ADVANCING OPTIMAL SUBSET ORACLE VIA LEARNING RELAXATION OF NEURAL SET FUNCTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning neural set functions is pivotal to a wide range of important applications, including compound selection in AI-driven drug discovery and product recommendation. Recent work has introduced optimal subset oracles to implicitly learn set functions under practical weakly supervised settings, where model parameters are optimized through mean-field variational inference. However, these frameworks rely on Monte Carlo sampling to estimate gradients of the evidence lower bound when updating the variational distribution. Repeated sampling across iterations incurs substantial computational overhead, while the resulting stochasticity can destabilize the optimization trajectory. In this work, we reinterpret the evidence lower bound as a continuous relaxation of the set function and learn a surrogate objective that replaces sampling-based gradient estimation during variational optimization. The learned surrogate provides stable and efficient gradients throughout the continuous domain, thereby reducing computational overhead and accelerating inference. Furthermore, we establish an approximation guarantee for the proposed framework under submodular maximization and characterize its connection to variational free energy. Experiments on a variety of real-world tasks demonstrate consistent improvements over existing baselines.

1 INTRODUCTION

Set-value prediction has a wide range of applications in real-world scenarios and plays a crucial role in many tasks. For example, recommendation systems select products that are likely to interest a user (Coppolillo et al., 2024), anomaly detection identifies outliers from the majority of observations (Zhang et al., 2020), and AI-driven drug discovery prioritizes promising compounds from large candidate databases (Gimeno et al., 2019). These tasks require explicitly or implicitly learning a set function (Rezatofighi et al., 2017; Zaheer et al., 2017) that assigns a utility value to each candidate subset, with more desirable subsets receiving higher values.

More formally, our objective is to select an optimal subset S^* from a given large ground set V, such that it attains the highest utility value among all candidate subsets according to a set function $F_\theta(S; V)$ parameterized by θ. This process can be understood as optimizing the following criteria:

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

A direct approach is to learn $F_\theta(S; V)$ in a supervised manner from tuples $(V_i, S_i, U_i)_{i=1}^N$, where U_i denotes the utility of the candidate subset S_i, a setting commonly referred to as a function-value (FV) oracle (Balcan & Harvey, 2018). However, this training paradigm is often prohibitively expensive, as it requires collecting utility annotations for a sufficiently large and diverse set of candidate subsets (Ou et al., 2022). To overcome this limitation, an alternative approach to optimizing objective (1) is to implicitly learn the set function from a probabilistic perspective (Tschiatschek et al., 2018). This approach estimates the parameter θ in a supervised manner using pairs $\{(V_i, S_i^*)\}_{i=1}^N$, where S_i^* denotes the optimal subset corresponding to V_i, serving as an optimal subset (OS) oracle. With limited data sampled from the underlying distribution $\mathbb{P}(S, V)$, the OS oracle learns latent patterns by maximizing the empirical log-likelihood $\sum_{i=1}^N \log p_\theta(S_i^* | V_i)$ over the observed data. Compared with the FV oracle, the OS oracle is generally more practical, as it avoids explicit utility labeling for candidate subsets, thereby reducing annotation costs in practice.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

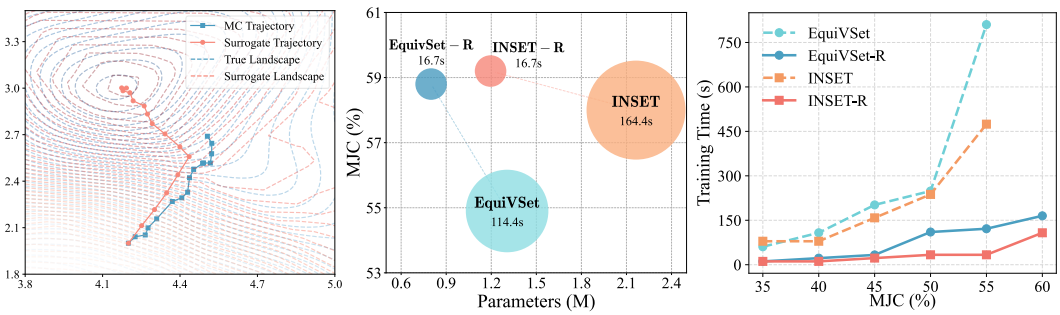


Figure 1: **(left)** Optimization trajectories over the same number of steps on a black-box objective: Monte Carlo sampling (blue) introduces stochastic noise, while surrogate-based optimization (red) converges faster and more smoothly. **(middle)** Comparison of different baselines on the CelebA dataset in terms of performance, per-iteration training time, and model size. **(right)** Time required by different baselines to reach the same validation performance on the CelebA dataset. Here, "$-$ R" denotes variants that replace Monte Carlo sampling with a learned set function relaxation.

Although the OS oracle is conceptually appealing, the distribution $p(S|V)$ is generally intractable in practice. Consequently, most existing OS oracles adopt a variational inference framework (Blei et al., 2017), in which a variational distribution $q_{\boldsymbol{\psi}}(S|V)$ is fitted to approximate the target distribution. Here, $\boldsymbol{\psi} \in [0,1]^{|V|}$ parameterizes $|V|$ independent Bernoulli variables, with ψ_i representing the probability that element i is included in the predicted subset S. Within this framework, optimizing the variational distribution requires maximizing the *evidence lower bound* (ELBO), which is equivalent to minimizing the divergence between the two distributions. Unfortunately, exact evaluation of the ELBO gradient is computationally prohibitive, as it involves an exponential summation over all possible subsets S. When the ground set V becomes moderately large (e.g., $|V| \geq 30$), enumerating the entire $2^{|V|}$-subset space becomes intractable.

Existing approaches attempt to address this challenge through the Monte Carlo (MC) sampling-based approximation (Xie et al., 2024a;b; Özcan et al., 2025); however, this strategy comes with limitations. Accompanied by the unrolled multi-step gradient ascent, extensive sampling incurs substantial computational overhead. Meanwhile, the induced stochasticity destabilizes the optimization trajectory, making the model prone to suboptimal solutions.

In practice, the ELBO can be viewed as a relaxation of the underlying set function. Motivated by this, our core idea is to learn a differentiable surrogate objective that replaces sampling-based ELBO gradient estimation during variational optimization. In this paper, we propose to learn a **Re**laxation of **Set** functions (ReSet) within optimal subset oracles, thereby replacing MC–based approximations with a flexible and learnable paradigm. By leveraging the learned relaxation, ReSet substantially reduces optimization overhead and enables faster updates of the variational distribution. We provide illustrative comparisons in Figure 1 to demonstrate this. This optimization process can be interpreted as a greedy strategy for continuous submodular maximization, and we derive an approximation ratio that provides a theoretical lower-bound guarantee. Furthermore, we explore the intrinsic relationship between the ELBO and variational free energy, revealing that our method offers advantages beyond merely optimizing a fixed objective by adaptively learning the trade-off between expected energy and entropy. Our main contributions can be summarized as follows:

- We propose to learn a set function relaxation that replaces sampling-based ELBO gradient estimation, enabling efficient and stable learning of neural set functions.
- We guarantee that the proposed method converges to a stationary point and derive a constant approximation ratio for the solution, ensuring theoretical soundness.
- We revisit the variational inference process through the lens of free energy, demonstrating that our method is able to learn an adaptive trade-off between expected energy and entropy.
- Experimental results indicate that the proposed method consistently outperforms baseline methods across diverse real-world tasks.

2 PRELIMINARIES

We begin by introducing the optimization framework for optimal subset oracles. From a probabilistic perspective, neural set function learning under the OS oracle can be formulated as a maximum likelihood problem (Stigler, 1990):

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

where $p_\theta(S|V)$ denotes the target set distribution we aim to identify. By treating the utility function $F_\theta(S;V)$ as the negative energy, the set distribution can be formulated via an energy-based model:

$$p_\theta(S|V) = \frac{\exp(F_\theta(S;V))}{Z}, \qquad Z = \sum_{S' \subseteq V} \exp(F_\theta(S';V)). \tag{3}$$

Such formulation follows the principle of maximum entropy (Guiasu & Shenitzer, 1985), thereby embodying the least prior assumptions. This endows the model with the ability to maintain the least possible bias toward unknown information, consistent with the notion of *non-informative priors* in Bayesian modeling (Jeffreys, 1946). Although this approach is conceptually attractive, learning the parameters θ is notoriously difficult. Following Ou et al. (2022), the training of energy-based model is cast in a variational inference framework, where a variational distribution $q(\boldsymbol{\psi})$ is introduced as an approximation to the set mass function $p_\theta(S|V)$, the fitting target can be expressed as[1]

$$\boldsymbol{\psi}^* = \arg\min_{\boldsymbol{\psi}} D[q(\boldsymbol{\psi})||p_\theta(S|V)], \tag{4}$$

where $D(\cdot||\cdot)$ is a discrepancy measure between two distributions, $\boldsymbol{\psi} \in [0,1]^{|V|}$ represents the probability that each element $s \in V$ is included in the optimal subset S^*. By achieving this objective, the variational distribution $q(\boldsymbol{\psi})$ can be regarded as a function of θ, which allows the parameter θ to be effectively learned in a straightforward manner through the cross-entropy loss:

$$\begin{aligned} \mathcal{L}(\theta;\boldsymbol{\psi}^*) &= \mathbb{E}_{\mathbb{P}(V,S)}[-\log q(\boldsymbol{\psi}^*)] \\ &\approx -\sum_{i \in S^*} \log \psi_i^* - \sum_{i \in V \setminus S^*} \log(1-\psi_i^*). \end{aligned} \tag{5}$$

In this way, the entire framework can be trained in a cooperative learning fashion (Xie et al., 2018): with θ fixed, the variational distribution $q(\boldsymbol{\psi})$ is optimized to approximate the energy model $p_\theta(S|V)$ via (4); then, θ is updated through gradient flow on the cross-entropy loss in (5).

3 METHOD

Although Section 2 outlines a feasible framework for learning set functions under energy-based modeling, the resulting optimization remains nontrivial. In this section, we show how this challenge can be effectively addressed through a straightforward end-to-end learnable paradigm.

3.1 PROBLEM SETUP

In order to optimize objective (4), we need to specify both the divergence measure $D(\cdot||\cdot)$ and the form of the variational distribution $q(\boldsymbol{\psi})$, while ensuring that $\boldsymbol{\psi}^*$ remains differentiable with respect to θ. A common choice is to define the variational distribution $q(\boldsymbol{\psi})$ as a product of $|V|$ independent Bernoulli variables, *i.e.*, $q(\boldsymbol{\psi}) = \prod_{i \in S} \psi_i \prod_{i \in V \setminus S}(1-\psi_i)$, which implies that $q(\boldsymbol{\psi})$ is fully factorized and can be regarded as a mean-field approximation of $p_\theta(S|V)$. By further choosing the Kullback-Leibler divergence as the measure, the objective (4) can be reformulated as maximizing the evidence lower bound (ELBO):

$$\min_{\psi} \mathbb{KL}(q(\boldsymbol{\psi})||p_\theta(S|V)) \iff \max_{\boldsymbol{\psi}} \underbrace{f_{\text{mt}}^{F_\theta}(\boldsymbol{\psi}) + \mathbb{H}(q(\boldsymbol{\psi}))}_{\text{ELBO}}, \tag{6}$$

[1] We use the shorthand $q(\boldsymbol{\psi})$ to denote $q_{\boldsymbol{\psi}}(S|V)$ here.

where $f_{\text{mt}}^{F_\theta}(\boldsymbol{\psi})$ denotes the multilinear extension of $F_\theta(S)$ (Calinescu et al., 2007), defined as

$$f_{\text{mt}}^{F_\theta}(\boldsymbol{\psi}) := \sum_{S \subseteq V} F_\theta(S) \prod_{i \in S} \psi_i \prod_{i \notin S} (1 - \psi_i), \tag{7}$$

and $\mathbb{H}(\cdot)$ is the entropy of a given distribution. To maximize the ELBO in (6), a straightforward approach is to apply fixed-point iteration, where the stationary condition is $\psi_i = \sigma(\nabla_{\psi_i} f_{\text{mt}}^{F_\theta}(\boldsymbol{\psi}))$[2], here $\sigma(\cdot)$ denotes the sigmoid function. In this way, $\boldsymbol{\psi}$ can be updated via the following procedure:

$$\boldsymbol{\psi}^{(0)} \leftarrow \text{Initialize in } [0,1]^{|V|}, \tag{8}$$

$$\boldsymbol{\psi}^{(k)} \leftarrow (1 + \exp(-\nabla_{\boldsymbol{\psi}^{(k-1)}} f_{\text{mt}}^{F_\theta}(\boldsymbol{\psi}^{(k-1)})))^{-1}, \tag{9}$$

$$\boldsymbol{\psi}^{*} \leftarrow \boldsymbol{\psi}^{(K)}, \tag{10}$$

which was referred to as differentiable mean-field variational inference by Ou et al. (2022).

