High-Order Contrastive Learning with Fine-grained Comparative Levels for Sparse Ordinal Tensor Completion

Yu Dai[*1] **Junchen Shen**[*1] **Zijie Zhai**[*1] **Danlin Liu**[*1] **Jingyang Chen**[*2] **Yu Sun**[3] **Ping Li**[4] **Jie Zhang**[2] **Kai Zhang**[1]

Abstract

Contrastive learning is a powerful paradigm for representation learning with wide applications in vision and NLP, but how to extend its success to high-dimensional tensors remains a challenge. This is because tensor data often exhibit high-order mode-interactions that are hard to profile and with negative samples growing combinatorially fast; besides, many real-world tensors have ordinal entries that necessitate more delicate comparative levels. We propose High-Order Contrastive Tensor Completion (HOCTC) to extend contrastive learning to sparse ordinal tensor regression. HOCTC employs a novel attention-based strategy with query-expansion to capture high-order mode interactions even in case of very limited tokens, which transcends beyond second-order learning scenarios. Besides, it extends two-level comparisons (positive-vs-negative) to fine-grained contrast-levels using ordinal tensor entries as a natural guidance. Efficient sampling scheme is proposed to enforce such delicate comparative structures, generating comprehensive self-supervised signals for high-order representation learning. Experiments show that HOCTC has promising results in sparse tensor completion in traffic/recommender applications.

1. Introduction

Self-supervised learning techniques, in particular contrastive learning, have emerged as a powerful paradigm for unsupervised representation learning with remarkable success in computer vision (He et al., 2020; Chen et al., 2020a), speech recognition (Kharitonov et al., 2021) and natural language processing (Gao et al., 2021; Chuang et al., 2022). The main idea is to push similar instances together in the feature space while pushing apart those dissimilar or irrelevant ones. In many domains, such positive/negative sample relation can be specified conveniently through simple rules, which allows representation learning to be performed in an unsupervised manner for large datasets.

There are a number of notable methods for contrastive learning. InstDisc (Wu et al., 2018) pioneers the use of instance-based discrimination as a pretext task. CMC (Tian et al., 2020) further uses multiple views of an image as positive samples, and those from distinct images as the negative samples, which enhances the discrimination. MoCo (He et al., 2020) increases negative samples through momentum contrast and a query encoder. PIRL and SimCLR (Misra & Maaten, 2020; Chen et al., 2020a) use more sophisticated strategies for selecting positive/negative samples, like jigsaw augmentation or random data augmentation (cropping, resizing, and re-coloring). In BYOL (Grill et al., 2020), a dual-network is used to alleviate the reliance on negative pairs. Besides, contrastive learning has also been applied successfully in natural language processing (Gao et al., 2021; Wang et al., 2021a), recommendation systems (Zhou et al., 2020; Liu et al., 2021; Xie et al., 2022; Chen et al., 2022), and graph learning (Veličković et al., 2019; Zhu et al., 2020; 2021; Hassani & Khasahmadi, 2020).

Current contrastive learning methods mainly exploit second-order relation, i.e., the proximity between a pair of entities like image patches (Chen et al., 2020a), time series (Van den Oord et al., 2018), or two variations of a sentence (Gao et al., 2021). In many applications, we have high-order (multi-way) interacting relations encoded by tensors. For example, in traffic monitoring, a three-mode *sensor-road-time* tensor describes the traffic flow across various sensors on different roads over time. In recommendation, a *user-item-tag* tensor describes the tag that a user assigns to an item. Modelling such high-order interactions among the modes of a tensor for tensor decomposition or tensor completion is useful in

[*]Equal contribution [1]School of Computer Science and Technology, East China Normal University, Shanghai, China. [2]Institute of Science and Technology for Brain Inspired Intelligence, Fudan University, Shanghai, China. [3]Indeed Inc., Sunnyvale, United States. [4]Southwest Petroleum University, Chengdu, China.. Correspondence to: Jie Zhang <jzhang080@gmail.com>, Kai Zhang <kzhang@cs.ecnu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

image and vision (Wu et al., 2009; Cao et al., 2016; Tao et al., 2017; Brandoni & Simoncini, 2020), social networks (Rettinger et al., 2012; Fernandes et al., 2021), and recommender systems (Taneja & Arora, 2018; Chen & Li, 2019).

The abundant availability and substantial volume of tensor data represent a valuable resource for self-supervised learning algorithms, enabling unsupervised representation learning with minimal human effort. This raises a natural question: *Can the success of contrastive learning be extended to the task of tensor completion?* This extension remains challenging for the following reasons.

First, capturing high-order nonlinear interactions among the modes of a tensor is challenging. Although advances in self-attention (Vaswani et al., 2017; Song et al., 2019a) have proven useful in capturing complex relation among a set of entities, their utility may be constrained in tensors. In this context, the number of queries/tokens equals the number of modes of a tensor, typically a modest number like 3 or 4. The limited queries thus hinder generation of informative context vectors for predicting missing tensor entries.

Second, handling tensors with ordinal entries is more challenging than binary values. Current contrastive learning primarily focuses on binary contrast levels (positive vs. negative). However, many tensor datasets describe relationships at a more detailed granularity. For instance, `Amazon Beauty` tensor signifies user-product-rating relation on a scale of 1 to 5, highlighting the ordinal nature of user preferences. The `Pems` tensor represents traffic-time-sensor relation with scores ranging from 3 to 83, capturing fluctuating traffic patterns. In these cases, enforcing fine-grained comparative levels becomes crucial to capture the subtle differences in the data, which requires thorough investigations.

Third, effective sampling for high-order, fine-grained contrastive relations in tensor data is challenging. Current contrastive learning algorithms mainly consider pairwise relation or its block-version (Arora et al., 2019) that are both second-order proximities. In case of higher-order proximity/interaction, the number of negative samples may grow exponentially. Besides, fine-grained contrastive levels in ordinal tensor data requires identifying "weakly positive" samples that are more difficult to find than negative samples due to the sparsity of tensor data (see Sec 3 for details).

To address these, we propose High-Order Contrastive Tensor Completion (HOCTC), an innovative contrastive learning network for sparse oridinal tensor completion. Figure 1 shows its three major deviations from conventional CL-framework. (1) We employ high-order interaction modelling beyond second-order contrastive leaning. In particular, a self-attention scheme with expanded queries (SAQE) is designed to model the nonlinear interactions among the modes of a tensor even in case of a limited number of tokens. (2)

	Traditional Contrastive Learning Scenario	High-Order Contrastive learning for Tensor Completion (HOCTC)
Order of Relation	Second-order: $\exp(f(z_i, z_j)/\tau)$	Higher-order: $\exp(f(u_i, v_j, w_k)/\tau)$
	for *a pair of* samples	for a triple of modes/factors (or more)
	$f(\cdot,\cdot)$: normalized inner product	$f(\cdot,\cdot,\cdot)$: self-attention with query expansion (SAQE)
Level of Contrast	Binary: positive / negative	Fine-grained: positive / weakly-positive / negative
Sampling Scheme	Random perturbation	Multi-level contrastive sampling

Figure 1. Difference between traditional contrastive learning and the proposed high-order contrastive learning for tensor completion.

We extend binary contrast levels (positive vs. negative) to more delicate comparisons especially for ordinal tensor data. (3) A multi-level contrastive sampling scheme is devised to pick both "weakly-positive" samples and negative samples efficiently to enrich fine-grained comparative structures.

The design of HOCTC is shown in Fig. 2 and detailed in Sec 3. Key innovations/advantages are highlighted below:

- **Contrastive High-order Interaction Modelling**. We integrate contrastive learning with high-order interaction modelling in the task of tensor completion, and devised innovative self-attention scheme with expanded queries to model tensor-mode interactions.
- **Fine-grained Contrastive Levels.** We extend contrastive learning from binary comparison (pos/neg) to fine-grained contrast levels (pos/weak-pos/neg),capturing rich, subtle distinctions to improve contrastive learning.
- **Accurate Completion of Sparse Ordinal Tensor.** We apply HOCTC on tensor data in spatiotemporal and recommendation tasks with promising results obtained.

In the following, Section 2 reviews related work; Section 3 introduces High-Order Contrastive Tensor Completion (HOCTC); experimental results are reported in Section 4, and Section 5 makes conclusions. Our code is released at `https://github.com/wuntunfisher/HOCTC`.

2. Related Work

2.1. Tensor Decomposition and Completion

Tensor decomposition aims at decomposing a tensor into product forms, and is widely used in knowledge graph (Trouillon et al., 2017; Balažević et al., 2019), recommendation (Taneja & Arora, 2018; Chen & Li, 2019), and anomaly

detection (Fanaee-T & Gama, 2016; Xie et al., 2017). Traditional decompositions like CP (Harshman et al., 1970) and Tucker (Tucker, 1966) are both linear. Later, various nonlinear methods are proposed to handle nonlinear high-order relations, such as NLTF (Fang et al., 2015) that uses Gaussian distributions to model the interactions between users, items, and tags, and InfTucker (Xu et al., 2011) that introduces latent Gaussian processes to model the intricate interactions in an infinite-dimensional feature space.

