SET REPRESENTATION AUXILIARY LEARNING WITH ADVERSARIAL ENCODING PERTURBATION AND OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sets are a fundamental data structure, and learning their vectorized representations is crucial for many computational problems. Existing methods typically focus on *intra-set* properties such as permutation invariance and cardinality independence. While effective at preserving basic *intra-set* semantics, these approaches may be insufficient in explicitly modeling *inter-set correlations*, which are critical for tasks requiring fine-grained comparisons between sets. In this work, we propose SRAL, a *Set Representation Auxiliary Learning* framework for capturing inter-set correlations that is compatible with various downstream tasks. SRAL conceptualizes sets as high-dimensional distributions and leverages the 2-Sliced-Wasserstein distance to derive their distributional discrepancies into set representation encoding. More importantly, we introduce a novel adversarial auxiliary learning scheme. Instead of manipulating the input data, e.g., element dropout/addition, our method introduces adversarial perturbations at the feature level. Through min-max optimization, we compel the model to achieve robustness against worst-case perturbations. Our theoretical analysis shows that this objective, in expectation, directly optimizes for the set-wise Wasserstein distances, forcing the model to learn highly discriminative representations. Comprehensive evaluations across four downstream tasks examine SRAL's performance relative to baseline methods, showing consistent effectiveness in both inter-set relation-sensitive retrieval and intra-set information-oriented processing tasks.

1 INTRODUCTION

Set-structured data are prevalent in practice, as they represent complex data objects composed of simpler units (Vargas-Calderón, 2025). With the rapid development of machine learning techniques, learning vectorized representations for sets is crucial (NaderiAlizadeh & Singh, 2025). It not only benefits a variety of emerging applications, e.g., similar group matching in social networks (Shen et al., 2012; Tang & Liu, 2022) and object retrieval in vector databases (Lee et al., 2019; Wang et al., 2022), but also demonstrates learning-based potential in addressing classical data management challenges, including data cleaning (Hadjieleftheriou et al., 2008; Wang & He, 2019), data integration (Dong & Rekatsinas, 2018; Ge et al., 2019), and set similarity processing (Li et al., 2021; Zeakis et al., 2022).

Traditional algorithms for retrieving set-structured data are typically *rule-based*. For example, Set Similarity Join (Arasu et al., 2006; Mann et al., 2016) computes pairs of sets from data partitions to filter candidates with a given score threshold. Methods in this area design iterative filter-verification frameworks (Deng et al., 2017; Zeakis et al., 2022) or propose index-based solutions (Li et al., 2021) to improve efficiency. However, these algorithm algorithms often lack *predictive capability* to generalize from data to make inferences. To address this limitation, *Set Representation Learning* has emerged. Unlike naive approaches that merely sum element features, its fundamental goal is to learn a holistic and fixed-size embedding that captures the intrinsic semantics of the whole unordered collection. This capability is essential for facilitating complex downstream tasks, for example, in E-commerce, enabling the model to interpret a product bundle as a cohesive semantic unit, e.g., a "camping kit", rather than a loose collection of items, thereby improving recommendation accuracy. While early machine learning approaches leveraged kernel methods (Jebara et al., 2004; Gretton et al., 2006; Boiman et al., 2008; Muandet et al., 2012), recent deep learning methods focus on capturing

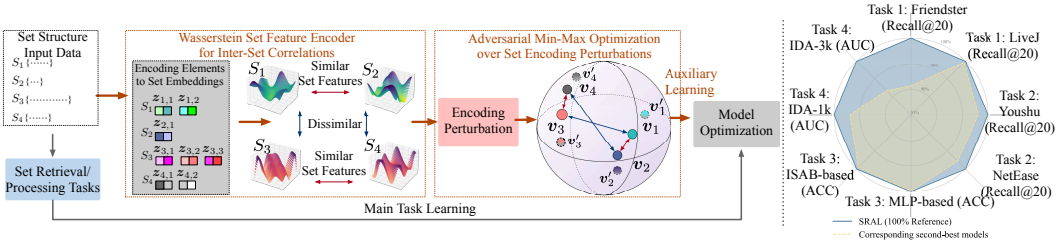


Figure 1: SRAL captures inter-set correlations for adversarial optimization (left); normalized ratios of second-best methods over SRAL are reported due to varying metric scales (right).

diverse set cardinalities and ensuring *invariance to element permutations* (Xu et al., 2025; Skianis et al., 2020; Zaheer et al., 2017; Lee et al., 2019; Kim, 2022; Wang et al., 2023). They design a composition of permutation-equivariant neural network backbones and aggregation mechanisms to preserve set semantics that outperform traditional feature pooling techniques (Mialon et al., 2021; Murphy et al., 2018; Zhang et al., 2020). Detailed discussions are reported in Appendix B.

Despite these advancements, existing methods predominantly focus on fulfilling *intra-set properties*. While permutation invariance and cardinality independence are essential, these models are often insufficient for *explicitly capturing the rich and complex inter-set correlations*. This however is crucial for comparing set-wise similarities and differences in certain scenarios. For instance, in set retrieval tasks, identifying the nearest neighbors for a query set inherently relies on a nuanced understanding of set-to-set relationships (Naderializadeh et al., 2021; Zhang et al., 2020). Similarly, in E-commerce applications, product bundles with overlapping items appeal to similar customer segments, where capturing the subtle relationships between different product sets can potentially enhance recommendation accuracy (Ma et al., 2022). Such set-wise correlation knowledge may not be naturally inherited through the mere encoding of intra-set properties, thus creating a notable gap in representation capability.

To bridge this gap, we introduce SRAL, a *Set Representation Auxiliary Learning* framework designed to learn representations with a focus to capture inter-set correlations. Our framework formulates model optimization as a flexible auxiliary objective built upon two synergistic components. ❶ First, we introduce a novel set encoder grounded in optimal transport theory. By conceptualizing sets as empirical distributions, this encoder measures the set-wise distributional discrepancy using the 2-Sliced-Wasserstein metric (Lahn et al., 2025; Rabin et al., 2011; Bonneel et al., 2015; Villani, 2009) and derives such distance information into the set embeddings. ❷ Second, and more critically, we propose an effective adversarial auxiliary learning scheme to forge discriminative representations by training the model to resist worst-case encoding perturbations. Specifically, our approach departs from conventional data manipulation strategies, e.g., element dropout/addition or subset sampling, by introducing adversarial perturbations directly to the set features. Our theoretical analysis demonstrates that the learning over such perturbation is, in expectation, equivalent to optimizing for the 2-Sliced-Wasserstein distances between the underlying perturbed distributions. Then by training the model to be robust against worst-case perturbations via a min-max optimization, we compel the encoder to learn high quality set representations. As illustrated in Figure 1(left), these components jointly work to consolidate the fine-grained learning of inter-set correlations within a unified auxiliary framework.

To validate SRAL, we conduct extensive experiments across four diverse downstream tasks. These tasks cover both inter-set relation-sensitive retrieval applications, i.e., *Learning to Rank Set Similarity* and *Bundle Recommendation*, and intra-set information-oriented processing ones, i.e., *Point Cloud Classification* and *Topic Set Expansion*. As visualized in Figure 1 (right), SRAL not only excels at its primary goal of capturing inter-set correlations but also performs well in processing intra-set information, which underscores the effectiveness and versatility of our framework.

2 Preliminaries

Problem Description. Let $\mathcal{S} = \{S_1, S_2, \cdots, S_m\}$ be a corpus of m sets. The elements in these sets are drawn from a finite universe $\mathcal{E} = \{e_1, e_2, \cdots, e_n\}$, which contains n unique elements. Each element $e_j \in \mathcal{E}$ is associated with a d-dimensional embedding vector $\boldsymbol{z}_j \in \{\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_n\}$. In practice, depending on the availability of raw features, these embeddings $\boldsymbol{z}_j$ are either initialized

using pre-trained feature extractors, e.g., word embeddings, or randomly initialized to be learned from scratch. The primary objective of *Set Representation Learning* is to learn an encoder function that maps any set S_i to a fixed-size vector embedding $\boldsymbol{v}_i$, which preserves inter-set correlation information to facilitate downstream set retrieval and processing tasks. All notations are explained in Appendix C.

Illustrative Example. Take bundle recommendation as a concrete example. A set S_i represents a product bundle, e.g., a "camping kit", containing items such as a tent, a sleeping bag, a kettle, etc. The necessity of Set Representation Learning lies in its ability to compress these irregular structures into a unified and fixed-size embedding. This enables downstream systems to interpret the bundle as a coherent semantic entity that captures the collective information of the items, rather than treating them as a loose collection of isolated products. Technically, the learned set representations should accommodate varying cardinalities, i.e., different numbers of items across bundles, and ensure permutation invariance, meaning the resulting embedding remains identical regardless of the order in which the items are listed. Beyond the specific e-commerce scenario illustrated above, Set Representation Learning has found broad applications across diverse domains. These include computer vision, for aggregating multi-view images or video frames (Wang et al., 2022), bioinformatics, where proteins are modeled as sets of residues for property prediction (NaderiAlizadeh & Singh, 2025), and computational pathology, where whole-slide images are treated as sets of patches for cancer diagnosis (Ilse et al., 2018; Carbonneau et al., 2018).

Distributional Distance Measurement. *Wasserstein distance*, derived from optimal transport (OT), provides a good measure for quantifying distributional distance (Tran et al., 2025; Rabin et al., 2011; Bonneel et al., 2015; Lv et al., 2024). It measures the minimum "cost" required to transform one probability distribution into another. Formally, given a probability distribution P, let the random variable $\boldsymbol{X}$ follow the distribution P, i.e., $\boldsymbol{X} \sim P$, $\boldsymbol{X} \in \mathbb{R}^d$. For the projection function $\theta : \mathbb{R}^d \to \mathbb{R}$, P^θ represents the push-forward of P with θ in a one-dimensional space, defined as $P^\theta(\boldsymbol{Y}) = P(\boldsymbol{x} : \theta(\boldsymbol{x}) \in \boldsymbol{Y}) = P(\theta^{-1}(\boldsymbol{Y}))$. The α-Wasserstein distance between P and Q is defined using L_α transport cost (Villani, 2009):

$$D_\alpha(P, Q) = \Big(\inf_{g \in Plans(P,Q)} \int \|\boldsymbol{x} - g(\boldsymbol{x})\|^\alpha dP(\boldsymbol{x}) \Big)^{\frac{1}{\alpha}}, \ \alpha \geq 1, \tag{1}$$

where the infimum is taken over all transport plans between P and Q. If a minimizer exists, denoted by g^+, it is the solution to the OT problem. For *one-dimensional* distributions, a closed-form tranport solution for g^+ from Q to P exists: $g^+(x) := F_P^{-1}\big(F_Q(x)\big)$, where F and $F_P^{-1}(x)$ denote the cumulative distribution function (CDF) and the quantile function of P, respectively. To prevent numerical intractability for *high-dimensional* cases (Kolouri et al., 2019), the alternative metric of *α-Sliced-Wasserstein distance* has been recently studied (Rabin et al., 2011; Bonneel et al., 2015; Deshpande et al., 2019):

$$SD_\alpha(P, Q) = \Big(\int_{\mathbb{S}^{d-1}} \big(D_\alpha(P^\theta, Q^\theta)\big)^\alpha d\theta \Big)^{\frac{1}{\alpha}}, \ \alpha \geq 1. \tag{2}$$

$\mathbb{S}^{d-1}$ denotes the unit d-dimensional hypersphere. The projection is $\theta(\boldsymbol{x}) = \boldsymbol{w}^\mathsf{T}\boldsymbol{x}$ where $\boldsymbol{w} \in \mathbb{S}^{d-1}$ is a unit vector in $\mathbb{R}^d$. P^θ is the push-forward of P using $\theta(\boldsymbol{x})$. This metric satisfies *positive-definiteness*, *symmetry*, and *triangle inequality* (Kolouri et al., 2016; 2019), qualifying it for similarity/distance measurement (Yang et al., 2024; Kantorovich, 1960).

3 SRAL Framework

3.1 Overview

Our SRAL framework is designed to capture inter-set correlations through the auxiliary learning objective, which seamlessly integrates with various downstream set-based tasks. The overall learning objective combines a scenario-specific main task loss $\mathcal{L}_{Main}$ with our auxiliary loss $\mathcal{L}_{Aux}$, weighted by a hyper-parameter λ_1. The complete objective function is:

$$\mathcal{L} = \mathcal{L}_{Main} + \lambda_1 \mathcal{L}_{Aux} + \lambda_2 ||\Xi||_2^2, \tag{3}$$

where $||\Xi||_2^2$ is an L2-regularizer on all trainable parameters Ξ to prevent over-fitting. The core of SRAL consists of two synergistic components. First, we introduce a novel set encoder based on the 2-Sliced-Wasserstein distance, which represents sets as empirical distributions and embeds them based on their distance to a learned reference. Second, we propose a potent adversarial auxiliary

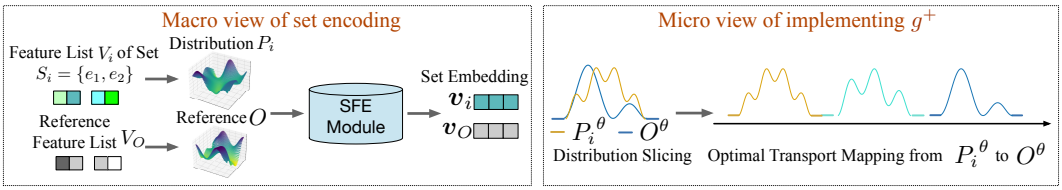


Figure 2: SFE module illustrations: a macro and micro views of set representation encoding.

learning scheme that perturbs the set features and encoding. By training the model to be robust against worst-case perturbations, this scheme forces the encoder to learn highly discriminative representations that capture fine-grained inter-set relationships.

