:1:2 ratio. For retrieval, models rank candidate tuples given a query entity; for completion, prediction errors are measured on held-out entries. Dataset-specific evaluation details and hyperparameter settings are reported in Appendices C.1 and C.2, respectively. Each method is evaluated over five independent runs, and we report the averaged results.

Table 3: Compatibility of DuET for the completion task. The triangles ▼ and ▲ indicate error reduction and error increase, respectively. A bold text indicates the best performance, and an underline indicates the second best. The last column shows the average reduction% of DuET over six baselines, computed as $(1 - (\text{avg. DuET+6 TF methods}/\text{avg. 6 TF methods})) * 100$, and the number of improved baselines (# of ▼/6).

Setting		CP		Tucker		CostCo		NCFT		M²DMTF		NeAT		Reduction
Dataset	Metric	w/o DuET	w/ DuET	w/o DuET	w/ DuET	w/o DuET	w/ DuET	w/o DuET	w/ DuET	w/o DuET	w/ DuET	w/o DuET	w/ DuET	x.xx% (▼/6)
ML-100k	RMSE	1.3870	0.9789 ▼	1.1482	0.9779 ▼	0.9503	**0.9367** ▼	0.9700	0.9614 ▼	0.9558	0.9566 ▲	0.9599	0.9458 ▼	9.52% (5/6)
	MAE	0.9651	0.7749 ▼	0.8947	0.7733 ▼	0.7521	**0.7407** ▼	0.7717	0.7660 ▼	0.7536	0.7564 ▲	0.7617	0.7476 ▼	6.81% (5/6)
	MAPE	34.39	29.22 ▼	32.37	29.47 ▼	29.02	**28.28** ▼	29.15	29.06 ▼	29.56	29.47 ▼	29.18	29.19 ▲	4.57% (5/6)
Yelp	RMSE	2.9263	1.9878 ▼	2.3838	1.9758 ▼	1.2096	1.1878 ▼	1.2173	1.1949 ▼	**1.1817**	1.2040 ▲	1.2101	1.1839 ▼	13.77% (5/6)
	MAE	2.2173	1.4455 ▼	1.8524	1.4255 ▼	0.9810	0.9603 ▼	1.0053	0.9754 ▼	0.9624	0.9666 ▲	0.9776	**0.9571** ▼	15.82% (5/6)
	MAPE	62.97	47.34 ▼	55.26	46.95 ▼	40.20	**38.97** ▼	40.71	39.06 ▼	39.59	39.45 ▼	39.70	39.54 ▼	9.73% (6/6)
Bitcoin	RMSE	3.5384	3.0285 ▼	3.6005	3.3099 ▼	3.0670	2.9699 ▼	3.0477	3.0219 ▼	3.0407	**2.9568** ▼	3.2395	2.9867 ▼	6.45% (6/6)
	MAE	2.4244	1.8305 ▼	2.4694	2.0094 ▼	1.7920	1.8094 ▲	1.8845	1.8485 ▼	1.7883	**1.7805** ▼	1.9115	1.8375 ▼	9.40% (5/6)
	MAPE	101.51	79.04 ▼	99.84	79.98 ▼	88.75	90.62 ▲	87.15	89.29 ▲	82.62	83.19 ▲	**74.32**	82.50 ▲	5.53% (2/6)
NIPS	RMSE	3.5819	3.5162 ▼	3.6128	3.5117 ▼	3.5571	**3.4491** ▼	3.5615	3.5754 ▲	3.6733	3.6589 ▼	3.5934	3.5319 ▼	1.56% (5/6)
	MAE	1.4592	1.4197 ▼	1.4665	1.4127 ▼	1.4260	**1.3553** ▼	1.4277	1.4284 ▲	1.4515	1.4553 ▲	1.4333	1.4170 ▼	2.02% (4/6)
	MAPE	56.74	54.35 ▼	55.88	53.34 ▼	52.64	**50.78** ▼	54.32	54.00 ▼	52.66	53.96 ▲	54.20	53.92 ▼	1.86% (5/6)

4.1 Compatibility of DuET (Q1)

We evaluate the compatibility of DuET by integrating it with various TF backbones across retrieval and completion tasks.

Retrieval Task. Table 2 shows that DuET consistently boosts performance in retrieval tasks, yielding substantial improvements ranging from **20.45%** to **59.51%** on average. Notably, integrating DuET enables TF methods to achieve either the best or second-best performance across nearly all scenarios, with the sole exception of CP on the FS-NYC dataset. CostCo combined with DuET secures the top performance on four out of five datasets (ranking second only on PPT-Ohmnet). This synergy arises since DuET explicitly denoises the input topology, providing high-quality structural priors that the convolutional architecture of CostCo can effectively leverage to distinguish positive interactions from negatives.

Completion Task. As shown in Table 3, DuET effectively reduces prediction errors in **58 out of 72** cases (80%). In most cases, integrating DuET enables TF methods to achieve either the best or second-best performance. The gains are particularly pronounced for classical multilinear models; for example, on ML-100k, DuET reduces the RMSE of CP and Tucker by **29.4%** (1.3870 → 0.9789) and **14.8%** (1.1482 → 0.9779), respectively. This indicates that DuET injects additional structural context into these models, alleviating their sensitivity to sparsity.

Retrieval vs. Completion Discrepancy: We analyze distinct behaviors between the two tasks. Relative improvements are consistently larger in retrieval (20–60%) than in completion (1–16%). This discrepancy is expected because retrieval is primarily a ranking task: propagation-based smoothing can cluster topologically related entities in the embedding space, which often improves the separability between positive and negative samples and boosts ranking metrics. In contrast, completion involves regression to precise scores. While smoothing provides beneficial regularization and improves generalization, it may also attenuate fine-grained signals that are important for accurate value prediction. Consequently, the gains in RMSE/MAE are more modest than the substantial improvements observed in ranking-based evaluations.

