On permutation-invariant neural networks

Anonymous authors
Paper under double-blind review

Abstract

Conventional machine learning algorithms have traditionally been designed under the assumption that input data follows a vector-based format, with an emphasis on vector-centric paradigms. However, as the demand for tasks involving set-based inputs has grown, there has been a paradigm shift in the research community towards addressing these challenges. In recent years, the emergence of neural network architectures such as Deep Sets and Transformers has presented a significant advancement in the treatment of set-based data. These architectures are specifically engineered to naturally accommodate sets as input, enabling more effective representation and processing of set structures. Consequently, there has been a surge of research endeavors dedicated to exploring and harnessing the capabilities of these architectures for various tasks involving the approximation of set functions. This comprehensive survey aims to provide an overview of the diverse problem settings and ongoing research efforts pertaining to neural networks that approximate set functions. By delving into the intricacies of these approaches and elucidating the associated challenges, the survey aims to equip readers with a comprehensive understanding of the field. Through this comprehensive perspective, we hope that researchers can gain valuable insights into the potential applications, inherent limitations, and future directions of set-based neural networks. Indeed, from this survey we gain two insights: i) Deep Sets and its variants can be generalized by differences in the aggregation function, and ii) the behavior of Deep Sets is sensitive to the choice of the aggregation function. From these observations, we show that Deep Sets, one of the well-known permutation-invariant neural networks, can be generalized in the sense of a quasi-arithmetic mean.

1 Introduction

In recent years, machine learning has achieved significant success in many fields, and many typical machine learning algorithms handle vectors as their input and output (Zhou, 2021; Islam, 2022; Erickson et al., 2017; Jordan & Mitchell, 2015; Mitchell, 1997). For example, some of these applications include image recognition (Sonka et al., 2014; Minaee et al., 2021; Guo et al., 2016), natural language processing (Strubell et al., 2019; Young et al., 2018; Otter et al., 2020; Deng & Liu, 2018), and recommendation systems (Portugal et al., 2018; Melville & Sindhwani, 2010; Zhang et al., 2019b).

However, with the advancement of the field of machine learning, there has been a growing emphasis on the research of algorithms that handle more complex data structures in recent years. In this paper, we consider machine learning algorithms that deal with sets (Hausdorff, 2021; Levy, 2012; Enderton, 1977) as one such data structure. Here is a couple of examples of set data structures:

- **Set of vector data.** For example a set of image vectors.
- **Point cloud data.** Point cloud data consists of a set of data points, each represented by its spatial coordinates in a multi-dimensional space.

Considering machine learning algorithms that handle sets allows us to leverage the representations of these diverse types of data. Certainly, the goal here is to utilize machine learning models to approximate set functions. Set functions, in this context, are mathematical functions that operate on sets of elements or data points. These functions capture various properties, relationships, or characteristics within a given set. However, when dealing with complex data and large sets, it can be challenging to directly model or compute these set functions using traditional model architectures. Therefore, we need to consider a specialized model architecture specifically designed for the approximation of set functions.

In particular, a notable characteristic of set functions, when compared to vector functions, is their permutation invariance. Permutation invariance, in the context of set functions or set-based data, means that the output of the function remains the same regardless of the order in which elements of the set are arranged. In other words, if you have a set of data points and apply a permutation (rearrangement) to the elements within the set, a permutation-invariant function will produce the same result. This property is crucial when dealing with sets of data where the order of elements should not affect the function's evaluation. In Section 2 we introduce a more formal definition. Popular conventional neural network architectures like VGG (Simonyan & Zisserman, 2014) and ResNet (He et al., 2016), or more recently proposed ResNeXt (Xie et al., 2017), EfficientNet (Tan & Le, 2019) and ResNest (Zhang et al., 2022b) do not inherently possess the permutation-invariant property. Hence, the primary research interest lies in determining what neural network architectures can be adopted to achieve the permutation-invariant property while maintaining performance and expressive capabilities similar to those achieved by conventional models. In Section 3, we introduce such model architectures and provide the overview of the objective tasks in Section 4. Furthermore, there have been several theoretical analyses of neural network approximations for such permutation-invariant functions, and Section 5 introduces them. To evaluate such research, datasets for performance assessment of approximate set functions are essential. In Section 6, we list some of the well-known datasets for this purpose. Finally, we propose a novel generalization of Deep Sets in Section 7.

Our contributions are summarized as follows.

- We provide a comprehensive survey of permutation-invariant neural network architectures and the tasks/datasets they handle (Section 3, 4 and 6).
- We also highlight existing results of theoretical analysis related to the generalization of Deep Sets in the sense of Janossy pooling (Section 5).
- Finally, based on the above survey results we remark that Deep Sets and its variants can be generalized in the sense of quasi-arithmetic mean. We use this concept to propose a special class of Deep Sets, namely Hölder's Power Deep Sets (Section 7).

2 Preliminary and related concepts

First, we introduce the necessary definitions and notation. Let $\mathcal{V} \coloneqq \{1, \dots, |\mathcal{V}|\}$ be the ground set.

Definition 2.1. Let $\varphi \colon \mathcal{V} \to \mathbb{R}^d$ be the mapping from each element of the ground set $\mathcal{V}$ to the corresponding d-dimensional vector with respect to the indices as $\varphi(s_i) = \boldsymbol{s}_i \in \mathbb{R}^d$ for all $s_i \in \mathcal{S} \subseteq \mathcal{V}$. Furthermore, when it is clear from the context, we identify $\mathcal{S} \subseteq \mathcal{V}$ with the set of vectors obtained by this mapping as $\mathcal{S} = \{s_1, \dots, s_{|\mathcal{S}|}\} = \{\varphi(s_1), \dots, \varphi(s_{|\mathcal{S}|})\} = \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_{|\mathcal{S}|}\}$.

For the case of dealing with subsets, we also provide the following definitions.

Definition 2.2. Denote the set of all subsets of a set $\mathcal{V}$, known as the power set of $\mathcal{V}$, by $2^{\mathcal{V}}$.

2.1 Permutation invariance

There are several key differences between functions that take general vectors as inputs and functions that take sets as inputs.

Definition 2.3 (Tuple)**.** A tuple, or an ordered n-tuple is a set of n elements with an order associated with them. If n elements are represented by x_1, $x_2, \dots, x_n$, then we write the ordered n-tuple as $(x_1, x_2, \to, x_n)$.

The concept of set permutations is important when dealing with set functions. We therefore introduce the following definition of the set of all permutations of any subset.

Definition 2.4. Let $\Pi_{\mathcal{S}}$ be the set of all permutations of a tuple $\mathcal{S}$.

Let $\Phi\colon \cup_{k=1}^{|\mathcal{V}|} \mathcal{V}^k \to \mathbb{R}$ be a function defined on tuples.

Definition 2.5 (Permutation invariant)**.** A function $\Phi\colon \cup_{k=1}^{|\mathcal{V}|} \mathcal{V}^k \to \mathbb{R}$ is said to be permutation invariant if $\Phi(\mathcal{S}) = \Phi(\pi_{\mathcal{S}}\mathcal{S})$ for any tuple $\mathcal{S}$ and its arbitrary permutation $\pi_{\mathcal{S}} \in \Pi_{\mathcal{S}}$.

Here, Φ can be perceived as a set function if and only if Φ is permutation invariant. Conversely, a set function $f\colon 2^{\mathcal{V}} \to \mathbb{R}$ induces a pertmutation invariant function. on tuples. On the other hand, function that changes its value depending on the permutation of the input set is called a permutation-sensitive function. Neural networks tasked with approximating set functions face unique challenges and requirements compared to conventional vector-input functions. In order to accurately model and capture the characteristics of sets, these networks need to fulfill the aforementioned properties. First, permutation invariance ensures that the output of the network remains consistent regardless of the order in which elements appear in the set. This is crucial for capturing the inherent structure and compositionality of sets, where the arrangement of elements does not affect the overall meaning or outcome. Second, equivariance to set transformations guarantees that the network's behavior remains consistent under operations such as adding or removing elements from the set. This property ensures that the network can adapt to changes in set size without distorting its output. By satisfying these properties, neural networks can effectively model and approximate set functions, enabling them to tackle a wide range of set-based tasks in various domains.

2.2 Permutation equivariance

Along with permutation invariance, another important concept is permutation equivariance. Permutation equivariant is defined as follows.

Definition 2.6 (Permutation equivariant)**.** Let $\mathcal{U} = \cup_{k=1}^{|\mathcal{V}|}\mathcal{V}^k$. A function $\Phi\colon \mathcal{U} \times \mathcal{U} \to \mathcal{U}$ is said to be permutation equivariant if permutation of the input instances permutes the output labels as

$$f(\{s_{\pi_{\mathcal{S}}(1)}, \dots, s_{\pi_{\mathcal{S}}(|\mathcal{S}|)}) = \{f_{\pi_{\mathcal{S}}(1)}(\mathcal{S}), \dots, f_{\pi_{\mathcal{S}}(|\mathcal{S}|)}(\mathcal{S})\}. \tag{1}$$

This equivariance property is important in the supervised setting (Zaheer et al., 2017). In recent studies, it has been theoretically shown that permutation invariant transformations can be constructed by combining multiple permutation equivariance transformations (Fei et al., 2022).

3 Model architectures for approximating set functions

In this section, we give an overview of neural network architectures approximating set functions. Table 1 summarizes architectures as well as the corresponding applications.

In particular, we focus on architectures following the idea of Deep Sets, which demonstrated universality results for permutation-invariant inputs. However, prior to that, several similar studies on related architectures also exist (Gens & Domingos, 2014; Cohen & Welling, 2016). For example, invariance can be achieved by pose normalization using an equivariant detector (Lowe, 2004; Jaderberg et al., 2015), or by averaging a possibly nonlinear function over a group (Reisert, 2008; Manay et al., 2006; Kondor, 2007).

3.1 Deep Sets

One seminal work for approximating set functions by neural networks is Deep Sets (Zaheer et al., 2017). The framework of Deep Sets is written as

$$f(\mathcal{S}) \coloneqq \rho\left(\sum_{s\in\mathcal{S}} \phi(\boldsymbol{s})\right), \tag{2}$$

Architecture	Novelties and contributions	Applied tasks
Deep Sets (Zaheer et al., 2017)	The universality result for permutation-invariance and sum-decomposability.	General set function approximation Point cloud classification Set expansion Set retrieval Image tagging Set anomaly detection
PointNet (Qi et al., 2017a)	The max-decomposition architecture.	Point cloud classification Point cloud segmentation
PointNet++ (Qi et al., 2017b)		
SetNet (Zhong et al., 2018)	Utilizing NetVLAD layer for the set retrieval task.	Set retrieval
Set Transformer (Lee et al., 2019)	Utilizing Transformer architecture for permutation-invariant inputs.	General set function approximation Point cloud classification Set anomaly detection
DSPN (Zhang et al., 2019b)	A model that predicts a set of vectors from another vector.	Set reconstruction Bounding box prediction
iDSPN (Zhang et al., 2021)	The novel concept of exclusive multiset-equivariance.	Class specific numbering Random multisets reconstruction Object property prediction
SetVAE (Kim et al., 2021b)	The VAE-based set generation model.	Set generation
Slot Attention (Locatello et al., 2020)	A new variant of permutation-invariant attention mechanism.	Object discovery Set prediction
Deep Sets++ & Set Transformer++ (Zhang et al., 2022c)	The Set Normalization as the alternative normalization layer.	General set function approximation Point cloud classification Set anomaly detection
PointCLIP (Zhang et al., 2022d)	CLIP for permutation-invariant inputs.	Point cloud classification
Perceiver (Jaegle et al., 2021b)	Introduction of the computationally efficient learnable query in the framework of Set Transformer.	General set function approximation Point cloud classification
Perceiver IO (Jaegle et al., 2021a)	Perceiver for multimodal outputs.	General set function approximation Multimodal set embedding
Perceiver VL (Tang et al., 2023)	Perceiver-based architecture for vision and language tasks.	Set retrieval

Table 1: Neural network architectures for approximating set functions.

for set $\mathcal{S}$, and two functions ϕ, ρ. We can see that Deep Sets architecture satisfies the permutation-invariant 2.5. Moreover it can represent the set function by using arbitrary neural networks ϕ and ρ. It is also known that Deep Sets have the universality for permutation-invariant and sum-decomposability (see Section 5.1 for more details).