3.2 2-Sliced-Wasserstein Set Representation Encoding

Our approaches begins by treating each set S_i as an empirical distribution. Specifically, a set S_i is represented by a list of feature vectors of its elements, $V_i = [\boldsymbol{z}_{i,k} \in \mathbb{R}^d]_{k=1}^{|S_i|}$, where $\boldsymbol{z}_{i,k}$ is the embedding of element $e_{i,k} \in S_i$. We assume these feature vectors are sampled from an underlying true data distribution P_i, and the observed features V_i define the empirical distribution $\widehat{P}_i$. In this work, we consider $\widehat{P}_i$ to be a valid approximation of P_i and unify them as P_i to simplify notation. To learn stable and discriminative set representations, our encoder leverages the distributional distance between an input set and a learnable *reference distribution* O. This reference design, characterized by H trainable embeddings $V_O = [\boldsymbol{z}_h \in \mathbb{R}^d]_{h=1}^H$, as a strategy shared by several previous methods (Naderializadeh et al., 2021; Guo et al., 2021a; Mialon et al., 2021), serves as a learnable "origin" in the set embedding space. Specifically, these embeddings are initialized as model parameters and are updated via backpropagation during the training process, adapting globally to minimize the overall objective function.

3.2.1 Set Feature Encoder (SFE)

Feature Mapping via Optimal Transport. Directly computing the Wasserstein distance between high-dimensional distributions P_i and O is computational intractable. Therefore, we employ the 2-Sliced-Wasserstein distance, which circumvents this issue by slicing the high-dimensional distributions into multiple one-dimensional ones. Each slice is defined by a linear projection $\theta(\boldsymbol{x}) = \boldsymbol{w}^\mathsf{T}\boldsymbol{x}$, where the unit vector $\boldsymbol{w}$ is uniformly sampled from the hypersphere $\mathbb{S}^{d-1}$. This projection reduces P_i and O to their one-dimensional counterpart ${P_i}^\theta$ and O^θ. Recall the early introduction in § 2, these sliced one-dimensional distributions are compatible with the closed-form solution. As stated in Peyré et al. (2019); Kolouri et al. (2019); Naderializadeh et al. (2021); Deshpande et al. (2019), the optimal transport map g^+, from the reference slice O^θ to the input slice ${P_i}^\theta$ can be defined as:

$$g^+(x^\theta|V_i^\theta) = F^{-1}_{{P_i}^\theta}\big(F_{O^\theta}(x^\theta)\big) \text{ where } x^\theta \in V_O^\theta, \tag{4}$$

where $V_i^\theta = [\boldsymbol{w}^\mathsf{T}\boldsymbol{z}_{i,k}]_{k=1}^{|S_i|}$ and $V_O^\theta = [\boldsymbol{w}^\mathsf{T}\boldsymbol{z}_h]_{h=1}^H$ denote the sliced features of ${P_i}^\theta$ and O^θ, respectively. Here the CDF of O^θ is: $F_{O^\theta}(x) = \frac{1}{H}\sum_{h=1}^H \delta(x \geq \boldsymbol{w}^\mathsf{T}\boldsymbol{z}_h)$, where δ returns 1 for zero input and 0 otherwise. We include the formal statement of Eq. (4) for readability in Appendix D.

For empirical distributions defined by samples, this theoretical solution can be intuitively interpreted as a *rank-matching* procedure. The term $F_{O^\theta}(x^\theta)$ essentially computes the rank percentile of x^θ within the sorted values of V_O^θ, and $F^{-1}_{{P_i}^\theta}(\cdot)$ then finds the value with the corresponding rank percentile in V_i^θ. This leads to the practical implementation of g^+ detailed as follows: $\forall x^\theta \in V_O^\theta$, let $\tau(x|V)$ be the rank of value x in the ordered set V. The mapping procedure for g^+ is executed as:

$$g^+(x^\theta|V_i^\theta) = \underset{x' \in V_i^\theta}{\arg\min}\left(\tau(x'|V_i^\theta) \geq \frac{|S_i|}{H} \cdot \tau(x^\theta|V_O^\theta)\right). \tag{5}$$

We provide its derivation in Appendix D. The indicator function $\tau(\cdot)$ can be pre-processed using the `argsort` function for V_i^θ and the `sort` function for V_O^θ. Note that to accommodate cardinality differences of set features (i.e., when $H \neq |S_i|$ in the second case of Eq. (3), we employ *linear interpolation*, which effectively preserves the data continuity.

Constructing SFE Module. To avoid the infinite projections as required by theory in Eq. (2), we employ a Monte Carlo approximation (Kingma et al., 2013; Metropolis et al., 1953) with R random projections. Let $\Theta = [\boldsymbol{w}_r]_{r=1}^{R}$ denote all R sampled projection vectors. Consequently, our Set Feature Encoder (SFE) module constructs the set embedding v_i by aggregating the results from all projections. For each projection $\boldsymbol{w}_r$, we implement the map $g^{+}(\boldsymbol{w}_r^{\mathsf{T}}\boldsymbol{z}_h|V_i^{\theta_r})$ for every point $\boldsymbol{z}_h$ in the reference set. Following these implementations with illustration in Figure 2, SFE concatenates along the innermost dimension to output the set embedding $\boldsymbol{v}_i$ as follows:

$$\texttt{SFE}(V_i, V_O|\Theta) = \underset{r=1..R;h=1..H}{\text{Concat}} \left(g^{+}(\boldsymbol{w}_r^{\mathsf{T}}\boldsymbol{z}_h|V_i^{\theta_r}) \right). \tag{6}$$

By capturing the inter-set distributional correlations, this SFE module provides the critical encoding process that we leverage in our adversarial auxiliary learning framework.

3.3 Adversarial Set Encoding Perturbation and Optimization

To provide adaptability across downstream tasks, we incorporate the auxiliary learning objective $\mathcal{L}_{Aux}$ within a self-supervised paradigm. The central idea is to produce stable representations when the constituent distribution is slightly perturbed. Rather than using simple random noise, we employ an adversarial framework that compels the model to be robust against worst-case perturbations, forcing it to learn more informative inter-set information.

3.3.1 Self-perturbation on Set Feature Encoding

We begin by generating perturbed samples for each set. For the input set S_i with features $V_i = [\boldsymbol{z}_{i,k}]_{k=1}^{|S_i|}$, the perturbation is constructed by adding small random noise to the element embeddings, where the norm of the noise is bounded by a hyper-parameter π:

$$\boldsymbol{z}'_{i,k} = \boldsymbol{z}_{i,k} + \boldsymbol{\epsilon}'_{i,k}, \text{ where } \boldsymbol{\epsilon}'_{i,k} \text{ is drawn from } \|\boldsymbol{\epsilon}\|_2 \leq \pi. \tag{7}$$

Based on the modification of $\boldsymbol{z}_{i,k}$, we generate a series of perturbed distribution features, i.e., $V'_i = [\boldsymbol{z}'_{i,k} \in \mathbb{R}^d]_{k=1}^{|S_i|}$. By feeding this perturbed set of elements into SFE, we obtain a perturbed set embedding $\boldsymbol{v}'_i = \texttt{SFE}(V'_i, V_O|\Theta)$. This would allow for a more fine-grained simulation of feature variations, thereby providing high-quality perturbated samples for the subsequent learning process.

3.3.2 Adversarial Min-Max Optimization

With the perturbed embeddings, we employ a self-supervised learning paradigm for optimization. To achieve this, we firstly construct two perturbed views for each set S_i, yielding a pair of positive set embeddings $\boldsymbol{v}'_i$ and $\boldsymbol{v}''_i$. We then implement with the InfoNCE loss (Oord et al., 2018) as follows:

$$\mathcal{L}_{wd} = \sum_{S_i \in \mathcal{S}} -\log \frac{\exp(-||\boldsymbol{v}'_i - \boldsymbol{v}''_i||_2/\psi)}{\sum_{S_j \in \mathcal{S}} \exp(-||\boldsymbol{v}'_i - \boldsymbol{v}''_j||_2/\psi)}, \tag{8}$$

where ψ is a hyper-parameter. Typically, this loss term promotes consistency between the perturbed representations of the same set S_i, while maximizing the Euclidean distance between embeddings of different sets, e.g., S_i and S_j. This however raises a question: does this learning objective, which operates on perturbed embeddings, disrupt the SFE module's fundamental capability derived from the Sliced-Wasserstein metric? Therefore, we introduce Remark 1, which demonstrates that optimizing this objective is, in expectation, equivalent to optimizing the objective directly on the underlying distributional distances, thus consolidating the learning capability of our set embedding approach.

Remark 1. *Let P'_i and P''_i denote two perturbed distributions corresponding to the input set S_i, yielding perturbed set embeddings $\boldsymbol{v}'_i$ and $\boldsymbol{v}''_i$. $\mathcal{S}$, ψ, SD_2 denote the set database, a hyper-parameter, and the 2-Sliced-Wasserstein distance. For $\forall S_i \in \mathcal{S}$, we have:*

$$\mathbb{E}\left[\frac{\exp(-||\boldsymbol{v}'_i - \boldsymbol{v}''_i||_2/\psi)}{\sum_{S_j \in \mathcal{S}} \exp(-||\boldsymbol{v}'_i - \boldsymbol{v}''_j||_2/\psi)}\right] = \frac{\exp(-||(SD_2(P'_i, P''_i))||_2/\psi)}{\sum_{S_j \in \mathcal{S}} \exp(-||(SD_2(P'_i, P''_j))||_2/\psi)}. \tag{9}$$

With proofs in Appendix D, Remark 1 demonstrates that, the Euclidean distance between the perturbed set embeddings is positively correlated in expectation with 2-Sliced-Wasserstein distance between their underlying distributions. This suggests that: by minimizing $\mathcal{L}_{wd}$ in the embedding space, we are implicitly optimizing for the alignment of distributional distances between sets, thereby enabling the model to capture fine-grained, distribution-based inter-set correlations.

To further enhance representation robustness, we go beyond merely resisting noise but actively seek "worst-case" perturbations that maximally disrupts the representation consistency. To this end, we elevate the self-supervised objective to an adversarial min-max problem. Specifically, we seek an adversarial perturbation increment $\boldsymbol{\sigma}$; $\boldsymbol{\sigma}$ is shared and applied to the perturbed features V_i' and V_i'' that are generated earlier from Eq. (7). Our goal is to find a "worst-case" perturbation that maximizes the loss in Eq. (8). Consequently, let Ξ denote the all trainable parameters, our final auxiliary learning objective, $\mathcal{L}_{Aux} = \max_{\|\boldsymbol{\sigma}\|_2 \leq \pi} \mathcal{L}_{wd}(\Xi, \boldsymbol{\sigma})$, is to be minimized under such worst-case perturbation. This derives the min-max optimization problem as follows:

$$\min_{\Xi} \max_{\|\boldsymbol{\sigma}\|_2 \leq \pi} \mathcal{L}_{wd}(\Xi, \boldsymbol{\sigma}) = \min_{\Xi} \max_{\|\boldsymbol{\sigma}\|_2 \leq \pi} \mathcal{L}_{wd}(\{\boldsymbol{v}_i^{\sigma}\}_{S_i \in \mathcal{S}}), \text{ where } \boldsymbol{v}_i^{\sigma} = \texttt{SFE}(V_i' + \sigma, V_O | \Theta). \quad (10)$$

However, in practice, it could be computational infeasible to exactly solve the min-max problem in Eq. (10). Therefore, we employ a first-order approximation (Goodfellow et al., 2015) to efficiently estimate the optimal perturbation. We linearize the loss function $\mathcal{L}_{wd}$ by taking its first-order Taylor expansion around $\boldsymbol{\sigma} = \mathbf{0}$:

$$\mathcal{L}_{wd}(\Xi, \boldsymbol{\sigma}) \approx \mathcal{L}_{wd}(\Xi, \mathbf{0}) + \sigma^{\mathsf{T}} \nabla_{\boldsymbol{\epsilon}} \mathcal{L}_{wd}(\Xi; \boldsymbol{\epsilon})\big|_{\boldsymbol{\epsilon}=\mathbf{0}}. \quad (11)$$

Maximizing this linear approximation under the norm constraint has a closed-form solution where the perturbation $\boldsymbol{\sigma}$ is aligned with the gradient direction. This motivates us to approximate the worst-case perturbation with a single step of gradient ascent and thus decompose the min-max problem with the following two alternating steps:

1. **Inner Maximization to Find $\boldsymbol{\sigma}$**: With the model parameters Ξ fixed, we compute the gradient of the loss with respect to a small perturbation $\boldsymbol{\epsilon}$ evaluated at $\boldsymbol{\epsilon} = 0$:
$$g_{\boldsymbol{\sigma}} = \nabla_{\boldsymbol{\epsilon}} \mathcal{L}_{wd}(\Xi; \boldsymbol{\epsilon})\big|_{\boldsymbol{\epsilon}=\mathbf{0}}. \quad (12)$$
We then update the perturbation along the gradient direction to obtain an initial adversarial perturbation $\hat{\boldsymbol{\sigma}}$ as $\hat{\boldsymbol{\sigma}} = \eta \cdot g_{\boldsymbol{\sigma}}$, where η is the ascent step size. To satisfy the constraint, we project this perturbation back onto the ℓ_2 ball of radius π:
$$\boldsymbol{\sigma} = \hat{\boldsymbol{\sigma}} \cdot \min\left(1, \frac{\pi}{\|\hat{\boldsymbol{\sigma}}\|_2}\right). \quad (13)$$
2. **Outer Minimization to Update Ξ**: After identifying the adversarial perturbation increment $\boldsymbol{\sigma}$, we apply it to generate the final perturbed embeddings, e.g., $\boldsymbol{v}_i^{\sigma} = \texttt{SFE}(V' + \boldsymbol{\sigma}, V_O | \Theta)$. Then, we compute the adversarial loss based on these examples, which is further combined with the main task loss $\mathcal{L}_{main}$. With β as the learning rate, the entire model's parameters Ξ are updated via the gradient descent:
$$\Xi \leftarrow \Xi - \beta \cdot \nabla_{\Xi}(\mathcal{L}_{Main} + \lambda_1 \mathcal{L}_{adv} + \lambda_2 ||\Xi||_2^2). \quad (14)$$

Through this procedure, our method learns to acquire stable set representations from deliberate feature perturbation, with the following remark to formalize this intuition (proofs are in Appendix D).

