HORSE: Hierarchical Representation for Large-Scale Neural Subset Selection

Binghui Xie, Yixuan Wang, Yongqiang Chen, Kaiwen Zhou, Yu Li, Wei Meng, James Cheng
Department of Computer Science and Engineering
The Chinese University of Hong Kong

Abstract

Subset selection tasks, such as anomaly detection and compound selection in AI-assisted drug discovery, are crucial for a wide range of applications. Learning subset-valued functions with neural networks has achieved great success by incorporating permutation invariance symmetry into the architecture. However, existing neural set architectures often struggle to either capture comprehensive information from the superset or address complex interactions within the input. Additionally, they often fail to perform in scenarios where superset sizes surpass available memory capacity. To address these challenges, we introduce the novel concept of the *Identity Property*, which requires models to integrate information from the originating set, resulting in the development of neural networks that excel at performing effective subset selection from large supersets. Moreover, we present the Hierarchical Representation of Neural Subset Selection (HORSE), an attention-based method that learns complex interactions and retains information from both the input set and the optimal subset supervision signal. Specifically, HORSE enables the partitioning of the input ground set into manageable chunks that can be processed independently and then aggregated, ensuring consistent outcomes across different partitions. Through extensive experimentation, we demonstrate that HORSE significantly enhances neural subset selection performance by capturing more complex information and surpasses the state-of-the-art methods in handling large-scale inputs by a margin of up to 20%.

1 Introduction

Set-valued functions are of great importance to a wide range of real-world applications. For example, anomaly detection aims to identify a set of outliers from a larger dataset that could be users or financial transactions [Zhang et al., 2020]. Another example is the recommender system, where the objective is to identify a set of products that better satisfy customer preferences [Ou et al., 2022]. In these scenarios, there is a need for implicitly learning a set function [Rezatofighi et al., 2017, Zaheer et al., 2017] that quantifies the usefulness of a given subset of the inputs. The set function assigns a utility value to each subset, and the subset with the highest utility corresponds to the most desired output.

To illustrate the concept, let us consider the task of a recommender system. In this task, we aim to recommend a subset of items S from a larger item pool V, denoted as $S \in V$, that maximizes the utility of S with respect to the satisfaction of the customers. The utility can be captured by a parameterized utility function, denoted as $F_\theta(S; V)$, and our goal is to optimize the following criteria:

$$S^* = \arg\max_{S \in 2^V} F_\theta(S; V). \tag{1}$$

One straightforward method [Balcan and Harvey, 2018] involves explicitly modeling the utility by learning the function $U = F_\theta(S; V)$ using supervised data. This data consists of pairs

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

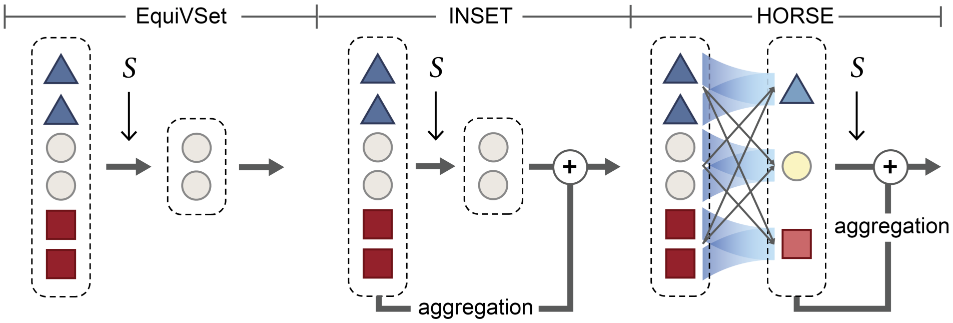

Figure 1: Comparison of HORSE to the state-of-the-arts EquiVSet and INSET in handling subsets. "S" represents the supervision, indicating the specific subset of interest. "+" refers to the aggregation of different vectors, which is implemented through concatenation in practice. Unlike EquiVSet and INSET, the HORSE model captures more complex information from V by employing attention mechanisms. Furthermore, HORSE facilitates the division of V into distinct partitions.

$\{(S_i, V_i), U_i\}_{i=1}^N$, where U_i represents the actual utility value of subset S_i given the respective item pool V_i. However, this training approach becomes challenging to implement due to the significant number of supervision signals required, which can be expensive and time-consuming to acquire [Ou et al., 2022]. To overcome this limitation, an alternative approach is to tackle Eq. 1 using an implicit learning method from a probabilistic perspective [Tschiatschek et al., 2018]. This approach requires utilizing data in the form of $\{(V_i, S_i^*)\}_{i=1}^N$, where S_i^* represents the optimal subset corresponding to V_i. The objective is to estimate the parameters θ such that Eq. 1 holds for all possible (V_i, S_i). In practical training, with limited data sampled from the underlying distribution $P(S, V)$, the empirical log-likelihood $\sum_{i=1}^N [\log p_\theta(S^*|V)]$ is maximized for all the data pairs $\{S, V\}$, where $p_\theta(S|V)$ is proportional to $F_\theta(S; V)$ for all $S \in 2^V$ [Ou et al., 2022, Xie et al., 2024]. Additional details are available in Appendix D.3.

The crux of the matter lies in determining the structure of neural networks that can effectively model $F_\theta(S, V)$ throughout the entire process. One commonly employed approach in the literature is to utilize an encoder to generate feature vectors for each element in V. These vectors are then inputted into DeepSets [Zaheer et al., 2017], along with the corresponding supervised subset S, in order to learn the permutation invariant set function $F(S)$. However, this methodology may neglect the interaction between S and V, leading to a reduction in the expressive capacity of the models. In the previous study, Xie et al. [2024] suggest incorporating the sum-pooling representation of V into S to enhance the performance. Yet, the simple integration in Xie et al. [2024], Wang et al. [2024b] limits its capacity to effectively model interactions among elements or subsets within these sets. Furthermore, the approach struggles with high-cardinality sets V, as encoding the entire set into memory may not be feasible [Bruno et al., 2021].

To address these problems, and inspired by [Willette et al., 2023], we introduce the notion of the *Identity Property*, a desirable concept for the effective functioning of the model $F(S, V)$. Identity Property requires $F(S, V)$ to accurately reflect which set V the information S originates from. In order to capture the interplay between S and V by adhering to the Identity Property, we propose a subset-based attentive set encoder. Additionally, this encoder facilitates the division of a large set V into smaller and manageable subsets. These subsets can be processed independently and later aggregated, ensuring no loss of information from V. Hence, our approach is able to efficiently handle large-scale subset representation learning. As depicted in Figure 1, our method is capable of modeling more complex information and managing large-scale inputs more effectively than the two state-of-the-art approaches in the field of Neural Subset Selection tasks, EquiVSet [Ou et al., 2022] and INSET [Xie et al., 2024].

In this work, we make several contributions to the field of neural subset selection, which can be summarized as follows:

- We introduce and rigorously define a critical concept termed as the *Identity Property* for neural subset selection. This property requires that models can reliably determine the source set V from which the information of the subset S is derived, which is a crucial requirement for neural subset selection tasks.

- To adhere to the Identity Property and model complex interaction, we present a subset-based attention mechanism. This mechanism is crafted to learn the Hierarchical Representation of Neural Subset Selection, denoted as HORSE. Our theoretical analysis confirms that HORSE not only upholds the Identity Property but also maintains Permutation Invariance.
- Through extensive empirical research, we validate the effectiveness of HORSE. Our experiments across a variety of datasets demonstrate the consistently superior performance of HORSE. Additionally, we specifically explore HORSE's capabilities in large set environments, further showcasing its practical applicability and efficiency compared with the baselines.

2 Related Work

2.1 Set Encoding.

The exploration of network architectures tailored for set-structured inputs has become a vibrant area of research in recent years. A number of key studies [Ravanbakhsh et al., 2017, Edwards and Storkey, 2017, Zaheer et al., 2017, Qi et al., 2017, Horn et al., 2020, Bloem-Reddy and Teh, 2020, Wang et al., 2023] have laid the groundwork in this domain, primarily focusing on creating models that are permutation equivariant using conventional feed-forward neural networks. These foundational models have been successful in universally approximating continuous permutation-invariant functions, primarily through the application of set-pooling layers to aggregate information across different elements of a set regardless of their order.