The crux of the matter is that computing the derivative of the multilinear extension $\nabla f_{\text{mt}}^{F_\theta}(\boldsymbol{\psi})$ entails summing over all possible subsets, which is computationally intractable. Existing methods (Xie et al., 2024a;b; Özcan et al., 2025) approximate using Monte Carlo (MC) sampling[3]:

$$\nabla_{\psi_i} f_{\text{mt}}^{F_\theta}(\boldsymbol{\psi}) \approx \frac{1}{M} \sum_{m=1}^{M} \left[F_\theta(S^{(m)} + i) - F_\theta(S^{(m)}) \right], \tag{11}$$

where $S^{(m)}$ is the m-th independent sample from $q(S; (\boldsymbol{\psi} | \psi_i \leftarrow 0))$, and $S^{(m)} + i$ denotes the union of the set $S^{(m)} \cup i$. However, the reliance on extensive sampling incurs substantial computational overhead, and the induced stochasticity can further destabilize the optimization trajectory.

3.2 Learning Adaptive Set Function Relaxation

During variational optimization, the entropy term depends only on $\boldsymbol{\psi}$, and its gradient can be evaluated in closed form. For a fixed $\boldsymbol{\psi}$, the multilinear extension $f_{\text{mt}}^{F_\theta}(\boldsymbol{\psi})$ depends only on θ, while evaluating its gradient remains the computational bottleneck. Motivated by this observation, our core idea is to learn a differentiable surrogate objective $g_\theta : [0,1]^{|V|} \to \mathbb{R}_+$ that plays the role of the multilinear extension, replacing sampling-based ELBO gradient estimation during variational optimization.[4] The surrogate enables stable gradient computation throughout the continuous domain without requiring repeated MC estimation at each optimization step. Notably, constructing this surrogate function is easy to implement in practice. According to Karalias et al. (2022), the extensions of set function, including $g_\theta(\cdot)$ here, can be modeled using simple neural networks. Consequently, the variational parameter $\boldsymbol{\psi}$ can be effectively optimized through gradient ascent

$$\boldsymbol{\psi}^{(k+1)} = \boldsymbol{\psi}^{(k)} + \alpha \left(\nabla g_\theta(\boldsymbol{\psi}^{(k)}) + \log \frac{\mathbf{1} - \boldsymbol{\psi}^{(k)}}{\boldsymbol{\psi}^{(k)}} \right), \tag{12}$$

where α is the step size. While the above equation illustrates a basic setting, in implementation, the continuous optimization can be carried out using different methods, such as projected gradient ascent or continuous greedy (Calinescu et al., 2007). Subsequently, the parameters θ can be updated through the marginal-based loss (Domke, 2013) defined in Equation (5). Note that, in this case, the loss is propagated through the entire optimization process, which involves computing second-order derivatives (i.e., gradients of gradients). These derivatives can be efficiently evaluated using Hessian–vector products, which scale linearly with the model size (Dagréou et al., 2024), similar to standard first-order backpropagation in feed-forward architectures. A brief description of our method is given in Algorithm 1, while a more comprehensive procedure is detailed in Appendix D.2.

In fact, maximizing the surrogate leads to the following continuous optimization problem:

$$\max_{\boldsymbol{\psi}} g_\theta(\boldsymbol{\psi}), \quad \text{s.t.} \quad \boldsymbol{\psi} \in \mathcal{X}, \tag{13}$$

[2]The detailed derivation of this stationary condition is provided in Appendix B.1.

[3]Further details on gradient approximation via Monte Carlo sampling are provided in Appendix B.2.

[4]Note that we do not introduce an additional neural network here. The original OS framework already uses a network to approximate F_θ for MC-based gradient estimation, while our method directly learns the relaxation.

Algorithm 1 Variational Optimization with ReSet

1: Initialize the variational parameter via the auxiliary recognition network
$\boldsymbol{\psi}^{(0)} \leftarrow \mathrm{RecNet}_{\phi}(V)$
2: **for** $k = 0$ to $K - 1$ **do**
3: Compute the surrogate $g_\theta(\boldsymbol{\psi}^{(k)})$ with Eq. (24)
4: Perform gradient ascent on $\boldsymbol{\psi}$ using Eq. (22)
$\boldsymbol{\psi}^{(k+1)} \leftarrow \boldsymbol{\psi}^{(k)} + \alpha(\nabla g_\theta(\boldsymbol{\psi}^{(k)}) + \log \frac{\mathbf{1}-\boldsymbol{\psi}^{(k)}}{\boldsymbol{\psi}^{(k)}}) + \boldsymbol{\epsilon}_k$
5: **end for**
6: Set $\boldsymbol{\psi}^* \leftarrow \boldsymbol{\psi}^{(K)}$

where the constraint set $\mathcal{X}$ is a box constraint, forming a standard compact convex set

$$\mathcal{X} = \left\{ \boldsymbol{\psi} \in \mathbb{R}^{|V|} \mid 0 \le \psi_i \le 1, \forall i \in \{1, \dots, |V|\} \right\}. \tag{14}$$

The gradient $\nabla g_\theta(\cdot)$ of the differentiable surrogate objective provides continuous surrogate marginal-gain signals, indicating which elements (probability components) should be increased to improve the objective. The entropy term in Eq. (12) balances these signals with uncertainty in the variational distribution. Since enforcing exact submodularity is difficult with neural parameterizations (Dolhansky & Bilmes, 2016), we instead consider weak DR-submodularity, under which the learned relaxation admits the following guarantees (Pedramfar et al., 2024; Hassani et al., 2017).

Theorem 3.1. *Suppose the relaxation surrogate $g_\theta : [0,1]^{|V|} \to \mathbb{R}_+$ is L-smooth[5] and γ-weakly DR-submodular [6]. Then the following results hold:*

1) For a solution $\boldsymbol{\psi}^* \in [0,1]^{|V|}$ of continuous greedy, we have

$$g_\theta(\boldsymbol{\psi}^*) \ge \gamma e^{-\gamma} g_\theta(\boldsymbol{\psi}_{\mathrm{OPT}}), \tag{15}$$

where $g_\theta(\boldsymbol{\psi}_{\mathrm{OPT}}) = \max_{\boldsymbol{\psi}} g_\theta(\boldsymbol{\psi})$ is the global optimum.

2) Projected gradient ascent with a fixed step size $0 < \alpha < \frac{1}{L}$ converges to first-order stationarity.

The above theorem provides theoretical support for our learning framework. It establishes convergence to first-order stationarity and shows that the relaxed objective can be optimized with a constant-factor approximation guarantee of $\gamma e^{-\gamma}$. In practice, we can refine the continuous-greedy solution by projected gradient ascent, preserving this approximation guarantee while converging to first-order stationarity. With entropy regularization, the utility bound includes an entropy-dependent correction, as detailed in Appendix A.4. As the final solution $\boldsymbol{\psi}^*$ is typically fractional, we can obtain a discrete solution by sampling $S^* \sim q(\boldsymbol{\psi}^*)$. Define the induced set function $F_\theta(S) := g_\theta(\mathbf{1}_S)$, where $\mathbf{1}_S$ is the indicator vector of S. The definition of the multilinear extension gives

$$\mathbb{E}_{S^* \sim q(\boldsymbol{\psi}^*)}[F_\theta(S^*)] = f^{F_\theta}_{\mathrm{mt}}(\boldsymbol{\psi}^*), \tag{16}$$

and when directional consistency between g_θ and the multilinear objective holds along the continuous-greedy trajectory, the resulting discrete solution satisfies[7]

$$\mathbb{E}_{S^* \sim q(\boldsymbol{\psi}^*)}[F_\theta(S^*)] \ge \gamma e^{-\gamma} F_\theta(S_{\mathrm{OPT}}), \tag{17}$$

where $S_{\mathrm{OPT}} \in \arg\max_{S \subseteq V} F_\theta(S)$.

3.3 Connection to Variational Free Energy

The energy function associated with Eq. (3) evaluates the plausibility of a specific state, reflecting a localized perspective. In contrast, the Helmholtz free energy provides a global measure:

$$\mathcal{F}_{\mathrm{H}} = -k_B T \log Z_T, \tag{18}$$

[5] L-smoothness is a standard regularity assumption widely adopted in first-order optimization and continuous DR-submodular maximization.

[6] This assumption holds in a broad range of practical settings, as $\gamma \in (0, 1]$ provides a continuous measure of the degree of submodularity, with $\gamma = 1$ corresponding to exact DR-submodularity.

[7] That is, the directions induced by g_θ also maximize the linearized multilinear objective; see Appendix A.3 for the derivation.

where k_B is the Boltzmann constant, $T > 0$ denotes the absolute temperature, and Z_T is the partition function, with $E_\theta(S; V) = -F_\theta(S; V)$. It aggregates all possible subsets of V. At unit temperature, $k_B T = 1$, Z_T coincides with the partition function Z in Eq. (3). Within the variational framework, for a given ground set V and a fixed temperature T, the corresponding variational free energy is defined as

$$\mathcal{F}_\theta(\boldsymbol{\psi}) = \mathbb{E}_{S\sim q(\boldsymbol{\psi})}\left[E_\theta(S; V)\right] + k_B T \sum_{i\in V}\left[\psi_i \log \psi_i + (1-\psi_i)\log(1-\psi_i)\right]. \tag{19}$$

Here, the first term denotes the expected energy under the mean-field variational distribution, while the second term is its negative entropy weighted by $k_B T$. The temperature therefore controls the trade-off between expected energy and entropy.

Proposition 3.2 (Variational Free Energy)**.** *Consider an energy function $E_\theta(S; V)$ and a mean-field variational distribution $q(\boldsymbol{\psi})$. The associated variational free energy admits a decomposition into expected energy and entropy:*

$$\mathcal{F}_\theta(\boldsymbol{\psi}) = \langle E\rangle_q - k_B T \mathcal{H}_q, \tag{20}$$

where $\langle E\rangle_q = \mathbb{E}_{S\sim q(\boldsymbol{\psi})}[E_\theta(S; V)]$ denotes the expected energy, and $\mathcal{H}_q = \mathbb{H}(q(\boldsymbol{\psi}))$ denotes the entropy of the variational distribution.