One fundamental property for approximating set functions is the sum-decomposability.

Definition 3.1. The function f is said to be sum-decomposable if there are functions ρ and ϕ such that

$$f(\mathcal{S}) = \rho\left(\sum_{\boldsymbol{s}\in\mathcal{S}} \phi(\boldsymbol{s})\right) \tag{3}$$

for any $\mathcal{S} \subseteq \mathcal{V}$. In this case, we say that (ρ, ϕ) is a sum-decomposition of f. Given a sum-decomposition (ρ, ϕ), we write $\Phi(\mathcal{S}) \coloneqq \sum_{\boldsymbol{s}\in\mathcal{S}} \phi(\boldsymbol{s})$. With this notation, we can write Eq. 3 as $f(\mathcal{S}) = \rho(\Phi(\mathcal{S}))$. We may also refer to the function $\rho \circ \Phi$ as a sum-decomposition.

Then, we can see that the architecture of Deep Sets satisfies the sum-decomposability.

3.2 PointNet and PointNet++

The most well-known architecture developed for processing point cloud data is PointNet (Qi et al., 2017a). One of the important differences between PointNet and Deep Sets is their pooling operation. For instance, PointNet employs global max pooling, while global sum pooling is adopted in Deep Sets. This implies that we can write PointNet architecture as follows:

$$f(\mathcal{S}) \coloneqq \rho\left(\max_{\boldsymbol{s}\in\mathcal{S}} \phi(\boldsymbol{s})\right). \tag{4}$$

Therefore, we can express the architectures of Deep Sets and PointNet in a unified notation.

A function that can be written in the form of Eq. 2 is referred to as sum-decomposable, and the function that can be written in the form of Eq. 4 is called max-decomposable.

The universality result of sum-decomposability in Deep Sets suggests that a similar result holds for max-decomposable functions, as indicated in subsequent studies (Wagstaff et al., 2022).

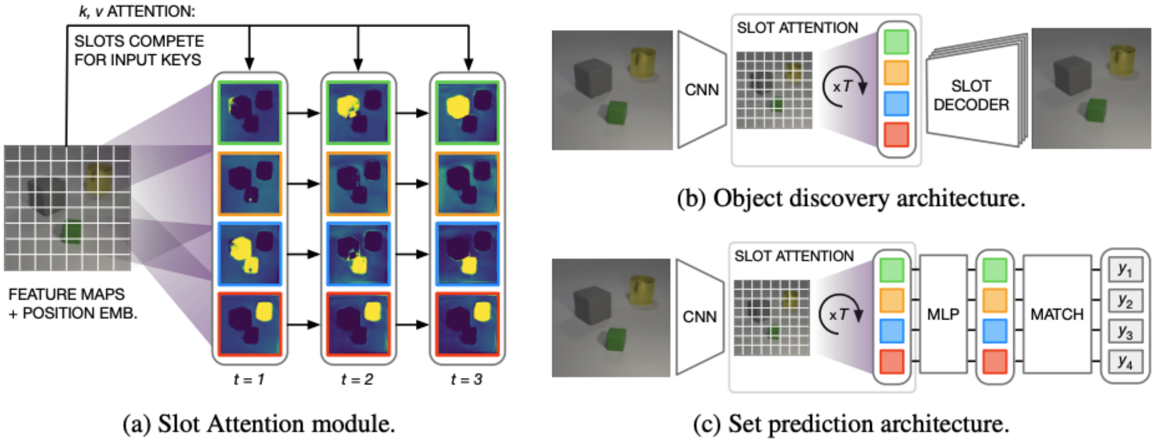
3.3 Set Transformer

In recent years, the effectiveness of Transformer architectures (Vaswani et al., 2017; Kimura & Tanaka, 2019; Lin et al., 2022) has been reported in various tasks that neural networks tackle, such as natural language processing (Kalyan et al., 2021; Wolf et al., 2019; Kitaev et al., 2020; Beltagy et al., 2020), computer vision (Han et al., 2022b; Khan et al., 2022; Dosovitskiy et al., 2020; Arnab et al., 2021; Zhou et al., 2021a), and time series analysis (Wen et al., 2022; Zhou et al., 2021b; Zerveas et al., 2021). Set Transformer (Lee et al., 2019) utilize the Transformer architecture to handle permutation-invariant inputs. Similar architectures have also been proposed for point cloud data (Guo et al., 2021; Park et al., 2022; Zhang et al., 2022a; Liu et al., 2023).

The architectures of Deep Sets and PointNet, as evident from Eq. 2 and 4, operate by independently transforming each element of the input set and then aggregating them. However, this approach neglects the relationships between elements, leading to a limitation. On the other hand, the Set Transformer addresses this limitation by introducing an attention mechanism, which takes into account the relationships between two elements in the input set. We can write this operation as follows.

$$f(\mathcal{S}) = \rho\left(\frac{1}{\tau(|\mathcal{S}|, 2)} \sum_{\mathcal{T}\in\mathcal{S}_{(2)}} \phi(\boldsymbol{t}_1, \boldsymbol{t}_2)\right), \quad (\boldsymbol{t}_1, \boldsymbol{t}_2 \in \mathcal{T}) \tag{5}$$

where $\tau(|\mathcal{S}|, 2) = \frac{|\mathcal{S}|}{(|\mathcal{S}|-2)!}$. Eq. 5 can be viewed as performing permutation-invariant operations on 2-subsets of permutations of the input set. Indeed, $\phi(\boldsymbol{t}_1, \boldsymbol{t}_2)$ is usually split up in attention weights $w(\boldsymbol{t}_1, \boldsymbol{t}_2)$ and values $v(\boldsymbol{t}_2)$ and a softmax acts on the weights.



(a) Slot Attention module.

(b) Object discovery architecture.

(c) Set prediction architecture.

Figure 1: Slot Attention module and example applications to unsupervised object discovery and supervised set prediction with labeled targets, from Figure 1 of Locatello et al. (2020). This figure is cited to illustrate the behavior of the Slot Attention module.

As introduced in Section 5.1, recent research has revealed that Deep Sets, PointNet, Set Transformer, and their variants can be regarded as special cases of a function class called Janossy Pooling (Murphy et al., 2018). Moreover, there have been several discussions regarding the generalization of Transformer and Deep Sets (Kim et al., 2021a; Maron et al., 2018).

3.4 Deep Sets++ and Set Transformer++

Many neural network architectures include normalization layers such as Layer Norm (Ba et al., 2016), BatchNorm (Ioffe & Szegedy, 2015; Bjorck et al., 2018) or others (Wu & He, 2018; Salimans & Kingma, 2016; Huang & Belongie, 2017). SetNorm (Zhang et al., 2022c) is used for neural networks that take sets as input, based on the result that the normalization layer is permutation-invariant only when the transformation part of the normalization layer deforms all the features with different scales and biases for each feature. Specifically, applying SetNorm to Deep Sets and Set Transformer, referred to as Deep Sets++ and Set Transformer++ respectively, has been shown experimentally to achieve improvements of original Deep Sets and Set Transformer. Furthermore, Zhang et al. (2022c) also releases a dataset called Flow-RBC, which comprises sets of measurement results of red blood cells of patients, aimed at predicting anemia.

3.5 DSPN and iDSPN

Deep Set Prediction Networks (DSPN) (Zhang et al., 2019b) propose a model for predicting a set of vectors from another set of vectors. The proposed decoder architecture leverages the fact that the gradients of the set functions with respect to the set are permutation-invariant, and it is effective for tasks such as predicting a set of bounding boxes for a single input image. Also, iDSPN (Zhang et al., 2021) introduced the concept of exclusive multiset-equivariance to allow for the arbitrary ordering of output elements with respect to duplicate elements in the input set. They also demonstrated that by constructing the encoder of DSPN using Fspool (Zhang et al., 2019c), the final output for the input set satisfies exclusive multiset-equivariance.

3.6 SetVAE

SetVAE (Kim et al., 2021b) is the set generation model based on Variational Auto-Encoder (VAE) (Kingma & Welling, 2013) that takes into account exchangeability, variable-size sets, interactions between elements, and hierarchy. Here, hierarchy refers to the relationships between subsets within a set, such as the hierarchical structure of elements in the set. The concept of the Hierarchical VAE was introduced in the context of high-resolution image generation (Sønderby et al., 2016; Vahdat & Kautz, 2020), and this study utilizes it for set generation. As an application of the idea of SetVAE, SCHA-VAE (Giannone & Winther, 2022) for generating a few shot image is also proposed.

3.7 PointCLIP

One of the learning strategies that has garnered significant attention in the field of machine learning in recent years is Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021; Shen et al., 2021; Luo et al., 2022). PointCLIP (Zhang et al., 2022d) adopts CLIP for permutation-invariant neural networks. PointCLIP encodes point cloud data using CLIP and achieves category classification for 3D data by examining their positional relationships with category texts. Furthermore, PointCLIPv2 (Zhu et al., 2023) is an improvement of PointCLIP, achieved through a dialogue system.

3.8 Slot Attention

The Slot Attention (Locatello et al., 2020) mechanism was proposed to obtain representations of arbitrary objects in images or videos in an unsupervised manner. Slot Attention employs an iterative attention mechanism to establish a mapping from its inputs to the slots (see Fig. 1). The slots are initially set at random and then refined at each iteration to associate with specific parts or groups of the input features. The process involves randomly sampling initial slot representations from a common probability distribution. It is proven that Slot Attention is

i) permutation invariance with respect to the input;

ii) permutation equivariance with respect to the order of the slots.

Zhang et al. (2022e) pointed out two issues with slot attention: the problem of single objects being bound to multiple slots (soft assignments) and the problem of multiple slots processing similar inputs, resulting in multiple slots having averaged information about the properties of a single object (Lack of tiebreaking). To address these issues, they leverage the observation that part of the slot attention processing can be seen as one step of the Sinkhorn algorithm (Sinkhorn, 1964), and they propose a method to construct slot attention to be exclusive multiset-equivariant without sacrificing computational efficiency.