Investigation of Failure Cases: We also analyze specific failure cases indicated by ▼ in retrieval and ▲ in completion. On the completion task, complex non-linear models such as M²DMTF occasionally exhibit increased error. Since M²DMTF applies mode-wise MLPs after the factor extraction stage, the benefits of topology-based regularization can be partially offset in completion tasks. Nevertheless, the performance differences for these models remain marginal in most cases, and DuET provides consistent improvements in the vast majority of scenarios. In retrieval, the slight degradation of CP on FS-NYC may reflect a structural mismatch between CP's low-rank multilinear structure and the strongly correlated, topology-induced priors in this dataset. Although our adaptive gating aims to downweight unhelpful signals, sigmoid-based gates may not fully shut off under stochastic end-to-end optimization. This allows a small residual signal injection that can slightly interfere with the learning process in this sensitive low-rank setting.

Table 4: Comparative study of graph-based operators on CostCo backbone.

		GNN-based Propagation				Graph Reg.		
Metric	Dataset	GAT	Cheb	APPNP	LEConv	Explicit	Spectral	DuET
NDCG@10 (↑)	FS-NYC	0.0102	0.0101	0.0099	0.0086	0.0070	0.0089	**0.0216**
	TFG	0.0202	0.0212	0.0246	0.0144	0.0246	0.0200	**0.0404**
RMSE (↓)	ML-100k	0.9383	0.9436	0.9455	0.9440	0.9489	0.9608	**0.9367**
	Yelp	1.2061	1.1859	1.1832	1.1862	1.1985	**1.1765**	1.1878

Table 5: Comparison with hypergraph neural networks on the CostCo backbone.

		Hypergraph Neural Networks			
Metric	Dataset	HyperConv	HGNN	HNHN	DuET
NDCG@10 (↑)	FS-NYC	0.0103	0.0037	0.0039	**0.0216**
	TFG	0.0185	0.0156	0.0183	**0.0404**
RMSE (↓)	ML-100k	0.9470	0.9462	1.0723	**0.9367**
	Yelp	1.2086	**1.1804**	1.1834	1.1878

4.2 Comparative Study of Propagation and Regularization Operators

GNN propagation and graph regularization. Table 4 compares representative graph-based operators on the CostCo backbone which demonstrates superior compatibility with DuET in most cases. We consider (i) GNN modules [9, 15, 44, 52] operating on dual-view adjacency and (ii) graph regularization baselines under

the dual-view setting, using either explicit adjacency or its spectral counterpart. Overall, DuET consistently outperforms these alternatives on retrieval tasks: it achieves the best NDCG@10 on both FS-NYC and TFG, yielding substantial gains over the strongest GNN baselines and graph-regularized variants. These results show that simply attaching a message-passing operator is insufficient under extreme sparsity, whereas our refinement strategy can better exploit the induced topology as a stable structural prior. On completion tasks (RMSE), DuET remains competitive and attains the best result on ML-100k. On Yelp, spectral regularization achieves the best RMSE, but DuET remains within 1% relative RMSE. These results indicate that operator choice can be dataset-dependent for completion, but DuET provides a robust improvement in general, particularly for retrieval.

The results highlight a practical trade-off between propagation and regularization under extreme sparsity. A GNN performs message passing over neighborhoods induced from sparse co-occurrences; when evidence is scarce, such neighborhoods can be noisy, which may amplify incidental links and lead to unstable updates. Graph regularization is often more stable by emphasizing low-frequency components, but it can also oversmooth and reduce adaptivity to heterogeneous signals. DuET is designed to balance these effects by extracting topology-derived structural priors and injecting them in a controlled manner, yielding consistent gains in retrieval and competitive performance in completion.

Hypergraph neural networks. We additionally compare DuET with representative hypergraph neural networks (HNNs), including HyperConv [3], HGNN [13], and HNHN [10], using the same CostCo backbone and experimental settings. These HNNs operate solely on the incidence view by performing message passing over hyperedges corresponding to observed tuples, without leveraging the marginal-view structures used in DuET. The resulting representations are then fed into the CostCo prediction module. As shown in Table 5, DuET achieves substantially higher performance than HNNs on most datasets, including FS-NYC, TFG, and ML-100k. On Yelp, HGNN achieves the best performance; however, the performance gap between DuET and HGNN is less than 1%. It is worth noting that many hypergraph neural networks are originally developed for feature-based inputs. Our setting focuses on refining learnable latent representations of tensor factorization, which corresponds to a different regime. These results suggest that hyperedge-aware propagation alone is insufficient for sparse-TF refinement; effective refinement additionally requires denoising noisy tuple-induced topology and controlling how structural signals are injected into learnable latent factors.

4.3 Ablation Study

We study how the choice of our component affects performance of DuET with CostCo on both tasks.

Dual-View vs. Marginal-View vs. Incidence-View We evaluate the performance of *incidence-view only*, *marginal-view only*, and *dual-view* (incidence + marginal) topology induction, denoted by **I**, **M**, and **D**, respectively. Table 6 shows that our dual view (**D**) with adaptive gating provides the most reliable performance by dynamically balancing the two views, thereby reducing the sensitivity to single-view choices. While **D** achieves the best average performance, the stronger single view between **M** and **I** depends on the dataset, task, and backbone.

Table 6: Ablation results. We report NDCG@10 for Retrieval and RMSE for Completion.