Notably, the set utility function naturally plays the role of negative energy here, *i.e.*, $F_\theta(\cdot) = -E_\theta(\cdot)$, which implies the expected energy term in (19) can be interpreted as the negative counterpart of the multilinear extension $f_{\text{mt}}^{F_\theta}$ in (6). Consequently, by setting $k_B T = 1$, the variational free energy becomes equivalent to the negative ELBO.

Corollary 3.3. *When the set utility is interpreted as negative energy, i.e., $E_\theta = -F_\theta$, and $k_B T = 1$, the variational free energy equals the negative evidence lower bound, $\mathcal{F}_\theta(\boldsymbol{\psi}) = -\text{ELBO}$.*

Therefore, maximizing the ELBO in Eq. (6) is equivalent to minimizing the variational free energy:

$$\arg\max_{\boldsymbol{\psi}\in\mathcal{X}} \text{ELBO} = \arg\min_{\boldsymbol{\psi}\in\mathcal{X}} \left[f_{\text{mt}}^{E_\theta}(\boldsymbol{\psi}) - \mathbb{H}(q(\boldsymbol{\psi}))\right] = \arg\min_{\boldsymbol{\psi}\in\mathcal{X}} \mathcal{F}_\theta(\boldsymbol{\psi}). \tag{21}$$

Conceptually, this equivalence indicates that variational inference can be interpreted as an energy minimization process that balances fitting the observed data with maintaining uncertainty. Minimizing $\mathcal{F}_\theta(\boldsymbol{\psi})$ favors low-energy configurations, while the entropy term discourages excessive concentration of the variational distribution. More importantly, this formulation provides flexibility beyond the objective in (6). In contrast to optimizing the ELBO under a fixed balancing relation (temperature), we can directly treat T in (19) as a learnable parameter, thereby allowing the model to identify an adaptive trade-off between expected energy and entropy.

4 Network Architecture

Although we have established theoretical guarantees above, realizing a smooth and well-behaved energy landscape remains notoriously difficult in practice (Gladstone et al., 2025), as real-world problems often involve high-dimensional objective surfaces. To address this, we leverage Langevin Dynamics (Du & Mordatch, 2019) by injecting a stochastic noise term, thereby promoting exploration of the energy landscape:

$$\boldsymbol{\psi}^{(k+1)} = \boldsymbol{\psi}^{(k)} + \alpha\left(\nabla g_\theta(\boldsymbol{\psi}^{(k)}) + \log\frac{\mathbf{1}-\boldsymbol{\psi}^{(k)}}{\boldsymbol{\psi}^{(k)}}\right) + \boldsymbol{\epsilon}_k, \qquad \boldsymbol{\epsilon}_k \sim \mathcal{N}(0, \tau), \tag{22}$$

where τ is the magnitude of the noise $\boldsymbol{\epsilon}$. Besides this, to ensure that the optimization trajectory of the predictive distribution can adapt to diverse problem settings, we make the gradient ascent step size α learnable, which improves generalization. Taken together, these improvements equip the model with a stronger capacity to navigate complex and rugged objective landscapes in real-world tasks. In addition, to further accelerate inference and improve scalability, we introduce an auxiliary recognition network[8] that amortizes inference by providing a reasonable initialization of the variational distribution $q(\boldsymbol{\psi})$ for a given set V before optimizing the objective in (6):

$$\boldsymbol{\psi}^{(0)} = \text{RecNet}_\phi(V). \tag{23}$$

[8]Following the setup of Ou et al. (2022); see Appendix B.3 for details.

Furthermore, modeling the surrogate g_θ requires permutation invariance, which cannot be achieved by classic feed-forward networks. We adopt the DeepSets architecture (Zaheer et al., 2017), which overcomes this limitation through the following proposition.

Proposition 4.1. *All permutation invariant set functions can be decomposed in the form* $f(S) = \rho\Big(\sum_{s_i \in S} \kappa(s_i)\Big)$*, for suitable transformations* κ *and* ρ*.*

Following the DeepSets framework, we further construct permutation-invariant sufficient representations[9] for both the superset V and the subset S by employing the soft assignment $\boldsymbol{\psi}$. These representations can be approximated by empirical estimations $R(s_i) = \sum_{s_i \in S} \delta(s_i)$ Bloem-Reddy & Teh (2020), where $\delta(s_i)$ represents an atom of unit mass located at s_i, such as one-hot embeddings. Hence, the neural network construction can be outlined as follows:

$$g_\theta(\boldsymbol{\psi}) = \sigma\left(\theta_1 \sum_{i=1}^{|V|} \zeta_1(\psi_i s_i) + \theta_2 \sum_{i=1}^{|V|} \zeta_2((1-\psi_i)s_i)\right). \tag{24}$$

Here, the feed-forward modules θ_1 and θ_2 are accompanied by a non-linear activation layer denoted by $\sigma(\cdot)$. Due to the limitations of the mean-field assumption, the variational distribution $q(\boldsymbol{\psi})$ is insufficient for capturing interactions among elements within the input set. To remedy this, we design the architecture of function ζ as the multi-head attention (Vaswani et al., 2017), enabling the capture of interactions among elements. The overall architecture of the proposed network is illustrated in Appendix D.1. Leveraging this network architecture, we show in the following section that our proposed framework empirically outperforms the baselines.

5 Empirical Studies

We evaluate the proposed method on a diverse range of tasks, including product recommendation, set anomaly detection, compound selection, and synthetic experiments. All experiments are conducted with five different random seeds, and we report the corresponding mean performance along with standard deviation. Detailed descriptions of the model architectures and training configurations are provided in Appendix E. Additional evaluations including ablation studies, hyperparameter sensitivity analyses, and experiments on varying ground set sizes are provided in Appendix F.

Evaluation Metric. The goal of optimal subset selection is to identify an optimal subset S^* from a given ground set V. In experiments, we adopt the mean Jaccard coefficient (MJC) as the evaluation metric to assess the performance of different methods. For each ground set, the Jaccard coefficient is defined as $\mathrm{JC}(S', S^*) = \frac{|S' \cap S^*|}{|S' \cup S^*|}$, which evaluates the similarity between the predicted subset S' and the ground-truth optimal subset S^*. Subsequently, the MJC is obtained by averaging the Jaccard coefficients across all test instances.

Baselines. Our method is designed as a plug-in module that can be integrated into existing optimal subset oracle frameworks. The resulting variants are denoted by the suffix "-R". We evaluate these variants against four standard baselines: Random, Probabilistic Greedy Model (PGM) (Tschiatschek et al., 2018), DeepSet (Zaheer et al., 2017) and Set Transformer (Lee et al., 2019); as well as two optimal subset oracles: EquiVSet (Ou et al., 2022) and INSET (Xie et al., 2024a). Detailed descriptions of these baselines are provided in Appendix E.3.

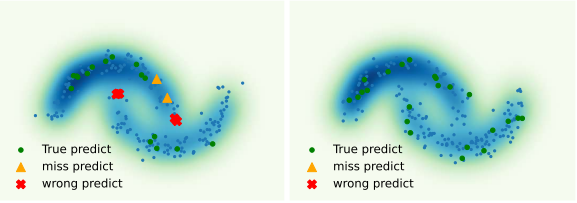
Synthetic Experiments. We first evaluate the methods on two synthetic datasets: the two-moons dataset (Pedregosa et al., 2011) with additional noise of variance $\sigma^2 = 0.1$, and the mixture of Gaussians represented as $\frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}) + \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}_0 = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^\top, \boldsymbol{\mu}_1 = -\boldsymbol{\mu}_0, \boldsymbol{\Sigma} = \frac{1}{4}\mathbf{I}$. Taking the Gaussian Mixture dataset as an example, the data are generated according to the following procedure: i) Initially, we sample an index $b \sim \mathrm{Bernoulli}(\frac{1}{2})$; ii) we then sample 10 points from the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_b, \boldsymbol{\Sigma})$ to construct the ground-truth subset S^*; iii) Finally, we sample additional 90 points from $\mathcal{N}(\boldsymbol{\mu}_{1-b}, \boldsymbol{\Sigma})$ to construct $V \backslash S^*$. For both datasets, we generate 1,000 samples for training, validation, and test, respectively.

From the quantitative results in Table 1, it can be observed that the proposed method achieves a significant improvement of nearly 70% over the baseline on the more challenging Two Moons dataset.

[9] Following Xie et al. (2024a), we incorporate background information from V into the set function; see Appendix C for details.

Table 1: Results on the synthetic datasets.

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	58.7 ± 0.2	90.9 ± 0.2
EquiVSet-R	99.2 ± 0.3	**91.0 ± 0.1**
INSET	59.0 ± 0.3	90.9 ± 0.2
INSET-R	**99.7 ± 0.1**	90.9 ± 0.1



(a) EquiVSet (b) EquiVSet-R

Figure 2: Visualization on Two-Moons.

In addition, we provide visualization results of the model predictions after training for 50 epochs in Figure 2. The green dots represent correct model predictions, the red crosses are incorrect model predictions, and the yellow triangles represent data points in subset oracle S^* that are missed by the model. One can see that EquiVSet tends to misclassify more points at the intersection, while also missing some in-distribution points. In comparison, EquiVSet-R learns a more cohesive and clearer distribution boundary, while correctly identifying all the difficult samples in the overlapping region.

Table 2: Experimental results (%) on product recommendation across categories in the Amazon dataset, where "Set-T" denotes Set Transformer.

Categories	Random	PGM	DeepSet	Set-T	EquiVSet	EquiVSet-R	INSET	INSET-R
Toys	8.3	44.1 ± 0.4	42.9 ± 0.5	62.6 ± 2.0	68.0 ± 2.0	**77.9 ± 1.0**	76.9 ± 0.5	**79.0 ± 0.6**
Furniture	6.5	17.5 ± 0.7	17.6 ± 0.7	17.6 ± 0.8	17.2 ± 0.9	**35.9 ± 1.3**	16.9 ± 5.0	**36.5 ± 1.5**
Gear	7.7	47.1 ± 0.4	38.1 ± 0.2	64.7 ± 0.6	70.0 ± 0.2	**78.7 ± 2.6**	80.8 ± 1.2	**82.1 ± 1.1**
Carseats	6.6	23.0 ± 1.0	21.0 ± 1.0	22.0 ± 1.0	21.0 ± 1.0	**26.9 ± 1.0**	23.1 ± 3.4	**30.2 ± 3.9**
Bath	7.6	56.4 ± 0.8	42.4 ± 0.6	71.6 ± 0.5	75.7 ± 0.9	**84.7 ± 0.9**	86.2 ± 0.5	**86.7 ± 0.6**
Health	7.6	44.9 ± 0.2	44.8 ± 0.4	69.0 ± 1.0	70.0 ± 2.0	**81.4 ± 0.6**	81.2 ± 0.5	**82.2 ± 0.2**
Diaper	8.4	58.0 ± 0.9	45.7 ± 0.5	78.9 ± 0.5	83.0 ± 1.0	**90.4 ± 0.6**	88.0 ± 0.7	**90.4 ± 0.5**
Bedding	7.9	48.0 ± 0.6	48.2 ± 0.8	76.0 ± 2.0	77.0 ± 1.0	**86.8 ± 1.2**	85.7 ± 1.0	**87.6 ± 1.7**
Safety	6.5	25.0 ± 0.6	22.1 ± 0.4	23.4 ± 0.9	25.0 ± 3.0	**32.1 ± 9.4**	23.8 ± 1.5	**38.3 ± 4.0**
Feeding	9.3	56.0 ± 0.8	43.0 ± 0.2	75.3 ± 0.6	81.0 ± 0.7	**87.2 ± 1.4**	88.2 ± 1.0	**88.6 ± 0.5**
Apparel	9.0	53.3 ± 0.5	50.7 ± 0.4	68.0 ± 2.0	75.0 ± 1.0	**82.2 ± 0.9**	**83.7 ± 0.3**	83.3 ± 1.3
Media	9.4	44.1 ± 0.9	42.0 ± 1.0	53.0 ± 2.0	57.0 ± 1.0	**67.0 ± 1.0**	62.0 ± 2.3	**68.1 ± 1.1**

Product Recommendation. We use the Amazon baby registry dataset (Gillenwater et al., 2014) as a benchmark for the product recommendation task, which contains multiple categories of sub-products selected by different customers. Each product in the dataset is registered under a specific category, such as "Diaper" or "Feeding", and is represented by a brief textual description. In practice, each product is represented by a 768-dimensional embedding obtained from a pre-trained BERT model (Devlin et al., 2019). Following Ou et al. (2022), we partition the data into training, validation, and test sets using a 1:1:1 ratio. The objective is to identify the subset of products that a customer finds most relevant within a candidate set of size 30. The performance of all methods across different categories is reported in Table 2. Overall, our approach outperforms the two optimal subset oracle baselines on most evaluated tasks. Notably, the performance improvements are more evident on challenging categories, such as Furniture, Carseats, Safety, and Media.