Chang et al. (2022) argue that slot attention suffers from the issue of unstable backward gradient computation because it performs sequential slot updates during the forward pass. Specifically, as training progresses, the spectral norm of the model increases. Therefore, they experimentally demonstrated that by replacing the iterative slot updates with implicit function differentiation at the fixed points, they can achieve stable backward computation without the need for ad hoc learning stabilization techniques, including gradient clipping (Pascanu et al., 2013; Zhang et al., 2019a), learning rate warmup (Goyal et al., 2017; Liu et al., 2019) or adjustment of the number of iterative slot updates. Kipf et al. (2021) point out that initializing slots through random sampling from learnable Gaussian distributions during the sequential updating process may lead to instability in behavior. Based on this, Jia et al. (2022) propose stabilizing the behavior by initializing slots with fixed learnable queries, namely BO-QSA. Vikström & Ilin (2022) propose the ViT architecture and a corresponding loss function within the framework of Masked Auto Encoder to acquire object-centric representations. Furthermore, DINOSAUR (Seitzer et al., 2022) learns object-centric representations based on higher-level semantics by optimizing the reconstruction loss with ViT features. Slotformer (Wu et al., 2022) is proposed as a transformer architecture that predicts slots autoregressively, and it has been reported to achieve high performance. There are many other variants of slot attention along with their respective applications, such as SIMONe (Kabra et al., 2021), EfficientMORL (Emami et al., 2021), OSRT (Sajjadi et al., 2022), OCLOC (Yuan et al., 2022), SAVi (Kipf et al., 2021) or SAVi++ (Elsayed et al., 2022).

3.9 Perceiver

In the attention mechanism, increasing the length of the sequence is an important issue to be addressed. Perceiver (Jaegle et al., 2021b) tackles this problem by introducing computationally efficient learnable queries.

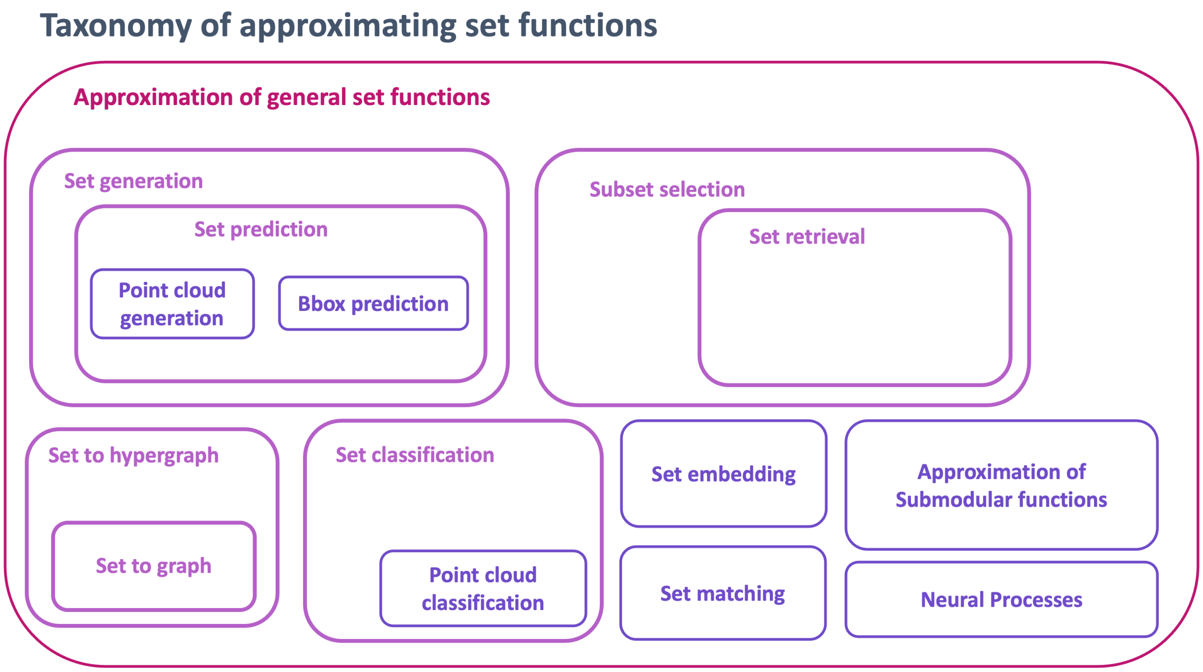


Figure 2: Taxonomy of approximating set functions. Several tasks can be considered as special cases of other tasks. For example, set retrieval, which is a set version of image retrieval, can be regarded as a kind of subset selection that extracts a subset from a set.

Although this framework can guarantee computational efficiency with the expressive power like Set Transformer, a limitation is that it is only applicable to unimodal classification tasks. Perceiver IO (Jaegle et al., 2021a) handles multimodal output by adding the decoder to Perceiver. Perceiver VL (Tang et al., 2023) also allows Perceiver-based architectures to be applied to vision and language tasks. In addition, Framingo (Alayrac et al., 2022), one of the foundation models for various vision and language downstream tasks, employs the iterative latent cross-attention proposed by Perceiver in its architecture.

4 Tasks of approximating set functions

In this section, we organize the tasks addressed by neural networks that approximate set functions. Figure 2 shows the taxonomy of approximating set functions.

4.1 Point cloud processing

Deep Sets, PointNet, and Set Transformer can be generalized in terms of the differences in the aggregation operations of elements within a set. However, specific aggregation operations are also proposed when the input set consists of point clouds. CurveNet (Xiang et al., 2021) proposes to treat point clouds as undirected graphs and represent curves as walks within the graph, thereby aggregating the points.

4.2 Set retrieval and subset selection

There exists a set retrieval task that generalizes the image retrieval task (Datta et al., 2008; Smeulders et al., 2000; Rui et al., 1999) to sets. The goal of the set retrieval system is to search and retrieve sets from the large pool of sets (Zhong et al., 2018).

Subset selection Subset selection is the task of selecting a subset of elements from a given set in a way that retains some meaningful criteria or properties. Ou et al. (2022) introduced the low-cost annotation method for subset selection and demonstrated its effectiveness.

SetNet (Zhong et al., 2018) is an architecture designed for set retrieval, which uses NetVLAD layer (Jin et al., 2021) instead of conventional pooling layers. In this paper, the Celebrity Together dataset is proposed specifically for set retrieval.

4.3 Set generation and prediction

Methods for set prediction can be broadly categorized into the following two approaches:

- distribution matching: approximates $P(\mathcal{Y}|\boldsymbol{x})$ for a set $\mathcal{Y}$ and an input vector $\boldsymbol{x}$;
- minimum assignment: calculates loss function between the assigned pairs.

Distribution matching Deep Set Prediction Networks (DSPN) (Zhang et al., 2019b) propose a model for predicting a set of vectors from another set of vectors. The proposed decoder architecture leverages the fact that the gradients of the set functions with respect to the set are permutation-invariant, and it is effective for tasks such as predicting a set of bounding boxes for a single input image. Also, iDSPN (Zhang et al., 2021) introduced the concept of exclusive multiset-equivariance to allow for the arbitrary ordering of output elements with respect to duplicate elements in the input set. PointGlow (Sun et al., 2020) applies the flow-based generative model for point cloud generation. With similar ideas, Biloš & Günnemann (2021) propose a continuous normalizing flow for sequential invariant vector data, and Zwartsenberg et al. (2023) propose Conditional Permutation Invariant Flows, which extend it to allow conditional generation.

Minimum assignment In the minimum assignment approach, there is freedom in choosing the distance function, but the latent set prediction (LSP) framework (Preechakul et al., 2021) relaxes this and provides convergence analysis. SetVAE (Kim et al., 2021b) is the set generation model based on VAE (Kingma & Welling, 2013) that takes into account exchangeability, variable-size sets, interactions between elements, and hierarchy.

Carion et al. (2020) consider object detection as a bounding box set prediction problem and propose an assignment-based method using a transformer, called Detection Transformer (DETR). Inspired by this identification of object detection and set prediction, many studies have been conducted using similar strategies (Hess et al., 2022; Carion et al., 2020; Ge et al., 2021; Misra et al., 2021). It has been reported that DETR can achieve SOTA performance, but its long training time is known to be a bottleneck. Sun et al. (2021) reconsidered the difficulty of DETR optimization and pointed out two causes of slow convergence: Hungarian loss and Transformer cross-attention mechanism. They also proposed several methods to solve these problems and showed their effectiveness through experiments.

Zhang et al. (2020) pointed out that methods optimizing assignment-based set loss inadvertently restrict the learnable probability distribution during the loss function selection phase and assume implicitly that the generated target follows a unimodal distribution on the set space, and proposed techniques to address these limitations.

4.4 Set matching

The task of estimating the degree of matching between two sets is referred to as set matching.

Set matching can be categorized into two cases: the homogeneous case and the heterogeneous case. In the homogeneous case, both input sets consist of elements from the same category or type. On the other hand, in the heterogeneous case, the input sets contain elements from different categories or types. Saito et al. (2020) proposes a novel approach for the heterogeneous case, which has not been addressed before. To solve set matching problems, it often relies on learning with negative sampling, and Kimura (2022) provides theoretical analyses for such problem settings. Furthermore, recent research has also reported the task-specific distribution shift in set matching tasks (Kimura, 2023).

4.5 Neural processes

The Neural Process family (Garnelo et al., 2018b;a; Jha et al., 2022), which is an approximation of probabilistic processes using neural networks, has been studied extensively. Garnelo et al. (2018a) first introduced the idea of conditional Neural Processes which model the conditional predictive distribution $p(f(T)|T, C)$, where C is the labeled dataset and T is the unlabeled dataset. One of the necessary conditions for defining a probabilistic process is permutation invariance. In the case of CNPs, to fulfill this condition, the encoder-decoder parts employ the architecture of Deep Sets. However, the output of CNPs consists of a pair of prediction mean and standard deviation, and it is not possible to sample functions like in actual probabilistic processes. Neural Processes (NPs) (Garnelo et al., 2018b) enable function sampling by introducing latent variables into the framework of CNPs.

Kim et al. (2018) showed that NPs are prone to underfitting. They argued that this problem can be solved by introducing an attention mechanism, and proposed Attentive Neural Processes.

4.6 Approximating submodular functions

Submodular set function (Fujishige, 2005; Lovász, 1983; Krause & Golovin, 2014) is one of the important classes of set functions, and there exist many applications. First, we introduce the definitions and known results for the submodular set function.

Definition 4.1. A function $f \colon 2^{\mathcal{V}} \to \mathbb{R}$ is called submodular if it satisfies

$$f(\mathcal{S}) + f(\mathcal{T}) \geq f(\mathcal{S} \cup \mathcal{T}) + f(\mathcal{S} \cap \mathcal{T}), \tag{6}$$

for any $\mathcal{S}, \mathcal{T} \subseteq \mathcal{V}$.

Definition 4.2. A function $f \colon 2^{\mathcal{V}} \to \mathbb{R}$ is supermodular if $-f$ is submodular.

Definition 4.3. A function that is both submodular and supermodular is called modular.

If $f \colon 2^{\mathcal{V}} \to \mathbb{R}$ is a modular function, we have

$$f(\mathcal{S}) + f(\mathcal{T}) = f(\mathcal{S} \cap \mathcal{T}) + f(\mathcal{S} \cup \mathcal{T}), \tag{7}$$

for any $\mathcal{S}, \mathcal{T} \subseteq \mathcal{V}$.