Recently, deep neural networks have drawn considerable interest in tensor decomposition. Many methods propose to replace the multi-linear multiplication with multi-layer perceptrons (MLPs) to fully exploit nonlinear activation layers in the neural network to better capture high-order interactions in tensor data (Dziugaite & Roy, 2015; Liu et al., 2018; Wu et al., 2019). For example, (Chen & Li, 2020) combines the MLP structure with traditional tensor algebra (CP-product) to obtain powerful nonlinear versions of tensor decomposition. CoSTCo (Liu et al., 2019) leverages the expressive power of CNN to model the complex interactions inside tensors and its parameter sharing scheme to preserve the desired low-rank structure, with promising results on a number of real world sparse tensors.

2.2. Contrastive Learning

Contrastive learning algorithms typically compute the loss function in a pairwise form, as

$$\mathcal{L} = \underset{\mathbf{z},\mathbf{z}^+,\mathbf{z}^-}{-\mathbb{E}}\left[-\log\left(\frac{e^{\langle \mathbf{z},\mathbf{z}^+\rangle/\tau}}{e^{\langle \mathbf{z},\mathbf{z}^+\rangle/\tau}+\sum_{\mathbf{z}^-\in \text{Neg}} e^{\langle \mathbf{z},\mathbf{z}^-\rangle/\tau}}\right)\right].$$

Here $(\mathbf{z}, \mathbf{z}^+)$ is a positive pair, and $(\mathbf{z}, \mathbf{z}^-)$'s are negative pairs where $\mathbf{z}^-$'s are often random points that are dissimilar to $\mathbf{z}$. In sequence recommendation, S^3-Rec (Zhou et al., 2020), CoSeRec (Liu et al., 2021) and CL4SRec (Xie et al., 2022) use cropping, masking, reordering or substitution to generate augmentations of the same user. In ICL (Chen et al., 2022), behaviour of a customer is aligned to a prototype by contrastive clustering, which is also applied in other domains (Li et al., 2021; Zhong et al., 2021; Zhou et al., 2022; Deng et al., 2023; Li et al., 2023a). In graph mining, GRACE (Zhu et al., 2020) and GCA (Zhu et al., 2021) generate augmented views of the same graph by feature masking and node/edge dropout. DGI (Veličković et al., 2019) maximizes mutual information between node-level and graph-level representations. MVGRL (Hassani & Khasahmadi, 2020) uses graph diffusion to create positive augmentations.

A series of algorithms exist in the literature related to multi-level/granularity contrastive learning. However, instead of referring to the varying strengths of relationship as quantified by ordinal tensors, the levels in those work represent very different concepts, like different attention-layers or representation stages in NoCL (Chen & Zhang, 2021) and MCVT (Mo et al., 2023); local and global representation of a graph in HGCL (Ju et al., 2023) or body movement in MAC-learning (Shu et al., 2022); different semantic units like characters and words in MCL (Zhao et al., 2023); different modalities (detailed visual features and holistic textual descriptions) in TGFR (Hasan et al., 2024), or instance-level versus class-level representations in MultiSCL (Hu et al., 2023). Few relates to tensor completion tasks either.

In summary, while there has been notable progress in both contrastive learning and tensor decomposition individually, their integration still requires thorough exploration. In (Yang et al., 2022), CP-decomposition is enhanced by a self-supervised loss and applied successfully in image-tensor classification. The proximity relation is defined on pairs of image-tensors and is a second-order scenario. In (Luo et al., 2022), contrastive learning is used to notably improve knowledge graph completion. It is a binary (0/1) tensor and the **h**ead-**R**elation-**t**ail coupling is modelled as a quadratic form $\mathbf{h}_i^\top \mathbf{R}_j \mathbf{t}_k$ instead of symmetric interactions among the three modes. In this paper, we are interested in modelling general-form and high-order interactions among the modes for sparse and ordinal tensors for their completion.

3. High-Order Contastive Tensor Completion

Without loss of generality, we use a 3rd-order tensor $\mathcal{T} \in \mathbb{R}^{I\times J\times K}$ for discussion. It has three dimensions (or modes). Along each dimension, the index ranges from 1 to their capital version, i.e., $i = 1, 2, ..., I$, $j = 1, 2, ..., J$, and $k = 1, 2, ..., K$. Element (i, j, k) of the tensor is denoted by $\mathcal{T}_{ijk} \in \mathbb{R}$. The goal of tensor completion is to fill the missing entries of the tensor based on its observed entries.

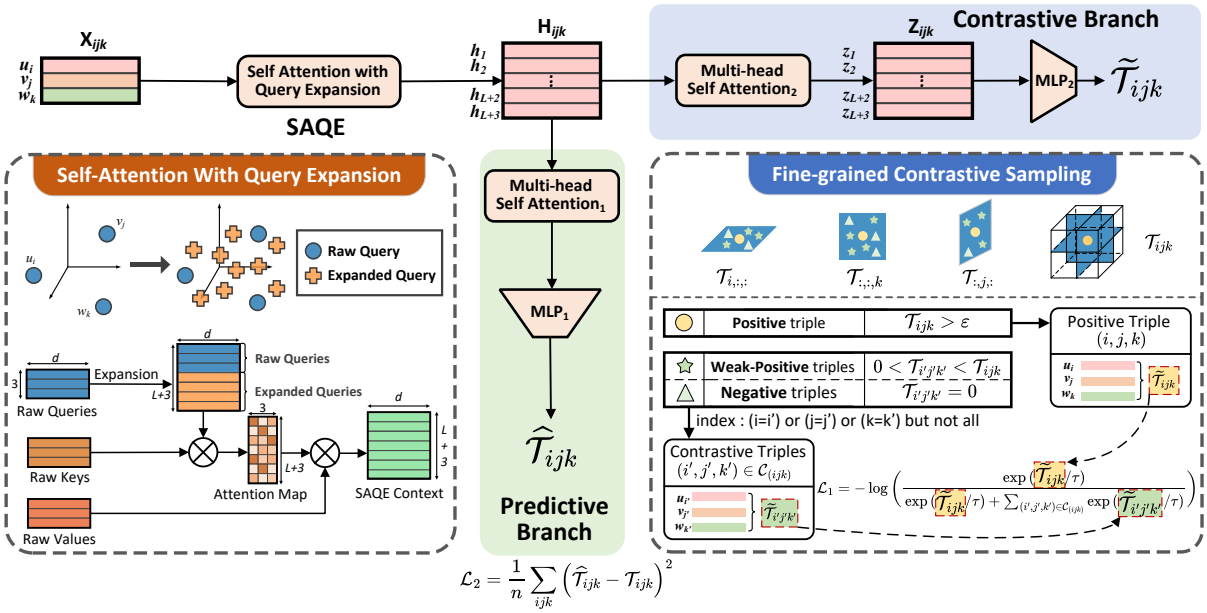
We use $\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k \in \mathbb{R}^{1\times d}$ to denote d-dimensional embeddings of the three indices corresponding to a triple (i, j, k). They are deemed as three *tokens*, concatenated in a matrix

$$\mathbf{X}_{ijk} = \begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_j \\ \mathbf{w}_k \end{bmatrix}, \quad \mathbf{X}_{ijk} \in \mathbb{R}^{3\times d}. \tag{1}$$

HOCTC network is shown in Fig. 2. First, the three tokens in $\mathbf{X}_{ijk}$ (1) will go through a novel attention-based module called Self-Attention with Query-Expansion (SAQE) to obtain an enhanced representation of the triple (i, j, k). Compared with standard self-attention, SAQE can generate a larger number of context vectors even in case of only a few tokens, which are denoted by

$$\mathbf{H}_{ijk}^\top = [\mathbf{h}_1^\top, \mathbf{h}_2^\top, \mathbf{h}_3^\top, ..., \mathbf{h}_{L+3}^\top]. \tag{2}$$

Here, with an abuse of notation, the first three rows in $\mathbf{H}_{ijk}$ are context vectors from $\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k$, while the remaining rows are based on the L expanded queries learned adaptively through SAQE, see (6) and (7) in Sec 3.1 for details.



$$\mathcal{L}_2 = \frac{1}{n}\sum_{ijk}\left(\widehat{\mathcal{T}}_{ijk} - \mathcal{T}_{ijk}\right)^2$$

Figure 2. The proposed HOCTC model using a 3-way tensor for illustration. Tensor modes $(\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k)$ are first fed into SAQE-module (self-attention with query-expansion) to obtain enhanced triple representation $\mathbf{H}_{ijk}$. Then it goes through two branches: (1) predictive branch to compute tensor completion error $\mathcal{L}_1$, and (2) contrastive branch to compute contrastive loss $\mathcal{L}_2$ to enforce fine-grained comparisons among different levels of tensor-mode interactions quantified by ordinal tensor entries. Two losses are added for training.