		Retrieval				Completion		
Backbone	**Data**	**I**	**M**	**D**	**Data**	**I**	**M**	**D**
CostCo	FS-NYC	0.020	0.009	**0.021**	ML-100k	0.943	0.947	**0.936**
	TFG	0.037	0.019	**0.040**	Yelp	**1.184**	1.186	1.187

Table 7: Performance comparison between Non-spectral and Spectral approaches across Retrieval and Completion tasks.

		NDCG@10			RMSE	
Backbone	**Data**	**Non-Spec.**	**Spectral**	**Data**	**Non-Spec.**	**Spectral**
CostCo	FS-NYC	0.018	**0.021**	ML-100k	0.943	**0.936**
	TFG	0.030	**0.040**	Yelp	1.188	**1.187**

Table 8: Performance comparison between Fixed and Adaptive gating across Retrieval and Completion tasks.

		NDCG@10			RMSE	
Backbone	**Data**	**Fixed**	**Adaptive**	**Data**	**Fixed**	**Adaptive**
CostCo	FS-NYC	0.014	**0.021**	ML-100k	0.952	**0.936**
	TFG	0.033	**0.040**	Yelp	**1.185**	1.187

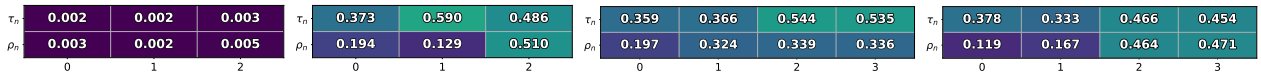
Spectral vs. Non-spectral. Table 7 compares adjacency-based propagation (Non-Spec.), instantiated from Eq. (5), against spectrally denoised propagation (Spectral). We find that spectral denoising consistently boosts NDCG@10 on both datasets, and for the completion, spectral denoising is also beneficial for CostCo.

Adaptive gating vs. Fixed gating. Table 8 examines whether adaptive fusion via learnable gating is necessary. Adaptive gating yields consistent gains in the retrieval task, highlighting that a static mixing coefficient can under-utilize structural priors. For the completion task, adaptive gating improves or matches performance in most cases, while one mild exception is CostCo on Yelp where Fixed gating is marginally better (1.185 vs. 1.187). Given the very small gap, the difference may be attributable to stochastic optimization noise (e.g., random initialization). Taken together, adaptive gating is crucial for retrieval robustness and typically helps completion, with rare cases where a simpler fixed gate remains competitive.

4.4 Analysis of Learned Gates: τ_n and ρ_n

We investigate the learned gate parameters τ_n and ρ_n on the FS-NYC (retrieval) and ML-100k (completion) datasets using CP and CostCo backbones. Figure 4 reports the learned mixing gates across four settings (retrieval/completion × CP/CostCo). Overall, τ_n provides consistently active tuple-level coupling, while ρ_n serves as a mode-adaptive filter that emphasizes temporal locality and suppresses noisy entity-centric co-occurrences, with backbone expressiveness determining how effectively this selectivity translates into gains.

τ_n vs. ρ_n. τ_n remains relatively balanced across modes, indicating stable incidence-view fusion in Eq. (10). This is because τ_n controls the strength of the tuple-level inter-mode coupling, which needs to remain consistently active to preserve inter-mode dependencies. In contrast, ρ_n is mode-selective in Eq. (11), taking larger values for temporal modes and smaller values for the other modes (e.g., user/item). This indicates that DuET exploits strong temporal locality for denoising while downweighting noisier co-occurrence



(a) DuET + CP on FS-NYC (b) DuET + CostCo on FS-NYC (c) DuET + CP on ML-100k (d) DuET + CostCo on ML-100k

Figure 4: τ_n and ρ_n for CP and CostCo on the retrieval and completion tasks.

neighborhoods in non-temporal modes to avoid over-smoothing and maintain discriminative representations.

CP vs. CostCo. We further examine how the underlying TF backbone affects the learned gates. As shown in Figure 4(a), the gates under CP stay close to zero, indicating that the model largely suppresses refinement signals in this regime. Ideally, the refinement contribution should vanish to recover the original CP baseline; however, the sigmoid parameterization can leave a small non-zero leakage, resulting in residual signal injection that may slightly interfere with optimization. In contrast, DuET + CostCo (Figure 4(b)) yields substantially larger and more adaptive gate activations, suggesting that CostCo's more expressive (non-linear) interaction modeling better accommodates structural refinement and helps the gates distinguish informative structure from noise. For ML-100k on the completion task, CP and CostCo exhibit the same τ_n-balanced / ρ_n-selective pattern (temporal ↑, user/item ↓).

5 Related Works

Tensor factorization methods. Many works have developed tensor factorization methods for analyzing higher-order tensors. CP (CANDECOMP/PARAFAC) decomposition [6, 19, 27] factorizes a given tensor into factor matrices which construct rank-one tensors as described in [30]. Tucker decomposition [51] decomposes a given tensor into factor matrices and core tensor. CostCo [35] is a CNN-based tensor factorization model that captures nonlinear interactions between factors. While NTF [55] is an RNN-based model designed to capture sequential patterns in temporal tensors, we focus on a more general-purpose framework that enhances diverse TF structures without assuming specific temporal dependencies. M^{2}DMTF [11] employs a nonlinear transformation for its factor matrices. We can view M^{2}DMTF as consisting of an MLP encoder and predictor that is the same as the predictor of Tucker decomposition; we experimentally show the compatibility of DuET with the MLP encoder of M^{2}DMTF. NeAT [2] is a non-linear tensor factorization method with interpretability. t-SVD [59] follows the t-product formulation over frontal slices rather than mode-wise embeddings, and is therefore outside the scope of DuET. Unlike these models that focus on specific decomposition structures, DuET is a model-agnostic framework that enhances existing TF methods by providing denoised latent representations through tensor-to-topology induction.