Proposition 4.1 ((Fujishige, 2005))**.** If $f \colon 2^{\mathcal{V}} \to \mathbb{R}$ is modular, it may be written as

$$\begin{aligned} f(\mathcal{S}) &= f(\emptyset) + \sum_{\boldsymbol{s} \in \mathcal{S}} \left(f(\{\boldsymbol{s}\}) - f(\emptyset)\right) &\quad (8)\\ &= c + \sum_{\boldsymbol{s} \in \mathcal{S}} \phi(\boldsymbol{s}), &\quad (9)\end{aligned}$$

for some $\phi \colon \mathcal{V} \to \mathbb{R}$.

From the above definitions and results, we have the following proposition for permutation-invariant neural networks.

Proposition 4.2. We assume that $\rho(\boldsymbol{s}) = \boldsymbol{s}$ for $\boldsymbol{s} \in \mathcal{S}$. Then, for permutation-invariant neural networks, we have

i) The architecture of Deep Sets takes the form of the modular function;

ii) The architecture of PointNet takes the form of the submodular function,

for all $\boldsymbol{x} \in \mathcal{V}$.

Proof. i) Let $c = 0$ and $\rho(\boldsymbol{s}) = \boldsymbol{s}$ in Eq. 9, we can confirm the statement.

ii) For $\mathcal{S}, \mathcal{T} \subseteq \mathcal{V}$, let

$$a \coloneqq \max_{\boldsymbol{s} \in \mathcal{S}} \phi(\boldsymbol{s}), \quad b \coloneqq \max_{\boldsymbol{s} \in \mathcal{T}} \phi(\boldsymbol{s}), \quad c \coloneqq \max_{\boldsymbol{s} \in \mathcal{S} \cap \mathcal{T}} \phi(\boldsymbol{s}), \quad d \coloneqq \max_{\boldsymbol{s} \in \mathcal{S} \cup \mathcal{T}} \phi(\boldsymbol{s}).$$

Then we need to show

$$a + b \geq c + d.$$

Here, we can see that $a, b \geq c$ and thus $\min(a, b) \geq c$. Similarly, we have $\max(a, b) = d$, and

$$\begin{aligned} a + b &= \min(a, b) + \max(a, b) \\ &\geq c + d. \end{aligned}$$

Then, we have the proof. □

Deep Submodular Functions (Dolhansky & Bilmes, 2016) is one of the seminal works on learning-based submodular functions. Furthermore, Tschiatschek et al. (2016) proposes a probability model where the energy function is represented by a parametric submodular function.

4.7 Person re-identification

Person re-identification (Zheng et al., 2015; Liao et al., 2015; Zheng et al., 2017) can be viewed as an approximation problem of set functions since it involves selecting the target element from a set of person images as input. The HAP2S loss (Yu et al., 2018) is proposed with the aim of efficient point-to-set metric learning for the Person re-identification task.

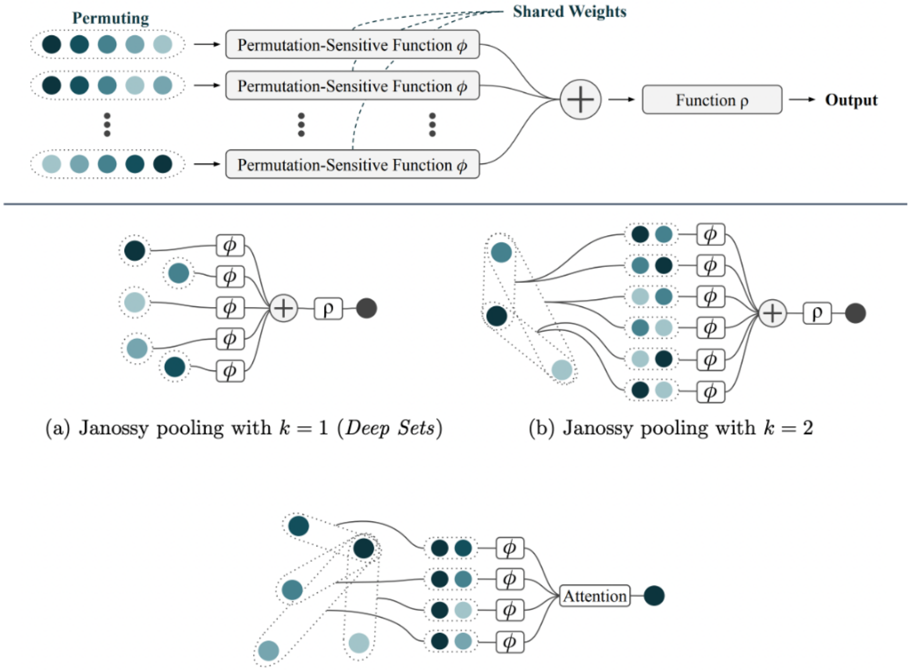
As a variant task of person re-identification, there is also group re-identification (Wei-Shi et al., 2009; Zheng et al., 2014; Lisanti et al., 2017), which involves identifying groups of individuals in images or videos. Lisanti et al. (2017) introduced a visual descriptor that achieves invariance both to the number of subjects and to their displacement within the image in this task. Xiong & Lai (2023) addresses GroupReID between RGB and IR images.

4.8 Other tasks

Metric learning In traditional video-based action recognition methods, task recognition often involves extracting subtasks and performing temporal alignment. However, Wang et al. (2022) suggests that, in some cases, the order of subtasks may not be crucial, and there could be alternative approaches that can achieve similar results. To perform distance learning between the query video and the contrastive videos, a set matching metric is introduced (Wang et al., 2022).

Sinha & Fleuret (2023) propose the permutation-invariant transformer-based model that can estimate the Earth Mover's Distance in quadratic order with respect to the number of elements. They report the effectiveness of the Sinkhorn algorithm in cases where there are constraints on computational costs, as increasing the number of iterations in the Sinkhorn algorithm improves accuracy compared to their proposed algorithm. Cuturi (2013) proposes a parallelizable Sinkhorn algorithm operating on multiple pairs of histograms that function within the GPU environment.

XAI Explainable Artificial Intelligence (XAI) aims to explain the behavior of machine learning models (Gunning et al., 2019; Tjoa & Guan, 2020; Kimura & Tanaka, 2020). Several studies are exploring the combination of approximating set functions and XAI techniques. Cotter et al. (2018) and Cotter et al. (2019) introduce an architecture for approximating interpretable set functions that maintain performance comparable to Deep Sets.


Permuting
Shared Weights
Permutation-Sensitive Function ϕ
Permutation-Sensitive Function ϕ
Permutation-Sensitive Function ϕ
Function ρ
Output
ϕ
ρ
Attention

(a) Janossy pooling with $k = 1$ (*Deep Sets*)

(b) Janossy pooling with $k = 2$

(c) Self-attention

Figure 3: Top panel: The Janossy pooling framework with the same permutation-sensitive network to each possible permutation of the input set, from Figure 1 of Wagstaff et al. (2022). Bottom panel: Different versions and variants of Janossy pooling, from Figure 2 of Wagstaff et al. (2022). These figures are cited to highlight that Janossy pooling is a generalization of Deep Sets and self-attention.

Federated learning Federated learning (Kairouz et al., 2021; Li et al., 2021; Konečný et al., 2016) is a framework of machine learning in which data is distributed without aggregation, and its use is expected to improve the efficiency of data processing. Federated Learning is motivated by a general interest in issues such as data privacy, data minimization, and data access rights. Amosy et al. (2024) proposed the application of Deep Sets to federated learning and reported its usefulness.

5 Theoretical analysis of approximating set functions

In this section we review several known theoretical results on the approximation of set functions.

5.1 Sum-decomposability and Janossy pooling

Definition 5.1. Let (ρ, ϕ) be a sum-decomposition. Write $\mathcal{Z}$ for the domain of ρ (which is also the codomain of ϕ, and the space in which the summation happens in Eq. 3. We refer to $\mathcal{Z}$ as the latent space of the sum-decomposition (ρ, ϕ).

Definition 5.2. Given a space $\mathcal{Z}$, we say that f is sum-decomposable via $\mathcal{Z}$ if f has a sum-decomposition whose latent space is $\mathcal{Z}$.

Definition 5.3. The function f is said to be continuously sum-decomposable when there exists a sum-decomposition (ρ, ϕ) of f such that both ρ and ϕ are continuous. (ϕ, ρ) is then a continuous sum-decomposition of f.

The objective is to construct models that can effectively represent diverse functions while ensuring that the output remains unchanged when the input elements are permuted. This delicate equilibrium guarantees that the models can capture the inherent complexities of the problem at hand while upholding the crucial aspect of permutation invariance.

One unifying framework of methods that learns either strictly permutation-invariant functions or suitable approximations is Janossy pooling (Murphy et al., 2018). Janossy pooling is renowned for its remarkable expressiveness, and its universality can be readily illustrated. It is capable of representing any permutation-invariant function, making it a highly versatile framework. This exceptional property highlights the ability of Janossy pooling to capture intricate relationships and patterns within sets. With its flexibility and effectiveness, Janossy pooling serves as a valuable tool for modeling and analyzing permutation-invariant functions, offering a broad spectrum of applications across diverse domains.

Definition 5.4 (Janossy pooling (Murphy et al., 2018))**.** For any set $\mathcal{S} \subseteq \mathcal{V}$ and its permutations $\Pi_{\mathcal{S}}$, Janossy pooling is defined as the aggregation of outputs of the permutation-sensitive function $\Phi(\mathcal{S})$ for all possible permutations:

$$\hat{f}(\mathcal{S}) = \frac{1}{|\Pi_{\mathcal{S}}|} \sum_{\pi_{\mathcal{S}} \in \Pi_{\mathcal{S}}} \Phi(\pi_{\mathcal{S}}(\mathcal{S})). \tag{10}$$

We can also consider the post-process ρ as

$$f(\mathcal{S}) = \rho\left(\hat{f}(\mathcal{S})\right), \tag{11}$$

and this is the form of sum-decomposable 3.1, and permutation-invariant 2.5.

It is obvious that the computational complexity of Janossy pooling scales at least linearly in the size of $\Pi_{\mathcal{S}}$, which is $|\mathcal{S}|!$. To address this problem, the following strategies are discussed (Murphy et al., 2018):

i) sorting: considering only a single canonical permutation, which is obtained by sorting the inputs;

ii) sampling: aggregating over a randomly-sampled subset of permutations;

iii) restricting permutation to k-subsets: for some $k < |\mathcal{S}|$, let $\mathcal{S}_{\{k\}}$ denote the set of all k-subsets from $\mathcal{S}$, and

$$\hat{f}(\mathcal{S}) = \frac{1}{\tau(|\mathcal{S}|, k)} \sum_{\mathcal{T} \in \mathcal{S}_{\{k\}}} \Phi(\mathcal{T}), \tag{12}$$

where $\tau(|\mathcal{S}|, k) = \frac{|\mathcal{S}|}{(|\mathcal{S}|-k)!}$.

The computational complexity of Eq. 12 is $\mathcal{O}(|\mathcal{S}|^k)$, and for sufficiently small k this gives far fewer that $|\mathcal{S}|!$. Note that the third strategy is the generalization of many practical models (Zaheer et al., 2017; Qi et al., 2017a;b; Lee et al., 2019). Indeed, the case of $k = 1$ is equivalent to Deep Sets Zaheer et al. (2017), and many other current neural network architectures resemble the case of $k = 2$ (see Fig. 3). The function $\hat{f}(\mathcal{S})$ in Eq. 12 is called k-ary Janossy pooling.