Set Anomaly Detection. Experiments on set anomaly detection are conducted using four widely adopted benchmarks: Double MNIST (Sun, 2019), F-MNIST (Xiao et al., 2017), CelebA (Liu et al., 2015b), and CIFAR-10 (Krizhevsky, 2009).[10] According to the protocol of Zaheer et al. (2017); Ou et al. (2022), we split each dataset into 10,000 training, 1,000 validation, and 1,000 test set instances. Taking the Double MNIST dataset as an example, we form the OS oracle S^* by sampling $n \in \{2, 3, 4\}$ images of the same class, with the complement $V \backslash S^*$ comprising $20 - |S^*|$ images sampled from non-target classes. From Table 3, it is evident that our proposed variants consistently outperform the baseline methods with notable improvements across all datasets.[11] For instance, on the Double MNIST dataset, our proposed variant EquiVSet-R achieves a relative improvement of approximately 83.9% compared to the original framework.

[10] Due to the absence of public code and the relative ambiguity in the protocol descriptions of Ou et al. (2022), we provide a reconstructed and well-defined experimental standard for Double MNIST, F-MNIST, and CIFAR-10; further discussions are provided in Appendix E.5.

[11] Results for PGM are unavailable as the official implementation has not been released.

Table 3: Experimental results (%) on anomaly detection and compound selection tasks.

	Anomaly Detection				Compound Selection	
	Double MNIST	**CelebA**	**F-MNIST**	**CIFAR10**	**PDBBind**	**BindingDB**
Random	9.3	21.9	20.9	20.1	9.9	0.9
PGM	-	48.1 ± 0.6	-	-	91.0 ± 1.0	69.0 ± 2.0
DeepSet	18.5 ± 1.0	44.0 ± 0.5	59.7 ± 1.2	29.0 ± 0.3	90.1 ± 1.1	71.0 ± 2.0
Set Transformer	22.0 ± 2.6	52.7 ± 0.8	69.8 ± 2.9	27.0 ± 1.7	91.9 ± 1.5	71.5 ± 1.0
EquiVSet	38.6 ± 2.5	54.9 ± 0.5	58.4 ± 1.2	30.4 ± 0.7	92.4 ± 1.1	72.1 ± 0.9
EquiVSet-R	**71.0 ± 1.6**	**58.8 ± 0.6**	**72.6 ± 2.1**	**32.4 ± 2.2**	**93.0 ± 2.6**	**78.3 ± 0.8**
INSET	46.2 ± 1.1	58.0 ± 1.2	66.0 ± 2.8	32.0 ± 0.9	93.5 ± 0.8	73.4 ± 1.0
INSET-R	**73.2 ± 1.8**	**59.2 ± 0.2**	**75.9 ± 1.4**	**35.8 ± 1.3**	**96.0 ± 0.6**	**77.8 ± 1.2**

Compound Selection in AI-aided Drug Discovery. In the drug discovery process, screening compounds with high bioactivity (Wallach et al., 2015; Li et al., 2021; Ji et al., 2023), structural diversity, and favorable ADME properties (Gimeno et al., 2019) is a critical step. Virtual screening pipelines typically proceed through multiple sequential filtering stages. For example, researchers first select compounds with high bioactivity, then extract diverse subsets, and finally remove candidates with unfavorable ADME properties. After these steps, a final subset of compounds is obtained. Due to privacy constraints in pharmaceutical practice, intermediate supervision signals are often inaccessible or costly to obtain. Consequently, models are expected to learn the composite screening process end-to-end, which aligns well with the OS oracle. Nevertheless, full end-to-end prediction is often more complex and requires additional domain expertise. Here, we focus on the high-bioactivity selection stage, following the experimental setting of Xie et al. (2024a). In the experiments, we evaluate the methods on PDBBind (Liu et al., 2015a) and BindingDB (Liu et al., 2007) datasets using the same high-bioactivity filter. As shown in Table 3, our method consistently improves upon both OS-oracle baselines across the two datasets. The gains are particularly pronounced on the more challenging BindingDB benchmark. This suggests that our model effectively captures essential patterns directly from the data. Consequently, it appears less sensitive to potential misguidance from inductive biases that might not perfectly align with the real-world complex distributions. Collectively, these results underscore the promise of the proposed framework for drug discovery applications.

6 Related Work

Set function learning is commonly studied through function-value (FV) oracles, which learn to predict the utility of a given set (Wendler et al., 2019; 2021; De & Chakrabarti, 2022). Representative architectures include DeepSets (Zaheer et al., 2017), Set Transformer (Lee et al., 2019), and neural models for submodular functions (Bilmes & Bai, 2017; Dolhansky & Bilmes, 2016; Balcan & Harvey, 2018). However, these methods typically require dense utility supervision, limiting their practicality. Optimal subset (OS) oracles address this issue by learning latent set functions from optimal subsets under an energy-based variational inference framework (Ou et al., 2022), with subsequent extensions incorporating contextual information, hierarchical attention, and implicit differentiation (Xie et al., 2024a;b; Özcan et al., 2025). Nevertheless, existing OS methods use Monte Carlo sampling for gradient estimation, leading to inefficient training. Our work builds on this line by learning a relaxation of neural set functions, while also connecting OS oracle learning with energy minimization and weakly DR-submodular maximization. Further discussion is provided in Appendix G.

7 Conclusion

In this paper, we propose ReSet, which learns a relaxation of neural set functions to advance optimal subset oracles. The central idea of our approach is to replace Monte Carlo sampling over the fixed landscape with a learnable paradigm. By constructing a surrogate of the evidence lower bound, our method enables efficient and stable gradient estimation across the continuous domain, thereby accelerating and improving the variational inference process. We further establish approximation guarantees for the resulting solutions, characterize their convergence to stationary points, and show that the proposed method can adaptively learn the trade-off between expected utility and entropy. As a general plug-in module, our method can be integrated into various optimal subset oracle frameworks, with extensive experiments demonstrating its effectiveness and scalability.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and proofreading. All AI-assisted edits were carefully reviewed by the authors and did not alter the technical content or scientific claims of the paper. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed model descriptions, training procedures, and experimental settings in the main paper and appendices. The assumptions and complete proofs of the theoretical results are included in the appendix. Dataset preprocessing, evaluation protocols, and hyperparameter settings are also provided to facilitate reproduction of the experiments. Anonymous source code is provided in the supplementary materials.

REFERENCES

Maria-Florina Balcan and Nicholas JA Harvey. Submodular functions: Learnability, structure, and optimization. *SIAM Journal on Computing*, 47(3):703–754, 2018.

Gantavya Bhatt, Arnav Das, and Jeff Bilmes. Deep submodular peripteral networks, 2024.

Jeffrey Bilmes and Wenruo Bai. Deep submodular functions. *arXiv preprint arXiv:1701.08939*, 2017.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Benjamin Bloem-Reddy and Yee Whye Teh. Probabilistic symmetries and invariant neural networks. *Journal of Machine Learning Research*, 21(90):1–61, 2020.

Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. In *Integer Programming and Combinatorial Optimization*, pp. 182–196, 2007.

Miguel A Carreira-Perpinan and Geoffrey Hinton. On contrastive divergence learning. In *International workshop on artificial intelligence and statistics*, pp. 33–40, 2005.

Erica Coppolillo, Giuseppe Manco, and Aristides Gionis. Relevance meets diversity: A user-centric framework for knowledge exploration through recommendations. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining*, pp. 490–501, 2024.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. How to compute hessian-vector products? In *The Third Blogpost Track at ICLR*, 2024.

Abir De and Soumen Chakrabarti. Neural estimation of submodular functions with applications to differentiable subset selection. In *Advances in Neural Information Processing Systems*, pp. 19537–19552, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, 2019.

Josip Djolonga and Andreas Krause. Differentiable learning of submodular models. In *Advances in Neural Information Processing Systems*, 2017.

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

Justin Domke. Learning graphical model parameters with approximate marginal inference. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2454–2467, 2013.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in neural information processing systems*, 32, 2019.

Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Learning iterative reasoning through energy minimization. In *International Conference on Machine Learning*, pp. 5570–5582, 2022.

Yilun Du, Jiayuan Mao, and Joshua B Tenenbaum. Learning iterative reasoning through energy diffusion. In *International Conference on Machine Learning*, pp. 11764–11776, 2024.

Jennifer A Gillenwater, Alex Kulesza, Emily Fox, and Ben Taskar. Expectation-maximization for learning determinantal point processes. *Advances in Neural Information Processing Systems*, 27, 2014.

Aleix Gimeno, María José Ojeda-Montes, Sarah Tomás-Hernández, Adrià Cereto-Massagué, Raúl Beltrán-Debón, Miquel Mulero, Gerard Pujadas, and Santiago Garcia-Vallvé. The light and dark sides of virtual screening: what is there to know? *International journal of molecular sciences*, 20 (6):1375, 2019.

Alexi Gladstone, Ganesh Nanduru, Md Mofijul Islam, Peixuan Han, Hyeonjeong Ha, Aman Chadha, Yilun Du, Heng Ji, Jundong Li, and Tariq Iqbal. Energy-based transformers are scalable learners and thinkers. *arXiv preprint arXiv:2507.02092*, 2025.

Joseph Gomes, Bharath Ramsundar, Evan N Feinberg, and Vijay S Pande. Atomic convolutional networks for predicting protein-ligand binding affinity. *arXiv preprint arXiv:1703.10603*, 2017.

Silviu Guiasu and Abe Shenitzer. The principle of maximum entropy. *The mathematical intelligencer*, 7(1):42–48, 1985.

Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jiajun He, Yuanqi Du, Francisco Vargas, Yuanqing Wang, Carla P Gomes, José Miguel Hernández-Lobato, and Eric Vanden-Eijnden. Feat: Free energy estimators with adaptive transport. *arXiv preprint arXiv:2504.11516*, 2025.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

Yuanfeng Ji, Lu Zhang, Jiaxiang Wu, Bingzhe Wu, Lanqing Li, Long-Kai Huang, Tingyang Xu, Yu Rong, Jie Ren, Ding Xue, et al. Drugood: Out-of-distribution dataset curator and benchmark for ai-aided drug discovery–a focus on affinity prediction problems with noise annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8023–8031, 2023.

Nikolaos Karalias, Joshua Robinson, Andreas Loukas, and Stefanie Jegelka. Neural set function extensions: Learning with discrete functions in high dimensions. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15338–15352, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Suraj Kothawade, Jiten Girdhar, Chandrashekhar Lavania, and Rishabh Iyer. Deep submodular networks for extractive data summarization, 2020.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

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, 2019.