However, these methodologies have primarily concentrated on learning representations at the aggregate set level, paying less attention to more nuanced interactions occurring at elements. Recognizing this gap, more recent research efforts have aimed at introducing more sophisticated interaction modeling within invariant set functions for various applications. A notable example is the work by [Lee et al., 2019b], which incorporates self-attention mechanisms to facilitate the processing of elements within sets, thereby effectively capturing element-wise interactions. Moreover, the concept of Janossy pooling, proposed by [Murphy et al., 2018], introduced a novel approach to incorporate higher-order interactions within the pooling process. Since then, subsequent studies have built upon this advancement, leading to further refinements and innovations in the field, e.g., [Kim, 2021, Li et al., 2020, Bruno et al., 2021, Willette et al., 2023].

2.2 Hierarchical Set Function.

The existing literature primarily concentrates on processing entire input sets, often overlooking the information provided by the sub-levels. Addressing this oversight, Maron et al. [2020] introduced an innovative approach that integrates the symmetry of elements to generate representations of an input set. This methodology was further expanded into a broader context by Wang et al. [2020]. Moreover, Bevilacqua et al. [2022] proposed a novel framework aimed at enhancing graph representations by including whole-graph representations to encode each subgraph. Along similar lines, Xie et al. [2024] developed an information aggregation module designed to learn $F(S, V)$ effectively.

Despite these advancements, a gap remains in the current research landscape. These methods tend to overlook more complex interactions between elements or subsets within sets. Furthermore, they often fall short in scenarios where the input set has a significantly large cardinality, indicating a need for more scalable and interaction-sensitive approaches in set processing. In Table 1, we compare our proposed method with importance baselines commonly used in subset selection tasks. Specifically, DeepSets

Table 1: Properties of Various Methods: "Attn" indicates the use of the attention mechanism, "V" signifies the explicit utilization of information from V, and 'Large-scale' denotes the capability of the methods to generalize effortlessly to large-scale settings.

Model	Attn	V	Large-Scale
DeepSets [Zaheer et al., 2017]	✗	✗	✓
Set Transformer [Lee et al., 2019a]	✓	✓	✗
EquiVSet [Ou et al., 2022]	✗	✗	✓
INSET [Xie et al., 2024]	✗	✓	✗
HORSE	✓	✓	✓

can handle large-scale settings but may lose complex information due to its simple pooling-based structure. Set Transformer excels at modeling complex information within sets but faces challenges with large input set sizes. Methods tailored for subset selection tasks, like EquiVSet and INSET, may struggle with learning intricate interactions and often overlook large-scale settings. `HORSE` is designed to address these drawbacks.

2.3 Core Subset Selection

Recent work has focused on extracting subsets from training datasets to decrease cost and improve effectiveness [Wei et al., 2015, Mirzasoleiman et al., 2020, Yang et al., 2023]. This research also highlights the importance of modeling the relationship between the original dataset and its subsets. Unlike neural subset selection, these core subsets are unlabeled, and typically, more data in the core subset enhances the performance of the models. Our approach differs in that our optimal subset is labeled within the training set, and its size is constrained.

3 Method

3.1 Preliminaries

In this paper, we concentrate on the development of neural networks for the purpose of modeling the hierarchical set function $F(S, V)$, a critical component for tasks involving sets, such as neural subset selection. For every ground set V, assumed to consist of n elements represented as x_i, that is, $V = \{x_1, x_2, ..., x_n\}$, each element x_i belonging to $\mathcal{X}$ is characterized by a d-dimensional tensor. Typically, the ground set V can be conceptualized as an assembly of multiple disjoint subsets, explicitly $V = S_1 \cup S_2 \cup \cdots \cup S_m$, where $S_i \cap S_j = \emptyset$ for $i \neq j$ and each S_i is a subset in $\mathbb{R}^{n_i \times d}$. In this context, n_i denotes the number of elements in subset S_i. Generally, $S \subseteq V$ acts as a supervisory signal in the form of a mask over V to indicate the elements to be selected. For the sake of clarity, we define S as the concrete subset derived from this mask.

In the context of neural subset selection, the task entails the encoding of subsets S_i into representative vectors to forecast the associated function value $Y \in \mathcal{Y}$. Traditional approaches, such as those documented in Zaheer et al. [2017] and Ou et al. [2022], involve directly selecting S_i based on the encoding embeddings of all elements within V, subsequently feeding S_i into feed-forward networks. Nonetheless, these methods model the function $F(S_i, V)$ solely based on the explicit subsets S_i, potentially leading to suboptimal results due to the omission of the broader context provided by the ground set V. This section introduces a novel attention-based method for encoding subset representations, which distinctively incorporates information from the entire input set V, thereby enhancing performance.

3.2 Identity Property

To effectively model $F(S, V)$, Xie et al. [2024] have proposed to combine the representations of V and S. In practice, this involves utilizing two DeepSets architectures, as proposed by Zaheer et al. [2017], to independently process S and V before merging their outputs, as presented by Figure 1. Given that set pooling operations process each element independently, certain information about the interactions among elements is inevitably lost. This omission can render some problems more challenging than necessary. To address this issue and facilitate the learning of complex interactions within sets, we introduce the following principles:

Property 3.1. *Consider $V \in \mathbb{R}^{n \times d}$ and $S \subseteq V$ where $S \in \mathbb{R}^{s \times d}$, assuming that V is partitioned into a random collection of disjoint subsets $V = S_1 \cup S_2 \cup \cdots \cup S_m$. Here, m varies within the range $[1, n]$, dependent on the chosen method of partitioning. The function F is said to satisfy the Identity Property if and only if there exist functions g and h such that*

$$F(h(S), h(V)) = F(h(S), g(h(S_1), \ldots, h(S_m))), \tag{2}$$

where g serves as an aggregation function that effectively combines the encoded representations of the subsets, ensuring that F leverages both the specific subset S and the ground set V through the transformations applied by h and the aggregation by g.

The method introduced by Xie et al. [2024] is notable for satisfying the Identity Property through its utilization of sum-pooling to simultaneously process all elements. However, this approach may not be practical for scenarios involving large inputs and may struggle to capture more complex information. In response to these limitations, we propose an attention-based method designed to fulfill the requirements of the Identity Property. Additionally, our interpretation of this property accommodates scenarios where S differs from the union of subsets $\{S_1 \cup S_2 \cup \cdots \cup S_m\}$. In practice, S is often chosen to be S_1 for simplicity. This nuanced approach allows for greater flexibility and effectiveness in encoding set information, especially in complex or large-scale settings.

3.3 Attention-Based Set Representation

In this section, we introduce a formulation for an attention-based set encoding function F, leveraging the concept of partitions (referred to as slots in [Bruno et al., 2021, Willette et al., 2023]). Given a ground set $V \in \mathbb{R}^{n\times d}$, we randomly divide it into m subsets. For each subset, we allocate a unique embedding $s_i \in \mathbb{R}^{d_s}$. Furthermore, we establish $\zeta = [s_1, \ldots, s_m]^T$ as a matrix in $\mathbb{R}^{m\times d_s}$. Similar to [Willette et al., 2023], we initialize ζ by sampling m embeddings s_i from a parameterized Gaussian distribution with random initialization. Following this setup, we calculate the unnormalized attention scores between ζ and V, facilitating a dynamic weighting of elements within V based on their relevance to each partition's embedding. This process aims to capture the nuanced interrelations within subsets and between elements and their corresponding subsets.

$$q = LN(\zeta W^q), \tag{3}$$

$$k = VW^k,$$

$$v = VW^v, \tag{4}$$

In this expression, "LN" represents Layer Normalization, and the matrices $W^q \in \mathbb{R}^{d_s\times d_h}$, $W^k \in \mathbb{R}^{d\times d_h}$, and $W^v \in \mathbb{R}^{d\times d_h}$ are introduced. These matrices serve to project V and ζ into a shared dimensional space d_h. Subsequently, we employ a dot product attention mechanism to assess the interactions between V and ζ. This process is governed by the specified formulation, strategically aligning elements of V with the embeddings in ζ through dimensional congruence, thus enabling a nuanced, attention-driven analysis of set elements in relation to their partitioned subsets.

$$\hat{M} = \sqrt{d_h^{-1}} \cdot qk^T, \tag{5}$$

$$\hat{A} = \sigma(\hat{M}) \in \mathbb{R}^{m\times n}, \tag{6}$$

where σ denotes an element-wise activation function. Utilizing the unnormalized attention scores, denoted by $\hat{A}$, we proceed to define the following mapping operation:

$$\bar{h} = \mathrm{nl}(\hat{A})v. \tag{7}$$

In this context, $\bar{h}$ signifies a transformation function mapping from $\mathbb{R}^{n\times d}$ to $\mathbb{R}^{m\times d_h}$. The term "nl" represents a normalization operation, defined as follows:

$$\mathrm{nl}(\hat{A})_{i,j} = \hat{A}_{i,j} / \sum_{i=1}^{m} \hat{A}_{i,j}, \tag{8}$$

which normalizes the column of $\hat{A}$. Then, we can apply a pooling function (such as sum, mean, min, or max) across the columns of $\bar{h}(V)$ and select the sigmoid function for σ, thereby establishing an attention mechanism akin to the SSE (Set Stream Embedding) method proposed by Bruno et al. [2021]. However, this approach has its limitations. Given that the attention score $\mathrm{nl}(\hat{A})_{i,j}$ is calculated independently of the other $n-1$ attention scores, it is not feasible for the rows of $\mathrm{nl}(\hat{A})$ to form convex coefficients, unlike the softmax outputs typically observed in conventional attention mechanisms, as described by Willette et al. [2023].