Tensor factorization with relational and structural priors. Many works have explored incorporating relational and structural priors into tensor factorization (TF). One line of research studies the interaction between TF and graph neural networks (GNNs), often using TF to improve the efficiency or expressiveness of GNN architectures [21, 26, 42, 47, 56]. These approaches primarily treat TF as a means to enhance graph representation learning. Another line of work introduces structural priors into TF using auxiliary relational cues, such as side information [17, 39, 49]. Related to this direction, several studies induce structural priors directly from tensors by constructing mode-specific affinities in the observation space [38, 43, 58]. However, these approaches are often coupled with specific constraints (e.g., non-negativity) or decomposition structures (e.g., CP or Tensor-Ring). In contrast, DuET induces structural priors solely from a sparse tensor without assuming specific constraints, making it applicable to various TF methods.

Tensor-based applications. Many previous works [4, 32, 37] developed TF-based methods for knowledge base completion. For instance, Lacroix et al. [31] developed a tensor factorization-based method for temporal knowledge completion. In addition, tensor factorization has been widely used for higher-order recommendation [7, 8, 12, 46] and time-range factor analysis [22, 25, 41]. Many works [1, 23, 36, 48] have focused on developing tensor completion methods that predict specific values of unobserved entries accurately. While we do not directly compare against these task-specific models, DuET serves as a general-purpose framework that stabilizes representation learning. Furthermore, DuET is potentially complementary to these specialized methods, as it refines latent embeddings in a task-agnostic manner and can be integrated with task-specific TF objectives, such as those incorporating temporal or domain-specific constraints.

6 Conclusion

In this paper, we propose DuET, a model-agnostic enhancement adapter that refines latent representations of entities for a wide range of tensor factorization (TF) methods. DuET induces dual-view structural priors directly from sparse tensor tuples, capturing both within-mode and cross-mode structural patterns. These priors are denoised via per-view spectral filtering and selectively injected into TF embeddings through mode-wise gated residual refinement, leading to more stable and reliable representations. Extensive experiments demonstrate that DuET consistently improves diverse TF backbones across retrieval and completion tasks, highlighting its generality and plug-and-play nature. Future work includes extending DuET to dynamic and streaming tensor settings, where structural priors must be continuously updated as new interactions arrive.

Acknowledgments

This work is supported by NSF (2316233 and 2416070). The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25490900). It was also supported by the DGIST Start-up Fund Program of the Ministry of Science and ICT (2026010244).

References

[1] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. 2011. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* 106, 1 (2011), 41–56.

[2] Dawon Ahn, Uday Singh Saini, Evangelos E Papalexakis, and Ali Payani. 2024. Neural Additive Tensor Decomposition for Sparse Tensors. In *33rd ACM International Conference on Information and Knowledge Management*.

[3] Song Bai, Feihu Zhang, and Philip HS Torr. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognition* 110 (2021), 107637.

[4] Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019. TuckER: Tensor Factorization for Knowledge Graph Completion. In *EMNLP-IJCNLP*. Association for Computational Linguistics, 5184–5193.

[5] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research* 7, 11 (2006).

[6] J Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* 35, 3 (1970), 283–319.

[7] Huiyuan Chen and Jing Li. 2020. Neural tensor model for learning multi-aspect factors in recommender systems. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 2020.

[8] Dongjin Choi, Jun-Gi Jang, and U Kang. 2019. S3 CMTF: Fast, accurate, and scalable method for incomplete coupled matrix-tensor factorization. *PloS one* 14, 6 (2019), e0217316.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* 29 (2016).

[10] Yihe Dong, Will Sawin, and Yoshua Bengio. 2020. Hnhn: Hypergraph networks with hyperedge neurons. *arXiv preprint arXiv:2006.12278* (2020).

[11] Jicong Fan. 2022. Multi-mode deep matrix and tensor factorization. In *international conference on learning representations*.

[12] Xiaomin Fang, Rong Pan, Guoxiang Cao, Xiuqiang He, and Wenyuan Dai. 2015. Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 29.

[13] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 3558–3565.

[14] Kunihiko Fukushima. 1975. Cognitron: A self-organizing multilayered neural network. *Biological cybernetics* 20, 3-4 (1975), 121–136.

[15] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997* (2018).

[16] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*. PMLR, 1263–1272.

[17] Yu Guan, Shuyu Dong, Bin Gao, P-A Absil, and François Glineur. 2020. Alternating minimization algorithms for graph regularized tensor completion. *arXiv preprint arXiv:2008.12876* (2020).

[18] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[19] Richard A Harshman et al. 1970. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multimodal factor analysis. (1970).

[20] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[21] Chenqing Hua, Guillaume Rabusseau, and Jian Tang. 2022. High-order pooling for graph neural networks with tensor decomposition. *Advances in Neural Information Processing Systems* 35 (2022), 6021–6033.

[22] Jun-Gi Jang and U Kang. 2021. Fast and memory-efficient tucker decomposition for answering diverse time range queries. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 725–735.

[23] Jun-Gi Jang, Jeongyoung Lee, Jiwon Park, and U Kang. 2022. Accurate PARAFAC2 decomposition for temporal irregular tensors with missing values. In *2022 IEEE international conference on big data (Big Data)*. IEEE, 982–991.

[24] Jun-Gi Jang, Jeongyoung Lee, Yong-chan Park, and U Kang. 2023. Fast and Accurate Dual-Way Streaming PARAFAC2 for Irregular Tensors - Algorithm and Application. In *KDD*. 879–890.