Remark 2. *Our min-max optimization objective in Eq. (10) is approximately equivalent to an implicit regularization of the* `SFE`*'s local Lipschitz continuity for representation stability.*

Generally, our method differentiates from conventional methods by adversarial perturbing the encoding process, and thus showcases its effectiveness and convergence efficiency in § 4.3.2.

4 Experiments

4.1 Setups

Tasks and Datasets. To thoroughly evaluate our proposed SRAL framework, we conduct comprehensive experiments across four diverse downstream tasks. These tasks are selected to span two primary categories: inter-set relation-sensitive tasks, i.e., Set Similarity Ranking, Bundle Recommendation, which require a deep understanding of correlations between different sets, and intra-set information-oriented tasks, i.e., Point Cloud Classification, Topic Set Expansion, which focus on processing the internal contents of a single set. All tasks and datasets conducted in our experiments are in a supervised setting. Due to the page limit, we supplement the detailed introduction of tasks and datasets in Appendix E.

- **Task 1: Learning to Rank Set Similarity.** This task evaluates the model's ability to learn from known similar set pairs and predict new associations by ranking sets based on the Euclidean

Table 1: Performance comparison for Tasks 1 (left) and 2 (right). Best and second-best cases are highlighted. Statistically significant improvements ($p < 0.05$) are marked with *.

Task 1: Set Similarity Learning

Model	Friendster				LIVEJ			
	R@20	N@20	R@100	N@100	R@20	N@20	R@100	N@100
SAP	72.41	68.13	85.32	72.15	79.86	77.75	86.94	85.34
SMP	70.78	68.99	81.61	72.22	78.95	76.70	86.83	84.65
DeepSet	63.20	60.75	76.60	69.89	75.45	74.55	83.31	79.76
RepSet	80.63	76.56	86.49	74.92	82.15	79.63	88.41	83.12
SAtt	77.52	71.92	87.51	75.21	83.79	81.73	91.39	85.07
PoT	82.44	81.85	86.96	81.47	83.18	84.25	89.33	86.45
Set2Box	67.35	69.73	73.46	70.33	77.24	75.89	85.12	82.34
OTKE	79.53	73.68	86.64	79.59	81.45	79.82	87.95	85.10
DIEM	82.49	81.40	88.36	81.56	83.95	84.92	89.88	87.15
PSWE	83.05	84.26	88.59	85.77	83.52	84.61	89.48	86.67
FSPool	79.90	81.96	87.76	84.41	85.36	87.17	93.07	90.29
FSW	83.58	84.39	88.52	85.81	84.19	85.04	89.95	87.23
SRAL	**91.57**	**92.22**	**94.53**	**93.01**	**87.56**	**89.31**	**92.93**	**91.25**
Gain	9.56%*	9.28%*	6.71%*	8.39%*	2.58%*	2.46%*	-0.15%	1.06%*

Task 2: Bundle Recommendation

Model	Youshu				NetEase			
	R@20	N@20	R@100	N@100	R@20	N@20	R@100	N@100
MFBPR	19.97	11.67	44.33	17.95	5.21	2.98	14.15	4.92
DSBRec	20.46	12.03	45.34	18.12	5.51	3.04	14.76	5.14
DAM	20.83	11.99	45.58	18.38	5.54	3.11	14.98	5.12
BundleNet	22.85	11.90	47.84	19.19	6.17	3.44	16.26	5.83
BGCN	25.22	14.54	49.38	21.18	7.04	3.91	17.25	6.51
CrossCBR	26.41	16.55	51.90	23.30	7.21	4.08	18.32	6.77
SRAL$^+$	**26.92**	**16.95**	**52.18**	**23.64**	**7.37**	**4.21**	**18.66**	**7.01**
Gain	1.93%*	2.42%*	0.54%*	1.46%*	2.22%*	3.19%*	1.86%*	3.54%*

distance of their learned embeddings. We use two large-scale, real-world social network datasets: *Friendster* (Yang & Leskovec, 2015) and *LIVEJ* (Mislove et al.).

- **Task 2: Bundle Representation Learning for Recommendation.** In this e-commerce scenario, the goal is to recommend bundles (sets of items) to users. Effective bundle representation is crucial for prediction accuracy. Experiments are conducted on the *Youshu* (Chen et al., 2019) book bundle dataset and the *NetEase* (Cao et al., 2017) for music playlist recommendation.
- **Task 3: Point Cloud Processing.** A point cloud is a set of 3D data points representing an object's surface. This task aims to classify the object category based on its point cloud representation. We utilize the standard *ModelNet40* (Wu et al., 2015) benchmark dataset.
- **Task 4: Topic Set Expansion.** Given a small seed set of keywords describing a topic, the objective is to expand this set with other semantically related keywords from a vocabulary. We use the *LDA-1k*, *LDA-3k*, and *LDA-5k* datasets (Zaheer et al., 2017) with different sizes and scopes.

Implementation Configurations. For all tasks, we employ a consistent procedure for data preparation and training: we partition the datasets into training and testing sets with an 8:2 ratio. The training set is then further subdivided into training and validation subsets using an 8:2 split to facilitate hyper-parameter tuning. All reported results are the average of five independent runs. Hyper-parameters and experiment configurations for reproducing are reported in Appendix E.1-E.2.

Competing Methods. We compare SRAL against a comprehensive list of baselines for four tasks. For the general set representation tasks, i.e., Set Similarity, Point Cloud Classification, and Topic Set Expansion, we include classic pooling methods *SAP* (Lin et al., 2013), *SMP* (Lin et al., 2013) and state-of-the-art deep learning models such as *DeepSet* (Zaheer et al., 2017), *RepSet* (Skianis et al., 2020), *SAtt* (Lee et al., 2019), *PoT* (Guo et al., 2021a), *Set2Box* (Lee et al., 2022), *OTKE* (Mialon et al., 2021), *DIEM* (Kim, 2022), *FSPool* (Zhang et al., 2020), *PSWE* (Naderializadeh et al., 2021), and *FSW* (Amir & Dym, 2025). For the specialized Bundle Recommendation task, we compare against established and recent recommendation models: *MFBPR* (Rendle et al., 2012), *DSBRec* (Zaheer et al., 2017), *DAM* (Chen et al., 2019), *BundleNet* (Deng et al., 2020), *BGCN* (Chang et al., 2020), and the state-of-the-art *CrossCBR* (Ma et al., 2022). For this task, we integrate SRAL into CrossCBR to enhance its bundle embedding module, denoting it as *SRAL+*. Detailed descriptions are in Appendix E.

4.2 Evaluation Results and Discussions

Task 1: Learning to Rank Set Similarity. We evaluate all models on Task 1, where sets are ranked based on Euclidean distance in their embedding space. The results for Recall (R@k) and NDCG (N@k) are summarized in Table 1(left) with threefold observations. ❶ Deep set encoders, e.g., DeepSet, RepSet, sAtt, generally outperform conventional pooling methods like SAP and SMP. Moreover, general OT-based methods, e.g., PoT and OTKE, and particularly Sliced-Wasserstein-based methods such as PSWE and FSW, further elevate the overall performance. ❷ Our proposed SRAL consistently achieves state-of-the-art performance across nearly all metrics and datasets. On the Friendster dataset, SRAL improvements over the best baseline ranging from 6.71% to 9.56%. On LIVEJ, it also demonstrates advantages, particularly in capturing top-ranked items with gains of 2.58% in R@20 and 2.46% in N@20. ❸ The vast majority of SRAL's performance gains are statistically significant, confirmed by a Wilcoxon signed-rank test (Conover, 1999) at a 95% confidence level.

Table 2: Performance comparison for Task 3: Point Cloud Processing.

Backbone	SAP	SMP	RepSet	SAtt	PoT	Set2Box	OTKE	DIEM	PSWE	FSPool	FSW	**SRAL**	**Gains**
MLP	57.65	86.35	83.45	85.89	85.20	82.15	85.92	85.58	86.41	85.76	86.38	**86.53**	+0.14%
ISAB	85.45	86.82	86.05	86.78	86.55	85.88	86.70	86.72	86.85	86.88	86.93	**87.31**	+0.44%*

Table 3: Performance comparison for Task 4: Topic Set Expansion.

Data	SAP	SMP	DeepSet	RepSet	SAtt	PoT	Set2Box	OTKE	DIEM	PSWE	FSPool	FSW	**SRAL**	**Gains**
LDA-1k	54.34	67.21	54.98	57.32	58.55	58.94	50.59	62.95	63.58	58.36	75.67	64.56	**80.94**	+6.96%*
LDA-3k	51.95	74.40	51.96	58.33	77.48	73.40	64.98	77.59	75.67	78.44	70.57	79.67	**87.93**	+10.37%*
LDA-5k	51.34	80.65	52.05	61.39	74.59	75.11	65.67	72.57	76.96	78.81	71.16	80.94	**86.20**	+6.50%*

This highlights the stability and reliability of our model's improvements. For complete results including standard deviations, please refer to Appendix E.3.3.

Task 2: Bundle Representation Learning for Recommendation. The problem of this task is formulated with a prediction function between bundle and user embeddings. Our analyses from Table 1(right) yield three key findings. ❶ SRAL$^+$ consistently outperforms baseline models on both Youshu and NetEase datasets, where the improvements are statistically significant across all metrics. ❷ We attribute this to the strength of CrossCBR backbone, which already excels at capturing user-bundle collaborative filtering signals (Rendle et al., 2012; He et al., 2017). Our SRAL$^+$ enhances this by providing a complementary signal, an explicit and semantically rich representation of the internal bundle structure. ❸ We recognize that this task ultimately aims to predict the user-bundle matching probability via learning the user-bundle interactions; therefore, this suggests that a more integrated learning framework is a promising direction for future work, where the both bundle semantics and collaborative patterns are co-optimized.

Task 3: Point Cloud Processing. We evaluate various set encoding methods on top of two distinct backbones: a Multi-layer Perceptron (MLP) and the more advanced Induced Set Attention Block (ISAB) (Lee et al., 2019). The test accuracies are reported in Table 2. ❶ We observe that SRAL achieves favorable results when paired with both backbones. With the standard MLP, SRAL yields a competitive accuracy of 86.53%. When the ISAB backbone is employed, the performance margin widens, and SRAL further reaches a statistically significant improvement of +0.44% over the second-best model FSW. ❷ While using ISAB backbone clearly improves performance for all methods, SRAL provides an additional performance lift in both settings. This indicates that SRAL is effective with less impact from the backbone model selection and compatible with different feature extractors.

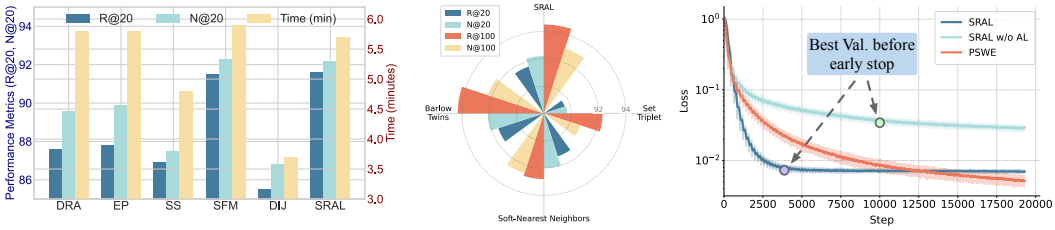
Task 4: Topic Set Expansion. With vocabularies encoded by word2vec (Mikolov, 2013), models classify elements based on their semantic similarity to a given query set. As shown in Table 3, SRAL consistently outperforms baselines in distinguishing intra-set semantics for topic expansion with AUC improvements ranging from 6.50% to 10.37%, demonstrating its effectiveness in capturing latent set semantics for classification. More importantly, in addition to its competitive performance on inter-set relation-sensitive applications (Task 1 & 2), SRAL also achieves superior results in intra-set information-oriented ones (Task 3 & 4), demonstrating its versatility as a good auxiliary representation learner.