To address this issue, we follow Willette et al. [2023] to aggregate information across all rows of $\mathrm{nl}(\hat{A})$, thereby incorporating dependencies among different set elements into the attention mechanism. This is achieved through a specific normalization process, outlined as follows:

$$M = diag(\mathrm{nl}(\hat{A})\mathbf{1}_n)^{-1} \tag{9}$$

where $\mathbf{1}_n = (1, \ldots, 1) \in \mathbb{R}^n$ represents a vector of ones of dimension n, and $M_2 \in \mathbb{R}^m$ signifies a vector in an m-dimensional space. Subsequently, we can compute $h(V)$ by applying the normalization

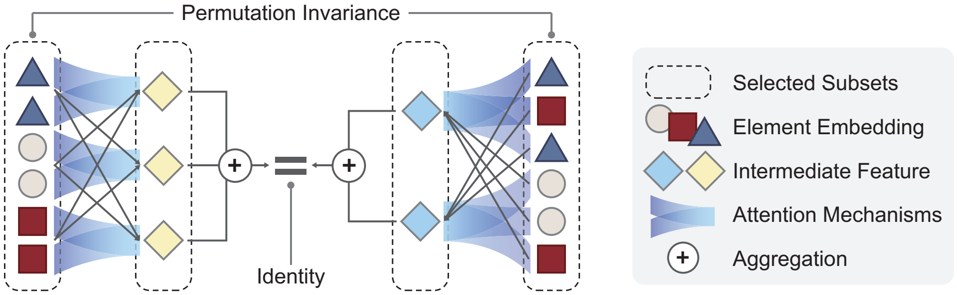

Figure 2: This figure illustrates the HORSE model's capability to achieve Permutation Invariance and satisfy the Identity Property in subset selection tasks. It demonstrates that HORSE maintains consistent output despite the permutation of input set elements and the partition if the ground set.

term M, as follows:

$$h(V) = M\text{nl}(\hat{A})VW^v. \tag{10}$$

Since S_i is a subset of V based on a partition method, $h_V(S_i)$ can be derived from $h(V)$. Detailed steps are provided in Algorithm 1. For simplicity, we omit V and use $h(S_i)$ going forward. This process ensures that h meets the criteria specified in Eq. 2. By constructing such an h function, we ensure that the model can recognize the input set V regardless of its partitioning, leading to the property that $g(h(S_1), h(S_2), \ldots, h(S_m))$ yields the same value for any partition of V.

Furthermore, it facilitates the learning of interactions among the partitioned segments of V, essentially enabling the model to identify the characteristics of the input ground set V. Specifically, we concatenate $h(S)$ with $g(h(S_1), h(S_2), \ldots, h(S_m))$. In practice, a Multilayer Perceptron (MLP) is utilized to process the concatenated tensor into a vector $Z \in \mathbb{R}^{d_o}$, achieving the following:

$$Z = F(h(S), g(h(S_1), h(S_2), \ldots h(S_m))) \in \mathbb{R}^{d_o}. \tag{11}$$

Given that the aforementioned process delineates the entire calculation in a matrix format, which may be complex for some readers, we have taken steps to enhance comprehension. To better illustrate how our method establishes an attention map between subsets S_i, we have detailed the procedural steps in the Appendix (see Algorithm 1), with a particular focus on the generation of $h(S_i)$. This will not only clarify the operational details but also emphasize the underlying methodology and thought process involved in constructing $h(S_i)$.

4 Theoretical Results

In the realm of machine learning, particularly within the scope of set-based tasks, a fundamental requirement is the invariance to the permutation of input set elements. This characteristic ensures that the computation or outcome of a task is unaffected by the order in which the set's elements are presented, a principle that is especially pertinent to neural subset selection tasks. To address and formalize this aspect within the context of our proposed method, we present a theorem that rigorously demonstrates the permutation invariance of our approach.

Theorem 4.1. *Let $\mathbb{S}_n$ denote the set of all permutations of a given set V. Since $V \in \mathbb{R}^{n\times d}$ is represented by a matrix, let $\pi_V \in \mathbb{R}^{n\times n}$ be a random permutation applied to V. Given that $S \subseteq V$ represents a subset of V, the permutation π_V naturally induces a corresponding permutation π_S on S. Under these conditions,* `HORSE` *exhibits permutation invariance, which is defined as:*

$$F(h(S), h(V)) = F(h(\pi_S \cdot S), h(\pi_V \cdot V)) \tag{12}$$

Theorem 4.1 assures that irrespective of how the elements in the input set V are ordered, the output generated by HORSE remains consistent. This property is crucial for ensuring the reliability and applicability of our method across a wide spectrum of set-based tasks, where the inherent order of data points should not influence the task outcome. Furthermore, an illustration of the underlying concept is provided in Figure 2.

Table 2: Product recommendation results for 12 different product categories. The best results are indicated in bold black, while the second-best results are highlighted in blue. Due to space limitation, we use Set-T to denote Set Transformer.

Categories	Random	PGM	DeepSet	Set-T	EquiVSet	INSET	HORSE
Gear	7.7	47.1±0.4	37.9±0.5	64.7±0.6	72.5±1.1	80.8±1.2	**83.2±1.3**
Bath	7.6	56.4±0.8	41.8±0.7	71.6±0.5	76.4±2.0	86.2±0.5	**87.6±1.0**
Toys	8.3	4.41±0.4	42.1±0.5	62.5±2.0	68.4±0.4	76.9±0.5	**77.4±0.9**
Media	9.4	44.1±0.9	42.6±0.4	53.0±2.0	55.4±0.5	62.0±2.3	**65.2±1.5**
Safety	6.5	25.0±0.6	22.1±0.4	23.4±0.9	23.1±2.0	23.8±1.5	**26.9±1.2**
Diaper	8.4	58.3±0.9	45.1±0.3	78.9±0.5	82.8±0.7	**88.3±0.7**	88.0±0.8
Health	7.6	44.9±0.2	45.2±0.1	69.2±1.2	70.5±0.9	81.2±0.5	**81.6±0.6**
Carseats	6.6	23.1±1.0	21.2±0.8	22.0±1.0	22.3±1.9	23.0±2.4	**24.8±2.2**
Bedding	7.9	48.5±0.6	48.1±0.2	76.2±2.2	76.2±0.5	85.7±1.1	**87.1±0.7**
Feeding	9.3	56.3±0.8	42.8±0.2	75.3±0.6	81.9±0.9	88.5±0.5	**90.3±1.1**
Apparel	9.0	53.3±0.5	50.8±0.4	68.0±2.0	76.4±0.5	83.7±0.3	**85.4±0.6**
Furniture	6.5	17.5±0.7	16.8±0.2	17.6±0.8	16.2±2.0	16.7±3.5	**18.1±1.5**

Theorem 4.2. *If σ represents a strictly positive element-wise activation function, then HORSE satisfies Property 3.1.*

By satisfying this property, HORSE ensures that its encoding captures both the individual characteristics of subsets within V and the overarching structure of the entire set, thereby facilitating a more nuanced and comprehensive understanding of set information. The proofs is inspired by [Willette et al., 2023] and can be found in Appendix A.

5 Experiments

In this section, we aim to demonstrate that HORSE significantly outperforms baseline models in a suite of benchmarks tailored to Neural Subset Selection tasks. Subsequently, we extend our investigation to scenarios involving large-scale input settings. Due to the page limitation, we have included additional experiments in Appendix D.