The triple representation $\mathbf{H}_{ijk}$ is in a highly adaptable feature space to capture high-order tensor-mode interactions. Based on it, two subsequent tasks of tensor completion and contrastive learning can be coordinated together as follows:

(1) **The predictive branch**, in which $\mathbf{H}_{ijk}$ is used to obtain estimated tensor entries $\widehat{\mathcal{T}}_{ijk}$'s for the task of tensor completion through a standard self-attention and MLP layer,

$$\widehat{\mathcal{T}}_{ijk} = \text{MLP}_1\left(\text{Self-attention}_1(\mathbf{H}_{ijk})\right), \tag{3}$$

which is used to compute tensor completion error (10).

(2) **The contrastive branch**, in which $\mathbf{H}_{ijk}$ is used to obtain an estimate of tensor entries $\widetilde{\mathcal{T}}_{ijk}$'s for the task of contrastive representation learning, as

$$\widetilde{\mathcal{T}}_{ijk} = \text{MLP}_2\left(\text{Self-attention}_2(\mathbf{H}_{ijk})\right), \tag{4}$$

which is used to compute the contrastive learning loss in (9). These two branches are integrated by summing their respective loss functions, as in (11).

3.1. Self-Attention with Expanded Queries for Modelling Tensor-Mode Interactions

Self-attention (Vaswani et al., 2017) is a powerful tool to capture complex nonlinear relations and is applied successfully in feature interactions in tabular data in AutoInt (Song et al., 2019b). However, an inherent difficulty exists in modelling the high-order coupling among the modes of a tensor. For example, in three-way tensors, the prediction of a tensor entry $\mathcal{T}_{ijk}$ involves only $N = 3$ modes/tokens and hence with only three queries. The limited number of queries, which equals the order of the tensor, may seriously limit the power of self-attention in representing the triple $(\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k)$ and predicting tensor entry $\mathcal{T}_{ijk}$.

To push the limit of self-attention in case of very few queries, we propose Self-Attention with Query Expansion (SAQE), a novel approach to increase the number of queries to enhance the learned representation. Suppose we have three tokens $\mathbf{u}_i, \mathbf{v}_j, \mathbf{w}_k$ associate with $\mathcal{T}_{ijk}$, which will generate three queries, keys and values as follows

$$\mathbf{Q}_{ijk} = \mathbf{X}_{ijk}\mathbf{W}^q, \mathbf{K}_{ijk} = \mathbf{X}_{ijk}\mathbf{W}^k, \mathbf{V}_{ijk} = \mathbf{X}_{ijk}\mathbf{W}^v, \tag{5}$$

where $\mathbf{W}^q, \mathbf{W}^k, \mathbf{W}^v \in \mathbb{R}^{d\times d'}$ are transform matrices for the query, key and values. Now we aim at learning an extra set of L queries to extend the query-set. To achieve this, we use the three raw queries in $\mathbf{Q}_{ijk}$ as dictionary, and use their linear combinations to generate extra queries, as

$$\widetilde{\mathbf{Q}}_{ijk} = \mathbf{M}\cdot\mathbf{Q}_{ijk}. \tag{6}$$

Here $\mathbf{M} \in \mathbb{R}^{L\times 3}$ has L rows, each specifying one linear combination of the three raw queries in $\mathbf{Q}_{ijk}$, and $\tilde{\mathbf{Q}}_{ijk}$ is the extended query matrix. The matrix $\mathbf{M}$ is subject to

row-wise ℓ_2-normalization. It can either be learned in a purely end-to-end fashion, or through offline approaches (see Appendix A for more details). Then we concatenate raw queries and extended ones together as

$$\mathbf{Q}^{\mathrm{E}}_{ijk} = \begin{bmatrix} \mathbf{Q}_{ijk} \\ \widetilde{\mathbf{Q}}_{ijk} \end{bmatrix}. \tag{7}$$

Here $\mathbf{Q}^{\mathrm{E}}_{ijk} \in \mathbb{R}^{(L+3)\times d'}$ contains altogether $L+3$ query vectors, which is much larger than the original three queries in $\mathbf{Q}_{ijk}$ (5), and is expected to encode the high-order mode interactions within the triple more effectively. As illustrated in Figure 2, these expanded queries serve as densely populated "sensors" to profile the attention landscape with a "higher resolution". Ablation studies in Sec 4.2 show that by extending the 3 raw queries to 30-40, the error of tensor completion drops significantly by up to 42.8% relatively.

We will use the "sensors" in $\mathbf{Q}^{\mathbf{E}}_{ijk}$ (7) as new queries, and the old keys in $\mathbf{K}_{ijk}$ and old values in $\mathbf{V}_{ijk}$ (5), and perform a standard cross-attention as follows:

$$\mathbf{H}_{ijk} = \mathrm{Att}\left(\mathbf{Q}^{\mathrm{E}}_{ijk}, \mathbf{K}_{ijk}, \mathbf{V}_{ijk}\right) = \mathrm{softmax}\left(\frac{\mathbf{Q}^{\mathrm{E}}_{ijk}\mathbf{K}^{\top}_{ijk}}{\sqrt{d'}}\right)\mathbf{V}_{ijk}$$

Here $\mathbf{H}_{ijk} \in \mathbb{R}^{(L+3)\times d}$ can then be deemed as an enriched representation for the triple (i,j,k). We will employ multiple attention heads to further enhance the representation power of the learned triple representations, as

$$\mathbf{H}^{\text{full}}_{ijk} = \mathrm{concat}\left[\mathbf{H}^{\text{head-1}}_{ijk}, \mathbf{H}^{\text{head-2}}_{ijk}, \ldots, \mathbf{H}^{\text{head-h}}_{ijk}\right], \quad \mathbf{H}^{\text{full}}_{ijk} = \mathrm{relu}\left(\mathbf{H}^{\text{full}}_{ijk}\right). \tag{8}$$

In SAQE, only a single attention layer is used, which eliminates concerns that the extended queries would generate noise in the 2rd (or deeper) layer (more detail in Appendix).

3.2. Fine-Grained Contrastive Relation for Ordinal Tensor Data

The performance of contrastive learning heavily relies on the design of its comparative structures. Current methods focus on two-level comparisons, i.e., the contrast between positive relations with negative relations. However, many real-world tensor are endowed with inherent ordinal relations and thus require fine-grained comparative structures.

Consider for example the Amazon Movie&TV tensor with triples (*user_i, item_j, time_k*), each associated with a rating from 0 to 5 (0 for unrated) to quantify the 3rd-order, user-item-time coupling. If we only contrast triples of non-zero rating (1-5) against triples of zero rating to distinguish between interaction and non-interaction, we may not leverage the full spectrum of user engagement. Instead, if we contrast a rating of 5 not just with 0 but also with ratings from 1 to 4, or a rating of 4 with ratings from 1 to 3, we could significantly enrich the comparative structure to capture a broader range of user preferences effectively. We also require that triples must share the same index for at least one dimension (or mode) to be contrasted against each other.

Next we propose the criteria for selecting fine-grained contrastive triples for a given positive triple (i,j,k).

Definition 3.1 (Fine-grained contrastive triples)**.** Given a positive triple (i,j,k) with $\mathcal{T}_{ijk} > 0$, then any other triple (i',j',k') that satisfies the following conditions can be treated as "contrastive triples" against the positive triple (i,j,k), which is denoted by $\mathcal{C}_{(ijk)}$:

1. Overlapped mode indices, $|(i',j',k') \cap (i,j,k)| \geq 1$.
2. Dominated coupling-strength, i.e., $\mathcal{T}_{i'j'k'} < \mathcal{T}_{ijk}$.

Criteria-1 states that (i,j,k) and (i',j',k') must share the same index for at least one mode ($i=i'$ or $j=j'$ or $k=k'$ but not all) to enhance the contextual relevance, or else the modes of $\mathcal{T}_{ijk}$ and $\mathcal{T}_{i'j'k'}$ will not have any overlap and thus become unsuited for contrastive learning. Criteria-2 states that the rating (or coupling) specified by $\mathcal{T}_{i'j'k'}$ must be less than that by $\mathcal{T}_{ijk}$, which is crucial for maintaining the correct optimization direction of the loss function.