4.3 Empirical Analyses of SRAL

In this section, we delve into the design choices of SRAL, through a series of empirical analyses. We use the Friendster dataset from Task 1 as the primary testbed for these studies.

4.3.1 Study of SRAL Set Encoding Design

Comparison with Other Designs. The design of our Set Feature Encoding (`SFE`) module is fundamentally based on the Wasserstein distance, which benefits from its implicit regularization to the embedding distance. To validate this design choice, we compare its performance against other common distributional distance metrics, namely the Kullback-Leibler (KL) Divergence, Jensen-Shannon (JS) Divergence, and the Sinkhorn Distance. Implementation details are reported in Appendix E.7. The results in Figure 3(A) lead to two primary conclusions. ❶ SRAL and the Sinkhorn-based method emerge as the most competitive approaches. This highlights the overall superiority of the Wasserstein distance for this task. Furthermore, SRAL achieves the best performance, which we attribute to our novel `SFE` module that can capture complex set similarities more effectively than simpler methods. ❷ We observe that our method incurs a relatively higher computational cost, i.e.,



(A) Results of varying perturbation approaches (B) Results of varying SSL losses (C) Loss curves

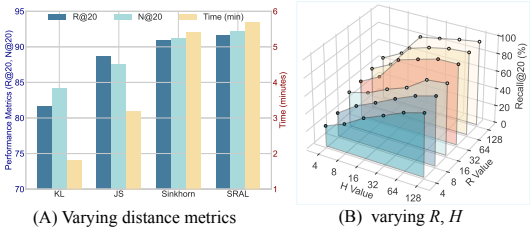
Figure 4: Results of varying (A) perturbation approaches, (B) SSL losses, and (C) loss curves.

Table 4: Ablation study.

Variant	Task 1		Task 2		Task 3		Task 4
	R@20	N@20	R@20	N@20	M-ACC	I-ACC	AUC
w/o SFE	67.02 (-26.81%)	69.38 (-24.77%)	25.31 (-5.98%)	15.14 (-10.68%)	66.45 (-23.20%)	72.48 (-16.99%)	73.39 (-16.54%)
w/o LI	75.45 (-17.60%)	74.29 (-19.44%)	25.96 (-3.57%)	15.83 (-6.61%)	74.77 (-13.59%)	79.75 (-8.66%)	72.44 (-17.62%)
w/o AEPO	77.13 (-15.77%)	79.42 (-13.88%)	26.22 (-2.60%)	16.38 (-3.36%)	86.56 (+0.03%)	86.46 (-0.97%)	66.21 (-24.70%)
w/o AL	87.38 (-4.58%)	88.86 (-3.64%)	26.94 (+0.07%)	16.97 (+0.11%)	86.37 (-0.18%)	87.27 (-0.04%)	83.53 (-5.00%)
SRAL	**91.57**	**92.22**	**26.92**	**16.95**	**86.53**	**87.31**	**87.93**

time cost per training epoch. However, considering the performance improvement it delivers, we view this as an acceptable trade-off between model effectiveness and computational efficiency.

Setting R and H. We investigate the impact of Monte Carlo trials R and the reference feature length H, with the results presented in Figure 3(B). ❶ As observed, the model's performance improves as both R and H increase. The performance is particularly sensitive to the R. For instance, when fixing H at 32, increasing R from 4 to 32 leads to a significant rise in Recall@20 from 41.23% to 91.57%, which is attributed to a more accurate approximation of the cumulative distribution. But the performance improvement shows diminishing marginal returns at larger parameter values, especially as the performance curve begins to plateau for R>32. ❷ We also notice that varying H based on fixed R provide less significant impact. Therefore, we select H=32 and R=128 as our final configuration to strike an ideal balance between model performance and resource consumption.



(A) Varying distance metrics (B) varying R, H

Figure 3: Results of varying (A) distance metrics and (B) R and H values.

4.3.2 Study of Adversarial Encoding Perturbation and Optimization

Perturbation Approaches. We introduce: ❶ Element-level Perturbation as random dropout or addition of set elements (DRA), and element replacement (EP). ❷ Set-level Perturbation as subset sampling (SS) (Yun et al., 2019), and set feature mixing (SFM) (Zhang et al., 2017). ❸ Noise Injection as direct injection of noise (DIJ) into the encoded set embeddings. As shown in Figure 4(A), both SRAL and SFM achieve competitive performance, outperforming simple data manipulation methods like EP and SS. This suggests that perturbing the set features yields more fine-grained and effective augmentations. Furthermore, compared to DIJ that directly perturbs final set embeddings, our strategy of perturbing the intermediate encoding process demonstrates superior effectiveness. Finally, while SRAL's performance is on par with SFM, it is slightly more efficient (5.7min vs. 5.9min). This is because our approach perturbs individual sets, whereas SFM requires mixing multiple sets, allowing SRAL to achieve an better balance between performance and efficiency.

Implementation of Self-supervised Learning Loss. While the specific self-supervised learning (SSL) objective in Eq. (8) is not the primary focus of this paper, we investigate the flexibility of our framework with other loss functions. We tested several alternatives, including Set Triplet Loss, Soft-Nearest Neighbors Loss (Frosst et al., 2019), and Barlow Twins Loss (Zbontar et al., 2021). The detailed formulations of these objectives can be found in the Appendix appendix E.7. As presented in Figure 4(B), the results demonstrate that our framework is compatible with various SSL objective functions, as these variants achieve competitive performance.

Convergence Analysis of Adversarial Encoding Optimization. Figure 4(C) presents the loss curves to illustrate model convergence. Although SRAL may have a higher complexity, compared to

its ablated version without adversarial learning ("w/o AL"), we observe that SRAL converges faster and deeper in practice, reaching its best validation performance early in training (we continued the training after this point to provide complete loss trajectories). The baseline method while requires a longer training duration and exhibits greater volatility. This experiment demonstrates SRAL's practical efficiency, effectively mitigating potential concerns about its computational overhead due to its rapid convergence. A more detailed analysis of model scalability is provided in Appendix E.8.

4.3.3 Ablation Study

We evaluate several ablation variants across four tasks. Due to space constraints, we report results for Friendster, Youshu, LDA-3k on Tasks 1, 3 (R@20 and N@20) and Task 4 (AUC), while Task 2 reports accuracy with MLP and ISAB backbones (M-ACC and I-ACC). From Table 4, we observe that: ❶ replacing `SFE` module with mean-pooling ("w/o `SFE`") results in substantial performance degradation despite the auxiliary learning with our encoding perturbation. This clearly validates the effectiveness and necessity of our `SFE` for capturing complex set features. ❷ Substituting linear interpolation in Eq. (3) with a two-layer MLP ("w/o `LI`") yields less reliable dimension completion compared to our implementation. ❸ Variant "w/o `AEPO`" retains the auxiliary learning objective but disables the adversarial optimization step. Specifically, we utilize the inner InfoNCE loss but remove the min-max strategy that generates worst-case perturbations via gradient ascent. The results show that it negatively impacts performance, particularly on Task 4 where the AUC drops by 24.70%. ❹ In contrast, removing the entire auxiliary learning ("w/o `AL`") by training the model solely with the main task supervision, results in a marginal performance improvement for Task 2.

We attribute this to the specific nature of SRAL$^+$, where its backcone model CrossCBR's graph structure is already highly optimized for capturing *user-bundle collaborative signals*. Our adversarial learning, which focuses on the bundle-side representation by modeling inter-bundle relationships, may provide limited complementary information for interaction prediction goal. This suggests a more integrated framework for this interaction-centric scenario.

5 Conclusion and Future Work

We introduced SRAL, an effective auxiliary learning framework for Set Representation Learning that is compatible with set-based retrieval and processing problems. We first introduce a set encoder based on the 2-Sliced-Wasserstein distance, which effectively captures distributional discrepancies between sets. We then propose a adversarial learning paradigm that strengthens representations by generating and optimizating worst-case perturbations to set features and encoding. Experiments demonstrate the performance superiority of SRAL over competing methods and the efficacy of its constituent design components. A promising future direction is to investigate the integration of SRAL with *approximate nearest neighbor* search algorithms, such as set-vector index construction algorithm and index-based search approaches (Johnson et al., 2019), for efficient online retrieval settings. Another direction is to explore the synergy with Large Language Models (LLMs), such as utilizing context-aware LLM representations as element features, or employing SRAL as a structural adapter to generate compact soft prompts for LLM-based set reasoning.

Statement

Ethics Statement. This research fully adheres to the ICLR Code of Ethics (`https://iclr.cc/public/CodeOfEthics`). We are committed to contributing to society and human well-being by developing AI methods that enhance drug safety assessment, ultimately benefiting public health. We have carefully considered potential harm and implemented safeguards to minimize negative consequences, particularly with regard to patient privacy and data protection. All datasets used are properly licensed (CC-BY-SA 4.0), and we have ensured appropriate attribution to original creators.

Reproducibility Statement. To ensure full reproducibility of our results, we provide comprehensive implementation details throughout the paper and appendices. Section 4 and Appendix E contains complete specifications of all experimental details, including data splits, hyperparameters, and optimizer configurations. We report error bars and statistical significance measures for all experimental results to ensure robust evaluation. Computational resource requirements, including hardware specifications,

memory usage, and execution times, are also reported to facilitate reproduction on similar systems. The codes are available here: https://anonymous.4open.science/r/SRAL_code-691E/.

REFERENCES

Tal Amir and Nadav Dym. Fourier sliced-wasserstein embedding for multisets and measures. In *The Thirteenth International Conference on Learning Representations*, 2025.

Arvind Arasu, Venkatesh Ganti, and Raghav Kaushik. Efficient exact set-similarity joins. In *VLDB*, pp. 918–929, 2006.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pp. 214–223. PMLR, 2017.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In *CVPR*, pp. 1–8. IEEE, 2008.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *JMIV*, 51(1):22–45, 2015.

Da Cao, Liqiang Nie, Xiangnan He, Xiaochi Wei, Shunzhi Zhu, and Tat-Seng Chua. Embedding factorization models for jointly recommending items and user generated lists. In *Proceedings of SIGIR*, pp. 585–594, 2017.

Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 77: 329–353, 2018.

Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. Bundle recommendation with graph convolutional networks. In *Proceedings of SIGIR*, pp. 1673–1676, 2020.

Liang Chen, Yang Liu, Xiangnan He, Lianli Gao, and Zibin Zheng. Matching user with item set: Collaborative bundle recommendation with deep attention network. In *IJCAI*, pp. 2095–2101, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pp. 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *NeurIPS*, 33:22243–22255, 2020b.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.

William Jay Conover. *Practical nonparametric statistics*, volume 350. john wiley & sons, 1999.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *NeurIPS*, 26, 2013.

Dong Deng, Albert Kim, Samuel Madden, and Michael Stonebraker. Silkmoth: An efficient method for finding related sets with maximum matching constraints. *VLDB*, 10(10), 2017.

Qilin Deng, Kai Wang, Minghao Zhao, Zhene Zou, Runze Wu, Jianrong Tao, Changjie Fan, and Liang Chen. Personalized bundle recommendation in online games. In *Proceedings of CIKM*, pp. 2381–2388, 2020.

Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. Max-sliced wasserstein distance and its use for gans. In *CVPR*, pp. 10648–10656, 2019.

Tzanko Donchev and Elza Farkhi. Stability and euler approximation of one-sided lipschitz differential inclusions. *SIAM journal on control and optimization*, 36(2):780–796, 1998.

Xin Luna Dong and Theodoros Rekatsinas. Data integration and machine learning: A natural synergy. In *SIGMOD*, pp. 1645–1650, 2018.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *CVPR*, pp. 9588–9597, 2021.

Dominik Maria Endres and Johannes E Schindelin. A new metric for probability distributions. *IEEE TIT*, 49(7):1858–1860, 2003.

Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *International conference on machine learning*, pp. 2012–2020. PMLR, 2019.

Chang Ge, Yinan Li, Eric Eilebrecht, Badrish Chandramouli, and Donald Kossmann. Speculative distributed csv data parsing for big data analytics. In *Proceedings of SIGMOD*, pp. 883–899, 2019.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *NeurIPS*, 19, 2006.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33:21271–21284, 2020.

Dandan Guo, Long Tian, Minghe Zhang, Mingyuan Zhou, and Hongyuan Zha. Learning prototype-oriented set representations for meta-learning. *arXiv preprint arXiv:2110.09140*, 2021a.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199, 2021b.

Marios Hadjieleftheriou, Xiaohui Yu, Nick Koudas, and Divesh Srivastava. Hashed samples: selectivity estimators for set similarity selection queries. *VLDB*, 1(1):201–212, 2008.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pp. 9729–9738, 2020.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW*, pp. 173–182, 2017.

Ernst Hellinger. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271, 1909.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pp. 2127–2136. PMLR, 2018.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, pp. 4651–4664. PMLR, 2021.

Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *JMLR*, 5:819–844, 2004.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Leonid V Kantorovich. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.