Theorem 5.1. *Let $f : \mathbb{R}^M \to \mathbb{R}$ be continuous and permutation invariant. Then f has a continuous k-ary Janossy representation via $\mathbb{R}^M$ for any choice of k.*

Theorem 5.2. *Let $f : \mathbb{R}^M \to \mathbb{R}$ be continuous and permutation invariant. Then f has a continuous M-ary Janossy representation via $\mathbb{R}$.*

The proof of the above theorem is given by examining the definition of Janossy representations.

5.2 Expressive power of Deep Sets and PointNet

Recall that the network architectures of PointNet f_{PN} and Deep Sets f_{DS} are given as

$$f_{\text{PN}} = \rho\left(\max_{\boldsymbol{s}\in\mathcal{S}} \phi(\boldsymbol{s})\right), \tag{13}$$

$$f_{\text{DS}} = \rho\left(\sum_{\boldsymbol{s}\in\mathcal{S}} \phi(\boldsymbol{s})\right). \tag{14}$$

In addition, we can consider the normalized version of Deep Sets $f_{\text{N}-\text{DS}}$ as

$$f_{\text{N-DS}} = \rho\left(\frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{s}\in\mathcal{S}} \phi(\boldsymbol{s})\right). \tag{15}$$

Bueno & Hylton (2021) provides the comparison of representation power and universal approximation theorems of PointNet and Deep Sets. Briefly,

- PointNet (normalized Deep Sets) has the capability to uniformly approximate functions that exhibit uniform continuity in relation to the Hausdorff (Wasserstein) metric.
- When input sets are allowed to be of arbitrary size, only constant functions can be uniformly approximated by both PointNet and normalized Deep Sets simultaneously.
- Even when the cardinality is fixed to a size of k, there exists a significant disparity in the approximation capabilities. Specifically, PointNet is unable to uniformly approximate averages of continuous functions over sets, such as the center-of-mass or higher moments, for $k \geq 3$. Furthermore, an explicit lower bound on the error for the learnability of these functions by PointNet is established.

In the following, we give the reproduction of the key statements of Deep Sets (Zaheer et al., 2017).

Theorem 5.3. The function $f\colon 2^{\mathcal{V}} \to \mathbb{R}$ is sum-decomposable if and only if f is a set function.

Proof. Let $\Phi(\mathcal{S}) \coloneqq \sum_{\boldsymbol{s}\in\mathcal{S}} \phi(\boldsymbol{s})$. Since $\mathcal{V}$ is countable, each $s \in \mathcal{V}$ can be mapped to a unique element in $\mathbb{N}$ by a bijective function $c\colon \mathcal{V} \to \mathbb{N}$. If we can choose ϕ so that Φ is invertible, then we can write $\rho = f \circ \Phi^{-1}$, and $f = \rho \circ \Phi$. Then f is sum-decomposable via $\mathbb{R}$. □

Theorem 5.4. Let $M \in \mathbb{N}$, and $f\colon [0,1]^M \to \mathbb{R}$ be a continuous permutation-invariant function. Then f is continuously sum-decomposable via $\mathbb{R}^{M+1}$.

Theorem 5.5. Deep Sets can represent any continuous permutation-invariant function of M elements if the dimension of the latent space is at least $M+1$.

In addition, later work (Han et al., 2022a) gives explicit bounds on the number of parameters with respect to the dimension and the target accuracy ϵ. In the following, a set function $f\colon 2^{\mathcal{V}} \to \mathbb{R}$ said to be permutation-invariant and differentiable if it satisfies Def. 2.5 and $f(\varphi(s_1), \dots, \varphi(s_{|\mathcal{S}|}))$ is differentiable with mapping $\varphi : \mathcal{V} \to \mathbb{R}^d$.

Theorem 5.6 (Han et al. (2022a))**.** Let $f\colon 2^{\mathcal{V}} \to \mathbb{R}$ be a continuously-differentiable, permutation-invariant function. Let $0 < \epsilon < \|\nabla f\|_2 \sqrt{|\mathcal{S}|d}\mathcal{S}^{-\frac{1}{d}}$, for any $\mathcal{S} \subseteq \mathcal{V}$ with the mapping $\varphi\colon [1, |\mathcal{V}|] \to \mathbb{R}^d$, where $\|\nabla f\|_2 = \max_{\mathcal{S}} \|f(\mathcal{S})\|_2$. Then, there exists $\phi\colon \mathbb{R}^d \to \mathbb{R}^M$, $\rho\colon \mathbb{R}^M \to \mathbb{R}$, such that

$$\left| f(\mathcal{S}) - \rho\left(\sum_{\boldsymbol{s}\in\mathcal{S}} \phi(\boldsymbol{s})\right)\right| = \left| f(\mathcal{S}) - \rho\left(\sum_{i=1}^{|\mathcal{S}|} \phi(\varphi(s_i))\right)\right| < \epsilon, \tag{16}$$

where M, the number of feature variables, satisfies the bound

$$M \leq \frac{2^N (\|\nabla f\|_2^2 |\mathcal{S}| d)^{|\mathcal{S}|d/2}}{\epsilon^{|\mathcal{S}|d} |\mathcal{S}|!}. \tag{17}$$

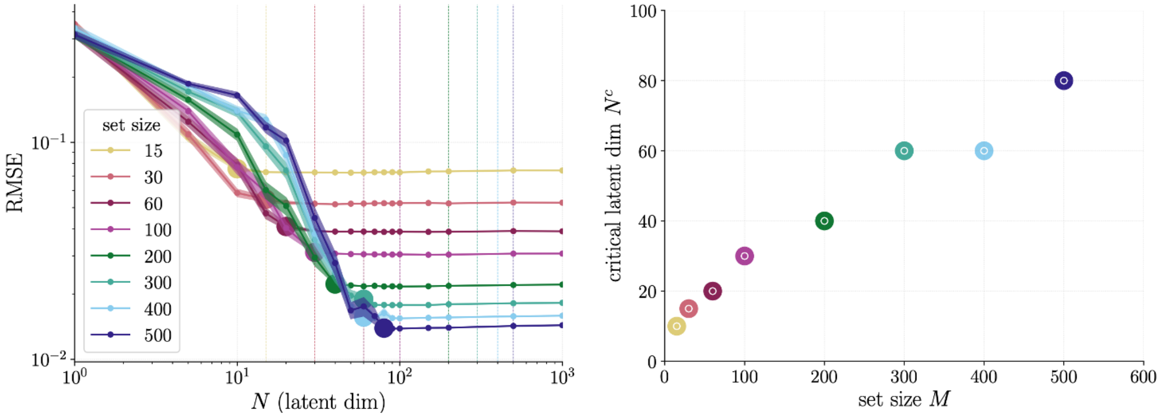


Figure 4: Illustrative toy example, from Figure 3 of Wagstaff et al. (2019). Right panel: Test performance of Deep Sets on median estimation depending on the latent dimension, and dashed lines indicate $N = M$. Here, RMSE is the Root Mean Squared Error. Left panel: Extracted critical points, and the colored data points depict minimum latent dimension for optimal performance for different set sizes. These figures are cited to show experimental results that confirm Theorem 5.7.

Furthermore, Wagstaff et al. (2022) provides a more precise analysis.

Theorem 5.7 (Wagstaff et al. (2022))**.** Let $M, N \in \mathbb{N}$, with $M > N$. Then, there exists continuous permutation-invariant functions $f : \mathbb{R}^M \to \mathbb{R}$ which are not continuously sum-decomposable via $\mathbb{R}^N$.

This implies that for Deep Sets to be capable of representing arbitrary continuous functions on sets of size M, the dimension of the latent space N must be at least M. A similar statement is also true for models based on max-decomposition, such as PointNet (Qi et al., 2017a).

Definition 5.5. The function f is said to be max-decomposable if there are functions ρ and ϕ such that

$$f(\mathcal{S}) = \rho\left(\max_{\boldsymbol{s}\in\mathcal{S}}(\phi(\boldsymbol{s}))\right), \tag{18}$$

where max is taken over each dimension independently in the latent space.

Theorem 5.8. Let $M > N \in \mathbb{N}$. Then there exist continuous permutation-invariant functions $f : \mathbb{R}^M \to \mathbb{R}$ which are not max-decomposable via $\mathbb{R}^N$.

Figure 4 shows the illustrative example for the above theorems. Theorem 5.7 implies that the number of input elements M has an influence on the required latent dimension N. The neural network, which has the architecture of Deep Sets, is trained to predict the median of a set of values. The input sets are randomly drawn from either a uniform, a Gaussian, or a Gamma distribution. This figure shows the relationship between different latent dimensions N, the input set size M, and the predictive performance, and it can be seen that

- The error monotonically decreases with the latent space dimension for every set size;
- Once a specific point is surpassed (referred to as the critical point), enlarging the dimension of the latent space no longer leads to a further reduction in error;
- As the set size increases, the latent dimension at the critical point also grows.

It should be noted that the critical points are observed when $N < M$. The reason behind this phenomenon lies in the fact that the models do not acquire an algorithmic solution for computing the median. Instead, they learn to estimate it based on samples drawn from the input distribution encountered during training.

5.2.1 The Choice of Aggregation

The Deep Set architecture exhibits invariance due to the inherent invariance of the aggregation function). Theoretical justification for summing the embeddings $\phi(\boldsymbol{s})$ is provided by the sum-decomposability (see Section 5.1 for more details). In practice, mean or max-pooling operations are commonly employed, offering simplicity and invariance as well as numerical advantages for handling varying population sizes and controlling input magnitude for downstream layers. This section explores alternative approaches and their respective properties.

Proposition 5.9 (Sum Isomorphism (Soelch et al., 2019))**.** Theorem 5.3 can be extended to aggregations of the form $\alpha_g = g \circ \sum \circ g^{-1}$, i. e. summations in an isomorphic space.

Proof. From $\rho \circ \sum \circ \phi = (\rho \circ g^{-1}) \circ g \circ \sum \circ g^{-1} \circ (g \circ \phi)$, sum decompositions can be constructed from α_g-decompositions and vice versa. □

This class includes mean (with $g((s_1, \ldots, s_{n+1})) = (s_1, \ldots, s_n)/s_{n+1}$, $g^{-1}(\boldsymbol{s}) = (\boldsymbol{s}^\top, 1)^\top$) and logsumexp ($L\Sigma E$) with $g = \ln$. Interestingly, $L\Sigma E$ can behave as max or linear function of summation.

We can observe that divide-and-conquer operations also yield invariant aggregations. Here, Soelch et al. (2019) claims as the commutative and associative binary operations like addition and multiplication yield invariant aggregations. This means that order invariance is equivalent to conquering being invariant to division. In the context of aggregation, order invariance is equivalent to the conquering step remaining invariant to division. This concept extends beyond the realm of basic arithmetic operations and includes logical operators such as any or all, as well as sorting operations that generalize max, min, and percentiles like the median. While these sophisticated aggregations may not be practical for typical first-order optimization, it is worth noting that aggregation techniques can encompass a wide range of complexities. Soelch et al. (2019) propose the learnable aggregation functions, namely recurrent aggregations.