Evaluations. To assess the performance of various methods, we employ the mean Jaccard coefficient (MJC) as the evaluation metric. This metric quantifies the similarity between the predicted subset S' and the true subset S^* for each data sample (S^*, V). The Jaccard coefficient is calculated as follows: $JC(S^*, S') = \frac{|S^* \cap S'|}{|S^* \cup S'|}$, where the intersection and union operations determine the size of the overlap and the total unique elements in both sets, respectively. The MJC is then derived by averaging the Jaccard coefficient across all test set samples. Please note that all the following reported performance metrics are presented in percentages, with a default multiplication factor of 100%.

Baselines. We conducted experiments compared with several approaches: Random, PGM [Tschiatschek et al., 2018], DeepSet [Zaheer et al., 2017], Set Transformer [Lee et al., 2019a], EquiVSet [Ou et al., 2022], and INSET [Xie et al., 2024]. The Random approach represents the expected performance of a random guess. DeepSet and Set Transformer are well-known methods or frameworks that satisfy permutation invariance, making them suitable for Neural Subset Selection tasks. PGM, EquiVSet and INSET are specifically designed for subset selection tasks. More comprehensive details available in Appendix D.

Table 3: Performance results on the Two-Moons and Gaussian-Mixture datasets. Bolded numbers denote the best performance on each dataset

Method	Two Moons	Gaussian Mixture
Random	5.5	5.5
PGM	36.0 ± 2.0	43.8 ± 0.9
DeepSet	47.2 ± 0.3	44.6 ± 0.2
Set Transformer	57.4 ± 0.2	90.5 ± 0.2
EquiVSet	58.5 ± 0.3	90.7 ± 0.2
INSET	58.2 ± 0.3	90.9 ± 0.2
HORSE	**60.2** ± **0.5**	**91.8** ± **0.2**

Table 4: Empirircal results of compound selection Tasks. Bolded numbers denote the best performance on each dataset. Due to space limitations, we use "Set-T" to denote Set Transformer.

	Random	PGM	DeepSet	Set-T	EquiVSet	INSET	HORSE
PDBBind	9.9	91.0±1.0	90.1±1.1	91.9±1.5	92.4±1.1	93.5±0.8	**94.1 ± 0.7**
BindingDB	9.0	69.0±2.0	71.0±2.0	71.5±1.0	72.1±0.9	73.4±1.0	**74.2 ± 1.1**
PDBBind-2	7.3	35.0±0.9	32.3±0.4	35.5±1.0	35.7±0.5	37.1±1.0	**43.2 ± 0.6**
BindingDB-2	2.7	17.6±0.6	16.5±0.5	18.3±0.4	18.8±0.6	19.8±0.5	**21.3 ± 0.5**
Average	7.23	53.15	52.48	54.30	54.75	55.95	**58.20**

5.1 Synthetic Experiments

Firstly, We validate the effectiveness of our models through experimental trials focused on learning set functions, using two synthetic datasets: the two-moons dataset [Pedregosa et al., 2011] with an added noise variance of $\sigma^2 = 0.1$, and and a Gaussian mixture represented as $\frac{1}{2}\mathcal{N}(\mu_0, \Sigma) + \frac{1}{2}\mathcal{N}(\mu_1, \Sigma)$.

Take the Gaussian mixture as an example, the data generation procedure as follows: i) Initially, we select an index, denoted as b, using a Bernoulli distribution with a probability of $\frac{1}{2}$. ii) Subsequently, we sample 10 points from the Gaussian distribution $\mathcal{N}(\mu_b, \Sigma)$ to construct the set S^*. iii) Further, we sample 90 points for $V \backslash S^*$ from the Gaussian distribution $\mathcal{N}(\mu_{1-b}, \Sigma)$. We follow the procedure of Ou et al. [2022] to obtain 1,000 samples, subsequently divided into training, validation, and test sets. We effectively demonstrate the efficacy of our approach in mastering complex set functions. Detailed results can be found in Table 3.

5.2 Product Recommendation

The task involves recommending the most suitable subset of 30 products to a customer within a specific category. For this experiment, we utilize the dataset from the Amazon baby registry, sourced from Gillenwater et al. [2014a]. This dataset includes numerous product subsets chosen by various customers, with Amazon categorizing each item on a baby registry into specific categories such as "Bath", "Health" and "Feeding". Detailed information can be found in Appendix D.

Table 2 presents the performance of all models across different categories. Notably, out of the twelve cases evaluated, HORSE outperforms other models in 11 of them. Even in the Diaper category, our method achieves results that are comparable to INSET, which is noteworthy. These significant improvements highlight the effectiveness and superiority of HORSE. Specifically, while EquiVSet and INSET struggle to surpass classical neural subset selection baselines in the Safety, Car Seats, and Furniture categories, HORSE consistently outperforms all baselines in a notable manner.

5.3 Compound Selection in AI-aided Drug Discovery

In drug discovery, the screening of compounds with diverse biological activities and favorable ADME (absorption, distribution, metabolism, and excretion) properties is a critical step [Li et al., 2021, Ji et al., 2022, Gimeno et al., 2019]. Virtual screening typically involves a sequential filtering process that employs multiple essential filters. However, neural networks encounter challenges when learning the complete screening process. This difficulty arises from the absence of intermediate supervision signals, which can be costly or impossible to obtain due to pharmaceutical protection policies. Therefore, we implement a single filter, namely, high bioactivity, to obtain the optimal subset of compound selection, following the methodology in [Ou et al., 2022]. Our experiments are conducted on two datasets: PDBBind [Liu et al., 2015a] and BindingDB [Liu et al., 2007]. To be more practical, we further enhance our approach with the inclusion of *two filters*: the high bioactivity filter and the diversity filter. This extended analysis is denoted as PDBBind-2 and BindingDB-2, representing the two-stage filtering process, for a more practical perspective.

Table 4 demonstrates that our method outperforms the baselines and significantly surpasses random guessing, especially on the BindingDB-2 and PDBBind-2 datasets. However, the improvement on PDBBind and BindingDB is less significant. This marginal enhancement is due to the informative structural characteristics of complexes (the elements within a set), which inherently provide substantial information for this task. Consequently, the model can effectively predict the activity values of

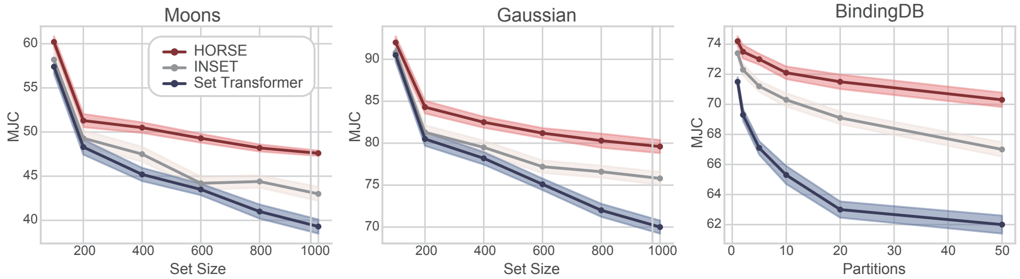

Figure 3: The performance of the methods on the Two-Moons and Gaussian Datasets with respect to the set size is examined in the left two subfigures. These subfigures provide insights into how the performance of the methods varies as the size of the input sets changes. The right subfigure focuses on the influence of the number of partitions on the performance using the BindingDB dataset.

complexes even without explicitly considering the interactions between the optimal subset and its complement in single filter scenarios. Nonetheless, our method still achieves superior results compared to other methods, confirming its effectiveness.

5.4 Large-Scale Setting

Set encoding mechanisms, as introduced by Zaheer et al. [2017] and further explored by Lee et al. [2019a], have fundamentally shifted the way neural networks perceive and process sets by emphasizing the importance of permutation invariance and the ability to handle variable-sized inputs. These models are designed to learn from the entire set in a single gradient step, ensuring that the learned representations encapsulate the holistic properties of the set. However, this approach encounters practical limitations when dealing with large-scale sets, where processing the entire set in a single step becomes computationally infeasible due to memory constraints or the sheer volume of data.