As can be seen, the fine-grained contrastive-triples $\mathcal{C}_{(ijk)}$ can be naturally divided into: (1) **weakly-positive triples**, for which $\mathcal{T}_{i'j'k'}$ is non-zero but smaller than $\mathcal{T}_{ijk}$; (2) **negative triples**, for which $\mathcal{T}_{i'j'k'}$ equals zero. We formalize this as,

$$\begin{aligned}
\mathcal{C}_{(ijk)} &= \mathcal{C}^{\text{weak-pos}}_{(ijk)} \bigcup \mathcal{C}^{\text{neg}}_{(ijk)} \\
\mathcal{C}^{\text{weak-pos}}_{(ijk)} &= \left\{ (i',j',k') \;\middle|\; \begin{array}{c} 0 < \mathcal{T}_{i'j'k'} < \mathcal{T}_{ijk} \\ (i',j',k') \cap (i,j,k)| \geq 1 \end{array} \right\} \\
\mathcal{C}^{\text{neg}}_{(ijk)} &= \left\{ (i',j',k') \;\middle|\; \begin{array}{c} \mathcal{T}_{i'j'k'} = 0 \\ |(i',j',k') \cap (i,j,k)| \geq 1 \end{array} \right\}
\end{aligned}$$

Division of $\mathcal{C}_{(ijk)}$ into weakly-positive set and negative set reveals the finer granularities of the contrast-levels than compared with the binary, positive-vs-negative comparisons. Such delicate comparisons allow probing into the difference not only between positive and negative relations, but also more delicately, between positive and weakly-positive relations. The enriched comparisons will lead to stronger self-supervised signals in enhancing the model's capability to capture the complex, high-order interactions among the modes of real-world tensors.

Upon determining the contrastive triple sets $(i',j',k') \in \mathcal{C}_{(ijk)}$ for a given positive triple (i,j,k), the contrastive loss function can then be defined as follows

$$\mathcal{L}_1 = -\log\left(\frac{\exp\left(\widetilde{\mathcal{T}}_{ijk}/\tau\right)}{\exp\left(\widetilde{\mathcal{T}}_{ijk}/\tau\right) + \sum\limits_{(i'j'k')\in\mathcal{C}_{(ijk)}} \exp\left(\widetilde{\mathcal{T}}_{i'j'k'}/\tau\right)}\right) \tag{9}$$

which shall be summed over the positive triples (i,j,k)'s

under consideration. The $\widetilde{\mathcal{T}}_{ijk}$ is the estimated tensor mode relations through the learned representation (4).

3.3. Multi-level Contrastive Sampling

We propose multi-level contrastive-sampling (MLCS) to obtain the contrastive triple set $\mathcal{C}_{(ijk)}$ for any positive triple (i, j, k) efficiently. It include two parts: weakly-positive set $\mathcal{C}^{\text{weak-pos}}_{(ijk)}$, and negative set $\mathcal{C}^{\text{neg}}_{(ijk)}$. The negative triples are easily obtained by perturbing one or two indices of the positive tripe (i, j, k). However, it's nontrivial to find enough weakly-positive triples for (i, j, k) due to the sparsity of real-world tensors (non-zero entries for recommendation tensors could be as low as $10^{-6}\%$ (Hegde et al., 2019)).

We use a simple mode-index-sorting scheme to reorder the list of observed triples, so that weakly-positive triples can be found efficiently. It begins by dividing observed triple list into b non-overlapping blocks. In each block, the triples are sorted as follows. First, we sort the i-index and reorder the triples; then for those triples with the same i-index, re-order them by sorting their j-index; finally, for triples with the same i, j-index, re-order them by their k-index. Such sorting increases the likelihood of a positive triple to encounter relevant weakly-positive samples in its vicinity within the triple-list, due to the effect of sorting. For each sorted block, we get mini-batches by sequentially cropping a segment from the list with desired size. We then pick weakly-positive triples $\mathcal{C}^{\text{weak-pos}}_{(ijk)}$ for any positive triple (i, j, k) in a minibatch efficiently by searching in its neighbors or the whole block.

We note that the index concentration effect is the strongest for the first mode that is sorted because it is an unconstrained sorting; for the two modes sorted afterwards, the concentration is less significant because they are subject to the ordering constraints of the first mode. For example, $(1, 50, 24)$ and $(1000, 50, 24)$ could be far away from each other due to the big difference in their i-index, but they share the same j-index and could still be meaningful contrastive triples for each other. Therefore, in practice we randomize the order of the three modes for sorting and integrate it with the mini-batch based training framework. Specifically, we shuffle the data at the beginning of each epoch and divide it into b non-overlapping blocks. Within each block, we randomly select an order—such as (i, j, k), (k, j, i), or (j, k, i)—to sort the data. This could promote a more robust and generalized learning process, see detailed discussions in the Appendix. A detailed pseudo-code for the mode-index sorting for obtaining the weakly-positive triples can be found in Algorithm 1.

Sorting n triples divided into b blocks takes $O(n \log \frac{n}{b})$ time. After index sorting, the cost of finding contrastive triples for each positive (i, j, k) in a minibatch will be $O(l+p)$, where l is the neighborhood size for searching for weakly-positive triples, and p is the number of negative triples needed for each positive triple. Empirically p is much smaller than the mini-batch size m, and negative sampling is done by randomly perturbing the triple indices, so the complexity of contrastive sampling in a minibatch, $O(lm + pm)$, is linear in m and independent of the order of the tensor data. Overall, HOCTC has a complexity between linear and log-linear over esample size, which is efficient.

Table 1. Statistics of the spatio-temporal tensors and the recommender tensors used in our experiments.

DATASET	SHAPE	#OBSERVED	SAMPLING_RATIO
PEMS	(228, 44, 288)	288922	10.00%
GZSPEED	(214, 61, 144)	187978	10.00%
METR	(207, 119, 288)	709431	10.00%
CITYTEMP	(1854, 24, 36)	106186	10.00%
SG	(2321, 5596, 1600)	105764	$5.09 \times 10^{-4}\%$
BEAUTY	(22279, 12079, 238)	192377	$3.00 \times 10^{-6}\%$
MOVIE & TV	(101916, 47975, 238)	984060	$8.45 \times 10^{-7}\%$
GOWALLA	(318608, 2857394, 8921)	19116507	$2.35 \times 10^{-7}\%$

3.4. Composite Loss Function

The estimated tensorial entries $\widehat{\mathcal{T}}_{ijk}$ in the predictive branch are used to compute the Mean Squared Error (MSE) loss:

$$\mathcal{L}_2 = \frac{1}{n} \sum_{ijk} \left(\mathcal{T}_{ijk} - \widehat{\mathcal{T}}_{ijk} \right)^2, \tag{10}$$

and the complete loss is defined as

$$\mathcal{L} = \mathcal{L}_2 + \alpha \cdot \mathcal{L}_1, \tag{11}$$

a mixture of MSE (10) with the contrastive loss term (9), as where α controls the trade-off between the two losses.

Note that HOCTC is a generic tensor completion method that does not take advantage of specific structures of the tensor data based on domain knowledge (e.g., temporal order or smoothness in the traffic data, or the head-tail symmetry in knowledge graphs). This means that HOCTC can be conveniently integrated with domain-based optimizations to further improve the performance of tensor completions.

4. Experiment

We have chosen two types of ordinal tensors in our experiments, i.e., spatio-temporal tensors and recommendation tensors[1], as listed in Table 2 and Table 3. Spatio-temporal tensor completion is a popular benchmark for evaluating sparse tensor completion models in many previous works (Liu et al., 2019; Chen et al., 2020b; Xie et al., 2020; Lei et al., 2022). Some datasets in the literature are not publicly disclosed. To ensure a faithful comparison, we have

[1]Binary tensors like knowledge graphs are therefore not the focus of the paper and will be studied in our future research.

Table 2. Tensor completion results of spatio-temporal tensors.

DATA	MODEL	MAE	MAPE	RMSE
PEMS	P-TUCKER	5.3976	13.6079	8.1234
	NEURALCP	4.3561	11.0245	6.9732
	COSTCO	4.1987	9.6052	6.8302
	NTM	4.6195	9.5847	7.1754
	NTC	4.2245	9.7545	6.9035
	LIGHTNESTLE	4.0680	9.0636	6.6864
	HOCTC	**3.4033**	**8.2663**	**5.7479**
GZ	P-TUCKER	4.9984	24.5495	6.0154
	NEURALCP	3.2768	15.2330	4.8482
	COSTCO	4.2937	15.6576	5.9790
	NTM	3.6004	15.4211	5.4166
	NTC	3.2760	11.6007	5.0013
	LIGHTNESTLE	3.0571	**9.8641**	4.5201
	HOCTC	**2.8937**	10.5821	**4.4491**
CITY	P-TUCKER	0.4784	1.5548	0.4754
	NEURALCP	0.3208	1.1124	0.4134
	COSTCO	0.2451	0.8534	0.3203
	NTM	0.3076	1.0720	0.3998
	NTC	0.2190	0.7625	0.2852
	LIGHTNESTLE	0.2902	0.9985	0.4011
	HOCTC	**0.1936**	**0.6735**	**0.2786**
METR	P-TUCKER	6.1558	17.8996	9.4813
	NEURALCP	4.4144	10.9632	7.2971
	COSTCO	4.1569	10.2143	7.3394
	NTM	4.4680	10.8168	7.5205
	NTC	4.0717	9.9937	7.0193
	LIGHTNESTLE	4.0454	10.3965	7.1485
	HOCTC	**3.4482**	**9.0951**	**6.1172**

selected four most widely-used, publicly-available spatio-temporal tensor datasets including Pems, GZspeed, Metr, and CityTemp. Among them, Pems, GZspeed, and Metr are tuples in the form of $(sensor\_id, day\_id, interval\_id)$; CityTemp is in the form $(day\_id, hour\_id, city\_id)$. Following (Li et al., 2023b), we randomly sample 10% tensor entries to construct a sparse tensor to evaluate tensor completion algorithms. For sparse tensors in recommender systems, we choose four widely used datasets: SG (Li et al., 2015) and Gowalla checkins (Liu et al., 2017) are $(user\_id, location\_id, poi\_id)$ tuples; Beauty and Movie&TV tensors are $(user\_id, item\_id, week\_id)$ tuples. More detailed description of the datasets can be found in the Appendix.