Definition 5.6 (Recurrent and Query Aggregation (Soelch et al., 2019))**.** A recurrent aggregation is a function $f(\mathcal{S}) = \boldsymbol{a}$ that can be written recursively as:

$$\begin{aligned} \boldsymbol{q}_t &= \text{query}(\boldsymbol{q}_{t-1}, \boldsymbol{a}_{t-1}) \\ \hat{w}_{i,t} &= \text{attention}(\boldsymbol{m}_i, \boldsymbol{q}_t) \\ \boldsymbol{w}_t &= \text{normalize}(\hat{\boldsymbol{w}}_t) \\ \boldsymbol{a}_t &= \text{reduce}(\{w_{i,t}, \boldsymbol{m}_i\}) \\ \boldsymbol{a} &= g(\boldsymbol{a}_{1:T}), \end{aligned}$$

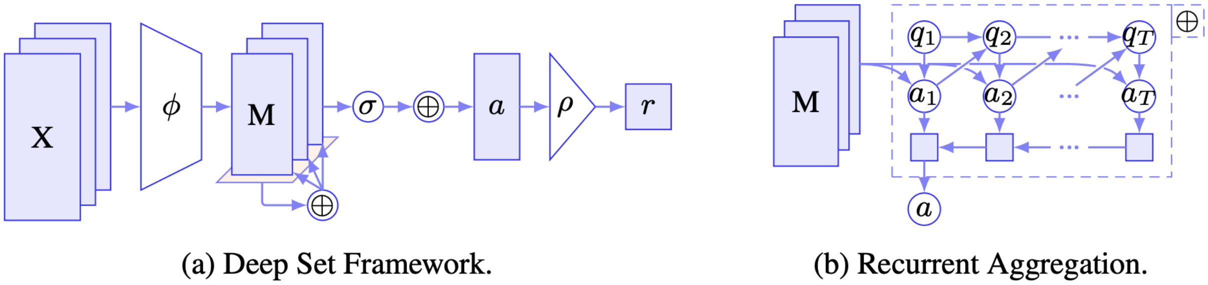
where $\boldsymbol{m}_i = \phi(\boldsymbol{s}_i)$ is an embedding of the input population $\{\boldsymbol{s}_i\}$ and $\boldsymbol{q}_1$ is a constant.

If the reduced operation remains invariant and the normalized operation is equivariant, both recurrent and query aggregations maintain invariance (see Figure 5). Empirical studies show that learnable aggregation functions introduced in this work are more robust in their performance and more consistent in their estimates with growing population sizes. The key insight of this work is that Deep Sets is highly sensitive to the choice of aggregation functions. From this observation, we propose a special class of Deep Sets that has not been studied before in Section 7.

6 Datasets

In this section, we introduce some commonly used datasets for evaluating the performance of neural networks that approximate set functions.

Flow-RBC (Zhang et al., 2022c): The Flow-RBC dataset comprises 98,240 training examples and 23,104 test examples. Each input set represents the distribution of 1,000 red blood cells (RBCs). Each RBC is characterized by volume and hemoglobin content measurements. The task involves regression, aiming to predict the corresponding hematocrit level measured on the same blood sample. In a blood sample, there are various components, including red blood cells, white blood cells, platelets, and plasma. The hematocrit level quantifies the percentage of volume occupied by red blood cells in the blood sample.



(a) Deep Set Framework.

(b) Recurrent Aggregation.

Figure 5: Deep Set architecture and Recurrent aggregation function from Figure 1 of Soelch et al. (2019). This figure is cited to give an overview of their proposed Recurrent aggregation module.

Celebrity Together dataset (Zhong et al., 2018): The Celebrity Together dataset consists of images depicting multiple celebrities together, making it a suitable choice for evaluating set retrieval methods. Unlike other face datasets that only include individual face crops, Celebrity Together comprises full images with multiple labeled faces. The dataset contains a total of 194k images and 546k faces, with an average of 2.8 faces per image.

SHIFT15M (Kimura et al., 2023) SHIFT15M is a dataset designed specifically for assessing models in set-to-set matching scenarios, considering distribution shift assumptions. It allows for evaluating model performance across different levels of dataset shifts by adjusting the magnitude. The dataset contains a total of 2.5m sets and 15m fashion items.

CLEVR (Johnson et al., 2017): CLEVR dataset is a synthetic Visual Question Answering dataset. It contains images of 3D-rendered objects; each image comes with a number of highly compositional questions that fall into different categories. Those categories fall into 5 classes of tasks: Exist, Count, Compare Integer, Query Attribute, and Compare Attribute. The CLEVR dataset consists of: a training set of 70k images and 700k questions, a validation set of 15k images and 150k questions, a test set of 15k images and 150k questions about objects, answers, scene graphs, and functional programs for all train and validation images and questions. Each object present in the scene, aside from position, is characterized by a set of four attributes: 2 sizes: large, and small, 3 shapes: square, cylinder, and sphere, 2 material types: rubber, and metal, 8 color types: gray, blue, brown, yellow, red, green, purple, cyan, resulting in 96 unique combinations.

ShapeNet (Chang et al., 2015): ShapeNet is a large scale repository for 3D CAD models. The repository contains over 300M models with 220,000 classified into 3,135 classes arranged using WordNet hypernym-hyponym relationships. ShapeNet Parts subset contains 31,693 meshes categorized into 16 common object classes (i.e. table, chair, plane, etc.). Each shape's ground truth contains 2-5 parts (with a total of 50 part classes).

ModelNet40 (Wu et al., 2015): ModelNet40 dataset contains 12,311 pre-aligned shapes from 40 categories, which are split into 9,843 for training and 2,468 for testing.

3D MNIST (Xu et al., 2016): 3D MNIST is a 3D version of the MNIST database of handwritten digits. This dataset contains 3D point clouds generated from the original images of the MNIST dataset, and it contains 6,000 instances.

7 A Special Class of Deep Sets: Hölder's Power Deep Sets

As mentioned in the previous sections, Deep Sets and PointNet can be generalized as differences in aggregation functions. Here we consider the following quasi-arithmetic mean:

$$M_f(x_1, x_2, \dots, x_n) \coloneqq f^{-1}\left(\frac{f(x_1) + f(x_2) + \cdots + f(x_n)}{n}\right), \tag{19}$$

for n numbers and f is the injective and continuous function. In particular, let $x_1, x_2, \dots, x_n$ be positive numbers and $f(\cdot) = (\cdot)^p$ for $p \in \mathbb{R}$, we can write the parametric form as

$$M_p(x_1, x_2, \dots, x_n) \coloneqq \left(\frac{1}{n}\sum_{i=1}^{n} x_i^p\right)^{1/p}. \tag{20}$$

The function M^p is called the power mean or Hölder mean (Bickel & Doksum, 2015), and has many applications in machine learning (Van Erven & Harremos, 2014; Kimura & Hino, 2021; Kimura, 2021; Kimura & Hino, 2022; Oh & Kwak, 2016; Nelson, 2017). By taking specific values of p, we have the following special cases:

- $M_{-\infty}(x_1, x_2, \dots, x_n) = \lim_{p \to -\infty} M_p(x_1, x_2, \dots, x_n) = \min\{x_1, x_2, \dots, x_n\}$.
- $M_{-1}(x_1, x_2, \dots, x_n) = n/(\frac{1}{x_1} + \frac{1}{x_2} + \cdots + \frac{1}{x_n})$ (harmonic mean).
- $M_0(x_1, x_2, \dots, x_n) = \lim_{p \to \infty}(x_1, x_2, \dots, x_n) = \sqrt[n]{x_1 + x_2 + \cdots + x_n}$ (geometric mean).
- $M_1(x_1, x_2, \dots, x_n) = (x_1 + x_2 + \cdots + x_n)/n$ (arithmetic mean).
- $M_{+\infty} = \lim_{p \to +\infty}(x_1, x_2, \dots, x_n) = \max\{x_1, x_2, \dots, x_n\}$.

By using this function, we can define the generalized Deep Sets as follows.

Definition 7.1 (Hölder's Power Deep Sets)**.** For a set $\mathcal{S} \subseteq \mathcal{V}$ and $\phi \colon \mathbb{R}^d \to \mathbb{R}_+$, we can define the Hölder's Power Deep Sets is defined as

$$f_{\mathrm{HP-DS}}(\mathcal{S}) \coloneqq \rho\left(M_p^{\phi}(\mathcal{S})\right), \tag{21}$$

where

$$M_p^{\phi}(\mathcal{S}) = \left(\frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{s} \in \mathcal{S}} \phi(\boldsymbol{s})^p\right)^{1/p}. \tag{22}$$

We can see that the Normalized Deep Sets and the PointNet are the special cases with $p = 1$ and $p = +\infty$, respectively.

7.1 Advantages of parametric form

In our Hölder's Power Deep Sets, choosing a power exponent p is equivalent to choosing an architecture such as Deep Sets or PointNet. The most straightforward way to obtain the best estimation is to compare the performance of several of these architectures. Then, for this formulation, we make the following expectations.

i) The optimal parameter p depends on the dataset and problem setting.

ii) With a good choice of p, better performance can be achieved than with Deep Sets or PointNet.

In the following numerical experiments, we confirm these expectations.

Table 2: Experimental results for different values of p in the Hölder's Power Deep Sets. Here, $p = 1$ and $p = +\infty$ give Deep Sets and PointNet, respectively.

dataset	$p = -1$	$p = 0$	$p = 1$ (Deep Sets)	$p = 2$	$p = 3$	$p = +\infty$ (PointNet)
CelebA (↑)	0.871(±**0.003**)	0.863(±0.009)	**0.873**(±0.008)	0.863(±0.004)	0.867(±0.012)	0.862(±0.004)
3D MNIST (↑)	0.746(±0.011)	0.809(±0.003)	0.810(±0.005)	**0.824**(±0.003)	0.820(±**0.002**)	0.823(±0.015)
Flow-RBC (↓)	32.18(±2.855)	30.59(±0.7857)	29.58(±0.997)	28.36(±**0.179**)	**28.34**(±0.301)	45.84(±15.34)

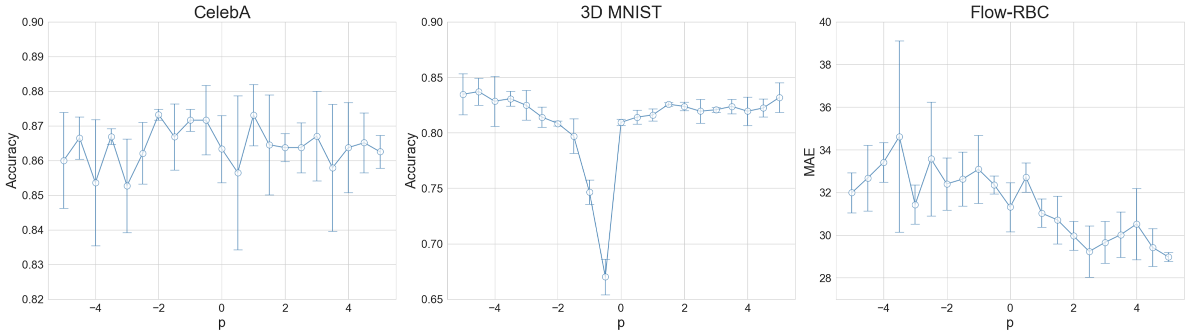


Figure 6: Performance evaluations with $p \in [-5, 5]$.

7.2 Numerical experiments on optimization of generalized Deep Sets

Next, we introduce the experimental results for our generalized Deep Sets.

7.2.1 Experimental settings

Datasets In our numerical experiments, we use three datasets, CelebA (Zhong et al., 2018), 3D MNIST (Xu et al., 2016) and Flow-RBC (Zhang et al., 2022c). See Section 6 for the overview of each dataset. The performance evaluation on CelebA and 3D MNIST is accuracy, while the performance evaluation on Flow-RBC is the mean absolute error. All experiments are conducted on three trials with different random seeds, and the means and standard deviations of the evaluations are reported.