To circumvent these challenges, an effective strategy involves training models on partitions of the set, sampled dynamically at each iteration of the optimization process [Lee et al., 2019a, Wang et al., 2024a]. This method allows for manageable subsets to be used for training, significantly reducing the computational load. However, this method will lose information [Bruno et al., 2021, Willette et al., 2023] since it does not process all the elements from V. Therefore, we instead partition the set elements into mini-batches, independently encode each batch, and aggregate them to obtain a single set encoding. By applying this methodology across both baseline models and HORSE in scenarios characterized by large-scale input sets, we can highlight the efficiency and scalability of our proposed solution. We conducted experiments on the Two-Moons and Gaussian-Mixture datasets. To ensure consistency, we set the size of the optimal subset S^* to be 10. Subsequently, we varied the size of the input ground set V within the range of $\{200, 400, 600, 800, 1000\}$. Notably, the memory capacity of the GeForce RTX 3090 is insufficient when the size reaches 600. The ground set was divided into 5 disjoint partitions, with each partition containing one-fifth of the elements in V. For the purpose of comparison, we selected INSER and Set Transformer as baselines alongside our proposed method, HORSE. INSET demonstrated the best performance among baselines, while Set Transformer is an alternative method that incorporates an attention mechanism. The results obtained from these experiments are presented in the left two subfigures of Figure 3. It is evident that HORSE outperforms both INSER and Set Transformer by a significant margin, demonstrating its superior effectiveness in handling large-scale sets.

Furthermore, to further enhance the practicality of our approach and investigate the potential impact of the partition numbers on the results, we conducted additional experiments on the BindingDB dataset. In this experiment, we set the size of the optimal subset S^* to be 15, while the size of the ground set remained fixed at 1000. We partitioned the ground set into a range of 2 to 50 partitions. The results of these experiments are presented in the right subfigure of Figure 3. Remarkably, it becomes evident that HORSE exhibits remarkable robustness with respect to the number of partitions considered. Regardless of the specific partitioning scheme employed, HORSE consistently delivers exceptional performance, which is more robust than our baselines.

6 Conclusion

In this paper, we have addressed the limitations observed in existing methods for neural subset selection tasks. These methods often struggle to effectively model complex information and lack scalability when dealing with large-scale inputs. To overcome these challenges, we propose an innovative and scalable approach called HORSE, which leverages the power of the attention mechanism. Theoretically, we establish that HORSE satisfies the Identity Property and Permutation Invariance, ensuring its soundness and effectiveness. Empirically, we thoroughly evaluate the performance of HORSE against various baselines in both standard and large-scale settings.

Limitation and Future Work. Our theoretical and empirical results demonstrate how the attention mechanism can enhance models for neural subset selection tasks in both standard and large-scale settings. However, in large-scale scenarios, our support is currently limited to a theoretical framework for partitioning the set into different groups within a synthetic distributed setting, rather than practical experimentation in a real distributed environment. Moving forward, we plan to implement and test our model in more practical, real-world scenarios to further validate its effectiveness.

7 Acknowledgements

We thank all the reviewers for their valuable comments. This work was supported by Research Grants 8601116, 8601594, and 8601625 from the UGC of Hong Kong.

References

M. Balcan and N. J. A. Harvey. Submodular functions: Learnability, structure, and optimization. *SIAM J. Comput.*, 47(3):703–754, 2018.

B. Bevilacqua, F. Frasca, D. Lim, B. Srinivasan, C. Cai, G. Balamurugan, M. M. Bronstein, and H. Maron. Equivariant subgraph aggregation networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

B. Bloem-Reddy and Y. W. Teh. Probabilistic symmetries and invariant neural networks. *J. Mach. Learn. Res.*, 21:90:1–90:61, 2020.

A. Bruno, J. Willette, J. Lee, and S. J. Hwang. Mini-batch consistent slot set encoder for scalable set encoding. *Advances in Neural Information Processing Systems*, 34:21365–21374, 2021.

H. Edwards and A. J. Storkey. Towards a neural statistician. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

J. Gillenwater, A. Kulesza, E. B. Fox, and B. Taskar. Expectation-maximization for learning determinantal point processes. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3149–3157, 2014a.

J. A. Gillenwater, A. Kulesza, E. Fox, and B. Taskar. Expectation-maximization for learning determinantal point processes. *Advances in Neural Information Processing Systems*, 27, 2014b.

A. Gimeno, M. J. Ojeda-Montes, S. Tomás-Hernández, A. Cereto-Massagué, R. Beltrán-Debón, M. Mulero, G. Pujadas, and S. Garcia-Vallvé. The light and dark sides of virtual screening: what is there to know? *International journal of molecular sciences*, 20(6):1375, 2019.

M. Horn, M. Moor, C. Bock, B. Rieck, and K. Borgwardt. Set functions for time series. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4353–4363. PMLR, 13–18 Jul 2020.

Y. Ji, L. Zhang, J. Wu, B. Wu, L. Huang, T. Xu, Y. Rong, L. Li, J. Ren, D. Xue, H. Lai, S. Xu, J. Feng, W. Liu, P. Luo, S. Zhou, J. Huang, P. Zhao, and Y. Bian. Drugood: Out-of-distribution (OOD) dataset curator and benchmark for ai-aided drug discovery - A focus on affinity prediction problems with noise annotations. *CoRR*, abs/2201.09637, 2022.

M. Kim. Differentiable expectation-maximization for set representation learning. In *International Conference on Learning Representations*, 2021.

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

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR, 2019a.

J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR, 2019b.

S. Li, J. Zhou, T. Xu, L. Huang, F. Wang, H. Xiong, W. Huang, D. Dou, and H. Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 975–985. ACM, 2021.

Y. Li, H. Yi, C. Bender, S. Shan, and J. B. Oliva. Exchangeable neural ode for set modeling. *Advances in Neural Information Processing Systems*, 33:6936–6946, 2020.

T. Liu, Y. Lin, X. Wen, R. N. Jorissen, and M. K. Gilson. Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, 35(suppl_1): D198–D201, 2007.

Z. Liu, Y. Li, L. Han, J. Li, J. Liu, Z. Zhao, W. Nie, Y. Liu, and R. Wang. Pdb-wide collection of binding data: current status of the pdbbind database. *Bioinformatics*, 31(3):405–412, 2015a.

Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015b.

H. Maron, O. Litany, G. Chechik, and E. Fetaya. On learning sets of symmetric elements. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6734–6744. PMLR, 2020.

B. Mirzasoleiman, J. Bilmes, and J. Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.

R. L. Murphy, B. Srinivasan, V. Rao, and B. Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. *arXiv preprint arXiv:1811.01900*, 2018.

Z. Ou, T. Xu, Q. Su, Y. Li, P. Zhao, and Y. Bian. Learning neural set functions under the optimal subset oracle. *NeurIPS*, 2022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 77–85. IEEE Computer Society, 2017.

S. Ravanbakhsh, J. Schneider, and B. Poczos. Equivariance through parameter-sharing. In *International conference on machine learning*, pages 2892–2901. PMLR, 2017.

S. H. Rezatofighi, V. K. BG, A. Milan, E. Abbasnejad, A. Dick, and I. Reid. Deepsetnet: Predicting sets with deep neural networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5257–5266. IEEE, 2017.

S.-H. Sun. Multi-digit mnist for few-shot learning, 2019.

S. Tschiatschek, A. Sahin, and A. Krause. Differentiable submodular maximization. *arXiv preprint arXiv:1803.01785*, 2018.

R. Wang, M. Albooyeh, and S. Ravanbakhsh. Equivariant networks for hierarchical structures. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13806–13817. Curran Associates, Inc., 2020.

Z. Wang, X. Li, J. Wang, Y. Kuang, M. Yuan, J. Zeng, Y. Zhang, and F. Wu. Learning cut selection for mixed-integer linear programming via hierarchical sequence model. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Zob4P9bRNcK`.

Z. Wang, Q. Y. Jie Wang, Y. Bai, X. Li, J. H. Lei Chen, M. Yuan, B. Li, Y. Zhang, and F. Wu. Towards next-generation logic synthesis: A scalable neural circuit generation framework. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Z. Wang, J. Wang, D. Zuo, Y. Ji, X. Xia, Y. Ma, J. Hao, M. Yuan, Y. Zhang, and F. Wu. A hierarchical adaptive multi-task reinforcement learning framework for multiplier circuit design. In *Forty-first International Conference on Machine Learning*. PMLR, 2024b.