Our codes were written in Tensorflow with Python 3.9. We have chosen six tensor decomposition methods including classic Tucker-decomposition (Oh et al., 2018) and five non-linear tensor decomposition methods proposed very recently using neural networks, like NeuralCP (Liu et al., 2018), CoSTCo (Liu et al., 2019), NTC (Xie et al., 2020), NTM (Chen & Li, 2020) and LightNestle (Li et al., 2023b). Note that LightNestle is designed mainly for spatio-temporal data, and will not be evaluated on recommender tensors with much larger dimensions. We also considered a number of GNN-based methods for recommendation systems (Li et al., 2019; Guo et al., 2021; Jiang et al., 2022) for comparison, (see Appendix for more results). Following (Liu et al., 2019), we report three evaluations metrics, including MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), and MAPE (Mean Absolute Percentage Error).

For HOCTC, the embedding dimension is $d = 20$ (and all others); both the standard self-attention module and the SAQE module have one attention layer with 5 heads; the MLP_1 in (3) and MLP_2 in (4) are both 3-layer MLPs; the number of blocks is chosen as 20. We use Adam and train HOCTC up to 200 epochs, adopting an early-stop strategy with a 20-epoch patience. Hyper-parameters: initial learning rate in $\{0.01, 0.001, 0.0001\}$; mini-batch size in $\{128, 256, 512, 1024\}$; α (11) in $\{1, 0.1, 0.01, 0.001\}$, the number of queries for SAQE in $\{5, 10, 20, 30\}$; all chosen by validation set. For competing methods, we follow their respective hyper-parameter tuning strategies (see Appendix for details). For all datasets, we have used a random 80%/10%/10% split as train/val/test split, following (Chen & Li, 2020). For spatio-temporal tensors, since they are complete before sampling, we only employ weakly-positive samples for contrastive learning in the loss function (9). For sparse tensors in recommendation tasks, we use both weakly-positive samples and negative samples (10 negative samples for each positive triple by random index perturbation).

4.1. Evaluation Results

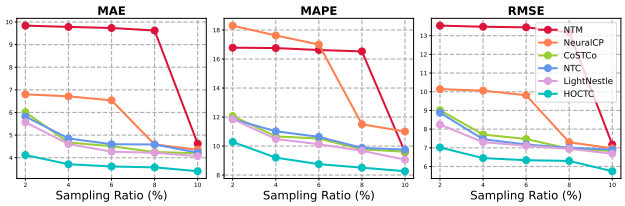
Results for spatio-temporal and recommender tensors are in Table 2 and Table 3, respectively. Overall, HOCTC has shown promising results. It attains the lowest error for 11 out of the 12 error-comparisons for the 4 spatio-temporal datasets, and 8 lowest errors among the 12 comparisons for the 4 recommendation tensors. In terms of MAE, HOCTC is relatively 16.3% and 14.8% better than best baseline on Pems and Metr datasets, respectively. On the largest sparse tensor Gowalla, it is relatively 26.7% and 57.9% better than the best baseline method for MAE and MAPE, respectively.

Following (Li et al., 2023b), Figure 3(a) shows the performance of neural network-based models on the Pems dataset using 5 different sampling ratios (0.02, 0.04, 0.06, 0.08, 0.1). As can be seen, all the methods have superior performance with more samples. Under the same sampling rate, HOCTC almost always attains the lowest error, and is also quite robust even at very low sampling rates. This evidence supports the efficacy of HOCTC in sparse tensor completion.

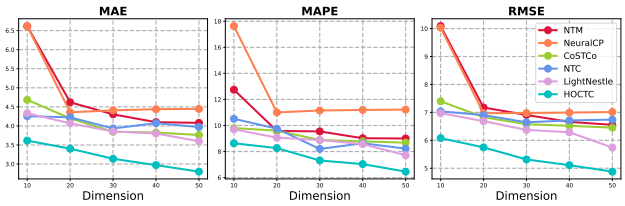
The embedding dimension of the latent factors (or modes) of the tensor influences both the computational demand and the model accuracy. Figure 3(b) presents the error of different methods by varying the dimension from 10 to 50,

Table 3. Tensor completion results of recommendation tensors.

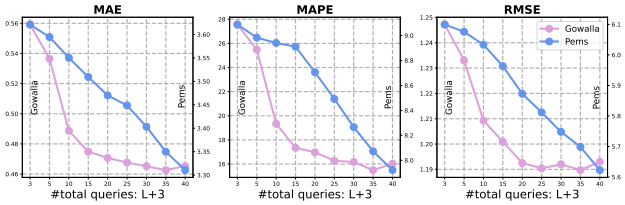
DATA	MODEL	MAE	MAPE	RMSE
SG	P-TUCKER	0.2454	167.5481	0.3954
	NEURALCP	0.1004	64.1548	0.1784
	COSTCO	0.0774	41.5384	**0.1546**
	NTM	0.0832	58.8020	0.1612
	NTC	0.0828	53.6845	0.1621
	HOCTC	**0.0668**	**25.8904**	0.1581
BEAUTY	P-TUCKER	1.4333	45.0231	1.8084
	NEURALCP	0.9324	37.5881	1.1814
	COSTCO	0.8237	33.1612	1.1566
	NTM	0.9590	**30.1363**	1.2141
	NTC	0.8986	31.0155	1.2073
	HOCTC	**0.7772**	33.6771	**1.1301**
MOVIE&TV	P-TUCKER	1.2246	36.9661	1.1653
	NEURALCP	0.7244	28.5476	1.0247
	COSTCO	0.6972	27.6130	1.0050
	NTM	1.0034	29.0678	1.2798
	NTC	0.7264	**27.5998**	1.0593
	HOCTC	**0.6693**	28.7005	**0.9960**
GOWALLA	P-TUCKER	1.0554	124.5485	2.2545
	NEURALCP	0.6538	41.2395	1.1643
	COSTCO	0.6472	40.2073	1.1818
	NTM	0.6954	48.4815	1.1836
	NTC	0.6345	38.4154	**1.1545**
	HOCTC	**0.4652**	**16.1613**	1.1920



(a) Predictive error v.s. sampling ratios on Pems dataset.



(b) Predictive error v.s. latent dimension on Pems dataset.



(c) Predictive error v.s. the number of expanded queries.

Figure 3. Behaviour of HOCTC. More examples are in Appendix.

using a 10% sampling rate on Pems dataset. Again, HOCTC is quite competitive for all target ranks and shows a stable performance improvement with increasing ranks.

In Figure 3(c) we analyze how varying the number of extra queries learned through SAQE can impact the performance of HOCTC, by plotting the errors versus the number of queries on two representative tensors (Pems and Gowalla). We can see that as more extra queries are incorporated to enrich the model, the error consistently decreases. Empirically, a tenfold increase in the number of queries (from 3 to 30 or 40) leads to a reduction in MAE/MAPE/RMSE by 4.8%-42.8%. This substantial improvement validates the effectiveness of SAQE in enhancing tensor completion.

It is worth noting that in the case of very sparse tensors such as Gowalla, there might be a slight fluctuation in performance when the number of queries surpasses a certain threshold. We hypothesize that this fluctuation could be due to overfitting, which can be effectively avoided by using the validation set to determine a suitable number of queries.

4.2. Ablation Study

We study how the two main modules, SAQE (Self Attention with Expanded Query) and MLCS (Multi-Level COntrastive Sampling), affect the performance. Table 4 reports ablation studies with MAE. As can be seen, when the two modules were replaced by their vanilla version, i.e., standard self-attention and binary-level contrastive learning, the performance is the worst. When the self-attention module is upgraded to SAQE, the error consistently drops by 3.14%-19.14% relatively across the 8 tensors; when the binary-level contrastive learning is replaced is upgraded to MLCS, the error consistency drops by up to 2.43%-16.77%. When both modules are upgraded, the error drops by 4.17% - 31.64%.