Minyoung Kim. Differentiable expectation-maximization for set representation learning. In *International Conference on Learning Representations*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Soheil Kolouri, Yang Zou, and Gustavo K Rohde. Sliced wasserstein kernels for probability distributions. In *CVPR*, pp. 5258–5267, 2016.

Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo Rohde. Generalized sliced wasserstein distances. *NeurIPS*, 32, 2019.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Nathaniel Lahn, Sharath Raghvendra, Emma Saarinen, and Pouyan Shirzadian. Scalable approximation algorithms for p-wasserstein distance and its variants. In *Forty-second International Conference on Machine Learning*, 2025.

Dong Bok Lee, Seanie Lee, Kenji Kawaguchi, Yunji Kim, Jihwan Bang, Jung-Woo Ha, and Sung Ju Hwang. Self-supervised set representation learning for unsupervised meta-learning. In *ICLR*, 2023.

Geon Lee, Chanyoung Park, and Kijung Shin. Set2box: Similarity preserving representation learning for sets. In *ICDM*, pp. 1023–1028. IEEE, 2022.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, pp. 3744–3753. PMLR, 2019.

Yifan Li, Xiaohui Yu, and Nick Koudas. Les3: learning-based exact set similarity search. *Proceedings of VLDB*, 14(11):2073–2086, 2021.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus, and Fabian-Robert Stöter. Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *ICML*, pp. 4104–4113. PMLR, 2019.

Jiaming Lv, Haoyuan Yang, and Peihua Li. Wasserstein distance rivals kullback-leibler divergence for knowledge distillation. *Advances in Neural Information Processing Systems*, 37:65445–65475, 2024.

Yunshan Ma, Yingzhi He, An Zhang, Xiang Wang, and Tat-Seng Chua. Crosscbr: cross-view contrastive learning for bundle recommendation. In *Proceedings of SIGKDD*, pp. 1233–1241, 2022.

Willi Mann, Nikolaus Augsten, and Panagiotis Bouros. An empirical evaluation of set similarity join techniques. *Proceedings of VLDB*, 9(9):636–647, 2016.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Grégoire Mialon, Dexiong Chen, Alexandre d'Aspremont, and Julien Mairal. A trainable optimal transport embedding for feature aggregation and its relationship to attention. In *ICLR*, 2021.

Tomas Mikolov. Efficient estimation of word representations in vector space. *ICLR*, 3781, 2013.

Alan Mislove, Massimiliano Marcon, P. Krishna Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In Constantine Dovrolis and Matthew Roughan (eds.), *Proceedings of SIGCOMM*, pp. 29–42.

Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. *NeurIPS*, 25, 2012.

Ryan L Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. In *ICLR*, 2018.

Navid NaderiAlizadeh and Rohit Singh. Aggregating residue-level protein language model embeddings with optimal transport. *Bioinformatics Advances*, 5(1):vbaf060, 2025.

Navid Naderializadeh, Joseph F Comer, Reed Andrews, Heiko Hoffmann, and Soheil Kolouri. Pooling by sliced-wasserstein embedding. *NeurIPS*, 34:3389–3400, 2021.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE CVPR*, pp. 652–660, 2017.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *SSVM*, pp. 435–446. Springer, 2011.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

S Hamid Rezatofighi, Roman Kaskman, Farbod T Motlagh, Qinfeng Shi, Daniel Cremers, Laura Leal-Taixé, and Ian Reid. Deep perm-set net: Learn to predict sets with unknown permutation and cardinality using deep neural networks. *arXiv preprint arXiv:1805.00613*, 2018.

Yilin Shen, Thang N Dinh, Huiyuan Zhang, and My T Thai. Interest-matching information propagation in multiple online social networks. In *CIKM*, pp. 1824–1828, 2012.

Konstantinos Skianis, Giannis Nikolentzos, Stratis Limnios, and Michalis Vazirgiannis. Rep the set: Neural networks for learning set representations. In *AISTATS*, pp. 1410–1420. PMLR, 2020.

Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (ToG)*, 34(4):1–11, 2015.

Lei Tang and Huan Liu. *Community detection and mining in social media*. Springer Nature, 2022.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *ICLR*, 2018.

Viet-Hoang Tran, Trang Pham, Tho Tran, Minh Khoi Nguyen Nhat, Thanh Chu, Tam Le, and Tan M Nguyen. Tree-sliced wasserstein distance: A geometric perspective. 2025.

Vladimir Vargas-Calderón. Quantum deep sets and sequences. *Quantum Machine Intelligence*, 7(2): 65, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

Pei Wang and Yeye He. Uni-detect: A unified approach to automated error detection in tables. In *SIGMOD*, pp. 811–828, 2019.

Peihao Wang, Shenghao Yang, Shu Li, Zhangyang Wang, and Pan Li. Polynomial width is sufficient for set representation with high-dimensional features. *arXiv preprint arXiv:2307.04001*, 2023.

Zhaoqing Wang, Qiang Li, Guoxin Zhang, Pengfei Wan, Wen Zheng, Nannan Wang, Mingming Gong, and Tongliang Liu. Exploring set similarity for dense self-supervised representation learning. In *CVPR*, pp. 16590–16599, 2022.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE CVPR*, pp. 1912–1920, 2015.

Fred Xu, Song Jiang, Zijie Huang, Xiao Luo, Shichang Zhang, Adrian Chen, and Yizhou Sun. Fuse: Measure-theoretic compact fuzzy set representation for taxonomy expansion. *arXiv preprint arXiv:2506.08409*, 2025.

Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.*, 42(1):181–213, 2015.

Peilun Yang, Hanchen Wang, Jianye Yang, Zhengping Qian, Ying Zhang, and Xuemin Lin. Deep learning approaches for similarity computation: A survey. *TKDE*, 2024.

Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *SIGIR*, pp. 1294–1303, 2022.

Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In *CVPR*, pp. 14263–14272, 2021.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *NeurIPS*, 30, 2017.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, pp. 12310–12320. PMLR, 2021.

Alexandros Zeakis, Dimitrios Skoutas, Dimitris Sacharidis, Odysseas Papapetrou, and Manolis Koubarakis. Tokenjoin: Efficient filtering for set similarity join with maximumweighted bipartite matching. *VLDB*, 16(4):790–802, 2022.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2017.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. Learning representations of sets through optimized permutations. In *ICLR*, 2019.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. Fspool: Learning set representations with featurewise sort pooling. *ICLR*, 2020.

A THE USE OF LARGE LANGUAGE MODELS (LLMS)

In preparing this manuscript, we employed a Large Language Model (LLM) as an assistive tool mainly for refining prose for grammar, clarity, and conciseness. The LLM did not contribute to research conception, methodological development, result analysis, or scientific conclusions. All content, including final text and code, was thoroughly reviewed, edited, and validated by the authors, who retain full responsibility for the work's accuracy and integrity.

B EXTENDED RELATED WORK

Set Representation Learning. Traditional set-based problems, e.g., *Set Similarity Join* (Arasu et al., 2006; Mann et al., 2016) and *Exact Set Similarity Search* (Li et al., 2021), leverage rule-based algorithms to process set data (Deng et al., 2017; Zeakis et al., 2022). For predictive capability, early machine learning approaches consider kernel-based methods (Jebara et al., 2004; Gretton et al., 2006; Boiman et al., 2008; Muandet et al., 2012). Other approaches learn permutation-invariant mapping into a Hilbert space with pooling operations (Zaheer et al., 2017; Skianis et al., 2020; Murphy et al., 2018; Lee et al., 2019; Zhang et al., 2020; 2019). While some models learn predictive optimal set permutation (Zhang et al., 2019; Rezatofighi et al., 2018), Zhang et al. (2020) defines canonical ordering via sorting techniques. Lee et al. (2019) leverages Transformer (Vaswani et al., 2017) for sets and Jaegle et al. (2021) designs pooling operations using multi-head attention where cross-attention serves as the permutation-invariant function. Mialon et al. (2021) learns set embeddings by reducing differences between references and inputs. Lee et al. (2022) leverages box embedding for sets and Xu et al. (2025) considers the fuzzy set representation concept. Additionally, (Skianis et al., 2020) is one of the earliest approaches to explicitly incorporate Optimal Transport into set representation learning. To capture complex set structures, RepSet generates embeddings by solving a Bipartite Matching problem that optimally aligns the elements of an input set with a collection of learnable hidden sets. However, unlike this high-dimensional formulation, our SRAL employs the 2-Sliced-Wasserstein metric to bypass the computational bottleneck of exact matching via effective approximation. Recent models consider meta-learning in addition (Guo et al., 2021a; Lee et al., 2023) and (Naderializadeh et al., 2021) proposes an effective pooling mechanism to achieve state-of-the-art performance. Amir & Dym (2025) proposes the FSW embedding to rigorously provide injective or bi-Lipschitz representations for multisets and measures. Vargas-Calderón (2025) proposes the quantum deep set modeling and NaderiAlizadeh & Singh (2025) leverages set representations for protein language modeling.

Self-supervised Learning. It generates augmented data samples and derives supervision signals to guide learning. Early methods (Chen et al., 2020c; He et al., 2020; Dwibedi et al., 2021; Chen et al., 2020a;b) establish contrastive frameworks to encourage similar views to cluster while pushing dissimilar ones apart. Non-contrastive methods effectively learn meaningful representations without feature collapse. For example, Grill et al. (2020) uses dual networks with a momentum encoder to minimize positive pair distances; Zbontar et al. (2021) optimizes cross-correlation matrices between views to approximate identity matrices. While most approaches manipulate input data, recent work modifies latent space representations directly, potentially reducing bias with less efforts (Yüksel et al., 2021; Yu et al., 2022). Our work investigates perturbating the set encoding process, providing theoretical guarantees and outperforming existing augmentation designs.

Distributional Distance Measurement. Existing approaches for quantifying distributional differences include Kullback-Leibler Divergence (Kullback & Leibler, 1951), Jensen-Shannon Divergence (Endres & Schindelin, 2003)) and Hellinger distance (Hellinger, 1909), etc. The Wasserstein metric (Kantorovich, 1960), due to its strong theoretical properties, has gained attention recently (Arjovsky et al., 2017; Tolstikhin et al., 2018). However, computing Wasserstein distances becomes computationally intractable for high-dimensional distributions. While optimization-based solutions exist (Cuturi, 2013; Solomon et al., 2015), the *Sliced-Wasserstein distance* (Bonneel et al., 2015; Kolouri et al., 2019) offers superior efficiency by projecting high-dimensional distributions onto multiple one-dimensional spaces and leveraging closed-form solutions. This approach has been successfully applied to numerous domains (Kolouri et al., 2016; Liutkus et al., 2019; Naderializadeh et al., 2021) and forms the foundation of our method for modeling inter-set distributional similarities.

C NOTATION EXPLANATIONS

We explain the key notations in Table 1.

Table 1: Notation Table.

Notation	Description
Basic Sets and Elements	
$\mathcal{E} = \{e_1, \cdots, e_n\}$	All elements.
$\mathcal{S} = \{S_1, \cdots, S_m\}$	A corpus of sets.
$S_i = \{e_{i,k} \in \mathcal{E}\}_{k=1}^{\lvert S_i \rvert}$	Set S_i containing $\lvert S_i \rvert$ elements.
$V_i = [\boldsymbol{z}_{i,k} \in \mathbb{R}^d]_{k=1}^{\lvert S_i \rvert}$	Features for the input set S_i.
$\boldsymbol{v}_i$	The final vector embedding for the set S_i.
Distributions and Optimal Transport	
P_i	The underlying distribution of elements for set S_i.
O	A trainable reference distribution.
$V_O = [\boldsymbol{z}_h \in \mathbb{R}^d]_{h=1}^{H}$	Features for the reference distribution O.
P_g	The push-forward distribution of P by function g.
$D_\alpha(P, Q)$	The α-Wasserstein distance between distributions P and Q.
$SD_\alpha(P, Q)$	The α-Sliced-Wasserstein Distance between P and Q.
g^+	The optimal transport plan.
$\mathbb{S}^{d-1}$	The unit d-dimensional hypersphere.
Set Feature Encoder (SFE)	
$\theta(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$	A linear projection function parameterized by $\boldsymbol{w} \in \mathbb{S}^{d-1}$.
P_i^θ, O^θ	1D distributions obtained by projecting P_i and O using θ.
$V_i^\theta = [\boldsymbol{w}^\top \boldsymbol{z}_{i,k}]_{k=1}^{\lvert S_i \rvert}$	Sliced features of P_i^θ.
$V_O^\theta = [\boldsymbol{w}^\top \boldsymbol{z}_h]_{h=1}^{H}$	Sliced features of O^θ.
$F_P(\cdot)$	The Cumulative Distribution Function (CDF) of a distribution P.
$F_P^{-1}(\cdot)$	The quantile function (inverse CDF) of a distribution P.
$\tau(x^\theta \vert V_O^\theta)$	The rank of a projected value x^θ within the sorted values of V_O^θ.
$g^+(x^\theta \vert V_i^\theta)$	The optimal transport solver from reference slice O^θ to input slice P_i^θ.
$\mathtt{SFE}(V_i, V_O \vert \Theta)$	The Set Feature Encoder function.
Adversarial Perturbation and Optimization	
P_i', P_i''	Two perturbed distributions generated from the input set S_i.
V_i'	Perturbed feature matrix of set S_i.
$\boldsymbol{v}_i', \boldsymbol{v}_i''$	A pair of perturbed set embeddings for set S_i.
$\boldsymbol{\epsilon}_{i,k}'$	Random noise vector added to an element embedding.
$\boldsymbol{\sigma}$	The adversarial perturbation increment vector.
$\hat{\boldsymbol{\sigma}}$	The initial adversarial perturbation before projection.
$\boldsymbol{g}_\sigma$	The gradient of the loss with respect to the perturbation.
$\boldsymbol{v}_i^\sigma$	The final adversarially perturbed set embedding.
Losses and Hyperparameters	
$\mathcal{L}_{Main}$	The main loss for the downstream task.
$\mathcal{L}_{Aux}$	The auxiliary loss from our framework.
$\mathcal{L}_{wd}$	The InfoNCE-based loss for self-supervised learning.
Ξ	The set of all trainable model parameters.
R, H	Number of distribution slices and feature size of the reference distribution.
π	The norm constraint (radius) for perturbations.
η, β	Step size for inner maximization and learning rate for outer minimization.
ψ	The temperature hyperparameter for the InfoNCE loss.
λ_1, λ_2	Hyperparameters to balance the main loss, auxiliary loss, and regularizer.