[25] Jun-Gi Jang, Yong-chan Park, and U Kang. 2024. Fast and accurate parafac2 decomposition for time range queries on irregular tensors. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 962–972.

[26] Baoyu Jing, Hanghang Tong, and Yada Zhu. 2021. Network of tensor time series. In *Proceedings of the Web Conference 2021*. 2425–2437.

[27] Henk AL Kiers. 2000. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society* 14, 3 (2000), 105–122.

[28] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[29] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[30] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009), 455–500.

[31] Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. Tensor Decompositions for Temporal Knowledge Base Completion. In *ICLR*. OpenReview.net.

[32] Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. 2018. Canonical tensor decomposition for knowledge base completion. In *International Conference on Machine Learning*. PMLR, 2863–2872.

[33] Richard B Lehoucq, Danny C Sorensen, and Chao Yang. 1998. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM.

[34] Xutao Li, Gao Cong, Xiao-Li Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. 2015. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 433–442.

[35] Hanpeng Liu, Yaguang Li, Michael Tsang, and Yan Liu. 2019. Costco: A neural tensor completion model for sparse tensors. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 324–334.

[36] Yuanyuan Liu, Fanhua Shang, Wei Fan, James Cheng, and Hong Cheng. 2014. Generalized higher-order orthogonal iteration for tensor decomposition and completion. *Advances in Neural Information Processing Systems* 27 (2014).

[37] Yu Liu, Quanming Yao, and Yong Li. 2020. Generalizing tensor decomposition for n-ary relational knowledge bases. In *Proceedings of the web conference 2020*. 1104–1114.

[38] Hayato Maki, Hiroki Tanaka, Sakriani Sakti, and Satoshi Nakamura. 2018. Graph regularized tensor factorization for single-trial EEG analysis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 846–850.

[39] Atsuhiro Narita, Kohei Hayashi, Ryota Tomioka, and Hisashi Kashima. 2012. Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery* 25, 2 (2012), 298–324.

[40] Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. 2017. CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*.

[41] Ruizhong Qiu, Jun-Gi Jang, Xiao Lin, Lihui Liu, and Hanghang Tong. 2024. TUCKET: A Tensor Time Series Data Structure for Efficient and Accurate Factor Analysis over Time Ranges. *Proc. VLDB Endow.* 17, 13 (2024), 4746–4759. https://doi.org/10.14778/3704965.3704980

[42] Shenghao Qiu, Chunwei Xia, and Zheng Wang. 2025. Accelerating Tensor-train Decomposition on Graph Neural Networks. In *2025 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 130–141.

[43] Yuning Qiu, Guoxu Zhou, Yanjiao Wang, Yu Zhang, and Shengli Xie. 2020. A generalized graph regularized non-negative tucker decomposition framework for tensor data representation. *IEEE transactions on cybernetics* 52, 1 (2020), 594–607.

[44] Ekagra Ranjan, Soumya Sanyal, and Partha Talukdar. 2020. Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 5470–5477.

[45] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[46] Steffen Rendle and Lars Schmidt-Thieme. 2010. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining*. 81–90.

[47] Nícolas Roque dos Santos, Dawon Ahn, Diego Minatel, Alneu de Andrade Lopes, and Evangelos E Papalexakis. 2025. Multi-view Graph Condensation via Tensor Decomposition. *arXiv preprint arXiv:2508.14330* (2025).

[48] Shaden Smith, Jongsoo Park, and George Karypis. 2016. An exploration of optimization algorithms for high performance tensor completion. In *SC'16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 359–371.

[49] Koh Takeuchi, Yoshinobu Kawahara, and Tomoharu Iwata. 2017. Structurally regularized non-negative tensor factorization for spatio-temporal pattern discoveries. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 582–598.

[50] Giorgio Tomasi and Rasmus Bro. 2005. PARAFAC and missing values. *Chemometrics and Intelligent Laboratory Systems* 75, 2 (2005), 163–180.

[51] Ledyard R Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31, 3 (1966), 279–311.

[52] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[53] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser,

Jonathan Bright, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* 17, 3 (2020), 261–272.
[54] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.
[55] Xian Wu, Baoxu Shi, Yuxiao Dong, Chao Huang, and Nitesh V Chawla. 2019. Neural tensor factorization for temporal interaction learning. In *Proceedings of the Twelfth ACM international conference on web search and data mining*. 537–545.
[56] Chunxing Yin, Da Zheng, Israt Nisa, Christos Faloutsos, George Karypis, and Richard Vuduc. 2022. Nimble gnn embedding with tensor-train decomposition. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2327–2335.
[57] Jaemin Yoo, Meng-Chieh Lee, Shubhranshu Shekhar, and Christos Faloutsos. 2023. Less is more: Slimg for accurate, robust, and interpretable graph mining. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3128–3139.
[58] Yuyuan Yu, Guoxu Zhou, Ning Zheng, Yuning Qiu, Shengli Xie, and Qibin Zhao. 2022. Graph-regularized non-negative tensor-ring decomposition for multiway representation learning. *IEEE Transactions on Cybernetics* 53, 5 (2022), 3114–3127.
[59] Zemin Zhang and Shuchin Aeron. 2016. Exact tensor completion using t-SVD. *IEEE Transactions on Signal Processing* 65, 6 (2016), 1511–1526.
[60] Hao Zhu and Piotr Koniusz. 2020. Simple spectral graph convolution. In *International conference on learning representations*.

Appendix

A Symbol Table

Table 9 summarizes the frequent symbols used in this paper.

Table 9: Frequently used notations.