5. Conclusion

We have introduced high-order contrastive learning (HOCTC) for sparse ordinal tensor completion. It extends traditional contrastive learning by modeling high-order contrastive relations, and by employing fine-grained comparative levels in case the target relation is no longer binary. Promising empirical results were observed against baseline methods in a number of widely used benchmark datasets. Future work includes theoretic analysis on multi-level contrastive learning, and how to achieve a good balance among different error metrics. We are also interested in combining the proposed method with spatio-temporal priors for more accurate prediction in spatio-temporal data.

Table 4. Impact of SAQE (self-attention with query-expansion) and MLCS (multi-level contrastive-sampling) module on MAE.

SAQE	MLCS	PEMS	GZ	CITY	METR	SG	BEAUTY	MOVIE&TV	GOWALLA
✗	✗	3.8511	3.0418	0.2832	3.9915	0.0724	0.8378	0.6984	0.6034
✓	✗	3.6643	2.9464	0.2290	3.6453	0.0689	0.8091	0.6725	0.5077
✗	✓	3.6213	2.9076	0.2357	3.8495	0.0680	0.8174	0.6766	0.5592
✓	✓	3.4033	2.8937	0.1936	3.4482	0.0668	0.7772	0.6693	0.4652

Acknowledgement

This work is partially supported by National Natural Science Foundation of China (Grant No. 62276099), Science and Technology Innovation 2030 - Brain Science and Brain-Inspired Intelligence Project (Grant No. 2021ZD0200204), and Special Fund for International Conferences of graduate students at East China Normal University.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning algorithms. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.

Balažević, I., Allen, C., and Hospedales, T. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5185–5194, 2019.

Brandoni, D. and Simoncini, V. Tensor-train decomposition for image recognition. *Calcolo*, 57:1–24, 2020.

Cao, W., Wang, Y., Sun, J., Meng, D., Yang, C., Cichocki, A., and Xu, Z. Total variation regularized tensor rpca for background subtraction from compressive measurements. *IEEE Transactions on Image Processing*, 25(9):4075–4090, 2016.

Chen, B., Wang, Y., Liu, Z., Tang, R., Guo, W., Zheng, H., Yao, W., Zhang, M., and He, X. Enhancing explicit and implicit feature interactions via information sharing for parallel deep ctr models. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 3757–3766, 2021.

Chen, H. and Li, J. Adversarial tensor factorization for context-aware recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pp. 363–367, 2019.

Chen, H. and Li, J. Neural tensor model for learning multi-aspect factors in recommender systems. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 2020, 2020.

Chen, Q. and Zhang, J. Multi-level contrastive learning for few-shot problems. *arXiv preprint arXiv:2107.07608*, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. Proceedings of Machine Learning Research, 2020a.

Chen, X., Yang, J., and Sun, L. A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 117:102673, 2020b.

Chen, Y., Liu, Z., Li, J., McAuley, J., and Xiong, C. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, pp. 2172–2182, 2022.

Cheng, Y. and Xue, Y. Looking at ctr prediction again: Is attention all you need? In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1279–1287, 2021.

Chuang, Y.-S., Dangovski, R., Luo, H., Zhang, Y., Chang, S., Soljačić, M., Li, S.-W., Yih, W.-t., Kim, Y., and Glass, J. Diffcse: Difference-based contrastive learning for sentence embeddings. *arXiv preprint arXiv:2204.10298*, 2022.

Deng, X., Huang, D., Chen, D.-H., Wang, C.-D., and Lai, J.-H. Strongly augmented contrastive clustering. *Pattern Recognition*, 139:109470, 2023.

Dziugaite, G. K. and Roy, D. M. Neural network matrix factorization. *arXiv preprint arXiv:1511.06443*, 2015.

Fanaee-T, H. and Gama, J. Tensor-based anomaly detection: An interdisciplinary survey. *Knowledge-Based Systems*, 98:130–147, 2016.

Fang, X., Pan, R., Cao, G., He, X., and Dai, W. Personalized tag recommendation through nonlinear tensor factorization using gaussian kernel. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015. doi: 10.1609/aaai.v29i1.9214. URL https://ojs.aaai.org/index.php/AAAI/article/view/9214.

Fernandes, S., Fanaee-T, H., and Gama, J. Tensor decomposition for analysing time-evolving social networks: An overview. *Artificial Intelligence Review*, 54:2891–2916, 2021.

Gao, T., Yao, X., and Chen, D. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing*, pp. 6894–6910, 2021.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Guo, W., Su, R., Tan, R., Guo, H., Zhang, Y., Liu, Z., Tang, R., and He, X. Dual graph enhanced embedding neural network for ctr prediction. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 496–504, 2021.

Harshman, R. A. et al. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1 – 84, 1970.

Hasan, M. M., Sami, S. M., and Nasrabadi, N. Text-guided face recognition using multi-granularity cross-modal contrastive learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5784–5793, 2024.

Hassani, K. and Khasahmadi, A. H. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pp. 4116–4126. Proceedings of Machine Learning Research, 2020.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Hegde, K., Asghari-Moghaddam, H., Pellauer, M., Crago, N., Jaleel, A., Solomonik, E., Emer, J., and Fletcher, C. W. Extensor: An accelerator for sparse tensor algebra. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 319–333, 2019.

Hu, X., Lin, L., Liu, A., Wen, L., Philip, S. Y., et al. A multi-level supervised contrastive learning framework for low-resource natural language inference. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

Jiang, W., Jiao, Y., Wang, Q., Liang, C., Guo, L., Zhang, Y., Sun, Z., Xiong, Y., and Zhu, Y. Triangle graph interest network for click-through rate prediction. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pp. 401–409, 2022.

Ju, W., Gu, Y., Luo, X., Wang, Y., Yuan, H., Zhong, H., and Zhang, M. Unsupervised graph-level representation learning with hierarchical contrasts. *Neural Networks*, 158:359–368, 2023.

Kharitonov, E., Rivière, M., Synnaeve, G., Wolf, L., Mazaré, P.-E., Douze, M., and Dupoux, E. Data augmenting contrastive learning of speech representations in the time domain. In *IEEE Spoken Language Technology Workshop*, pp. 215–222, 2021.

Lang, L., Zhu, Z., Liu, X., Zhao, J., Xu, J., and Shan, M. Architecture and operation adaptive network for online recommendations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3139–3149, 2021.

Lei, M., Labbe, A., Wu, Y., and Sun, L. Bayesian kernelized matrix factorization for spatiotemporal traffic data imputation and kriging. *IEEE Transactions on Intelligent Transportation Systems*, 23(10):18962–18974, 2022.

Li, W., Zhu, E., Wang, S., and Guo, X. Graph clustering with high-order contrastive learning. *Entropy*, 25(10): 1432, 2023a.

Li, X., Cong, G., Li, X.-L., Pham, T.-A. N., and Krishnaswamy, S. 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*, pp. 433–442, 2015.

Li, Y., Hu, P., Liu, Z., Peng, D., Zhou, J. T., and Peng, X. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8547–8555, 2021.

Li, Y., Liang, W., Xie, K., Zhang, D., Xie, S., and Li, K. Lightnestle: quick and accurate neural sequential tensor completion via meta learning. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pp. 1–10. IEEE, 2023b.

Li, Z., Cui, Z., Wu, S., Zhang, X., and Wang, L. Fi-gnn: Modeling feature interactions via graph neural networks for ctr prediction. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 539–548, 2019.

Liu, B., He, L., Li, Y., Zhe, S., and Xu, Z. Neuralcp: Bayesian multiway data analysis with neural tensor decomposition. *Cognitive Computation*, 10:1051–1061, 2018.

Liu, H., Li, Y., Tsang, M., and Liu, Y. Costco: A neural tensor completion model for sparse tensors. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 324–334, 2019.

Liu, Y., Pham, T.-A. N., Cong, G., and Yuan, Q. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment*, 10(10):1010–1021, 2017.

Liu, Z., Chen, Y., Li, J., Yu, P. S., McAuley, J., and Xiong, C. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*, 2021.

Luo, Z., Xu, W., Liu, W., Bian, J., Yin, J., and Liu, T.-Y. KGE-CL: Contrastive learning of tensor decomposition based knowledge graph embeddings. In *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 2598–2607. International Committee on Computational Linguistics, 2022. URL https://aclanthology.org/2022.coling-1.229.

Mao, K., Zhu, J., Su, L., Cai, G., Li, Y., and Dong, Z. Finalmlp: an enhanced two-stream mlp model for ctr prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 4552–4560, 2023.

Misra, I. and Maaten, L. v. d. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6707–6717, 2020.

Mo, S., Sun, Z., and Li, C. Multi-level contrastive learning for self-supervised vision transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2778–2787, 2023.

Oh, S., Park, N., Lee, S., and Kang, U. Scalable tucker factorization for sparse tensors-algorithms and discoveries. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 1120–1131. IEEE, 2018.