K. Wei, R. Iyer, and J. Bilmes. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR, 2015.

J. Willette, S. Lee, B. Andreis, K. Kawaguchi, J. Lee, and S. J. Hwang. Scalable set encoding with universal mini-batch consistency and unbiased full set gradient approximation. In *International Conference on Machine Learning*, pages 37008–37041. PMLR, 2023.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

B. Xie, Y. Bian, K. zhou, Y. Chen, P. Zhao, B. Han, W. Meng, and J. Cheng. Enhancing neural subset selection: Integrating background information into set representations, 2024.

Y. Yang, H. Kang, and B. Mirzasoleiman. Towards sustainable learning: Coresets for data-efficient deep learning. In *International Conference on Machine Learning*, pages 39314–39330. PMLR, 2023.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola. Deep sets. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3391–3401, 2017.

D. W. Zhang, G. J. Burghouts, and C. G. Snoek. Set prediction without imposing structure as conditional density estimation. *arXiv preprint arXiv:2010.04109*, 2020.

A Proof

A.1 Proof of Theorem 4.1

Proof. Consider a given set $V = \{x_1, x_2, \dots, x_n\} \in \mathbb{R}^{n \times d}$, where n represents the number of elements in V. Let $\mathbb{S}_n$ represent the set of all permutations of V. Now, suppose π_V denotes a random permutation applied to V. By utilizing this permutation π_V, we can construct a permutation matrix $M \in \mathbb{R}^{n \times n}$:

$$MV = \begin{bmatrix} -x_{\pi_V(1)}^\top- \\ \vdots \\ -x_{\pi_V(n)}^\top- \end{bmatrix}.$$

Assuming the use of an activation function σ that is strictly positive for each element,

$$\begin{aligned} \sigma(\sqrt{d^{-1}} \cdot q(MVW^K)^\top) &= \sigma(\sqrt{d^{-1}} \cdot q(VW^K)^\top M^\top) \\ &= \sigma(\sqrt{d^{-1}} \cdot qk^\top)M^\top \\ &= \hat{A}M^\top. \end{aligned}$$

The normalized attention score $\text{nl}(\hat{A})$ can be computed using the given permutation π_V, resulting in

$$\begin{aligned} \text{nl}(\hat{A}M^\top) = &\begin{bmatrix} \hat{A}_{1,\pi(1)} / \sum_{i=1}^{k} \hat{A}_{i,\pi(1)} & \cdots & \hat{A}_{1,\pi(N)} / \sum_{i=1}^{k} \hat{A}_{i,\pi(n)} \\ \vdots & \ddots & \vdots \\ \hat{A}_{k,\pi(1)} / \sum_{i=1}^{k} \hat{A}_{i,\pi(1)} & \cdots & \hat{A}_{k,\pi(n)} / \sum_{i=1}^{k} \hat{A}_{i,\pi(n)} \end{bmatrix} \\ &= \text{nl}(\hat{A})M^\top. \end{aligned} \tag{13}$$

Now, we consider the matrix multiplication of

$$\text{nl}(\hat{A})M^\top = \begin{bmatrix} \text{nl}(\hat{A})_{1,\pi(1)} & \cdots & \text{nl}(\hat{A})_{1,\pi(n)} \\ \vdots & \ddots & \vdots \\ \text{nl}(\hat{A})_{k,\pi(1)} & \cdots & \text{nl}(\hat{A})_{k,\pi(n)} \end{bmatrix}$$

and

$$MVW^V = \begin{bmatrix} \mathbf{x}_{\pi(1)}^\top W^V \\ \vdots \\ \mathbf{x}_{\pi(n)}^\top W^V \end{bmatrix}.$$

Since

$$\begin{bmatrix} \text{nl}(\hat{A})_{1,\pi(1)} & \cdots & \text{nl}(\hat{A})_{1,\pi(n)} \\ \vdots & \ddots & \vdots \\ \text{nl}(\hat{A})_{k,\pi(1)} & \cdots & \text{nl}(\hat{A})_{k,\pi(n)} \end{bmatrix} \begin{bmatrix} x_{\pi(1)}^\top W^V \\ \vdots \\ x_{\pi(N)}^\top W^V \end{bmatrix}$$

is equal to

$$\begin{bmatrix} \sum_{j=1}^{N} \text{nl}(\hat{A})_{1,\pi(j)} x_{\pi(j)}^\top W^V \\ \vdots \\ \sum_{j=1}^{N} \text{nl}(\hat{A})_{m,\pi(j)} x_{\pi(j)}^\top W^V \end{bmatrix},$$

which can also be formulated as:

$$
\begin{aligned}
&= \begin{bmatrix} \sum_{j=1}^{N} \text{nl}(\hat{A})_{1,j} x_j^\top W^V \\ \vdots \\ \sum_{j=1}^{N} \text{nl}(\hat{A})_{m,j} x_j^\top W^V \end{bmatrix} \\
&= \text{nl}(\hat{A}) v. \qquad (14)
\end{aligned}
$$

Therefore, $\text{nl}(\hat{A})v$ is permutation invariant under the permutation group of $\mathbb{S}_n$. Since

$$
\text{nl}(\hat{A})\mathbf{1}_n = \sum_{j=1}^{n} \text{nl}(\hat{A})_{i,j} = \sum_{j=1}^{N} \text{nl}(\hat{A})_{i,\pi_V(j)},
$$

thus $\text{diag}\left(\text{nl}(\hat{A})\mathbf{1}_n\right)^{-1}$ is also invariant with respect to the permutation of input V, which leads to the conclusion that

$$
h(MV) = h(V).
$$

Similarly, we can construct the permutation matrix $M \in R^{s \times s}$ for a given S and permutatation π_S, such that:

$$
MS = \begin{bmatrix} -x_{\pi_S(1)}^\top - \\ \vdots \\ -x_{\pi_S(s)}^\top - \end{bmatrix}.
$$

with the same process as Equation 13 and 14, we can have the following conclusion:

$$
F(h(S), h(V)) = F(h(\pi_S \cdot S), h(\pi_V \cdot V)).
$$

□

B Proof of Theorem 4.2

Proof. Consider the input set $V \in \mathbb{R}^{n \times d}$ and let $V = S_1 \cup S_2 \cup \cdots \cup S_m$ represent a partition of V with $|S_i| = n_i$. In other words, V can be expressed as the union of all S_i and each S_i is disjoint from S_j for $i \neq j$. Without loss of generality, we can make the assumption that.

$$
k = VW^k = \begin{bmatrix} S_1 W^k \\ \vdots \\ S_m W^k \end{bmatrix}, \quad v = VW^v = \begin{bmatrix} S_1 W^v \\ \vdots \\ S_m W^v \end{bmatrix}
$$

where $S_i W^k \in \mathbb{R}^{n_i \times d_h}$ and $S_i W^v \in \mathbb{R}^{n_i \times d_h}$ for $i = 1, 2 \ldots, m$. Then we can express the matrix $\text{nl}(\hat{A})$ as follows:

$$
\text{nl}(\hat{A}) = \left[\text{nl}(\hat{A}^{(1)}) \cdots \text{nl}(\hat{A}^{(m)})\right],
$$

where $\hat{A}^{(i)} = \sigma(\sqrt{d^{-1}} \cdot q(S_i W^k)^\top) \in \mathbb{R}^{m \times n_i}$ for $i = 1, 2 \ldots, m$ since $\text{nl}(\hat{A})_{i,j}$ is independent to $\text{nl}(\hat{A})_{i,t}$ for all $t \neq j$.