Shuangli Li, Jingbo Zhou, Tong Xu, Liang Huang, Fan Wang, Haoyi Xiong, Weili Huang, Dejing Dou, and Hui Xiong. Structure-aware interactive graph neural networks for the prediction of protein-ligand binding affinity. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 975–985, 2021.

Tiqing Liu, Yuhmei Lin, Xin Wen, Robert N Jorissen, and Michael K Gilson. Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic acids research*, 35(suppl_1):D198–D201, 2007.

Zhihai Liu, Yan Li, Li Han, Jie Li, Jie Liu, Zhixiong Zhao, Wei Nie, Yuchen Liu, and Renxiao Wang. Pdb-wide collection of binding data: current status of the pdbbind database. *Bioinformatics*, 31 (3):405–412, 2015a.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015b.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Haggai Maron, Or Litany, Gal Chechik, and Ethan Fetaya. On learning sets of symmetric elements, 2020.

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

Gözde Özcan, Chengzhi Shi, and Stratis Ioannidis. Learning set functions with implicit differentiation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 19777–19785, 2025.

Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

Mohammad Pedramfar, Christopher John Quinn, and Vaneet Aggarwal. A unified approach for maximizing continuous dr-submodular functions, 2024.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

S Hamid Rezatofighi, Vijay Kumar Bg, Anton Milan, Ehsan Abbasnejad, Anthony Dick, and Ian Reid. Deepsetnet: Predicting sets with deep neural networks. In *International Conference on Computer Vision*, pp. 5257–5266, 2017.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in artificial intelligence*, pp. 574–584, 2020.

Stephen M Stigler. *The history of statistics: The measurement of uncertainty before 1900*. Harvard University Press, 1990.

Shao-Hua Sun. Multi-digit mnist for few-shot learning. *GitHub repository*, 2019.

Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071, 2008.

Sebastian Tschiatschek, Aytunc Sahin, and Andreas Krause. Differentiable submodular maximization. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 2731–2738, 2018.

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

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015.

Runqian Wang and Yilun Du. Equilibrium matching: Generative modeling with implicit energy-based models. *arXiv preprint arXiv:2510.02300*, 2025.

Chris Wendler, Markus Püschel, and Dan Alistarh. Powerset convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Chris Wendler, Andisheh Amrollahi, Bastian Seifert, Andreas Krause, and Markus Püschel. Learning set functions that are sparse in non-orthogonal fourier bases. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10283–10292, 2021.

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

Binghui Xie, Yatao Bian, Yongqiang Chen, Peilin Zhao, Bo Han, Wei Meng, James Cheng, et al. Enhancing neural subset selection: Integrating background information into set representations. In *International Conference on Learning Representations*, 2024a.

Binghui Xie, Yixuan Wang, Yongqiang Chen, Kaiwen Zhou, Yu Li, Wei Meng, and James Cheng. Horse: hierarchical representation for large-scale neural subset selection. In *Advances in neural information processing systems*, pp. 4852–4877, 2024b.

Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via mcmc teaching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

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.

David W Zhang, Gertjan J Burghouts, and Cees GM Snoek. Set prediction without imposing structure as conditional density estimation. In *International Conference on Learning Representations*, 2020.

Appendix for "Advancing Optimal Subset Oracle via Learning Relaxation of Neural Set Functions"

CONTENTS

A Proofs and Derivations **15**

A.1 Proof of Theorem 3.1 . . . 15

A.2 Proof of Corollary 3.3 . . . 18

A.3 Derivation of the Discrete Approximation Guarantee . . . 19

A.4 Entropy-Regularized Refinement . . . 19

B Details of Differentiable Mean Field Variational Inference in OS Oracle **20**

B.1 Derivations of the Fixed Point Iteration . . . 20

B.2 Details of Monte Carlo Gradient Estimation . . . 21

B.3 Accelerating Inference with Equivariant Neural Networks . . . 21

C Incorporate Sufficient Invariant Statistics of Background Information **21**

D Detailed Procedure of ReSet **23**

D.1 Overall Framework . . . 23

D.2 Detailed Pseudo Code . . . 23

E Experimental Details **23**

E.1 The Architecture of ReSet . . . 23

E.2 Implementation Details . . . 25

E.3 Baselines . . . 26

E.4 Detailed Experimental Settings for Product Recommendation . . . 26

E.5 Detailed Experimental Settings for Set Anomaly Detection . . . 26

E.6 Detailed Experimental Settings for Compound Selection . . . 28

F Additional Experiments **29**

F.1 Gradient Alignment of the Learned Surrogate . . . 29

F.2 Ablation Study . . . 29

F.3 Comparison with iDiffMF . . . 30

F.4 Computational Efficiency . . . 31

F.5 Comparison with Low-Variance Gradient Estimators . . . 31

F.6 Sensitivity Analysis of Hyperparameters . . . 32

F.7 Set Size Transferability Analysis . . . 35

G Related Work **35**

G.1 Set Function Learning . . . 35

G.2 Energy-based Modeling . . . 36

G.3 Submodular Function Learning . . . 37

A Proofs and Derivations

A.1 Proof of Theorem 3.1

Theorem 3.1. *Suppose the relaxation surrogate $g_\theta : [0,1]^{|V|} \to \mathbb{R}_+$ is L-smooth and γ-weakly DR-submodular . Then the following results hold:*

1) For a solution $\boldsymbol{\psi}^* \in [0,1]^{|V|}$ of continuous greedy, we have

$$g_\theta(\boldsymbol{\psi}^*) \geq \gamma e^{-\gamma} g_\theta(\boldsymbol{\psi}_{\mathrm{OPT}}), \tag{15}$$

where $g_\theta(\boldsymbol{\psi}_{\mathrm{OPT}}) = \max_{\boldsymbol{\psi}} g_\theta(\boldsymbol{\psi})$ is the global optimum.

2) Projected gradient ascent with a fixed step size $0 < \alpha < \frac{1}{L}$ converges to first-order stationarity.

Proof. Recall that we consider the continuous relaxation

$$\max_{\boldsymbol{\psi} \in \mathcal{X}} g_\theta(\boldsymbol{\psi}), \qquad \mathcal{X} = [0,1]^{|V|}.$$

Here, $g_\theta(\cdot)$ is a differentiable surrogate of the multilinear extension induced by the discrete set function $F_\theta(\cdot)$. Under the assumptions of the theorem, $g_\theta : [0,1]^{|V|} \to \mathbb{R}_+$ is L-smooth and γ-weakly DR-submodular for some $\gamma \in (0,1]$. Specifically, for any $\boldsymbol{x}, \boldsymbol{y} \in [0,1]^{|V|}$ satisfying $\boldsymbol{x} \leq \boldsymbol{y}$,

$$\nabla g_\theta(\boldsymbol{x}) \geq \gamma \nabla g_\theta(\boldsymbol{y}), \tag{25}$$

where all vector inequalities are understood coordinate-wise. Let

$$\boldsymbol{\psi}_{\mathrm{OPT}} \in \arg\max_{\boldsymbol{\psi} \in \mathcal{X}} g_\theta(\boldsymbol{\psi})$$

denote a global maximizer.

For brevity, let $d \triangleq |V|$, $\eta \triangleq 1/T$, and define

$$G_t \triangleq g_\theta\left(\boldsymbol{\psi}^{(t)}\right), \qquad G_{\mathrm{OPT}} \triangleq g_\theta\left(\boldsymbol{\psi}_{\mathrm{OPT}}\right).$$

For part 1, initialize $\boldsymbol{\psi}^{(0)} = \mathbf{0}$ and consider the continuous-greedy iteration

$$\begin{aligned} \boldsymbol{v}^{(t)} &\in \arg\max_{\mathbf{0} \leq \boldsymbol{v} \leq \mathbf{1} - \boldsymbol{\psi}^{(t)}} \left\langle \nabla g_\theta(\boldsymbol{\psi}^{(t)}), \boldsymbol{v} \right\rangle, \\ \boldsymbol{\psi}^{(t+1)} &= \boldsymbol{\psi}^{(t)} + \frac{1}{T}\boldsymbol{v}^{(t)}, \qquad t = 0, \ldots, T-1, \end{aligned} \tag{26}$$

where $T \geq 2$. We first verify that the iterates generated by Eq. (26) remain feasible. The claim holds at $t = 0$ because $\boldsymbol{\psi}^{(0)} = \mathbf{0}$. Moreover, since

$$\mathbf{0} \leq \boldsymbol{v}^{(t)} \leq \mathbf{1} - \boldsymbol{\psi}^{(t)}$$

and $\eta \in (0,1]$, we have

$$\mathbf{0} \leq \boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} + \eta \boldsymbol{v}^{(t)} \leq (1-\eta)\boldsymbol{\psi}^{(t)} + \eta \mathbf{1} \leq \mathbf{1}.$$

Therefore, $\boldsymbol{\psi}^{(t)} \in [0,1]^d$ for every t.

We next establish two consequences of γ-weak DR-submodularity that will be used below. For any $\boldsymbol{x}, \boldsymbol{y} \in [0,1]^d$ satisfying $\boldsymbol{x} \leq \boldsymbol{y}$, Eq. (25) implies

$$\langle \nabla g_\theta(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \geq \gamma \left(g_\theta(\boldsymbol{y}) - g_\theta(\boldsymbol{x}) \right). \tag{27}$$

Indeed, letting $\boldsymbol{z} \triangleq \boldsymbol{y} - \boldsymbol{x} \geq \mathbf{0}$ gives

$$\begin{aligned} g_\theta(\boldsymbol{y}) - g_\theta(\boldsymbol{x}) &= \int_0^1 \langle \nabla g_\theta(\boldsymbol{x} + s\boldsymbol{z}), \boldsymbol{z} \rangle \, ds \\ &\leq \frac{1}{\gamma} \langle \nabla g_\theta(\boldsymbol{x}), \boldsymbol{z} \rangle, \end{aligned} \tag{28}$$

where the inequality follows from $\boldsymbol{x} \le \boldsymbol{x} + s\boldsymbol{z}$, $\boldsymbol{z} \ge \mathbf{0}$, and Eq. (25).

In addition, nonnegativity and γ-weak DR-submodularity imply that, for any $\boldsymbol{x}, \boldsymbol{y} \in [0,1]^d$,

$$g_\theta(\boldsymbol{x} \vee \boldsymbol{y}) \ge (1 - \gamma\|\boldsymbol{x}\|_\infty)\, g_\theta(\boldsymbol{y}). \tag{29}$$

To establish this inequality, let $r \triangleq \|\boldsymbol{x}\|_\infty$. The result is immediate when $r = 0$. Otherwise, define

$$\boldsymbol{z} \triangleq \boldsymbol{x} \vee \boldsymbol{y} - \boldsymbol{y} \ge \mathbf{0}.$$

For every $s \in [0, 1/r]$, both $\boldsymbol{y} + rs\boldsymbol{z}$ and $\boldsymbol{y} + s\boldsymbol{z}$ belong to $[0,1]^d$, and $\boldsymbol{y} + rs\boldsymbol{z} \le \boldsymbol{y} + s\boldsymbol{z}$. It follows that

$$\begin{aligned} g_\theta(\boldsymbol{x} \vee \boldsymbol{y}) - g_\theta(\boldsymbol{y}) &= r \int_0^{1/r} \langle \nabla g_\theta(\boldsymbol{y} + rs\boldsymbol{z}), \boldsymbol{z} \rangle \, ds \\ &\ge r\gamma \int_0^{1/r} \langle \nabla g_\theta(\boldsymbol{y} + s\boldsymbol{z}), \boldsymbol{z} \rangle \, ds \\ &= r\gamma \left[g_\theta\left(\boldsymbol{y} + \frac{\boldsymbol{z}}{r}\right) - g_\theta(\boldsymbol{y}) \right] \\ &\ge -r\gamma g_\theta(\boldsymbol{y}), \end{aligned} \tag{30}$$

where the last inequality follows from $g_\theta(\cdot) \ge 0$. Rearranging proves Eq. (29).