Rettinger, A., Wermser, H., Huang, Y., and Tresp, V. Context-aware tensor decomposition for relation prediction in social networks. *Social Network Analysis and Mining*, 2:373–385, 2012.

Shu, X., Xu, B., Zhang, L., and Tang, J. Multi-granularity anchor-contrastive representation learning for semi-supervised skeleton-based action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., and Tang, J. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1161–1170, 2019a.

Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., and Tang, J. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1161–1170, 2019b.

Taneja, A. and Arora, A. Cross domain recommendation using multidimensional tensor factorization. *Expert Systems with Applications*, 92:304–316, 2018.

Tao, D., Guo, Y., Li, Y., and Gao, X. Tensor rank preserving discriminant analysis for facial recognition. *IEEE transactions on image processing*, 27(1):325–334, 2017.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pp. 776–794. Springer, 2020.

Trouillon, T., Dance, C. R., Gaussier, É., Welbl, J., Riedel, S., and Bouchard, G. Knowledge graph completion via complex tensor factorization. *Journal of Machine Learning Research*, 18(130):1–38, 2017.

Tucker, L. R. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

Van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv e-prints*, pp. arXiv–1807, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. Deep Graph Infomax. In *International Conference on Learning Representations*, 2019.

Wang, D., Ding, N., Li, P., and Zheng, H.-T. Cline: Contrastive learning with semantic negative examples for natural language understanding. *arXiv preprint arXiv:2107.00440*, 2021a.

Wang, F., Gu, H., Li, D., Lu, T., Zhang, P., and Gu, N. Towards deeper, lighter and interpretable cross network for ctr prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 2523–2533, 2023.

Wang, Z., She, Q., and Zhang, J. Masknet: Introducing feature-wise multiplication to ctr ranking models by instance-guided mask. *arXiv preprint arXiv:2102.07619*, 2021b.

Wu, F., Liu, Y., and Zhuang, Y. Tensor-based transductive learning for multimodality video semantic concept detection. *IEEE Transactions on Multimedia*, 11(5):868–878, 2009.

Wu, X., Shi, B., Dong, Y., Huang, C., and Chawla, N. V. Neural tensor factorization for temporal interaction learning. In *Proceedings of the Twelfth ACM international conference on web search and data mining*, pp. 537–545, 2019.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Xie, K., Li, X., Wang, X., Xie, G., Wen, J., Cao, J., and Zhang, D. Fast tensor factorization for accurate internet anomaly detection. *IEEE/ACM transactions on networking*, 25(6):3794–3807, 2017.

Xie, K., Lu, H., Wang, X., Xie, G., Ding, Y., Xie, D., Wen, J., and Zhang, D. Neural tensor completion for accurate network monitoring. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 1688–1697. IEEE, 2020.

Xie, X., Sun, F., Liu, Z., Wu, S., Gao, J., Zhang, J., Ding, B., and Cui, B. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, pp. 1259–1273. IEEE, 2022.

Xu, Z., Yan, F., et al. Infinite tucker decomposition: Nonparametric bayesian models for multiway data analysis. *arXiv preprint arXiv:1108.6296*, 2011.

Yang, C., Qian, C., Singh, N., Xiao, C. D., Westover, M., Solomonik, E., and Sun, J. Atd: Augmenting cp tensor decomposition by self supervision. *Advances in Neural Information Processing Systems*, 35:32039–32052, 2022.

Zhao, S., Wang, C., Hu, M., Yan, T., and Wang, M. Mcl: Multi-granularity contrastive learning framework for chinese ner. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 14011–14019, 2023.

Zhong, H., Wu, J., Chen, C., Huang, J., Deng, M., Nie, L., Lin, Z., and Hua, X.-S. Graph contrastive clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9224–9233, 2021.

Zhou, K., Wang, H., Zhao, W. X., Zhu, Y., Wang, S., Zhang, F., Wang, Z., and Wen, J.-R. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pp. 1893–1902, 2020.

Zhou, T., Wang, W., Konukoglu, E., and Van Gool, L. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2582–2593, 2022.

Zhu, J., Jia, Q., Cai, G., Dai, Q., Li, J., Dong, Z., Tang, R., and Zhang, R. Final: Factorized interaction layer for ctr prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2006–2010, 2023.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., and Wang, L. Deep Graph Contrastive Representation Learning. In *ICML Workshop on Graph Representation Learning and Beyond*, 2020.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., and Wang, L. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference*, pp. 2069–2080, 2021.

Algorithm 1 Mode Index Sorting for Obtaining Weakly-Positive Triples $\mathcal{C}_{(ijk)}$ for a positive triple (i, j, k).

Input: A set of observed tensor entries $\mathcal{T}_{ijk}$'s with index sets $\mathcal{I}, \mathcal{J}, \mathcal{K}$; number of blocks b, mini-batch size m, neighborhood size l;
Output: Weakly-positive triples $\mathcal{C}^{\text{weak-pos}}_{(ijk)}$ for positive triple (i, j, k)

initialize $\mathcal{C}^{\text{weak-pos}}_{(ijk)} = \emptyset$
// partition the input triples into b non-overlapping chunks
$\{\pi_1, \pi_2, \ldots, \pi_b\} \leftarrow$ cut-into-block($\mathcal{T}, b$)
// sort by index for each block π_l
for $l = 1$ to b **do**
  $\pi_l \leftarrow$ sort-by-index(i, π_l) *// sort by first index*
  for $i \in \mathcal{I}$ **do**
    $same_{(i)} \leftarrow$ get-entries(i, π_l)
    sort-by-index(j, $same_{(i)}$) *// sort by second index for entries sharing same i*
    for $j \in \mathcal{J}$ **do**
      $same_{(ij)} \leftarrow$ get-entries(j, $same_{(i)}$)
      sort-by-index(k, $same_{(ij)}$) *// sort by third index for entries sharing same i and j*
    end for
  end for
end for
// pick a mini-batch of size m from the (sorted) blocks
$\mathcal{B} \leftarrow$sample-mini-batch($m, \{\pi_1, \pi_2, \ldots, \pi_b\}$)
// get indices from mini-batch $\mathcal{B}$
$\mathcal{I}', \mathcal{J}', \mathcal{K}' \leftarrow$get-index-set($\mathcal{B}$)
// select $\mathcal{C}^{\text{weak-pos}}_{(ijk)}$ for a positive triple (i, j, k) within its vicinity in the mini-batch
for $i' \in \mathcal{I}', j' \in \mathcal{J}', k' \in \mathcal{K}'$ and (i', j', k') within a neighborhood window $2l + 1$ of (i, j, k) in the sorted triple list **do**
  if $1 \leq |\{i, j, k\} \cap \{i', j', k'\}| < 3$ and $\mathcal{T}_{ijk} > \mathcal{T}_{i'j'k'}$ **then**
    $\mathcal{C}^{\text{weak-pos}}_{(ijk)} \leftarrow \mathcal{C}^{\text{weak-pos}}_{(ijk)} \cup \{(i', j', k')\}$
  end if
end for
return $\mathcal{C}^{\text{weak-pos}}_{(ijk)}$

A. Self-Attention with Query Expansion (SAQE)

The SAQE aims at extending the amount of queries in case of limited queries. Here, we only increase the number of queries, but the number of keys and values remain the same. This is because the keys and values have to be strictly paired and they represent the "source" signal. In comparison, the queries serve as "sensors" to reflect the impact of the source signals; obviously, having more sensors would increase the "resolution" of the signal but will not introduce undesired signals. We only implement one attention layer in SAQE, and so we don't have to worry about introducing undesired noise into the second or deeper layer of attention due to the extra context vectors generated in the first layer from the extra queries.

The key of SAQE is the construction of matrix $\mathbf{M}$ in e.q. (6). We can surely learn $\mathbf{M}$ in a purely supervised fashion. However, we found that pre-defining a suitable candidate of $\mathbf{M}$ and then removing the redundancy in it could deliver better results, which we describe as follows. Here, instead of using only the three vectors in $\mathbf{Q}_{ijk}$ as the basis for generating extra (query) vectors, we first extend it to a larger set of vectors. To do this, we first compute the mean of the three vectors and obtain altogether 4 vetors. Then we pick out all possible vector pairs from these four vectors (with $C_4^2 = 6$ pairs), compute the mean vector for each pair, and finally end up with having altogether 10 vectors as our basis. Finally, we generate a $\mathbb{R}^{1000\times 10}$ matrix whose entries are drawn from a standard Gaussian distribution (with each row normalized by its ℓ_2-norm), and then use k-means clustering to pick k most representative rows as the reconstruction matrix. Using these representative rows, we generate altogether k vectors based on the 10 basis vectors as dictionary. Here, k is the desired number of queries vectors. Such a procedure is implemented off-line and is quite efficient. Empirically, it generates better results than optimizing the $\mathbf{M}$ matrix in an entirely unconstrained manner.