Since

$$
\bar{h} = \left[\text{nl}(\hat{A}^{(1)}) \cdots \text{nl}(\hat{A}^{(m)})\right] \begin{bmatrix} S_1 W^v \\ \vdots \\ S_m W^v \end{bmatrix},
$$

the following equality holds

$$\bar{h} = \sum_{i=1}^{m} \mathrm{nl}(\hat{A}^{(i)}) S_i W^v \tag{15}$$

Since

$$M_i = \sum_{t=1}^{m} \sum_{j=1}^{n_i} \mathrm{nl}(\hat{A}^{(t)})_{i,j},$$

we can decompose $\mathrm{nl}(\hat{A})\mathbf{1}_n$ as

$$\begin{aligned} \mathrm{nl}(\hat{A})\mathbf{1}_n &= \sum_{i=1}^{m} \left(\sum_{j=1}^{n_i} \mathrm{nl}(\hat{A}^{(i)})_{1,j}, \dots, \sum_{j=1}^{n_i} \mathrm{nl}(\hat{A}^{(i)})_{m,j} \right)^{\top} \\ &= \sum_{i=1}^{m} \mathrm{nl}(\hat{A}^{(i)}) \mathbf{1}_{n_i} \end{aligned} \tag{16}$$

where $\mathbf{1}_{n_i} = (1, \dots, 1) \in \mathbb{R}^{n_i}$. Combining Equation 15 and 16

$$h(V) = \mathrm{diag}\left(\sum_{i=1}^{m} \mathrm{nl}(\hat{A}^{(i)}) \mathbf{1}_{n_i} \right)^{-1} \left(\sum_{i=1}^{m} \mathrm{nl}(\hat{A}^{(i)}) S_i W^v \right).$$

We define $h_1(S_i) = \mathrm{nl}(\hat{A}^{(i)})\mathbf{1}_{n_i}$ and $h_2(S_i) = \mathrm{nl}(\hat{A}^{(i)}) S_i W^v$. Moreover, $h(S_i) = \mathrm{diag}(h_1(S_i))^{-1} h_2(S_i)$ Now, we define a function,

$$\begin{aligned} &g(\{h(S_1), \dots, h(S_m)\}) \coloneqq \\ &\qquad g_1(\{h_1(S_1), \dots, h_1(S_m)\}) \cdot g_2(\{h_2(S_1), \dots, h_2(S_m)\}), \end{aligned}$$

where

$$\begin{aligned} g_1(\{h_1(S_1), \dots h_1(S_m)\} &\coloneqq \mathrm{diag}\left(\sum_{i=1}^{m} h_1(S_i) \right)^{-1} \\ g_2(\{h_2(S_1), \dots, h_2(S_m)\}) &\coloneqq \sum_{i=1}^{m} h_2(S_i). \end{aligned}$$

Then $h(V) = g(\{h(S_1), \dots, h(S_m)\})$. Since the partition is arbitrary, h satisfies Property 3.1.

□

C Pseudo Code of HORSE

In the main text, we present HORSE using matrix calculations, which may be challenging to comprehend. To improve understanding of how our method establishes an attention map between subsets S_i, we detail the procedural steps in Algorithm 1, with a special emphasis on the generation of $h(S_i)$. This approach is designed to elucidate the operational details and highlight the methodology involved in constructing $h(S_i)$.

Algorithm 1 HORSE. $V = \{S_1, S_2, \ldots, S_m\}$ is the input set partitioned into m subsets. $\xi \in \mathbb{R}^{m \times d_s}$ is the initialized embedding and g is the choice of aggregation function.

```
Input: $V = \{S_1, S_2, \ldots, S_m\}$, $S = S_1$, $\zeta \in \mathbb{R}^{m \times d_s}$, $g$
Output: $Z \in \mathbb{R}^{d_o}$
Initialize $\zeta$
$q = LN(\zeta W^q)$
for $i = 1, 2, \ldots, m$ do
    $k_i = S_i W^k$
    $v_i = S_i W^v$
end for
$k = [k_1, k_2, \ldots, k_m]^T$
$v = [v_1, v_2, \ldots, v_m]^T$
$\hat{M} = \sqrt{d_h^{-1}} \cdot qk^T$
$\hat{A} = \sigma(M_1)$
$A = \mathrm{nl}(\hat{A}) = [A_1, A_2, \ldots, A_m]^T$
$M = diag(\mathrm{nl}(A)\mathbf{1}_n)^{-1} = [M_1, \ldots, M_m]^T$
for $i = 1, 2, \ldots, m$ do
    $h(S_i) = M_i A_i S_i W^v$
end for
$\hat{S}_i = g(h(S_1), \ldots h(S_m))$
$Z = F(h(S), \hat{S}_i)$
return $Z$
```

D Experimental Details and Additional Experiments

D.1 Detailed Description of Tasks.

Product Recommendation. The task involves recommending the most suitable subset of 30 products to a customer within a specific category. For this experiment, we utilize the dataset from the Amazon baby registry, sourced from Gillenwater et al. [2014a]. This dataset includes numerous product subsets chosen by various customers, with Amazon categorizing each item on a baby registry into specific categories such as "Bath", "Health" and "Feeding". Additionally, each product is represented by a 768-dimensional vector generated by a pre-trained BERT model, based on its textual description.

The Amazon baby registry data [Gillenwater et al., 2014b] comprises various datasets collected from Amazon, encompassing different categories such as toys, furniture, and more. Within each category, there exist $|V|$ sets of products that have been selected by different customers. To create a sample (S^*, V), we follow a specific procedure. Initially, we remove any subset with an optimal subset size $|S^*|$ greater than or equal to 30. The remaining subsets are then divided into training, validation, and test folds using a 1:1:1 ratio. Additionally, we randomly select an additional $30 - |S^*|$ products from the same category to construct (S^*, V). This process allows us to create a data point (S^*, V). For comprehensive information, please refer to Table 5 in Ou et al. [2022], which presents the statistics of the categories.

Set Anomaly Detection. We tackle set anomaly detection tasks on four real-world datasets: double MNIST [Sun, 2019], CelebA [Liu et al., 2015b], F-MNIST [Xiao et al., 2017], and CIFAR-10 [Krizhevsky, 2009]. Each dataset is partitioned into training, validation, and test sets, each comprising

Table 5: The statistical properties of the Amazon product dataset.

Categories	$\lvert\mathcal{D}\rvert$	$\lvert V\rvert$	$\sum \lvert S^*\rvert$	$\mathbb{E}[\lvert S^*\rvert]$	$\min_{S^*} \lvert S^*\rvert$	$\max_{S^*} \lvert S^*\rvert$
Gear	4,277	30	16,288	3.80	3	10
Bath	3,195	30	12,147	3.80	3	11
Toys	2,421	30	9,924	4.09	3	14
Media	1,485	30	6,723	4.52	3	19
Safety	267	30	846	3.16	3	5
Diaper	6,108	30	25,333	4.14	3	15
Health	2,995	30	11,053	3.69	3	9
Carseats	483	30	1,576	3.26	3	6
Bedding	4,524	30	17,509	3.87	3	12
Feeding	8,202	30	37,901	4.62	3	23
Apparel	4,675	30	21,176	4.52	3	21
Furniture	280	30	892	3.18	3	6

10,000, 1,000, and 1,000 samples, respectively. In each dataset, we randomly select n images from the dataset to create the OS Oracle S^*, where n can be either 2, 3, or 4. This setup aligns with the approach outlined in [Zaheer et al., 2017, Ou et al., 2022].

Let's take CelebA as an illustrative example. In this scenario, the goal is to identify anomalous faces solely through visual observation, without using any attribute values. The CelebA dataset consists of 202,599 face images, each annotated with 40 boolean attributes. When constructing sets, for each ground set V, we randomly select n images from the dataset to create the OS Oracle S^*, ensuring that none of the selected images contain any of the two attributes. Additionally, we ensure that no individual person's face appears in both the training and test sets.

Regarding the results presented in Table 6, it is evident that our model exhibits a significant performance advantage over all the baseline methods. This substantial improvement underscores the superior capabilities of our model in addressing the given task.

Compound Selection in AI-aided Drug Discovery. In drug discovery, the screening of compounds with diverse biological activities and favorable ADME (absorption, distribution, metabolism, and excretion) properties is a critical step [Li et al., 2021, Ji et al., 2022, Gimeno et al., 2019]. Virtual screening typically involves a sequential filtering process that employs multiple essential filters. These filters initially select diverse subsets from highly active compounds and subsequently eliminate compounds with unfavorable ADME characteristics. After passing through several filtering stages, an optimal subset of compounds is identified. However, neural networks encounter challenges when learning the complete screening process. This difficulty arises from the absence of intermediate supervision signals, which can be costly or impossible to obtain due to pharmaceutical protection policies. Consequently, models are expected to learn this intricate selection process in an end-to-end manner. In other words, models must predict S^* based solely on the optimal subset supervision signals, without knowledge of the intermediate steps. Therefore, we simulate the optimal subset oracle of compound selection by applying one or two filters by using PDBBind and BindingDB, as [Ou et al., 2022].

PDBBind offers an extensive compilation of experimentally measured binding affinity data for biomolecular complexes. We utilized the ”refined” portion of the complete PDBBind dataset, which consists of 179 complexes, to construct our subsets. To create a data point (V, S^*), we randomly sampled 30 complexes from the dataset to form the ground set V. The subset S^* was then generated by selecting the five most active complexes within V. We constructed separate training, validation, and test splits, comprising 1000, 100, and 100 data points, respectively.