D THEORETICAL PROOFS

Complete Formulation for Eq. (4). Let $V_i^\theta = [\boldsymbol{w}^\top \boldsymbol{z}_{i,k}]_{k=1}^{\lvert S_i \rvert}$ and $V_O^\theta = [\boldsymbol{w}^\top \boldsymbol{z}_h]_{h=1}^{H}$ denote the sliced features of ${P_i}^\theta$ and O^θ, respectively. Let any projected input $x^\theta \in V_O^\theta$, the OT solver is formulated as:

$$g^+(x^\theta \vert V_i^\theta) = F_{{P_i}^\theta}^{-1}\big(F_{O^\theta}(x^\theta)\big). \tag{1}$$

Proof. The transport distance, as one candidate plan for the α-Wasserstein distance (Eq.(1) in § 2), can be computed as:

$$\begin{aligned}\text{Distance}(g^+) &= \Big(\int_{\mathbb{R}} \|x^\theta - g^+(x^\theta|V_i^\theta)\|^\alpha dO^\theta(x^\theta)\Big)^{\frac{1}{\alpha}} \\ &= \Big(\int_{\mathbb{R}} \|x^\theta - F^{-1}_{P_i{}^\theta}\big(F_{O^\theta}(x^\theta)\big)\|^\alpha dO^\theta(x^\theta)\Big)^{\frac{1}{\alpha}} \\ &= \Big(\int_0^1 \|F^{-1}_{O^\theta}(y) - F^{-1}_{P_i{}^\theta}(y)\|^\alpha dy\Big)^{\frac{1}{\alpha}} \\ &= D_\alpha(O^\theta, P_i{}^\theta),\end{aligned} \tag{2}$$

which equals to $D_\alpha(P_i{}^\theta, O^\theta)$ because of the symmetry property. This proves to be the optimal distance for these two slices. □

Mapping Procedure for g^+. $\forall x^\theta \in V_O^\theta$, let $\tau(x|V)$ be the rank of value x in the ordered set V. The mapping procedure for g^+ is executed as follows:

$$g^+(x^\theta|V_i^\theta) = \begin{cases} \underset{x'\in V_i^\theta}{\arg\min}\Big(\tau(x'|V_i^\theta) = \tau(x^\theta|V_O^\theta)\Big) \text{ if } H = |S_i|; \\ \underset{x'\in V_i^\theta}{\arg\min}\Big(\tau(x'|V_i^\theta) \geq \dfrac{|S_i|}{H}\cdot\tau(x^\theta|V_O^\theta)\Big) \text{ if } H \neq |S_i|. \end{cases} \tag{3}$$

Proof. The empirical distributions of V_i^θ and V_O^θ, e.g.,

$$F_{O^\theta}(x^\theta) = \frac{1}{H}\sum\nolimits_{h=1}^{H}\delta(x^\theta \geq \boldsymbol{w}^\mathsf{T}\cdot\boldsymbol{z}_h), \tag{4}$$

are monotonically increasing. If $H = |S_i|$, we can firstly modify the original form of the optimal transport map $F^{-1}_{P_i{}^\theta}\big(F_{O^\theta}(x^\theta)\big)$ to:

$$g^+(x^\theta|V_i^\theta) = \underset{x'\in V_i^\theta}{\arg\min}\big(F_{P_i{}^\theta}(x') = y\big) \text{ where } y = F_{O^\theta}(x^\theta). \tag{5}$$

$\tau(x^\theta|V_O^\theta)$ is the ranking of each input x^θ in the *ascending sorting* of V_O^θ, and then we can quantitatively replace the term $F_{P_i{}^\theta}(\cdot)$.

$$g^+(x^\theta|V_i^\theta) = \underset{x'\in V_i^\theta}{\arg\min}\Big(\tau(x'|V_i^\theta) = \tau(x^\theta|V_O^\theta)\Big) \text{ if } H = |S_i|. \tag{6}$$

If $H \neq |S_i|$, in this work, we applies the linear interpolation to the sorted positions. Therefore, $\frac{|S_i|}{H}$ is set as the interpolation point to complete the proof. □

Remark 1. *Let P_i' and P_i'' denote two perturbed distributions corresponding to the input set S_i, yielding perturbed set embeddings $\boldsymbol{v}_i'$ and $\boldsymbol{v}_i''$. $\mathcal{S}$, ψ, SD_2 denote the set dataset, a hyper-parameter, and the 2-Sliced-Wasserstein distance. For $\forall S_i \in \mathcal{S}$, we have the following expected equation:*

$$\mathbb{E}\left[\frac{\exp(-||\boldsymbol{v}_i' - \boldsymbol{v}_i''||_2/\psi)}{\sum_{S_j\in\mathcal{S}}\exp(-||\boldsymbol{v}_i' - \boldsymbol{v}_j''||_2/\psi)}\right] = \frac{\exp(-||(SD_2(P_i', P_i''))||_2/\psi)}{\sum_{S_j\in\mathcal{S}}\exp(-||(SD_2(P_i', P_j''))||_2/\psi)}. \tag{7}$$

Proof of Remark 1. Let $\boldsymbol{v}_j^{\theta_r}$ denote the intermediate embedding derived by the function θ_r. $||$ denotes the concatenation operation. Firstly, after setting α=2, we have:

$$\begin{aligned}\|\boldsymbol{v}_i' - \boldsymbol{v}_j''\|_2 \propto &\Big\|\frac{1}{R}\sum\nolimits_{r=1}^{R}\frac{1}{H}\sum\nolimits_{h=1}^{H}\Big(g^+(\boldsymbol{w}_r^\mathsf{T}\boldsymbol{z}_h|V_i'^{\theta_r}) - g^+(\boldsymbol{w}_r^\mathsf{T}\boldsymbol{z}_h|V_j''^{\theta_r})\Big)^2\Big\|_2 \\ = &\Big\|\frac{1}{R}\sum\nolimits_{r=1}^{R}\frac{1}{H}\sum\nolimits_{h=1}^{H}\Big(F^{-1}_{P_i'^{\theta_r}}\big(F_{O^{\theta_r}}(w_r^\mathsf{T}z_h)\big) - F^{-1}_{P_j''^{\theta_r}}\big(F_{O^{\theta_r}}(w_r^\mathsf{T}z_h)\big)\Big)^2\Big\|_2\end{aligned} \tag{8}$$

The inner summation is substituted by recognizing the relationship between the sample mean and the integral for an empirical distribution. Given an empirical distribution $P(x) = \frac{1}{H}\sum_{h=1}^{H}\delta(x \geq X_h)$,

the sample mean equals the integral: $\frac{1}{H}\sum_{h=1}^{H} f(X_h) = \int f(x)dP(x)$. Then we apply this to the inner summation, with $t = w_r^{\mathsf{T}} z_h$ being a sample from O^{θ_r}.

$$\begin{aligned} &\frac{1}{H}\sum\nolimits_{h=1}^{H}\left(F^{-1}_{P_i'^{\theta_r}}\left(F_{O^{\theta_r}}(t)\right) - F^{-1}_{P_j''^{\theta_r}}\left(F_{O^{\theta_r}}(t)\right)\right)^2 \\ &= \int\left(F^{-1}_{P_i'^{\theta_r}}\left(F_{O^{\theta_r}}(t)\right) - F^{-1}_{P_j''^{\theta_r}}\left(F_{O^{\theta_r}}(t)\right)\right)^2 dO^{\theta_r}(t) \end{aligned} \tag{9}$$

For the outer summation, as $R \to \infty$, we apply the Law of Large Numbers for integration over the hypersphere: $\frac{1}{R}\sum_{r=1}^{R} f(\theta_r) = \int_{\mathbb{S}^{d-1}} f(\theta)d\theta$. This indicates that the continuous form in expectation of above formulation can be derived as:

$$\begin{aligned} &= \left(\int_{\mathbb{S}^{d-1}}\int_{\mathbb{R}}\left(F^{-1}_{P_i'^{\theta}}\left(F_{O^{\theta}}(t)\right) - F^{-1}_{P_j''^{\theta}}\left(F_{O^{\theta}}(t)\right)\right)^2 dO^{\theta}(t)d\theta\right)^{1/2} \\ &= \left(\int_{\mathbb{S}^{d-1}}\int_{0}^{1}\left(F^{-1}_{P_i'^{\theta}}(y) - F^{-1}_{P_j''^{\theta}}(y)\right)^2 dyd\theta\right)^{1/2} \\ &= \left(\int_{\mathbb{S}^{d-1}} D_2(P_i'^{\theta}, P_j''^{\theta})^2 d\theta\right)^{1/2} \\ &= SD_2(P_i', P_j''). \end{aligned} \tag{10}$$

And the computation for P_i', P_i'', and P_j'' are independent which thus complete the proofs. □

Remark 2. *Our min-max optimization objective in Eq. (10) is approximately equivalent to an implicit regularization of the* SFE*'s local Lipschitz continuity for representation stability.*

Proof of Remark 2. To find the perturbation $\boldsymbol{\epsilon}$ that maximizes the linear approximation under the constraint $||\boldsymbol{\sigma}||_2 \leq \pi$, we align the perturbation $\boldsymbol{\sigma}$ with the gradient direction. This yields the optimal perturbation as:

$$\boldsymbol{\sigma} = \pi \cdot \frac{g_{\boldsymbol{\sigma}}}{\|g_{\boldsymbol{\sigma}}\|_2} = \pi \cdot \frac{\nabla_{\boldsymbol{\epsilon}}\mathcal{L}_{wd}(\Xi;\boldsymbol{\epsilon})|_{\boldsymbol{\epsilon}=0}}{||\nabla_{\boldsymbol{\epsilon}}\mathcal{L}_{wd}(\Xi;\boldsymbol{\epsilon})|_{\boldsymbol{\epsilon}=0}||2}. \tag{11}$$

Substituting this $\boldsymbol{\sigma}$ back into the Taylor expansion of Eq. (11) gives the approximate value of the maximized loss as follows:

$$\max_{\|\boldsymbol{\sigma}\|_2 \leq \pi} \mathcal{L}_{wd}(\Xi, \boldsymbol{\sigma}) \approx \mathcal{L}_{wd}(\Xi, \mathbf{0}) + \pi \cdot ||\nabla\boldsymbol{\epsilon}\mathcal{L}_{wd}(\Xi;\boldsymbol{\epsilon})|_{\epsilon=0}||_2. \tag{12}$$

Thus, the original min-max objective can be approximated by the following minimization problem:

$$\min_{\Xi}\left(\mathcal{L}_{wd}(\Xi, \mathbf{0}) + \pi||\nabla\boldsymbol{\epsilon}\mathcal{L}_{wd}(\Xi;\mathbf{0})||_2\right). \tag{13}$$

This formulation explicitly shows that the adversarial objective encourages the minimization of not only the standard loss $\mathcal{L}_{wd}$, but also penalizes the norm of the loss's gradient $||\nabla\boldsymbol{\epsilon}\mathcal{L}_{wd}(\Xi;\mathbf{0})||_2$. This second term encourages the model to be less sensitive to input perturbations.

As for $\nabla\boldsymbol{\epsilon}\mathcal{L}_{wd}$ term, the loss $\mathcal{L}_{wd}$ is essentially a function of the set embeddings, e.g., $\boldsymbol{v}_i$, which depend on the perturbation $\boldsymbol{\epsilon}$ via the SFE module: $\boldsymbol{v}_i(\boldsymbol{\epsilon}) = \mathtt{SFE}(V' + \boldsymbol{\epsilon}, V_O|\Theta)$. We can apply the multivariable chain rule to decompose the gradient. For simplicity, we consider the gradient's dependence on a single embedding $\boldsymbol{v}_i(\boldsymbol{\epsilon})$. The total derivative of $\mathcal{L}_{wd}$ with respect to $\boldsymbol{\epsilon}$, evaluted at $\boldsymbol{\epsilon} = \mathbf{0}$, can be expressed as:

$$\nabla_{\boldsymbol{\epsilon}}\mathcal{L}_{wd}(\Xi;\mathbf{0}) = \sum_{i\in\text{Batch}} (J_{\mathtt{SFE}}(V_i'))^{\mathsf{T}}\nabla_{\boldsymbol{v}_i}\mathcal{L}_{wd}. \tag{14}$$

Here $\nabla_{\boldsymbol{v}_i}\mathcal{L}_{wd}$ is the loss gradient with respect to the vector embedding $\boldsymbol{v}_i$, and $J_{\mathtt{SFE}}(V_i')$ is the Jacobian matrix of the SFE function with respect to its input features. The regularizer in Eq. (13) penalizes the norm of this sum. For this norm to be small, the model is incentivized to reduce the norms of its constituent components, most notably the spectral norm of the SFE Jacobian, $\|J_{\mathtt{SFE}}\|_2$.