Architecture of neural networks In our experiments, we use simple Deep Sets (Zaheer et al., 2017) consisting of three layers. For this base architecture, we apply the generalized aggregation function in Eq. equation 20.

Computing resources Our experiments use a 16-core CPU with 60 GB memory and a single NVIDIA Tesla T4 GPU.

7.2.2 Experimental results

First, we confirm that changes in the power exponent p in our formulation have an effect on performance. Table 2 shows the experimental results for different values of p in the Hölder's Power Deep Sets. We use $p = -1, 0, 1, 2, 3, +\infty$ in this experiments, and they correspond to the harmonic mean, geometric mean, arithmetic mean, quadratic mean, cubic mean and maximum. Here, $p = 1$ and $p = +\infty$ give Deep Sets and PointNet, respectively. This experiment shows that different p achieve different evaluation performances. In particular, we can observe that the parameters that achieve the best performance in terms of bias differ from those that are best in terms of variance. More precisely, Figure 6 shows the performance evaluations with $p \in [-5, 5]$. Experimental results are reported for different values of p for every 0.5 within this range. From this figure, we can see that the optimal parameter p depends on the dataset, and it does not necessarily correspond to well-known special cases such as Deep Sets or PointNet.

Table 3: Optimization of power exponent for the Hölder's Power Deep Sets. Each cell shows the obtained p, the achieved evaluation performance and the required computation time, respectively.

dataset	linear search	Bayesian optimization	gradient descent
CelebA (↑)	$p = -2.000$ $0.873(\pm 0.002)$ $120,000$ [sec]	$p = -0.521$ $0.867(\pm 0.015)$ $33,000$ [sec]	$p = 0.106$ $0.870(\pm 0.009)$ $3,000$ [sec]
3D MNIST (↑)	$p = -5.500$ $0.843(\pm 0.006)$ $32,000$ [sec]	$p = -6.405$ $0.842(\pm 0.004)$ $3,200$ [sec]	$p = 0.091$ $0.822(\pm 0.011)$ 800 [sec]
Flow-RBC (↓)	$p = 3.000$ $28.34(\pm 0.301)$ $280,800$ [sec]	$p = -0.374$ $33.54(\pm 0.441)$ $203,580$ [sec]	$p = 2.850$ $30.32(\pm 2.685)$ $7,020$ [sec]

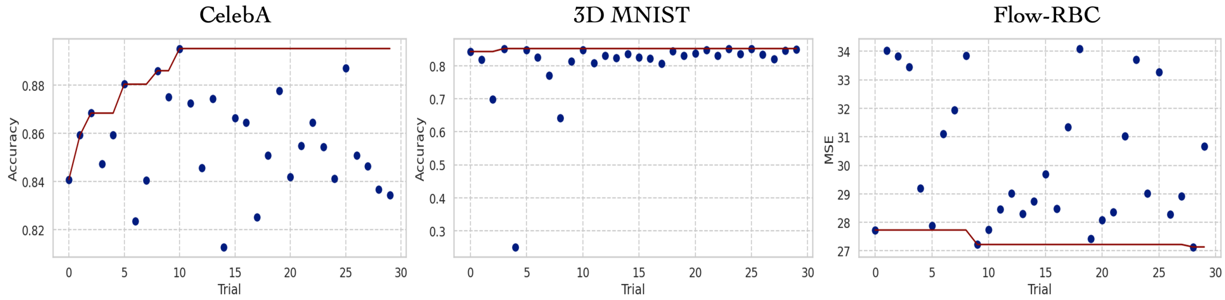


Figure 7: Bayesian optimization history for each dataset.

The above experimental results confirm expectations i) and ii). The next question is how to find a good parameter p efficiently. Enumerating several special forms and comparing their performance, as in the experiment in Table 1, is the simplest way to do this, but it seems to provide too few choices. An alternative way is to try all the values of p within a certain range, as shown in Figure 6, but this is computationally expensive when the search range is wide or when small changes in p are examined. To solve these problems, we consider directly optimizing the power exponent p using Bayesian optimization and gradient descent. In this experiments, we set $p \in [-10, 10]$. For linear search, we compare the performance with different p for every 0.5 within this range. For Bayesian optimization, we use the Optuna (Akiba et al., 2019) and set the number of trials to 30. For the gradient descent, we use the Adam optimizer (Kingma & Ba, 2014) with $lr = 0.001$.

Table 3 shows the experimental results on the optimization of power exponent p for the Hölder's Power Deep Sets. In this table, we report the obtained p, the achieved evaluation performance and the required computation time. In addition, Fig. 7 shows the history of Bayesian optimization for each dataset. From these results, we can see that comparable performance to linear search can be achieved in reasonable computation time by Bayesian optimization and gradient method.

7.3 Limitations of Hölder's Power Deep Sets

Here we discuss several limitations of Hölder's Power Deep Sets. Our architecture leverages the concept of the quasi-arithmetic mean to connect Deep Sets and PointNet in the linear case. However, in order to establish the practical advantages, thorough experimental and theoretical analysis is required when nonlinear ρ is used.

8 Conclusion and discussion

Unlike typical machine learning models that handle vector data, when dealing with set data, it is crucial to ensure permutation-invariance. In this survey, we discussed various neural network architectures that satisfy permutation-invariance and how they are beneficial for a range of tasks. Permutation-invariant architectures not only enhance the ability of models to learn from and make predictions on set data but also open the door to more sophisticated handling of complex structures in machine learning. As we delve deeper into the potential of permutation-invariant neural networks and explore their adaptations for specific tasks, future research will likely focus on refining these models, addressing challenges, and uncovering novel applications. This dynamic landscape promises further advancements in machine learning, bridging the gap between vector and set data to unlock new opportunities for understanding and processing complex information structures.

8.1 Future directions

Future research topics for machine learning models that deal with sets or permutation-invariant vectors may include the following.

- **Research in specific areas is still insufficient**: For example, there is limited work on XAI and federated learning for neural networks dealing with sets mentioned in Section 4.8. The connection between subfields, which are well studied in general machine learning, and models that approximate set functions is meaningful.
- **Availability of more datasets**: As reviewed in Section 6, there are many available set datasets for machine learning, but they are still underdeveloped compared to computer vision and natural language processing. In particular, a de-facto standard dataset of the scale of ImageNet (Deng et al., 2009; Krizhevsky et al., 2012; Russakovsky et al., 2015) in image classification would contribute greatly to the development of the field.

We also consider the following research questions concerning our generalized Deep Sets as follows.

- **Efficient parameter optimization**: The range of parameters introduced in our generalized Deep Sets is $\mathbb{R}$. In our experiments, however, we have a finite number of candidates or ranges and optimize within them. We expect that the usefulness of our formulation would be enhanced if a wider range of parameters could be searched efficiently.
- **Further generalization**: Our Hölder's Power Deep Sets given in Eq. equation 20 can be further generalized in the form of weighted mean as $(\frac{1}{|S|}\sum_{\boldsymbol{s}\in\mathcal{S}} w(\boldsymbol{s})\cdot\phi(\boldsymbol{s})^p)^{1/p}$, with $\sum_{\boldsymbol{s}\in\mathcal{S}} w(\boldsymbol{s}) = 1$. Considering such a generalization allows for more freedom in the model and may allow for further improvements.
- **Theoretical analysis**: In this work, we investigated the behavior of generalized Deep Sets from numerical experiments. The theoretical analysis of this formulation is expected to contribute to further research.
- **Comprehensive experiment**: We performed numerical experiments on three different datasets. Experimental results show that the optimal parameters vary depending on the dataset. We could potentially obtain further insights by conducting similar experiments on a larger number of datasets.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Ohad Amosy, Gal Eyal, and Gal Chechik. Late to the party? on-demand unlabeled personalized federated learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2184–2193, 2024.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6836–6846, 2021.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Peter J Bickel and Kjell A Doksum. *Mathematical statistics: basic ideas and selected topics, volumes I-II package*. CRC Press, 2015.

Marin Biloš and Stephan Günnemann. Scalable normalizing flows for permutation invariant densities. In *International Conference on Machine Learning*, pp. 957–967. PMLR, 2021.

Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. *Advances in neural information processing systems*, 31, 2018.

Christian Bueno and Alan Hylton. On the representation power of set pooling networks. *Advances in Neural Information Processing Systems*, 34:17170–17182, 2021.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Michael Chang, Tom Griffiths, and Sergey Levine. Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation. *Advances in Neural Information Processing Systems*, 35:32694–32708, 2022.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Andrew Cotter, Maya Gupta, Heinrich Jiang, James Muller, Taman Narayan, Serena Wang, and Tao Zhu. Interpretable set functions. *arXiv preprint arXiv:1806.00050*, 2018.

Andrew Cotter, Maya Gupta, Heinrich Jiang, Erez Louidor, James Muller, Tamann Narayan, Serena Wang, and Tao Zhu. Shape constraints for set functions. In *International conference on machine learning*, pp. 1388–1396. PMLR, 2019.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)*, 40(2):1–60, 2008.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Li Deng and Yang Liu. *Deep learning in natural language processing*. Springer, 2018.

Brian W Dolhansky and Jeff A Bilmes. Deep submodular functions: Definitions and learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Gamaleldin Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. *Advances in Neural Information Processing Systems*, 35:28940–28954, 2022.

Patrick Emami, Pan He, Sanjay Ranka, and Anand Rangarajan. Efficient iterative amortized inference for learning symmetric and disentangled multi-object representations. In *International Conference on Machine Learning*, pp. 2970–2981. PMLR, 2021.

Herbert B Enderton. *Elements of set theory*. Academic press, 1977.

Bradley J Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L Kline. Machine learning for medical imaging. *Radiographics*, 37(2):505–515, 2017.

Jiajun Fei, Ziyu Zhu, Wenlei Liu, Zhidong Deng, Mingyang Li, Huanjun Deng, and Shuo Zhang. Dumlp-pin: a dual-mlp-dot-product permutation-invariant network for set feature extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 598–606, 2022.

Satoru Fujishige. *Submodular functions and optimization*. Elsevier, 2005.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International conference on machine learning*, pp. 1704–1713. PMLR, 2018a.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018b.

Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 303–312, 2021.

Robert Gens and Pedro M Domingos. Deep symmetry networks. *Advances in neural information processing systems*, 27, 2014.

Giorgio Giannone and Ole Winther. Scha-vae: Hierarchical context aggregation for few-shot generation. In *International Conference on Machine Learning*, pp. 7550–7569. PMLR, 2022.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. Xai—explainable artificial intelligence. *Science robotics*, 4(37):eaay7120, 2019.

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, 2021.

Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016.

Jiequn Han, Yingzhou Li, Lin Lin, Jianfeng Lu, Jiefu Zhang, and Linfeng Zhang. Universal approximation of symmetric and anti-symmetric functions. *Communications in Mathematical Sciences*, 20(5):1397–1408, 2022a.

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022b.

Felix Hausdorff. *Set theory*, volume 119. American Mathematical Soc., 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Georg Hess, Christoffer Petersson, and Lennart Svensson. Object detection as probabilistic set prediction. In *European Conference on Computer Vision*, pp. 550–566. Springer, 2022.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

ABM Rezbaul Islam. Machine learning in computer vision. In *Applications of Machine Learning and Artificial Intelligence in Education*, pp. 48–72. IGI Global, 2022.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2021a.