Symbol	Description
$\mathcal{X}$	sparse N-th order tensor
Ω_o	observed tuple set
I_n	number of entities in mode n
$\mathbf{U}^{(n)}$	base TF factor matrix for mode n
$\mathbf{V}$	concatenated embedding matrix (Eq. (7))
$\mathbf{W}$	sparse binary view matrix
$\mathbf{G}(\mathbf{W})$	diagonal-free co-occurrence adjacency (Eq. (4))
$\mathbf{M}(\mathbf{W})$	symmetric normalized operator (Eq. (5))
q	number of retained eigenpairs
$\mathbf{P}(\mathbf{W})$	denoised spectral operator (Eq. (6))
$\mathbf{F}_{\text{inc}}$	incidence-view denoised filter
$\mathbf{F}_{\text{marg}}$	marginal-view denoised filter
$\mathbf{\Phi}_\tau, \mathbf{\Phi}_\rho$	incidence-/marginal-view gate matrices
$\mathbf{Z}, \mathbf{Y}$	incidence-/marginal-view stage outputs
L	number of refinement steps
$f(\cdot;\theta)$	base TF predictor/scorer

B Lemmas, Theorems, and Proofs

B.1 Proof of Proposition 1

Proof of Proposition 1. We show that one matrix–vector product with $\mathbf{M}(\mathbf{W})$ costs $O(\text{nnz}(\mathbf{W}))$ time. Let $\mathbf{s} = \mathbf{D}^{-1/2}\mathbf{v}$. Using $\mathbf{G}(\mathbf{W}) = \mathbf{W}\mathbf{W}^\top - \text{Diag}(\text{diag}(\mathbf{W}\mathbf{W}^\top))$, we can write

$$\mathbf{M}(\mathbf{W})\mathbf{v} = \mathbf{D}^{-1/2}\Big(\mathbf{W}(\mathbf{W}^\top\mathbf{s}) - \text{diag}(\mathbf{W}\mathbf{W}^\top) \odot \mathbf{s}\Big).$$

The two sparse multiplies $\mathbf{W}^\top\mathbf{s}$ and $\mathbf{W}(\mathbf{W}^\top\mathbf{s})$ each take $O(\text{nnz}(\mathbf{W}))$ time, while the diagonal removal term and degree normalization are elementwise and cost $O(N)$. Moreover, $\text{diag}(\mathbf{W}\mathbf{W}^\top)$ and $\mathbf{D}^{-1/2}$ can be precomputed once using sparse operations in $O(\text{nnz}(\mathbf{W}))$ time and stored as N-vectors. Therefore, each matrix-vector multiplication is $O(\text{nnz}(\mathbf{W}))$ time up to lower-order terms, and an iterative eigensolver using T matvecs runs in $O(T \cdot \text{nnz}(\mathbf{W}))$ time. The memory is $O(\text{nnz}(\mathbf{W}))$ for storing $\mathbf{W}$ plus $O(Nq)$ for the top-q eigenvectors, yielding $O(\text{nnz}(\mathbf{W}) + Nq)$ space. □

B.2 Proof of Proposition 2

Proof. Each propagation step consists of an incidence-view update (Eq. (10)) and a marginal-view update (Eq. (11)). Both stages apply spectrally denoised filters using the implicit operator $\mathbf{P}(\mathbf{W})$, implemented as $\mathbf{P}(\mathbf{W})\mathbf{V} = \mathbf{U}_q(g(\mathbf{\Lambda}_q)(\mathbf{U}_q^\top\mathbf{V}))$. For $\mathbf{V} \in \mathbb{R}^{V_{\text{total}} \times K}$, computing $\mathbf{U}_q^\top\mathbf{V}$ and multiplying back by $\mathbf{U}_q$ each require $O(V_{\text{total}}qK)$ time. Diagonal gating and residual mixing incur only linear-time overhead. The marginal-view operator is block-diagonal, and summing the per-mode costs $O(I_n qK)$ over all modes also yields $O(V_{\text{total}}qK)$. Thus, one propagation step runs in $O(V_{\text{total}}qK)$ time, and repeating it for L steps gives $O(L \cdot V_{\text{total}} \cdot q \cdot K)$. □

B.3 Proofs for Proposition 3

Setup and notation. We prove Proposition 3 for the two-stage update in Eqs. (10)–(11). $\mathbf{F}$ denotes a symmetric spectral filter whose eigenvalues lie in $[0, 1]$; equivalently, $\mathbf{F} = \mathbf{U}\,\text{diag}(\mu)\mathbf{U}^\top$ with $\mu_i \in [0, 1]$, hence $\|\mathbf{F}\|_2 \le 1$. The gates satisfy $0 \preceq \mathbf{\Phi}_\rho, \mathbf{\Phi}_\tau \preceq \mathbf{I}$, implying $\|\mathbf{\Phi}_\tau^{1/2}\|_2 \le 1$. For the analysis, we rewrite the two stages as linear operators. The incidence-view stage in Eq. (10) can be expressed as

$$\mathbf{Z} = C_\tau(\mathbf{V}) \quad \text{with} \quad C_\tau(\mathbf{V}) = (\mathbf{I} - \mathbf{\Phi}_\tau)\mathbf{V} + \mathbf{\Phi}_\tau^{1/2}\mathbf{F}\mathbf{\Phi}_\tau^{1/2}\mathbf{V},$$

and the marginal-view stage in Eq. (11) as

$$\mathbf{Y} = \mathbf{M}_\rho\mathbf{Z} \quad \text{with} \quad \mathbf{M}_\rho = (\mathbf{I} - \mathbf{\Phi}_\rho) + \mathbf{\Phi}_\rho\mathbf{F}.$$

For the full model, $\mathbf{F}$ and $\mathbf{\Phi}_\rho$ are block-diagonal across modes; the arguments below apply block-wise.