We now apply these inequalities to bound the progress at each iteration. Consider the comparison direction

$$\boldsymbol{d}^{(t)} \triangleq \boldsymbol{\psi}_{\mathrm{OPT}} \vee \boldsymbol{\psi}^{(t)} - \boldsymbol{\psi}^{(t)} = \left(\boldsymbol{\psi}_{\mathrm{OPT}} - \boldsymbol{\psi}^{(t)}\right)_+. \tag{31}$$

Since

$$\mathbf{0} \le \boldsymbol{d}^{(t)} \le \mathbf{1} - \boldsymbol{\psi}^{(t)},$$

the vector $\boldsymbol{d}^{(t)}$ is feasible for the linear optimization problem in Eq. (26). The optimality of $\boldsymbol{v}^{(t)}$, together with Eqs. (27) and (29), therefore yields

$$\begin{aligned} \left\langle \nabla g_\theta(\boldsymbol{\psi}^{(t)}), \boldsymbol{v}^{(t)} \right\rangle &\ge \left\langle \nabla g_\theta(\boldsymbol{\psi}^{(t)}), \boldsymbol{d}^{(t)} \right\rangle \\ &\ge \gamma \left[g_\theta\left(\boldsymbol{\psi}^{(t)} \vee \boldsymbol{\psi}_{\mathrm{OPT}}\right) - G_t \right] \\ &\ge \gamma \left[\left(1 - \gamma\|\boldsymbol{\psi}^{(t)}\|_\infty\right) G_{\mathrm{OPT}} - G_t \right]. \end{aligned} \tag{32}$$

For every coordinate i, the direction constraint and $\gamma \le 1$ imply

$$v_i^{(t)} \le 1 - \psi_i^{(t)} \le 1 - \gamma\psi_i^{(t)}.$$

Consequently,

$$\begin{aligned} 1 - \gamma\psi_i^{(t+1)} &= 1 - \gamma\psi_i^{(t)} - \eta\gamma v_i^{(t)} \\ &\ge (1 - \eta\gamma)\left(1 - \gamma\psi_i^{(t)}\right). \end{aligned} \tag{33}$$

Since $\boldsymbol{\psi}^{(0)} = \mathbf{0}$, induction gives

$$1 - \gamma\psi_i^{(t)} \ge (1 - \eta\gamma)^t$$

for every coordinate i. Hence,

$$1 - \gamma\|\boldsymbol{\psi}^{(t)}\|_\infty \ge (1 - \eta\gamma)^t. \tag{34}$$

Combining Eqs. (32) and (34), we obtain

$$\left\langle \nabla g_\theta(\boldsymbol{\psi}^{(t)}), \boldsymbol{v}^{(t)} \right\rangle \ge \gamma \left[(1 - \eta\gamma)^t G_{\mathrm{OPT}} - G_t \right]. \tag{35}$$

By L-smoothness and Eq. (35),

$$\begin{aligned} G_{t+1} &\ge G_t + \eta \left\langle \nabla g_\theta(\boldsymbol{\psi}^{(t)}), \boldsymbol{v}^{(t)} \right\rangle - \frac{L\eta^2}{2}\|\boldsymbol{v}^{(t)}\|_2^2 \\ &\ge (1 - \eta\gamma) G_t + \eta\gamma(1 - \eta\gamma)^t G_{\mathrm{OPT}} - \frac{L\eta^2 d}{2}, \end{aligned} \tag{36}$$

where the last inequality also uses $\|\boldsymbol{v}^{(t)}\|_2^2 \le d$.

Letting $a \triangleq 1 - \eta\gamma$ and recursively applying Eq. (36) yields

$$G_T \ge a^T G_0 + T\eta\gamma a^{T-1} G_{\mathrm{OPT}} - \frac{L\eta^2 d}{2} \sum_{t=0}^{T-1} a^{T-1-t}$$
$$\ge T\eta\gamma a^{T-1} G_{\mathrm{OPT}} - \frac{TL\eta^2 d}{2}, \tag{37}$$

where we used $G_0 \ge 0$, $a \in [0,1]$, and $\sum_{t=0}^{T-1} a^{T-1-t} \le T$. Substituting $\eta = 1/T$ gives

$$G_T \ge \gamma \left(1 - \frac{\gamma}{T}\right)^{T-1} G_{\mathrm{OPT}} - \frac{Ld}{2T}. \tag{38}$$

For $\gamma \in (0,1]$ and $T \ge 2$, applying $\log(1-u) \ge -u/(1-u)$ for $u \in [0,1)$ gives

$$(T-1)\log\left(1 - \frac{\gamma}{T}\right) \ge -\frac{\gamma(T-1)}{T-\gamma} \ge -\gamma,$$

and hence

$$\left(1 - \frac{\gamma}{T}\right)^{T-1} \ge e^{-\gamma}.$$

Substituting this inequality into Eq. (38) and recalling that $d = |V|$, we conclude that

$$g_\theta\left(\boldsymbol{\psi}^{(T)}\right) \ge \gamma e^{-\gamma} g_\theta\left(\boldsymbol{\psi}_{\mathrm{OPT}}\right) - \frac{L|V|}{2T}. \tag{39}$$

Let $\boldsymbol{\psi}_T$ denote the terminal output obtained with step size $1/T$. By compactness of $\mathcal{X}$, the sequence of terminal outputs has accumulation points. Let $\boldsymbol{\psi}^*$ be any such point and choose $T_j \to \infty$ such that $\boldsymbol{\psi}_{T_j} \to \boldsymbol{\psi}^*$. By continuity of g_θ and Eq. (39),

$$g_\theta(\boldsymbol{\psi}^*) = \lim_{j\to\infty} g_\theta(\boldsymbol{\psi}_{T_j}) \ge \gamma e^{-\gamma} g_\theta(\boldsymbol{\psi}_{\mathrm{OPT}}). \tag{40}$$

This proves part 1 for the limiting continuous-greedy output.

We now prove part 2 for projected gradient ascent. Specifically, consider

$$\boldsymbol{\psi}^{(k+1)} = \Pi_{\mathcal{X}}\left(\boldsymbol{\psi}^{(k)} + \alpha \nabla g_\theta(\boldsymbol{\psi}^{(k)})\right), \qquad 0 < \alpha < \frac{1}{L}, \tag{41}$$

where $\Pi_{\mathcal{X}}(\cdot)$ denotes the Euclidean projection onto $\mathcal{X}$. The first-order optimality condition of the projection gives

$$\left\langle \boldsymbol{\psi}^{(k)} + \alpha \nabla g_\theta(\boldsymbol{\psi}^{(k)}) - \boldsymbol{\psi}^{(k+1)}, \boldsymbol{\psi} - \boldsymbol{\psi}^{(k+1)} \right\rangle \le 0, \qquad \forall\, \boldsymbol{\psi} \in \mathcal{X}. \tag{42}$$

Setting $\boldsymbol{\psi} = \boldsymbol{\psi}^{(k)}$ in Eq. (42) yields

$$\left\langle \nabla g_\theta(\boldsymbol{\psi}^{(k)}), \boldsymbol{\psi}^{(k+1)} - \boldsymbol{\psi}^{(k)} \right\rangle \ge \frac{1}{\alpha} \left\| \boldsymbol{\psi}^{(k+1)} - \boldsymbol{\psi}^{(k)} \right\|_2^2. \tag{43}$$

Combining Eq. (43) with L-smoothness gives

$$g_\theta(\boldsymbol{\psi}^{(k+1)}) \ge g_\theta(\boldsymbol{\psi}^{(k)}) + \left\langle \nabla g_\theta(\boldsymbol{\psi}^{(k)}), \boldsymbol{\psi}^{(k+1)} - \boldsymbol{\psi}^{(k)} \right\rangle - \frac{L}{2} \left\| \boldsymbol{\psi}^{(k+1)} - \boldsymbol{\psi}^{(k)} \right\|_2^2$$
$$\ge g_\theta(\boldsymbol{\psi}^{(k)}) + \left(\frac{1}{\alpha} - \frac{L}{2}\right) \left\| \boldsymbol{\psi}^{(k+1)} - \boldsymbol{\psi}^{(k)} \right\|_2^2$$
$$\ge g_\theta(\boldsymbol{\psi}^{(k)}) + \frac{1}{2\alpha} \left\| \boldsymbol{\psi}^{(k+1)} - \boldsymbol{\psi}^{(k)} \right\|_2^2, \tag{44}$$

where the last inequality follows from $\alpha < 1/L$. Thus, the objective values are monotonically nondecreasing. Since g_θ is continuous and $\mathcal{X}$ is compact, g_θ is bounded from above on $\mathcal{X}$. Summing Eq. (44) over k therefore gives

$$\sum_{k=0}^{\infty} \left\| \boldsymbol{\psi}^{(k+1)} - \boldsymbol{\psi}^{(k)} \right\|_2^2 < \infty, \tag{45}$$

and consequently

$$\lim_{k\to\infty} \left\| \boldsymbol{\psi}^{(k+1)} - \boldsymbol{\psi}^{(k)} \right\|_2 = 0. \tag{46}$$

To express this result in terms of first-order stationarity, define the projected gradient mapping

$$\mathcal{G}_\alpha(\boldsymbol{\psi}) \triangleq \frac{1}{\alpha} \left[\Pi_{\mathcal{X}} \left(\boldsymbol{\psi} + \alpha \nabla g_\theta(\boldsymbol{\psi}) \right) - \boldsymbol{\psi} \right]. \tag{47}$$

By Eqs. (41) and (46),

$$\lim_{k\to\infty} \left\| \mathcal{G}_\alpha(\boldsymbol{\psi}^{(k)}) \right\|_2 = 0. \tag{48}$$

Because $\mathcal{X}$ is compact, the sequence $\{\boldsymbol{\psi}^{(k)}\}$ admits at least one accumulation point. Let $\bar{\boldsymbol{\psi}}$ be any such point. The continuity of the projected gradient mapping and Eq. (48) imply $\mathcal{G}_\alpha(\bar{\boldsymbol{\psi}}) = \mathbf{0}$. Equivalently,

$$\left\langle \nabla g_\theta(\bar{\boldsymbol{\psi}}), \boldsymbol{\psi} - \bar{\boldsymbol{\psi}} \right\rangle \le 0, \qquad \forall\, \boldsymbol{\psi} \in \mathcal{X}. \tag{49}$$

Therefore, projected gradient ascent converges to first-order stationarity in the sense that its projected gradient mapping vanishes and every accumulation point is stationary. Together with Eq. (39), this establishes both the approximation and convergence guarantees stated in the theorem.