This penalty on the Jacobian norm is directly related to the local Lipschitz continuity of the SFE function. For a differentiable function, its local Lipschitz constant L over a region $\mathcal{V}$ is bounded by the supremum of the spectral norm of its Jacobian within that region:

$$L(\mathtt{SFE}, \mathcal{V}) = \sup_{V\in\mathcal{V}} ||J_{\mathtt{SFE}}(V)||_2. \tag{15}$$

By encouraging a smaller Jacobian norm, our adversarial optimization implicitly regularizes the SFE to have a smaller local Lipschitz constant. A smaller constant ensures that for any two nearby feature sets V_i and V_j, the distance between their embeddings is bounded:

$$||\texttt{SFE}(V_i) - \texttt{SFE}(V_j)||_2 \leq L(\texttt{SFE}) \cdot ||V_i - V_j||_2. \tag{16}$$

This property means that small, non-semantic perturbations to the input features will only result in small, bounded changes to the output embedding, which defines the representation stability Donchev & Farkhi (1998). □

E Supplementary Details of Experiments

E.1 Hyper-parameter Settings

We report all hyper-parameter settings in Table 2.

Table 2: Hyper-parameter settings.

	Task 1		**Task 2**		**Task 3**		**Task 4**		
	Friendster	LIVEJ	Youshu	NetEase	MLP	ISAB	LDA-1k	LDA-3k	LDA-5k
d	128	128	32	32	256	256	128	128	128
H	128	128	32	32	1024	1024	128	128	128
R	32	32	64	128	256	256	32	32	32
π	$5 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$5 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-2}$	$5 \cdot 10^{-1}$	$5 \cdot 10^{-1}$	$5 \cdot 10^{-1}$
ψ	$1 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$5 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$5 \cdot 10^{-1}$	$5 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-1}$
η	$1 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-1}$
β	$5 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{-3}$
λ_1	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	$5 \cdot 10^{-1}$	$5 \cdot 10^{-1}$	$5 \cdot 10^{-1}$
λ_2	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-5}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-5}$	$1 \cdot 10^{-5}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$	$1 \cdot 10^{-4}$

E.2 Experiment Configurations

We implement all models by Python 3.8 and PyTorch 1.12.0 with non-distributed training. We run on a Linux machine with 4 NVIDIA A100-PCIE-40GB GPUs and 10 vCPU Intel Xeon Processor (Skylake, IBRS). We adhere to the officially reported hyperparameter settings of all baselines and conduct a grid search to models without prescribed configurations. The learning rate is tuned in the range $\{10^{-4}, 10^{-3}, 10^{-2}\}$. Optimization for all models is performed using the default Adam optimizer (Kingma & Ba, 2015). All results are averaged based on five-fold evaluations.

E.3 Task 1: Learning to Rank Set Similarity

E.3.1 Dataset Descriptions

We incorporate two large real-world datasets, i.e., Friendster (Yang & Leskovec, 2015) and LIVEJ (Mislove et al.) for set similarity learning evaluation. We filter out sets with fewer than three elements. Specifically, Friendster is the dataset from an online social gaming site[1] where each set corresponds to a group membership. LIVEJ is a dataset from a free online community LiveJournal[2].

With coefficients w_{1-4} randomly sampled from $(0, 1)$, we incorporate four widely-used similarity metrics, i.e., Jaccard, Cosine, Normalized Overlap (denoted as NOverlap), and Dice, as the score functions to construct similar sets. Their formulations are reported in in Table 3. Given a set, e.g., s_i, its similarity value $\mathcal{M}$ to anther set, e.g., s_j, are then computed with the linear weighted sum:

$$\begin{aligned}\mathcal{M}(s_i, s_j) =& w_1 \cdot \mathcal{M}_{Jaccard}(s_i, s_j) + w_2 \cdot \mathcal{M}_{Cosine}(s_i, s_j) + \\ & w_3 \cdot \mathcal{M}_{Overlap}(s_i, s_j) + w_4 \cdot \mathcal{M}_{Dice}(s_i, s_j).\end{aligned} \tag{17}$$

This construction process ensures that the evaluation is free from selection bias and captures a wide spectrum of possible similar set configurations. With # S and # E denoting the numbers of sets and elements, the result data statistics are reported in Table 4.

[1] http://www.friendster.com/

[2] http://www.livejournal.com/

Table 3: Similarity metrics.

Jaccard	Cosine	NOverlap	Dice
$\frac{\lvert s_1 \cap s_2 \rvert}{\lvert s_1 \cup s_2 \rvert}$	$\frac{\lvert s_1 \cap s_2 \rvert}{\sqrt{\lvert s_1 \rvert \cdot \lvert s_2 \rvert}}$	$\frac{\lvert s_1 \cap s_2 \rvert}{\max(\lvert s_1 \rvert, \lvert s_2 \rvert)}$	$\frac{2 \cdot \lvert s_1 \cap s_2 \rvert}{\lvert s_1 \rvert + \lvert s_2 \rvert}$

Table 4: Dataset statistics of Task 1.

	# S	# E	# Avg. E/S
Friendster	889,839	5,501,401	11.29
LIVEJ	1,205,816	1,975,812	9.90

Table 5: Detailed performance comparison for Task 1.

Task 1: Set Similarity Learning								
Model	**Friendster**				**LIVEJ**			
	R@20	N@20	R@100	N@100	R@20	N@20	R@100	N@100
SAP	72.41 ± 0.18	68.13 ± 0.29	85.32 ± 0.15	72.15 ± 0.26	79.86 ± 0.11	77.75 ± 0.14	86.94 ± 0.09	85.34 ± 0.12
SMP	70.78 ± 0.28	68.99 ± 0.30	81.61 ± 0.11	72.22 ± 0.25	78.95 ± 0.12	76.70 ± 0.08	86.83 ± 0.10	84.65 ± 0.09
DeepSet	63.20 ± 0.67	60.75 ± 0.69	76.60 ± 1.64	69.89 ± 0.60	75.45 ± 0.09	74.55 ± 0.11	83.31 ± 0.05	79.76 ± 0.06
RepSet	80.63 ± 0.19	76.56 ± 0.21	86.49 ± 0.14	74.92 ± 0.23	82.15 ± 0.18	79.63 ± 0.22	88.41 ± 0.15	83.12 ± 0.19
SAtt	77.52 ± 0.45	71.92 ± 0.47	87.51 ± 0.32	75.21 ± 0.42	83.79 ± 0.06	81.73 ± 0.11	91.39 ± 0.02	85.07 ± 0.10
PoT	82.44 ± 0.12	81.85 ± 0.16	86.96 ± 0.15	81.47 ± 0.18	83.18 ± 0.14	84.25 ± 0.17	89.33 ± 0.11	86.45 ± 0.13
Set2Box	67.35 ± 0.22	69.73 ± 0.25	73.46 ± 0.18	70.33 ± 0.20	77.24 ± 0.20	75.89 ± 0.23	85.12 ± 0.16	82.34 ± 0.21
OTKE	79.53 ± 0.14	73.68 ± 0.21	86.64 ± 0.11	79.59 ± 0.17	81.45 ± 0.16	79.82 ± 0.19	87.95 ± 0.13	85.10 ± 0.15
DIEM	82.49 ± 0.16	81.40 ± 0.13	88.36 ± 0.09	81.56 ± 0.15	83.95 ± 0.12	84.92 ± 0.15	89.88 ± 0.08	87.15 ± 0.11
PSWE	83.05 ± 0.09	84.26 ± 0.15	88.59 ± 0.05	85.77 ± 0.14	83.52 ± 0.09	84.61 ± 0.13	89.48 ± 0.02	86.67 ± 0.09
FSPool	79.90 ± 0.13	81.96 ± 0.15	87.76 ± 0.11	84.41 ± 0.14	85.36 ± 0.03	87.17 ± 0.08	93.07 ± 0.06	90.29 ± 0.06
FSW	83.58 ± 0.13	84.39 ± 0.10	88.52 ± 0.07	85.81 ± 0.12	84.19 ± 0.05	85.04 ± 0.06	89.95 ± 0.09	87.23 ± 0.10
SRAL	**91.57** ± 0.22	**92.22** ± 0.22	**94.53** ± 0.11	**93.01** ± 0.19	**87.56 ± 0.31**	**89.31 ± 0.31**	**92.93 ± 0.02**	**91.25 ± 0.02**
Gain	9.56%*	9.28%*	6.71%*	8.39%*	2.58%*	2.46%*	-0.15%	1.06%*

E.3.2 Method Descriptions

The methods for set similarity learning are introduced as follows:

- *SAP* (Lin et al., 2013) denotes the classic implementation with global average pooling methodology for sets.
- *SMP* (Lin et al., 2013) is the implementation with set max pooling.
- *DeepSet* (Zaheer et al., 2017) is another classic set embedding method that learns permutation-invariant functions with deep neural networks. We implement global mean pooling as the permutation-invariant function of DeepSet.
- *RepSet* Skianis et al. (2020) extracts set representations by computing the bipartite matching costs between the input set and a collection of learnable reference sets.
- *SAtt* (Lee et al., 2019) is a state-of-the-art method to embed set structures with self-attention mechanism and Transformer architecture.
- *PoT* (Guo et al., 2021a) introduces a prototype-oriented optimal transport framework that learns set representations by minimizing the transport distance between the set's empirical distribution and learnable global prototypes.
- *Set2Box* (Lee et al., 2022) maps sets into hyper-rectangular box embeddings to effectively capture logical relationships and set boundaries.
- *OTKE* (Mialon et al., 2021) introduces a trainable embedding scheme based on kernelized optimal transport to aggregate set features.
- *DIEM* (Kim, 2022) proposes a differentiable framework to learn informative set interactions and enhance representation distinctiveness.
- *FSPool* (Zhang et al., 2020) is a representative set embedding framework with carefully-designed deep learning architecture.
- *PSWE* (Naderializadeh et al., 2021) is one of the state-of-the-art deep learning model for set representation learning.
- *FSW* (Amir & Dym, 2025) is another state-of-the-art model that utilizes the Fourier transform in the frequency domain for "multisets".

E.3.3 Detailed Experimental Results

For completeness, we present the detailed experimental results for Task 1 in Table 5. These results including mean scores and standard deviations over multiple runs, substantiate the findings discussed in the main text, where our model SRAL consistently outperforms all baselines.

E.4 Task 2: Bundle Representation Learning for Recommendation

E.4.1 Dataset Descriptions

Following recent works (Ma et al., 2022; Chang et al., 2020; Deng et al., 2020), we include two real-world datasets: Youshu[3] (Chen et al., 2019) for book list recommendation and NetEase (Cao et al., 2017) for music playlist recommendation. Dataset statistics are reported in Table 6.

Table 6: Data statistics of Task 2. S, U, and E denote sets (bundles), users, and elements.

Dataset	# S	# U	# E	# Avg. E/S
Youshu	4,771	8,039	32,770	37.03
NetEase	22,864	18,528	123,623	77.80

E.4.2 Method Descriptions

We include the following bundle recommender models:

- *MFBPR* (Rendle et al., 2012) utilizes the Bayesian Personalized Ranking (BPR) loss within a Matrix Factorization framework to model collaborative filtering between users and bundles.
- *DSBRec* is a specialized implementation with DeepSet (Zaheer et al., 2017) and optimize with BPR loss.
- *DAM* (Chen et al., 2019) attentively captures bundle representations from associated items and utilizes multi-task learning to optimize interactions between users and both items and bundles.
- *BundleNet* (Deng et al., 2020) is a traditional framework for bundle recommendation that constructs a tripartite graph of user-bundle-element relationships and utilizes Graph Convolution Network (GCN) for representation learning alongside multi-task learning.
- *BGCN* (Chang et al., 2020), a bundle recommendation method, dissects user-bundle-element relationships into two distinct perspectives: bundle-view and item-view graphs.
- *CrossCBR* (Ma et al., 2022) leverages contrastive learning to achieve cross-view alignment in the latent space.

E.4.3 Detailed Experimental Results

Table 7 catalogs the precise outcomes on both the Youshu and NetEase datasets, with standard deviations included to underscore the stability of our findings. The results show that SRAL$^+$ consistently outperforms other rival methods with low standard deviations across all runs.

E.5 Task 3: Point Cloud Processing

E.5.1 Dataset Descriptions

For this study, we utilize ModelNet40 dataset (Wu et al., 2015), comprising 3D point clouds extracted from triangular meshes of 12,311 computer-aided design models across 40 distinct object categories. Each object is represented by a set of 1024 points, following methodologies outlined in (Guo et al., 2021b; Qi et al., 2017).