B. Datasets

The Pems, GZspeed, and Metr datasets each record traffic speed data, with entries corresponding to specific sensors and time intervals, represented as $(sensor\_id, day\_id, interval\_id)$. Pems captures data from 228 sensors over 44 days at 288 intervals per day. GZspeed encompasses traffic data from 214 road segments in Guangzhou, China, collected over 61 days at 144 intervals daily. Metr reflects traffic conditions in the Los Angeles Metropolitan area via 207 loop detectors, with data

aggregated at 5-minute intervals. The CityTemp dataset records hourly temperature variations across 36 cities, formatted as $(day\_id, hour\_id, city\_id)$. Follow (Li et al., 2023b), we utilize a randomized extraction method, retaining 10% of the non-zero entries from these complete tensors to form our datasets for analysis.

We also examine social check-in data from two distinct sources: SG checkins, sourced from Foursquare in Singapore, and Gowalla checkins, a global dataset collected by Gowalla. For both SG and Gowalla datasets, we adopted the methodological framework proposed by (Liu et al., 2019). Each tensor index is a tuple of $(user\_id, location\_id, poi\_id)$.

The Beauty and Movies & TV tensors are from Amazon datasets. These datasets comprise $(user\_id, item\_id, week\_id)$ tuples. Following the methodology of (Chen & Li, 2020), we employ the standard 5-core dataset approach, removing purchase records prior to 2010 and categorizing timestamps by weeks.

C. Baselines

We here describe more detail on the optimization of the baseline comparison models. Most models have a number of hyperparameters which need to be optimized. We use the default hyperparameters values and hyperparameter grid search settings specified in their own study if any.

For NTM, the batch size is in [32, 64, 128, 256, 512, 1024]; the learning rate is in [0.0005, 0.001, 0.005, 0.01]; the number of T-MLP blocks is in [2, 3, 4, 5]. For CoSTCo, the batch size is in [32, 64, 128, 256, 512, 1024]; the learning rate is in [0.0001, 0.001, 0.005, 0.03, 0.01, 0.1]. For NTC, the number of Convolutional Kernels is in [8, 16, 32, 64]; the number of CNN layers is in [2, 3, 4, 5]. For LightNestle, the batch size is in [32, 64, 128, 256, 512, 1024]; the learning rate is in [0.0001, 0.001, 0.005, 0.01, 0.1]; the weight decay rate is in [1e-6, ..., 1e-2]. For NeuralCP, the batch size is in [32, 64, 128, 256, 512, 1024]; the learning rate is in [0.0001, 0.001, 0.005, 0.01, 0.1].

D. Additional Results

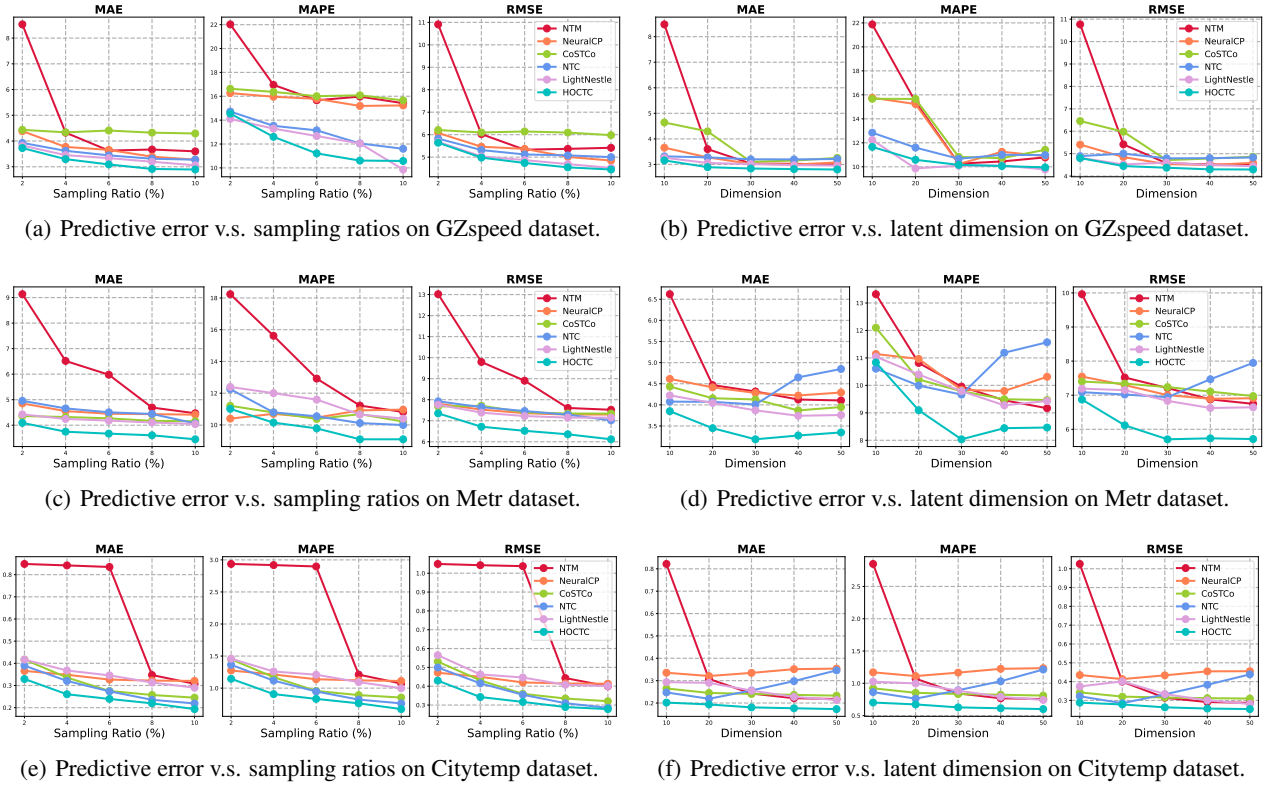
To further evaluate our model's performance on recommendation tensors, We have added 7 recently proposed methods (Lang et al., 2021; Chen et al., 2021; Wang et al., 2021b; Cheng & Xue, 2021; Mao et al., 2023; Zhu et al., 2023; Wang et al., 2023) in the field of recommendation and CTR prediction in 2021-2023, and report their results (together with ours) in the Table 5. As can be seen, even when compared with these advanced approaches proposed recently, our approach still maintains a competitive performance, with the MAE metric being 3-25% lower across 4 datasets.

Table 5. Comparative Analysis of Model Performance(MAE) Across Different Recommendation Datasets

DATA/MODEL	AOANET	EDCN	MASKNET	SAM	FINALMLP	FINALNET	GDCN	HOCTC
SG	0.0811	0.0814	0.0838	0.0711	0.0801	0.0832	0.0726	**0.0668**
BEAUTY	0.8330	0.8226	0.8348	0.8646	0.7986	0.8213	0.7945	**0.7772**
MOVIES&TV	0.6858	0.6938	0.6973	0.7818	0.6941	0.7285	0.7001	**0.6693**
GOWALLA	0.6382	0.6401	0.6410	0.6289	0.5700	0.6390	0.5466	**0.4652**

The new attention mechanism devised in our work, i.e., attention with expanded queries, can be deemed as a general type of GNN defined on bipartite graphs: the expanded query vectors (nodes) are one partite, the key vectors (nodes) are the other partite, and the weights of the edges crossing the two partites are exactly the attention weights. Then our attention scheme in fact enforces message passing between the L queries and the N keys. To enrich our comparisons, we therefore selected several GNN based models (Li et al., 2019; Guo et al., 2021; Jiang et al., 2022). As shown in the Table 6, when compared with these recent GNN models developed for recommendation tasks, our method continues to demonstrate competitive performance. Specifically, the MAE of our method is 3-27% lower compared to these models across four sparse recommendation tensor data widely utilized in the literature.

Figure 4 reports how the performance of our method varies with some hyper-parameters, including embedding dimensions, sampling ratios, and the number of extra queries.



(a) Predictive error v.s. sampling ratios on GZspeed dataset.

(b) Predictive error v.s. latent dimension on GZspeed dataset.

(c) Predictive error v.s. sampling ratios on Metr dataset.

(d) Predictive error v.s. latent dimension on Metr dataset.

(e) Predictive error v.s. sampling ratios on Citytemp dataset.

(f) Predictive error v.s. latent dimension on Citytemp dataset.

Figure 4. Hyper-Parameter Study.

Table 6. Comparative Analysis of Model Performance(MAE) with GNN-based models

MODEL/DATA	SG	BEAUTY	MOVIE&TV	GOWALLA
FI-GNN	0.0856	0.8271	0.8181	0.6426
DG-ENN	0.0744	0.8049	0.7731	0.6014
TGIN	0.0702	0.8041	0.7092	0.5687
HOCTC	**0.0668**	**0.7772**	**0.6693**	**0.4652**