The two guarantees can also be combined through initialization. Let $\widehat{\boldsymbol{\psi}}$ be the continuous-greedy solution from part 1. With θ fixed, initialize projected gradient ascent in Eq. (41) at $\boldsymbol{\psi}^{(0)} = \widehat{\boldsymbol{\psi}}$, using $0 < \alpha < 1/L$. By Eq. (44), the objective values are nondecreasing, so for every $k \ge 0$,

$$g_\theta(\boldsymbol{\psi}^{(k)}) \ge g_\theta(\widehat{\boldsymbol{\psi}}) \ge \gamma e^{-\gamma} g_\theta(\boldsymbol{\psi}_{\mathrm{OPT}}). \tag{50}$$

The preceding stationarity argument applies to this initialization. Thus, every accumulation point $\bar{\boldsymbol{\psi}}$ is first-order stationary and, by continuity of g_θ, satisfies

$$g_\theta(\bar{\boldsymbol{\psi}}) \ge \gamma e^{-\gamma} g_\theta(\boldsymbol{\psi}_{\mathrm{OPT}}), \qquad \mathcal{G}_\alpha(\bar{\boldsymbol{\psi}}) = \mathbf{0}. \tag{51}$$

Hence, projected gradient refinement preserves the continuous-greedy approximation guarantee while attaining first-order stationarity. □

A.2 Proof of Corollary 3.3

Corollary 3.3. *When the set utility is interpreted as negative energy, i.e., $E_\theta = -F_\theta$, and $k_B T = 1$, the variational free energy equals the negative evidence lower bound, $\mathcal{F}_\theta(\boldsymbol{\psi}) = -\mathrm{ELBO}$.*

Proof. At unit temperature, $k_B T = 1$, the variational free energy in Eq. (19) becomes

$$\mathcal{F}_\theta(\boldsymbol{\psi}) = \mathbb{E}_{S\sim q(\boldsymbol{\psi})}[E_\theta(S;V)] - \mathbb{H}(q(\boldsymbol{\psi})). \tag{52}$$

Since $E_\theta(S;V) = -F_\theta(S;V)$, the definition of the multilinear extension in Eq. (7) gives

$$\mathcal{F}_\theta(\boldsymbol{\psi}) = -f_{\mathrm{mt}}^{F_\theta}(\boldsymbol{\psi}) - \mathbb{H}(q(\boldsymbol{\psi})) = -\mathrm{ELBO}(\boldsymbol{\psi}),$$

where the last equality follows from Eq. (6). Therefore, maximizing the ELBO is equivalent to minimizing the variational free energy. □

A.3 Derivation of the Discrete Approximation Guarantee

Let $F_\theta(S) := g_\theta(\mathbf{1}_S)$ and write $f := f_{\mathrm{mt}}^{F_\theta}$ for brevity. Under the assumptions of Theorem 3.1, consider the continuous-greedy trajectory $\boldsymbol{\psi}(t)$, $t \in [0,1]$, starting from $\mathbf{0}$ and ending at $\boldsymbol{\psi}^*$. Let $\boldsymbol{v}_g(t)$ be the direction selected by g_θ. Directional consistency assumes that this direction also maximizes the linearized objective f along the trajectory:

$$\boldsymbol{v}_g(t) \in \arg\max_{\mathbf{0} \le \boldsymbol{v} \le \mathbf{1} - \boldsymbol{\psi}(t)} \langle \nabla f(\boldsymbol{\psi}(t)), \boldsymbol{v} \rangle . \tag{53}$$

For the box constraints, matching gradient signs is sufficient for this condition. Appendix F.1 provides complementary empirical evidence of local gradient alignment, with median cosine similarities of 0.845–0.982 at sampled probability vectors on synthetic sets. These results support the use of g_θ to guide local optimization, while the guarantee below relies on the exact condition in Eq. (53).

We first show that f satisfies the structural conditions required by Theorem 3.1. Since $g_\theta \ge 0$, both F_θ and f are nonnegative. For $A \subseteq V \setminus \{i\}$, define the discrete marginal gain $\Delta_i F_\theta(A) := F_\theta(A \cup \{i\}) - F_\theta(A)$. For any $A \subseteq B \subseteq V \setminus \{i\}$, the weak DR condition in Eq. (25) implies

$$\Delta_i F_\theta(A) = \int_0^1 \partial_i g_\theta(\mathbf{1}_A + s\boldsymbol{e}_i)\, ds \ge \gamma \int_0^1 \partial_i g_\theta(\mathbf{1}_B + s\boldsymbol{e}_i)\, ds = \gamma \Delta_i F_\theta(B),$$

where $\boldsymbol{e}_i$ is the i-th standard basis vector. For any $\boldsymbol{x} \le \boldsymbol{y}$, couple the independent random subsets $R_{\boldsymbol{x}}, R_{\boldsymbol{y}} \subseteq V \setminus \{i\}$ with inclusion probabilities x_j and y_j, respectively, using the same uniform random variables. Then $R_{\boldsymbol{x}} \subseteq R_{\boldsymbol{y}}$ almost surely, and the derivative formula for the multilinear extension gives

$$\partial_i f(\boldsymbol{x}) = \mathbb{E}\left[\Delta_i F_\theta(R_{\boldsymbol{x}})\right] \ge \gamma \mathbb{E}\left[\Delta_i F_\theta(R_{\boldsymbol{y}})\right] = \gamma \partial_i f(\boldsymbol{y}).$$

Thus, f is γ-weakly DR-submodular. Moreover, as a multilinear polynomial on the compact box $\mathcal{X}$, it is L_f-smooth for some finite L_f.

By Eq. (53), $\boldsymbol{\psi}(t)$ is also a continuous-greedy trajectory for f. Applying Theorem 3.1(1) to f therefore yields

$$f(\boldsymbol{\psi}^*) \ge \gamma e^{-\gamma} \max_{\boldsymbol{\psi} \in \mathcal{X}} f(\boldsymbol{\psi}). \tag{54}$$

The multilinear extension is an expectation of discrete function values, so it cannot exceed $\max_{S \subseteq V} F_\theta(S)$. Equality is attained at the indicator vector of an optimal subset, giving

$$\max_{\boldsymbol{\psi} \in \mathcal{X}} f(\boldsymbol{\psi}) = \max_{S \subseteq V} F_\theta(S) = F_\theta(S_{\mathrm{OPT}}).$$

Finally, independent Bernoulli rounding $S^* \sim q(\boldsymbol{\psi}^*)$ preserves the multilinear value in expectation:

$$\mathbb{E}_{S^* \sim q(\boldsymbol{\psi}^*)}[F_\theta(S^*)] = f(\boldsymbol{\psi}^*) \ge \gamma e^{-\gamma} F_\theta(S_{\mathrm{OPT}}).$$

This establishes Eq. (17).

A.4 Entropy-Regularized Refinement

Under the assumptions of Theorem 3.1, fix θ and $T > 0$, and define

$$h(\boldsymbol{\psi}) := g_\theta(\boldsymbol{\psi}) + k_B T \mathbb{H}(q(\boldsymbol{\psi})), \qquad \mathcal{X}_\delta = [\delta, 1-\delta]^{|V|}, \quad 0 < \delta < \tfrac{1}{2}. \tag{55}$$

The initialization and each displayed update in Eqs. (12) and (22) are projected onto $\mathcal{X}_\delta$; below, $\boldsymbol{\psi}^{(k)}$ denotes the feasible iterate. We analyze deterministic refinement, with $k_B T = 1$ recovering the entropy coefficient used in the main text.

Convergence. The Bernoulli entropy has Hessian $-\operatorname{diag}(1/[\psi_i(1-\psi_i)])$. Hence h is L_δ-smooth on $\mathcal{X}_\delta$, where

$$L_\delta = L + \frac{k_B T}{\delta(1-\delta)}. \tag{56}$$

For a fixed $0 < \alpha < 1/L_\delta$, consider

$$\boldsymbol{\psi}^{(k+1)} = \Pi_{\mathcal{X}_\delta}\left(\boldsymbol{\psi}^{(k)} + \alpha \nabla h(\boldsymbol{\psi}^{(k)})\right). \tag{57}$$

Applying the ascent argument in Appendix A.1 to h gives

$$h(\boldsymbol{\psi}^{(k+1)}) \geq h(\boldsymbol{\psi}^{(k)}) + \frac{1}{2\alpha}\|\boldsymbol{\psi}^{(k+1)} - \boldsymbol{\psi}^{(k)}\|_2^2. \tag{58}$$

Since h is bounded above, summing this inequality shows that the projected gradient mapping vanishes. By compactness and continuity, every accumulation point $\bar{\boldsymbol{\psi}}$ satisfies

$$\langle \nabla h(\bar{\boldsymbol{\psi}}), \boldsymbol{\psi} - \bar{\boldsymbol{\psi}} \rangle \leq 0, \qquad \forall\, \boldsymbol{\psi} \in \mathcal{X}_\delta. \tag{59}$$

Utility after refinement. Let $\boldsymbol{\psi}^*$ be the continuous-greedy solution in Theorem 3.1(1), and initialize at $\boldsymbol{\psi}^{(0)} = \Pi_{\mathcal{X}_\delta}(\boldsymbol{\psi}^*)$. Write $\varepsilon_\delta = \max\{0, g_\theta(\boldsymbol{\psi}^*) - g_\theta(\boldsymbol{\psi}^{(0)})\}$ for the initialization loss. Since $\|\boldsymbol{\psi}^{(0)} - \boldsymbol{\psi}^*\|_2 \leq \delta\sqrt{|V|}$, smoothness yields

$$\varepsilon_\delta \leq \delta\sqrt{|V|}\,\|\nabla g_\theta(\boldsymbol{\psi}^*)\|_2 + \frac{L|V|\delta^2}{2}. \tag{60}$$

Monotonicity and continuity give $h(\bar{\boldsymbol{\psi}}) \geq h(\boldsymbol{\psi}^{(0)})$. Using $\mathbb{H}(q(\boldsymbol{\psi})) \leq |V|\log 2$ and Theorem 3.1(1), we obtain

$$\begin{aligned} g_\theta(\bar{\boldsymbol{\psi}}) &\geq g_\theta(\boldsymbol{\psi}^{(0)}) + k_B T\left[\mathbb{H}(q(\boldsymbol{\psi}^{(0)})) - \mathbb{H}(q(\bar{\boldsymbol{\psi}}))\right] \\ &\geq \gamma e^{-\gamma} g_\theta(\boldsymbol{\psi}_{\mathrm{OPT}}) - \varepsilon_\delta - k_B T\left[|V|\log 2 - \mathbb{H}(q(\boldsymbol{\psi}^{(0)}))\right]. \end{aligned} \tag{61}$$

Thus the same limit point is stationary for h and retains a utility guarantee with an entropy-dependent correction. A finite-step continuous-greedy initializer additionally incurs the discretization loss in Eq. (39).