Lemma 1 (Incidence-view stage is non-expansive). *Let $C_\tau(\mathbf{V}) = (\mathbf{I} - \mathbf{\Phi}_\tau)\mathbf{V} + \mathbf{\Phi}_\tau^{1/2}\mathbf{F}\mathbf{\Phi}_\tau^{1/2}\mathbf{V}$ with $0 \preceq \mathbf{\Phi}_\tau \preceq \mathbf{I}$ and symmetric $\mathbf{F}$ whose eigenvalues lie in $[0, 1]$. Then $\|C_\tau(\mathbf{V})\|_F \le \|\mathbf{V}\|_F$ for all $\mathbf{V}$. Equivalently, in Eq. (10), $\|\mathbf{Z}\|_F \le \|\mathbf{V}\|_F$.*

Proof. The mapping is linear and can be written as $C_\tau(\mathbf{V}) = \mathbf{T}\mathbf{V}$ where

$$\mathbf{T} = (\mathbf{I} - \mathbf{\Phi}_\tau) + \mathbf{\Phi}_\tau^{1/2}\mathbf{F}\mathbf{\Phi}_\tau^{1/2}.$$

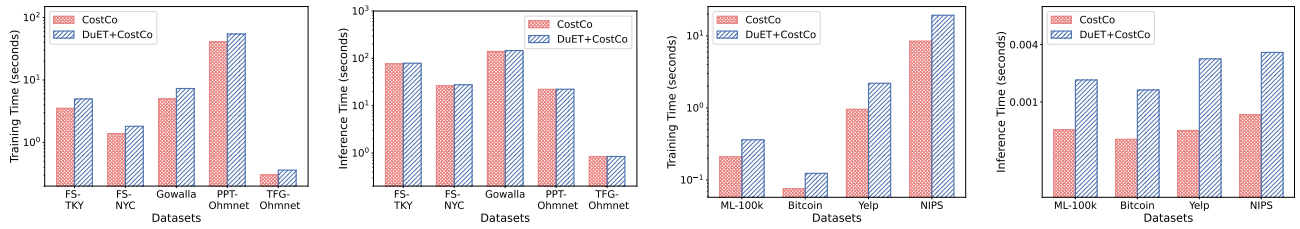
Note that $\mathbf{T}$ is symmetric. For any vector $\mathbf{x}$, let $\mathbf{y} = \mathbf{\Phi}_\tau^{1/2}\mathbf{x}$. Using $\|\mathbf{F}\|_2 \le 1$, we have $\mathbf{y}^\top\mathbf{F}\mathbf{y} \le \|\mathbf{y}\|_2^2$. Therefore,

$$\mathbf{x}^\top\mathbf{T}\mathbf{x} = \mathbf{x}^\top(\mathbf{I} - \mathbf{\Phi}_\tau)\mathbf{x} + \mathbf{y}^\top\mathbf{F}\mathbf{y} \le \|\mathbf{x}\|_2^2 - \|\mathbf{y}\|_2^2 + \|\mathbf{y}\|_2^2 = \|\mathbf{x}\|_2^2.$$

Moreover, since $\mathbf{I} - \mathbf{\Phi}_\tau \succeq 0$ and $\mathbf{\Phi}_\tau^{1/2}\mathbf{F}\mathbf{\Phi}_\tau^{1/2} \succeq 0$, we have $\mathbf{T} \succeq 0$. Thus all eigenvalues of $\mathbf{T}$ lie in $[0, 1]$, and hence $\|\mathbf{T}\|_2 \le 1$. Hence, for any matrix $\mathbf{V}$, $\|C_\tau(\mathbf{V})\|_F = \|\mathbf{T}\mathbf{V}\|_F \le \|\mathbf{T}\|_2\|\mathbf{V}\|_F \le \|\mathbf{V}\|_F$. □

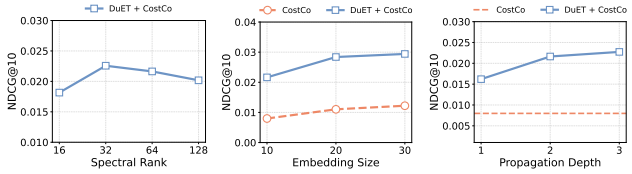
Lemma 2 (Marginal-view mixer is non-expansive). *Let $\mathbf{M}_\rho = (\mathbf{I} - \mathbf{\Phi}_\rho) + \mathbf{\Phi}_\rho\mathbf{F}$ with $0 \preceq \mathbf{\Phi}_\rho \preceq \mathbf{I}$ and symmetric $\mathbf{F}$ with eigenvalues in $[0, 1]$. Then $\|\mathbf{M}_\rho\|_2 \le 1$. Consequently, $\|\mathbf{M}_\rho\mathbf{Z}\|_F \le \|\mathbf{Z}\|_F$ for all $\mathbf{Z}$.*

Proof. In the mode-wise setting used in our model, $\mathbf{F}$ and $\mathbf{\Phi}_\rho$ are block-diagonal with the same block partition, and within each block $\mathbf{\Phi}_\rho = \rho\mathbf{I}$. Thus each block of $\mathbf{M}_\rho$ is symmetric and can be analyzed independently. For a single block with scalar gate $\rho \in [0, 1]$, $\mathbf{M}_\rho = (1 - \rho)\mathbf{I} + \rho\mathbf{F}$ shares eigenvectors with $\mathbf{F}$, and its eigenvalues are $\nu_i = (1 - \rho) + \rho\mu_i \in [1 - \rho, 1] \subseteq [0, 1]$. Thus $\|\mathbf{M}_\rho\|_2 = \max_i |\nu_i| \le 1$, and $\|\mathbf{M}_\rho\mathbf{Z}\|_F \le \|\mathbf{M}_\rho\|_2\|\mathbf{Z}\|_F$. □



(a) Training time on retrieval (b) Inference time on retrieval (c) Training time on completion (d) Inference time on completion

Figure 5: Training and inference time for CostCo and DuET + CostCo on retrieval and completion tasks.