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

Saurav Jha, Dong Gong, Xuesong Wang, Richard E Turner, and Lina Yao. The neural process family: Survey, applications and perspectives. *arXiv preprint arXiv:2209.00517*, 2022.

Baoxiong Jia, Yu Liu, and Siyuan Huang. Improving object-centric learning with query optimization. In *The Eleventh International Conference on Learning Representations*, 2022.

Xuebo Jin, Zhi Tao, and Jianlei Kong. Multi-stream aggregation network for fine-grained crop pests and diseases image recognition. *International Journal of Cybernetics and Cyber-Physical Systems*, 1(1):52–67, 2021.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349 (6245):255–260, 2015.

Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matt Botvinick, Alexander Lerchner, and Chris Burgess. Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. *Advances in Neural Information Processing Systems*, 34:20146–20159, 2021.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*, 2021.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations*, 2018.

Jinwoo Kim, Saeyoon Oh, and Seunghoon Hong. Transformers generalize deepsets and can be extended to graphs & hypergraphs. *Advances in Neural Information Processing Systems*, 34:28016–28028, 2021a.

Jinwoo Kim, Jaehoon Yoo, Juho Lee, and Seunghoon Hong. Setvae: Learning hierarchical composition for generative modeling of set-structured data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15059–15068, 2021b.

Masanari Kimura. Generalized t-sne through the lens of information geometry. *IEEE Access*, 9:129619–129625, 2021.

Masanari Kimura. Generalization bounds for set-to-set matching with negative sampling. In *International Conference on Neural Information Processing*, pp. 468–476. Springer, 2022.

Masanari Kimura. On the decomposition of covariate shift assumption for the set-to-set matching. *IEEE Access*, 11:120728–120740, 2023. doi: 10.1109/ACCESS.2023.3324044.

Masanari Kimura and Hideitsu Hino. α-geodesical skew divergence. *Entropy*, 23(5):528, 2021.

Masanari Kimura and Hideitsu Hino. Information geometrically generalized covariate shift adaptation. *Neural Computation*, 34(9):1944–1977, 2022.

Masanari Kimura and Masayuki Tanaka. Interpretation of feature space using multi-channel attentional sub-networks. In *CVPR Workshops*, pp. 36–39, 2019.

Masanari Kimura and Masayuki Tanaka. New perspective of interpretability of deep neural networks. In *2020 3rd International Conference on Information and Computer Technologies (ICICT)*, pp. 78–85. IEEE, 2020.

Masanari Kimura, Takuma Nakamura, and Yuki Saito. Shift15m: Fashion-specific dataset for set-to-set matching with several distribution shifts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3507–3512, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *International Conference on Learning Representations*, 2021.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

Risi Kondor. A novel set of rotationally and translationally invariant features for images based on the non-commutative bispectrum. *arXiv preprint cs/0701127*, 2007.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability*, 3(71-104):3, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

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 *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Azriel Levy. *Basic set theory*. Courier Corporation, 2012.

Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2197–2206, 2015.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI Open*, 2022.

Giuseppe Lisanti, Niki Martinel, Alberto Del Bimbo, and Gian Luca Foresti. Group re-identification via unsupervised transfer of sparse features encoding. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2449–2458, 2017.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. Flatformer: Flattened window attention for efficient point cloud transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1200–1211, 2023.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

László Lovász. Submodular functions and convexity. *Mathematical Programming The State of the Art: Bonn 1982*, pp. 235–257, 1983.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022.

Siddharth Manay, Daniel Cremers, Byung-Woo Hong, Anthony J Yezzi, and Stefano Soatto. Integral invariants for shape matching. *IEEE Transactions on pattern analysis and machine intelligence*, 28(10): 1602–1618, 2006.

Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and equivariant graph networks. *arXiv preprint arXiv:1812.09902*, 2018.

Prem Melville and Vikas Sindhwani. Recommender systems. *Encyclopedia of machine learning*, 1:829–838, 2010.

Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542, 2021.

Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2906–2917, 2021.

Tom M Mitchell. Machine learning, 1997.

Ryan L Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. *arXiv preprint arXiv:1811.01900*, 2018.

Kenric P Nelson. Assessing probabilistic inference by comparing the generalized mean of the model and source probabilities. *Entropy*, 19(6):286, 2017.

Jiyong Oh and Nojun Kwak. Generalized mean for robust principal component analysis. *Pattern Recognition*, 54:116–127, 2016.

Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2):604–624, 2020.

Zijing Ou, Tingyang Xu, Qinliang Su, Yingzhen Li, Peilin Zhao, and Yatao Bian. Learning neural set functions under the optimal subset oracle. *Advances in Neural Information Processing Systems*, 35:35021–35034, 2022.

Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16949–16958, 2022.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pp. 1310–1318. Pmlr, 2013.

Ivens Portugal, Paulo Alencar, and Donald Cowan. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97:205–227, 2018.

Konpat Preechakul, Chawan Piansaddhayanon, Burin Naowarat, Tirasan Khandhawit, Sira Sriswasdi, and Ekapol Chuangsuwanich. Set prediction in the latent space. *Advances in Neural Information Processing Systems*, 34:25516–25527, 2021.

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 conference on computer vision and pattern recognition*, pp. 652–660, 2017a.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017b.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Marco Reisert. *Group integration techniques in pattern analysis: a kernel view*. PhD thesis, Freiburg (Breisgau), Univ., Diss., 2008, 2008.

Yong Rui, Thomas S Huang, and Shih-Fu Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, 10(1):39–62, 1999.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

Yuki Saito, Takuma Nakamura, Hirotaka Hachiya, and Kenji Fukumizu. Exchangeable deep neural networks for set-to-set matching and learning. In *European Conference on Computer Vision*, pp. 626–646. Springer, 2020.

Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. *Advances in Neural Information Processing Systems*, 35:9512–9524, 2022.

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, 29, 2016.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, et al. Bridging the gap to real-world object-centric learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Atul Kumar Sinha and François Fleuret. Deepemd: A transformer-based fast estimation of the earth mover's distance. 2023.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12):1349–1380, 2000.

Maximilian Soelch, Adnan Akhundov, Patrick van der Smagt, and Justin Bayer. On deep set learning and the choice of aggregations. In *Artificial Neural Networks and Machine Learning–ICANN 2019: Theoretical Neural Computation: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part I 28*, pp. 444–457. Springer, 2019.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *Advances in neural information processing systems*, 29, 2016.

Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision.* Cengage Learning, 2014.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 61–70, 2020.

Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris M Kitani. Rethinking transformer-based set prediction for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3611–3620, 2021.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Zineng Tang, Jaemin Cho, Jie Lei, and Mohit Bansal. Perceiver-vl: Efficient vision-and-language modeling with iterative latent attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4410–4420, 2023.

Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE transactions on neural networks and learning systems*, 32(11):4793–4813, 2020.

Sebastian Tschiatschek, Josip Djolonga, and Andreas Krause. Learning probabilistic submodular diversity models via noise contrastive estimation. In *Artificial Intelligence and Statistics*, pp. 770–779. PMLR, 2016.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020.

Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

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

Oscar Vikström and Alexander Ilin. Learning explicit object-centric representations with vision transformers. *arXiv preprint arXiv:2210.14139*, 2022.

Edward Wagstaff, Fabian Fuchs, Martin Engelcke, Ingmar Posner, and Michael A Osborne. On the limitations of representing functions on sets. In *International Conference on Machine Learning*, pp. 6487–6494. PMLR, 2019.

Edward Wagstaff, Fabian B Fuchs, Martin Engelcke, Michael A Osborne, and Ingmar Posner. Universal approximation of functions on sets. *Journal of Machine Learning Research*, 23(151):1–56, 2022.

Xiang Wang, Shiwei Zhang, Zhiwu Qing, Mingqian Tang, Zhengrong Zuo, Changxin Gao, Rong Jin, and Nong Sang. Hybrid relation guided set matching for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19948–19957, 2022.

Zheng Wei-Shi, Gong Shaogang, and Xiang Tao. Associating groups of people. In *Proceedings of the British Machine Vision Conference*, pp. 23–1, 2009.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

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 conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. In *The Eleventh International Conference on Learning Representations*, 2022.

Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 915–924, 2021.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

Jianghao Xiong and Jianhuang Lai. Similarity metric learning for rgb-infrared group re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13662–13671, 2023.

Xiaofan Xu, Alireza Dehghani, David Corrigan, Sam Caulfield, and David Moloney. Convolutional neural network for 3d object recognition using volumetric representation. In *2016 first international workshop on sensing, processing and learning for intelligent machines (SPLINE)*, pp. 1–5. IEEE, 2016.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

Rui Yu, Zhiyong Dou, Song Bai, Zhaoxiang Zhang, Yongchao Xu, and Xiang Bai. Hard-aware point-to-set deep metric for person re-identification. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 188–204, 2018.

Jinyang Yuan, Bin Li, and Xiangyang Xue. Unsupervised learning of compositional scene representations from multiple unspecified viewpoints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8971–8979, 2022.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2114–2124, 2021.

Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Patchformer: An efficient point transformer with patch attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11799–11808, 2022a.

David W Zhang, Gertjan J Burghouts, and Cees GM Snoek. Set prediction without imposing structure as conditional density estimation. *arXiv preprint arXiv:2010.04109*, 2020.

Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2736–2746, 2022b.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*, 2019a.

Lily Zhang, Veronica Tozzo, John Higgins, and Rajesh Ranganath. Set norm and equivariant skip connections: Putting the deep in deep sets. In *International Conference on Machine Learning*, pp. 26559–26574. PMLR, 2022c.

Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8552–8562, 2022d.

Yan Zhang, Jonathon Hare, and Adam Prugel-Bennett. Deep set prediction networks. *Advances in Neural Information Processing Systems*, 32, 2019b.

Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. Fspool: Learning set representations with featurewise sort pooling. In *International Conference on Learning Representations*, 2019c.

Yan Zhang, David W Zhang, Simon Lacoste-Julien, Gertjan J Burghouts, and Cees GM Snoek. Multiset-equivariant set prediction with approximate implicit differentiation. *arXiv preprint arXiv:2111.12193*, 2021.

Yan Zhang, David W Zhang, Simon Lacoste-Julien, Gertjan J Burghouts, and Cees GM Snoek. Unlocking slot attention by changing optimal transport costs. In *NeurIPS'22 Workshop on All Things Attention: Bridging Different Perspectives on Attention*, 2022e.

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pp. 1116–1124, 2015.

Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Group association: Assisting re-identification by visual context. *Person Re-Identification*, pp. 183–201, 2014.

Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE international conference on computer vision*, pp. 3754–3762, 2017.

Yujie Zhong, Relja Arandjelovic, and Andrew Zisserman. Compact deep aggregation for set retrieval. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pp. 0–0, 2018.

Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021a.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021b.

Zhi-Hua Zhou. *Machine learning*. Springer Nature, 2021.

Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2639–2650, 2023.

Berend Zwartsenberg, Adam Scibior, Matthew Niedoba, Vasileios Lioutas, Justice Sefas, Yunpeng Liu, Setareh Dabiri, Jonathan Wilder Lavington, Trevor Campbell, and Frank Wood. Conditional permutation invariant flows. *Transactions on Machine Learning Research*, 2023.