BindingDB is an openly accessible database that provides measured binding affinities for a collection of $52,273$ drug-target pairs involving small, drug-like molecules. Similar to PDBBind, we randomly selected 300 drug-target pairs from the BindingDB database to form the ground set V. From this dataset, we carefully identified the 15 most active drug-target pairs and designated them as S^*. To ensure comprehensive evaluation and robust model training, we subsequently created distinct training, validation, and test sets. These comprised 1000, 100, and 100 data points respectively.

Table 6: Empirical results of set anomaly detection Tasks. Bolded numbers denote the best performance. HORSE outperforms all the baselines on the four datasets.

	Random	PGM	DeepSet	Set-T	EquiVSet	INSET	HORSE
Double MNIST	8.2	30.0±1.0	11.1±0.3	51.2±0.5	57.5±1.8	69.7±1.0	**72.3 ± 1.2**
CelebA	2.2	48.1±0.6	44.0±0.6	52.7±0.8	54.9±0.5	57.5±1.2	**59.3 ± 1.0**
F-MNIST	1.9	54.0±2.0	49.0±2.0	58.1±1.0	65.0±1.0	70.1±2.1	**73.5 ± 1.6**
CIFAR-10	1.9	45.0±2.0	32.0±0.8	65.0±2.3	60.0±1.2	71.2±2.1	**74.3 ± 1.2**
Average	3.55	44.28	34.03	56.75	59.35	67.13	**69.85**

D.2 Descriptions of Baselines

Random. This represents the expected performance of a random guess, serving as a baseline to help us gauge the actual difficulty of the tasks.

PGM [Tschiatschek et al., 2018]. PGM, which stands for Probabilistic Greedy Model, tackles the optimization Problem 1 by employing a differentiable extension of the greedy maximization algorithm. For a deeper understanding of this approach, please refer to the origianl paper or Appendix A in [Ou et al., 2022].

DeepSet [Zaheer et al., 2017]. Here, we employ DeepSet as a baseline model. We use DeepSet to predict the probabilities of including specific instances in S^*, essentially learning an invariant permutation mapping from the power set 2^V to the interval $[0, 1]$ of size $|V|$. This model was also used as a backbone for set function learning in EquiVSet. Moreover, it is suitable for subset selection tasks, as detailed in its original paper.

Set Transformer [Lee et al., 2019a]. Set Transformer extends DeepSet's capabilities by integrating the self-attention mechanism. This addition allows the model to consider pairwise interactions between elements, enabling it to capture dependencies and relationships among different elements more effectively. It can be also utilized for subset selection tasks, similar to DeepSet.

EquiVSet [Ou et al., 2022]. EquiVSet employs an energy-based model (EBM) to establish the set mass function, denoted as $P(S|V)$ from a probabilistic standpoint. Their primary objective lies in learning a distribution $P(S|V)$ that monotonically increases with respect to the utility function $F(S, V)$. It's worth noting that their framework focuses on approximating the symmetric function $F(S)$ rather than the symmetric function $F(S, V)$, with DeepSet serving as the foundational component of their model to approximate the set function.

INSET [Xie et al., 2024]. As discussed in the Introduction, EquiVSet faces limitations in incorporating information from the ground set V. In response to this challenge, Xie et al. [2024] present an innovative solution. They propose the generation of embeddings for V and subsequently concatenate these embeddings with the representations of S, which has been presented in Figure 1.

Among these baselines, DeepSets and Set Transformer are two crucial model structures widely used in set-based tasks, including subset selection tasks. On the other hand, PGM, EquiVSet, and INSET are methods specifically designed for neural subset selection tasks. Notably, both INSET and HORSE are implemented based on the EquiVSet framework, yet they significantly outperform EquiVSet.

D.3 The Objective of Neural Subset Selection in Optimal Subset Oracle

Our method, HORSE, is applicable for learning $F(S, V)$ across a range of tasks. In this paper, we primarily utilize the framework established in [Ou et al., 2022] and modify it with our approach to model $F(S, V)$. The optimization objective aims to solve Equation 1 by employing an implicit learning strategy based on probabilistic reasoning. This approach can be formulated concisely as follows:

$$\begin{aligned}&\arg\max_{\theta}\ \mathbb{E}_{\mathbb{P}(V,S)}[\log p_\theta(S^*|V)]\\&\mathrm{s.t.}\ \ p_\theta(S|V) \propto F_\theta(S;V), \forall S \in 2^V,\end{aligned}$$

Table 7: In the table, we report the performance of different sample numbers denoted by ”k” and compare them against the best-performing baselines.

	Media	Safety
Best Baseline	62.0 ± 2.3	25.0 ± 0.6
k=2	63.1 ± 1.2	24.8 ± 1.3
k=4	64.4 ± 1.0	25.9 ± 1.2
k=6	65.8 ± 1.2	26.8 ± 0.9
k=8	**66.8 ± 1.3**	27.4 ± 1.0
k=10	66.2 ± 0.9	**27.7** ± **0.8**

Constructing a suitable set mass function $p_\theta(S|V)$ that exhibits monotonicity with respect to the utility function $F_\theta(S;V)$ is a crucial aspect of tackling this problem. To accomplish this, we can utilize the Energy-Based Model (EBM):

$$p_\theta(S|V) = \frac{\exp(F_\theta(S;V))}{Z},\ Z := \sum\nolimits_{S' \subseteq V} \exp(F_\theta(S';V)),$$

In practice, we approximate the Energy-Based Model (EBM) through a variational approximation. Due to the scope of this paper, we omit the detailed explanation for the sake of simplicity. We kindly invite readers to refer to [Ou et al., 2022] for further information on this topic.

D.4 Implementation Details

In this subsection, we present the implementation details of HORSE. The setup closely follows that of [Ou et al., 2022]. The proposed models are trained using the Adam optimizer [Kingma and Ba, 2014] with a fixed learning rate of $1e-4$ and a weight decay rate of $1e-5$. To accommodate different model sizes across various datasets, we select the batch size from the set $\{4, 8, 16, 32, 64, 128\}$. Importantly, we choose the largest batch size that allows efficient training on a single GeForce RTX 3090 GPU.

To enhance training efficiency and mitigate overfitting, we utilize an early stopping strategy for both the baselines and our proposed models. In this strategy, if there is no performance improvement over 10 consecutive epochs, we terminate the training process prematurely. For each dataset, the maximum number of epochs allowed for training is set to 80. At the end of each epoch, we assess the model’s performance on the validation set and save the model with the best performance. Finally, we evaluate the saved models on the test set to determine their performance.

In order to consider the effect of randomness and ensure the reliability of the findings, we conduct all experiments five times using different random seeds. The average performance metrics, along with their corresponding standard deviations, are reported as the final performance measures. This approach allows for a comprehensive evaluation of the models’ performance while accounting for the influence of random variations.

D.5 Ablation Study

To further investigate the robustness of INSET, we have conducted ablation studies specifically focusing on the Monte-Carlo sample numbers k for each input pair (S^*, V). In our framework, the model θ is trained to accurately predict the optimal subset S^* from a given ground set V. Following the Energy-Based Method (EBM) proposed in [Ou et al., 2022], we incorporate a necessary hyperparameter. During the training process, we sample k subsets from V in order to optimize the model parameters θ, thereby maximizing the conditional probability distribution $p_\theta(S^*|V)$ among all pairs of (S, V) for a given V.

To evaluate the robustness of HORSE across various values of k, we perform experiments on Media and Safety Categories of Product Recommendation. The results of these experiments are presented in Table 7, providing a comprehensive overview of the performance achieved with different k. Our findings clearly demonstrate that HORSE consistently outperforms all other baselines across the entire range of k values considered.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We summarize our contributions in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide a discussion of the limitation in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the proof in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the detailed algorithm in the Methods section. Additionally, our method is based on an open-source repository, enhancing its reproducibility.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provided part of our code at the time of submission. Furthermore, our method is based on the open-source repository detailed in Ou et al. [2022]. We have also comprehensively documented our experimental settings in the Experiment section and the Appendix. The integration of the open-source repository with our code ensures the reproducibility of our empirical results. We will also release our code once our paper becomes open access.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to our Experiment section and Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide the error bars and standard deviation in our experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to our Appendix, and we follow the settings of [Ou et al., 2022].

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The experiments in this paper are conducted with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The focus and experimental finding of this paper do not relevant to societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper does not have such a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All baselines used are properly cited. Sources of the datasets are also stated and all datasets are under CC-BY 4.0.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing experiments nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.