(a) Spectral Rank q (b) Embedding Size r (c) Propagation Depth L

Figure 6: Hyperparameter sensitivity of DuET +CostCo on the FS-NYC dataset (the retrieval task).

Theorem 1 (Two-stage refinement is non-expansive and preserves Frobenius balls). *Let $T(\mathbf{V}) = \mathbf{M}_\rho\, C_\tau(\mathbf{V})$. Then $\|T(\mathbf{V})\|_F \le \|\mathbf{V}\|_F$ for all $\mathbf{V}$. In particular, $\Omega_C = \{\mathbf{V} : \|\mathbf{V}\|_F \le C\}$ is invariant under T, i.e., $T(\Omega_C) \subseteq \Omega_C$.*

Proof. By Lemmas 1 and 2,

$$\|T(\mathbf{V})\|_F = \|\mathbf{M}_\rho\, C_\tau(\mathbf{V})\|_F \le \|\mathbf{M}_\rho\|_2\, \|C_\tau(\mathbf{V})\|_F \le \|\mathbf{V}\|_F.$$

If $\|\mathbf{V}\|_F \le C$, then $\|T(\mathbf{V})\|_F \le C$, proving invariance of Ω_C. □

Remark. In DuET, $\mathbf{F}_{\text{marg}}$ is block-diagonal across modes and $\mathbf{\Phi}_\rho = \text{blkdiag}(\rho_1 \mathbf{I}_{I_1}, \ldots, \rho_N \mathbf{I}_{I_N})$. Thus Lemma 2 applies block-wise. Similarly, $\mathbf{\Phi}_\tau = \text{blkdiag}(\tau_1 \mathbf{I}_{I_1}, \ldots, \tau_N \mathbf{I}_{I_N})$ satisfies $0 \preceq \mathbf{\Phi}_\tau \preceq \mathbf{I}$, so Lemma 1 directly implies the contraction in Eq. (10).

C Experimental Settings & Results

C.1 Evaluation Procedure

We randomly split observed interactions into train, validation, and test data with the ratio 7:1:2. For the completion task, the trained model predicts the values of the entries corresponding to the interactions in the test set, and the prediction errors are measured against the ground-truth values. For the retrieval task, given a query entity (e.g., a user), the model ranks candidate combinations of the remaining dimensions (e.g., location and time) based on the predicted scores, and the ranking quality is evaluated against the ground-truth interactions in the test set. For the FS-TKY, FS-NYC, and Gowalla datasets, given a user query, the TF model predicts the top-k (location, time) pairs. For the PPT-Ohmnet and TFG-Ohmnet datasets, given a tissue query, the TF model predicts the top-k (protein, protein) pairs and (function, gene) pairs, respectively.

C.2 Hyperparameter Setting

We set the batch size, the number of epochs, and the embedding size of entities to 2048, 20, and 10, respectively. We initially explore several learning rates ($[10^{-1}, 10^{-2}, 10^{-3}]$) and weight decays ($[0, 10^{-3}, 10^{-5}]$). Based on preliminary results showing that a learning rate of 10^{-2} and a weight decay of 0 generally perform well, we adopt these settings for all subsequent experiments. We learn all models using the Adam optimizer [28]. We perform $l = 2$ layer propagation in DuET, and use a ReLU function [14] as an activation function. We evaluate the performance of each model five times and report the average results.

C.3 Hyperparameter Sensitivity

We investigate the hyperparameter sensitivity for the spectral rank q, embedding size r, and propagation depth L.

Spectral rank q. We examine DuET with CostCo by varying $q \in \{16, 32, 64, 128\}$ on the FS-NYC dataset. As shown in Figure 6(a), the performance peaks at $q = 32$ and remains comparable at $q = 64$, whereas it slightly degrades at $q = 16$ and 128. This implies that too small or too large a spectral rank can hurt performance by underfitting structural signals or introducing noisy components.

Embedding size r. We investigate DuET with CostCo by varying $r \in \{10, 20, 30\}$. Figure 6(b) shows the performance of DuET with CostCo and base CostCo as r varies. For the retrieval task, the performance gap slightly increases as r grows; DuET with CostCo and base CostCo benefit from larger r. DuET allows the model to leverage growing representational capacity and learn denoised latent vectors.

Propagation depth L. We evaluate DuET with CostCo by varying $L \in \{1, 2, 3\}$ on the FS-NYC dataset. In Figure 6(c), performance increases as L grows. The performance significantly improves when L increases from 1 to 2, demonstrating the importance of capturing multi-hop dependencies. However, the performance gain becomes marginal as L further increases from 2 to 3.

C.4 Computational Cost of DuET Adapter

We evaluate the impact of the DuET adapter on training and inference time. Figure 5 shows that DuET +CostCo introduces only marginal overhead on retrieval tasks, where the dominant cost comes from evaluating many candidate tuples for top-k ranking. For completion, training overhead remains modest, while inference overhead is more visible because the baseline inference time is already extremely small. Even in this lightweight regime, the absolute latency remains sub-millisecond (e.g., < 0.004 seconds), making the overhead practically acceptable relative to the accuracy gains of DuET.