E.5.2 Detailed Experimental Results

For Task 3, Table 8 summarizes the comparative test accuracies using both MLP and ISAB backbones. The inclusion of standard deviations further illustrates the stability of these outcomes.

E.6 Task 4: Topic Set Expansion

E.6.1 Dataset Descriptions

We leverage three datasets[4], i.e., LDA-1k, LDA-3k, and LDA-5k, from previous work (Zaheer et al., 2017) that are originally processed from latent Dirichlet allocation (Blei et al., 2003). They

[3] https://github.com/mysbupt/CrossCBR/blob/master/dataset.tgz

[4] https://github.com/manzilzaheer/DeepSets/tree/master/SetExpansion/data/lda

Table 7: Detailed performance comparison for Task 2.

Task 2: Bundle Recommendation								
Model	Youshu				NetEase			
	R@20	N@20	R@100	N@100	R@20	N@20	R@100	N@100
MFBPR	19.97 ± 0.45	11.67 ± 0.52	44.33 ± 0.33	17.95 ± 0.41	5.21 ± 0.15	2.98 ± 0.11	14.15 ± 0.18	4.92 ± 0.12
DSBRec	20.46 ± 0.41	12.03 ± 0.31	45.34 ± 0.32	18.12 ± 0.28	5.51 ± 0.10	3.04 ± 0.08	14.76 ± 0.13	5.14 ± 0.14
DAM	20.83 ± 0.36	11.99 ± 0.22	45.58 ± 0.26	18.38 ± 0.33	5.54 ± 0.13	3.11 ± 0.09	14.98 ± 0.16	5.12 ± 0.10
BundleNet	22.85 ± 0.35	11.90 ± 0.25	47.84 ± 0.22	19.19 ± 0.31	6.17 ± 0.12	3.44 ± 0.07	16.26 ± 0.11	5.83 ± 0.08
BGCN	25.22 ± 0.12	14.54 ± 0.10	49.38 ± 0.26	21.18 ± 0.29	7.04 ± 0.10	3.91 ± 0.08	17.25 ± 0.15	6.51 ± 0.09
CrossCBR	26.41 ± 0.42	16.55 ± 0.23	51.90 ± 0.44	23.30 ± 0.25	7.21 ± 0.11	4.08 ± 0.06	18.32 ± 0.13	6.77 ± 0.07
SRAL$^+$	**26.92 ± 0.09**	**16.95 ± 0.08**	**52.18 ± 0.19**	**23.64 ± 0.06**	**7.37 ± 0.11**	**4.21 ± 0.05**	**18.66 ± 0.01**	**7.01 ± 0.06**
Gain	1.93%*	2.42%*	0.54%*	1.46%*	2.22%*	3.19%*	1.86%*	3.54%*

Table 8: Test accuracy (%) of SRAL and competing methods for Task 3.

Task 3: Point Cloud Processing												
Backbone	SAP	SMP	RepSet	SAtt	PoT	Set2Box	OTKE	DIEM	PSWE	FSPool	FSW	SRAL	Gains
MLP	57.65 ± 0.52	86.35 ± 0.43	83.45 ± 0.55	85.89 ± 0.41	85.20 ± 0.48	82.15 ± 0.64	85.92 ± 0.38	85.58 ± 0.42	86.41 ± 0.39	85.76 ± 0.32	86.38 ± 0.35	**86.53** ± 0.36	+0.14%
ISAB	85.45 ± 0.16	86.82 ± 0.49	86.05 ± 0.31	86.78 ± 0.28	86.55 ± 0.25	85.88 ± 0.34	86.70 ± 0.29	86.72 ± 0.26	86.85 ± 0.30	86.88 ± 0.53	86.93 ± 0.21	**87.31** ± 0.23	+0.44%*

respectively contain 2,000, 6,000, and 10,000 sets and 17,016, 37,718 and 61,127 vocabulary elements. Their average elements per set are around 25.

E.6.2 Detailed Experimental Results

The complete results with standard deviations of Task 4 evaluation are reported in Table 9.

E.7 Implementation of Self-supervised Learning Loss

We implement the following self-supervised learning losses for comparison:

- **Set Triplet Loss**:

$$\mathcal{L} = \sum_{S_i, S_j \in \mathcal{S}} \max\left(d\left(\boldsymbol{v}_i, \boldsymbol{v}_i'\right) - d\left(\boldsymbol{v}_i, \boldsymbol{v}_j'\right) + \alpha, 0\right). \tag{18}$$

  where α is the hyperparameter.

- **Soft-Nearest Neighbors Loss**:

$$\mathcal{L} = -\log \frac{\exp(\text{sim}(\boldsymbol{v}_i', \boldsymbol{v}_i'')/\tau)}{\sum_{j \neq i} \exp(\text{sim}(\boldsymbol{v}_i', \boldsymbol{v}_j')/\tau)} - \log \frac{\exp(\text{sim}(\boldsymbol{v}_i'', \boldsymbol{v}_i')/\tau)}{\sum_{j \neq i} \exp(\text{sim}(\boldsymbol{v}_i'', \boldsymbol{v}_j'')/\tau)}, \tag{19}$$

  where we use inverse Euclidean distance to implement the sim function. For this experiment, it is important to note that these metrics cannot be directly integrated into our SFE architecture. Therefore, we made adaptations for them as: for sets of varying sizes, we first aggregate their element embeddings using mean pooling to obtain a single vector representation for each set. Subsequently, we employ each respective metric to calculate the distributional distance. The training objective is to align these distances with the ground-truth similarity rankings from Task 1.

- **Barlow Twins Loss**:

$$\mathcal{L} = \sum_{k=1}^{D} (1 - C_{kk})^2 + \lambda \sum_{k=1}^{D} \sum_{l \neq k} (C_{kl})^2, \tag{20}$$

  where $C = \frac{1}{BatchSize} norm(V')^\mathsf{T} \cdot norm(V'')$. Here V' and V'' denote the perturbed batch set embeddings.

E.8 Scalability Study of SRAL

To assess the scalability of SRAL, we derived nine proportionally-sized sub-datasets from Friendster. Figure 1 presents a scalability analysis of SRAL on subsets of the Friendster dataset. We have two main observations. First, the training time per epoch (blue bars, left axis) exhibits a gradually growing trend as the data volume increases. This prevents prohibitive computational costs on larger datasets and confirms the model's acceptable efficiency. Second, the model's performance, measured by Recall@20 (orange line, right axis), remains generally consistent with slight fluctuation. These results demonstrate the effectiveness of our SRAL model, making it a practical solution for large-scale settings.

Table 9: AUC results (%) of SRAL and competing methods for Task 4.

Task 4: Topic Set Expansion													
Data	SAP	SMP	DeepSet	RepSet	SAtt	PoT	Set2Box	OTKE	DIEM	PSWE	FSPool	FSW	SRAL	Gains
LDA-1k	54.34 ± 3.91	67.21 ± 6.69	54.98 ± 5.11	57.32 ± 2.34	58.55 ± 2.70	58.94 ± 3.12	50.59 ± 4.05	62.95 ± 3.88	63.58 ± 3.56	58.36 ± 5.49	75.67 ± 4.01	64.56 ± 3.41	**80.94** ± 1.38	+6.96%*
LDA-3k	51.95 ± 1.87	74.40 ± 2.12	51.96 ± 3.91	58.33 ± 1.20	77.48 ± 3.84	73.40 ± 2.45	64.98 ± 3.22	77.59 ± 2.91	75.67 ± 2.15	78.44 ± 2.04	70.57 ± 1.38	79.67 ± 2.35	**87.93** ± 1.92	+10.37%*
LDA-5k	51.34 ± 1.34	80.65 ± 1.18	52.05 ± 1.28	61.39 ± 2.31	74.59 ± 3.37	75.11 ± 2.08	65.67 ± 2.54	72.57 ± 2.76	76.96 ± 2.33	78.81 ± 2.92	71.16 ± 1.78	80.94 ± 2.34	**86.20** ± 0.67	+6.50%*

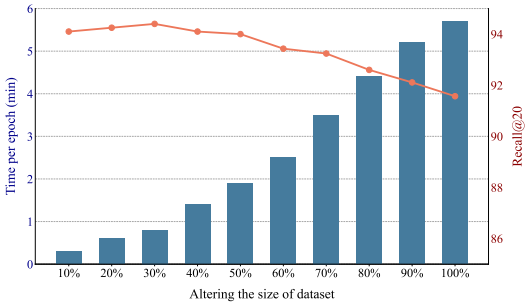


Figure 1: Results with varying data sizes.

Table 10: Runtime comparison for Task 1 and Task 2.

Task 1: Set Similarity Learning Runtime				
Model	Friendster		LIVEJ	
	In Total	Per-epoch	In Total	Per-epoch
PSWE	3.1h	1.6min	5.1h	2.7min
FSPool	2.9h	1.5min	6.5h	2.1min
SRAL	3.9h	5.6min	7.7h	7.9min

Task 2: Bundle Recommendation Runtime				
Model	Youshu		NetEase	
	In Total	Per-epoch	In Total	Per-epoch
BGCN	12.6min	6.3s	23.4min	9.7s
CrossCBR	10.4min	4.5s	19.5min	8.4s
SRAL$^+$	24.5min	42.2s	44.8min	57.4s

Table 11: Runtime comparison for Task 3 and Task 4.

Task 3: Point Cloud Processing Runtime				
Model	MLP		ISAB	
	In Total	Per-epoch	In Total	Per-epoch
PSWE	1.6h	27.4s	1.8h	29.5s
FSPool	1.2h	23.9s	1.6h	26.3s
SRAL	2.3h	1.6min	2.7h	1.7min

Task 4: Topic Set Expansion Runtime						
Model	LDA-1k		LDA-3k		LDA-5k	
	In Total	Per-epoch	In Total	Per-epoch	In Total	Per-epoch
PSWE	10.5s	0.3s	15.6s	0.3s	18.5s	0.3s
FSPool	9.8s	0.2s	14.4s	0.2s	17.4s	0.3s
SRAL	14.3s	0.5s	23.5s	0.5s	31.6s	0.6s

E.9 Computation Efficiency Analysis

E.9.1 Computation Cost Comparison

We report the training time cost of SRAL and compare it with two most competitively performing models across all tasks. As shown in Tables 10 and 11, although our SRAL model incurs a higher "Per-epoch" computational cost compared to the baselines, its "In Total" training time remains comparable. This is because our proposed Adversarial Encoding Perturbation and Optimization mechanism well promotes the model convergence, as we analyzed earlier in § 4.3.2.

E.9.2 Module Computation Cost

To offer deeper insights into the computational overhead, we break down the per-epoch training time for individual modules using the Friendster dataset from Task 1 as follows:

As shown in Table 12, the Adversarial Encoding Perturbation and Optimization (AEPO) module incurs the highest computational cost among all components. This overhead primarily arises from the adversarial perturbation generation and the iterative min-max optimization process. Nevertheless, considering its substantial contributions to accelerating convergence and significantly improving training stability, we believe that this computational expenditure constitutes a worthwhile trade-off.

E.10 Comparison with Vanilla OT-based Solutions

While Optimal Transport (OT) has been explored in set representation learning, e.g., *RepSet* Skianis et al. (2020), our approach distinguishes itself through the utilization of the Sliced-Wasserstein (SW) distance and a tailored adversarial enhancement mechanism. *RepSet* formulates the set distance as a bipartite matching problem, which is equivalent to exact OT, and offers an approximation variant (*ApproxRepSet*) by relaxing constraints to a semi-relaxed OT problem. To provide the

Table 12: Training time cost of each major module.

	SRAL (complete)	SRAL (SFE Module)	SRAL (AEPO)	SRAL (Main Loss)
Youshu	5.6min	1.8min	3.3min	0.1min

comparison, we conducted experiments on the Friendster dataset (Task 1) covering three aspects: (1) direct performance comparison against *RepSet* and *ApproxRepSet*; (2) evaluating *RepSet* variants as encoders within our SRAL framework. The results are summarized in Table 13.

Performance Analysis. We observe that our SRAL outperforms the vanilla *RepSet* baseline; and the approximation variant *ApproxRepSet* exhibits inferior performance. We attribute this gap to the nature of the "semi-relaxed" approximation employed in *ApproxRepSet*. As noted in Skianis et al. (2020), dropping constraints to achieve computational efficiency may sacrifice rigorous metric properties, such as the triangle inequality, and makes the optimization prone to local optima. In contrast, our SW-based approach preserves key geometric properties while remaining computationally efficient, resulting in superior representation stability.

Encoder Compatibility. When integrating *RepSet* as the encoder within our framework, we observe a performance gain but still falls short of our native SRAL model. This empirical finding validates our theoretical analysis in Remark 1: our AEPO mechanism is specifically tailored for the Set Feature Encoder (SFE) based on Sliced-Wasserstein metric. The adversarial perturbations generated by AEPO aim to maximize discrepancies in the embedding space, which are positively correlated in expectation with the Sliced-Wasserstein distance. In contrast, such a direct perturbation may not hold for the bipartite matching objective in *RepSet*, rendering the adversarial optimization less effective in capturing distributional semantics.

Table 13: Performance comparison with RepSet and its variants.

	RepSet	SRAL (RepSet)	ApproxRepSet	SRAL (ApproxRepSet)	**SRAL**
Recall@20	80.63	82.25	77.81	78.93	**91.57**