Discrete output. For $S \sim q(\bar{\boldsymbol{\psi}})$ and $F_\theta(S) = g_\theta(\mathbf{1}_S)$, smoothness gives

$$\mathbb{E}[F_\theta(S)] \geq g_\theta(\bar{\boldsymbol{\psi}}) - \frac{L}{2}\sum_{i\in V}\bar{\psi}_i(1-\bar{\psi}_i). \tag{62}$$

Indeed, the linear term in the smoothness inequality has zero expectation because $\mathbb{E}[\mathbf{1}_S] = \bar{\boldsymbol{\psi}}$, and the expected squared deviation is the sum of the Bernoulli variances. Combining this with Eq. (61) and $g_\theta(\boldsymbol{\psi}_{\mathrm{OPT}}) \geq F_\theta(S_{\mathrm{OPT}})$ yields

$$\begin{aligned} \mathbb{E}[F_\theta(S)] \geq \gamma e^{-\gamma} F_\theta(S_{\mathrm{OPT}}) - \varepsilon_\delta - k_B T\left[|V|\log 2 - \mathbb{H}(q(\boldsymbol{\psi}^{(0)}))\right] \\ - \frac{L}{2}\sum_{i\in V}\bar{\psi}_i(1-\bar{\psi}_i). \end{aligned} \tag{63}$$

B Details of Differentiable Mean Field Variational Inference in OS Oracle

B.1 Derivations of the Fixed Point Iteration

Following Ou et al. (2022), we present a detailed derivation of the fixed-point iteration (FPI) associated with Eq. (9):

$$\psi_i^{(k+1)} \leftarrow \left(1 + \exp\left(-\nabla_{\psi_i^{(k)}} f_{\mathrm{mt}}^{F_\theta}(\boldsymbol{\psi}^{(k)})\right)\right)^{-1}. \tag{64}$$

We begin by recalling that the objective is to maximize the evidence lower bound (ELBO),

$$\max_{\boldsymbol{\psi}\in[0,1]^{|V|}} \underbrace{\sum_{S\subseteq V} F_\theta(S)\prod_{i\in S}\psi_i\prod_{i\notin S}(1-\psi_i)}_{f_{\mathrm{mt}}^{F_\theta}(\boldsymbol{\psi})} - \underbrace{\sum_{i=1}^{|V|}\left[\psi_i\log\psi_i + (1-\psi_i)\log(1-\psi_i)\right]}_{-\mathbb{H}(q(S;\boldsymbol{\psi}))}. \tag{65}$$

The update in Eq. (64) is obtained by enforcing stationarity of the ELBO with respect to coordinate ψ_i. Specifically, setting the partial derivative to zero yields

$$\nabla\psi_i f_{\mathrm{mt}}^{F_\theta}(\boldsymbol{\psi}) + \nabla_{\psi_i}\mathbb{H}(q(S;\boldsymbol{\psi})) = \nabla_{\psi_i} f_{\mathrm{mt}}^{F_\theta}(\boldsymbol{\psi}) + \log\frac{1-\psi_i}{\psi_i} = 0.$$

Solving for ψ_i leads to

$$\psi_i = \left(1+\exp\left(-\nabla_{\psi_i} f_{\mathrm{mt}}^{F_\theta}(\boldsymbol{\psi})\right)\right)^{-1},$$

which coincides with the fixed-point update employed in mean-field variational inference.

B.2 Details of Monte Carlo Gradient Estimation

According to Eq. (64), the variational parameters can be updated via fixed-point iteration. Nevertheless, a major computational challenge arises from evaluating the gradient of the multilinear extension $f_{\mathrm{mt}}^{F_\theta}(\boldsymbol{\psi})$ defined in Eq. (7), as it involves summation over all $2^{|V|}$ possible subsets. To address this issue, Ou et al. (2022) showed that the gradient $\nabla_{\boldsymbol{\psi}} f_{\mathrm{mt}}^{F_\theta}$ admits an expectation form, which enables efficient estimation through Monte Carlo sampling. Specifically, the partial derivative with respect to ψ_i can be derived as

$$\begin{aligned}
\nabla_{\psi_i} f_{\mathrm{mt}}^{F_\theta} &= \nabla_{\psi_i}\sum_{S\subseteq V} F_\theta(S)\prod_{i\in S}\psi_i\prod_{i\notin S}(1-\psi_i) \\
&= \mathbb{E}_{q(S;(\boldsymbol{\psi}|\psi_i\leftarrow 1))}[F_\theta(S)] - \mathbb{E}_{q(S;(\boldsymbol{\psi}|\psi_i\leftarrow 0))}[F_\theta(S)] \\
&= \sum_{S\subseteq V, i\in S} F_\theta(S)\prod_{j\in S\setminus\{i\}}\psi_j\prod_{j'\notin S}(1-\psi_{j'}) - \sum_{S\subseteq V\setminus\{i\}} F_\theta(S)\prod_{j\in S}\psi_j\prod_{j'\in V\setminus(S\cup\{i\})}(1-\psi_{j'}) \\
&= \sum_{S\subseteq V\setminus\{i\}}[F_\theta(S+i)-F_\theta(S)]\prod_{j\in S}\psi_j\prod_{j'\in V\setminus S\setminus\{i\}}(1-\psi_{j'}) \\
&= \mathbb{E}_{q(S;(\boldsymbol{\psi}|\psi_i\leftarrow 0))}\left[F_\theta(S+i)-F_\theta(S)\right]. \qquad (66)
\end{aligned}$$

Here, $S+i$ denotes the union $S\cup\{i\}$. Based on Eq. (66), the gradient $\nabla_{\psi_i} f_{\mathrm{mt}}^{F_\theta}$ can be approximated using Monte Carlo estimation: i) sample m subsets $S_n, n=1,\ldots,m$ from the variational distribution $q(S;(\boldsymbol{\psi}|\psi_i\leftarrow 0))$; ii) approximate the expectation by the average $\frac{1}{m}\sum_{n=1}^{m}[F_\theta(S_n+i)-F_\theta(S_n)]$. This procedure provides a tractable and unbiased estimator of the gradient required for the fixed-point updates.

B.3 Accelerating Inference with Equivariant Neural Networks

Although Monte Carlo–based gradient estimation makes the optimization of the variational parameters $\boldsymbol{\psi}$ feasible, it is accompanied by a notorious drawback: the computational cost is often prohibitively high. In particular, the DiffMF procedure in Eq. (8) typically involves an inner sampling loop for each data instance, which substantially increases the overall training overhead. To improve scalability on large-scale datasets, prior works (Ou et al., 2022; Xie et al., 2024a) propose to amortize the approximate inference process by introducing an auxiliary recognition network. This network directly predicts the variational parameters $\boldsymbol{\psi}$ of the distribution $q_\phi(S;\boldsymbol{\psi})$, where ϕ denotes the learnable parameters of the recognition model. Crucially, the recognition network must be permutation equivariant: permuting the input elements should permute the output probabilities in the same way, preserving the correspondence between each element and its variational parameter. In our framework, optimizing each data point similarly requires multiple gradient ascent steps as in Eq. (12). Therefore, we likewise employ an auxiliary recognition network to accelerate the inference procedure. Concretely, the recognition network provides an initialization $\boldsymbol{\psi}^{(0)} = \mathrm{RecNet}(V;\phi)\in[0,1]^{|V|}$. Each output coordinate specifies the inclusion probability of the corresponding element in V, jointly defining the initial variational distribution $q_\phi(S;\boldsymbol{\psi}^{(0)})$.

C Incorporate Sufficient Invariant Statistics of Background Information

Within our framework, we adopt the architecture proposed by Xie et al. (2024a) and implement its relaxed variant, INSET-R. The central idea of INSET is to incorporate contextual information from

the ground set V into the set function, yielding a formulation of the form $F(V, S)$, in contrast to EquiVSet, which models the set function solely as $F(S)$. This mechanism is referred to by Xie et al. (2024a) as the fusion of *invariant sufficient representations* of the subset S and the ground set V. Specifically, Bloem-Reddy & Teh (2020) show that the empirical measure $M_S(S) = \sum_{s_i \in S} \delta(s_i)$ can be served as a valid invariant sufficient representation, where $\delta(s_i)$ denotes a unit-mass atom located at s_i, for example instantiated via one-hot encodings. Furthermore, building on the result of Zaheer et al. (2017), this empirical measure can be efficiently approximated using the aggregation form $\rho\left(\sum_{s \in S} \phi(s)\right)$. This formulation provides a practical and effective means of realizing invariant sufficient representations in neural architectures.

Proposition C.1. *If f is a valid permutation invariant function on S, it can be approximated arbitrarily close in the form of $f(S) = \rho\left(\sum_{s \in S} \phi(s)\right)$, for suitable transformations ϕ and ρ.*

In practice, an encoder $\phi(\cdot)$ is employed to produce element-wise embeddings, while the mapping $\rho(\cdot)$ may be instantiated by a variety of feedforward architectures, including fully connected layers equipped with nonlinear activations. In a related line of work, Maron et al. (2020) establish that universal approximation of invariant sufficient representations can be achieved via the aggregation form $\sum_{S \in V} \sum_{s \in S} \phi(s)$, which simplifies to the equivalent expression $\sum_{x_j \in V} \phi(x_j)$. Therefore, the neural network architecture for set functions injected with superset background information can be expressed as

$$\theta(S, V) = \sigma\left(\theta_1\left(\sum_{i=1}^{n_i} \phi(x_i)\right) + \theta_2\left(\sum_{j=1}^{n} \phi(x_j)\right)\right). \tag{67}$$

Here, the feed-forward modules θ_1 and θ_2 are accompanied by a non-linear activation layer denoted by σ. As a comparison, we provide the set function architectures for four different optimal subset oracles in Table 4.

Table 4: Comparison of set function architectures across four different optimal subset oracles.

EquiVSet	**EquiVSet-**R	**INSET**	**INSET-**R
$\mathrm{FC}_S(256, 500, \mathrm{ReLU})$	$\mathrm{MHA}_S(256, 4)$	$\mathrm{FC}_V(256, 500, \mathrm{ReLU})$	$\mathrm{MHA}_V(256, 4)$
$\mathrm{FC}_S(500, 1, -)$	$\mathrm{FC}_S(256, 512, \mathrm{ReLU})$	$\mathrm{FC}_V(500, 1, -)$	$\mathrm{MHA}_S(256, 4)$
-	$\mathrm{FC}_S(512, 1, -)$	$\mathrm{FC}_S(256, 500, \mathrm{ReLU})$	$\mathrm{FC}_V(256, 512, \mathrm{ReLU})$
-	-	$\mathrm{FC}_S(500, 1, -)$	$\mathrm{FC}_V(512, 1, -)$
-	-	-	$\mathrm{FC}_S(256, 512, \mathrm{ReLU})$
-	-	-	$\mathrm{FC}_S(512, 1, -)$

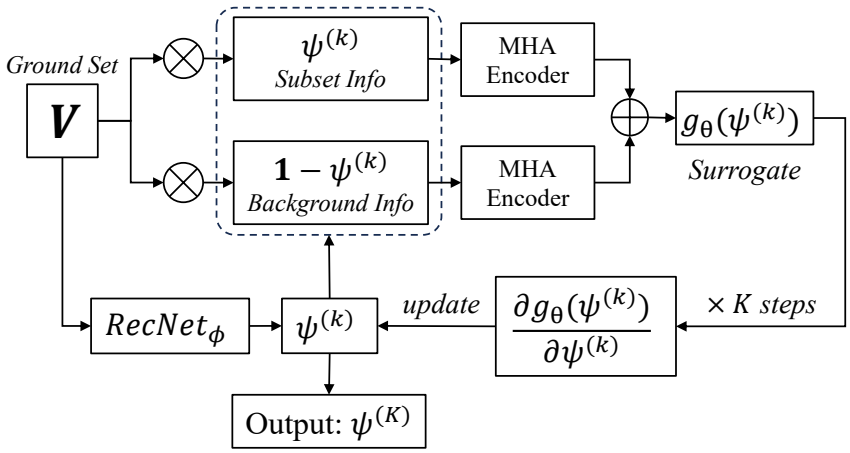


Figure 3: Overall framework of the proposed network architecture.

D Detailed Procedure of ReSet

D.1 Overall Framework

Figure 3 illustrates the overall architecture of ReSet. Given a ground set V, an auxiliary recognition network RecNet_ϕ first produces an initial variational parameter $\boldsymbol{\psi}^{(0)}$, which amortizes the inference process. At each optimization step, the current soft assignment $\boldsymbol{\psi}^{(k)}$ and its complement $1-\boldsymbol{\psi}^{(k)}$ are used to construct subset-aware and background-aware representations of the input set. These two branches are encoded by multi-head attention modules to capture element-wise dependencies, and are then combined to produce the surrogate $g_\theta(\boldsymbol{\psi}^{(k)})$. The surrogate gradient is combined with the analytic entropy gradient to iteratively update the variational parameter for K steps, yielding the final prediction $\boldsymbol{\psi}^{(K)}$. This design enables efficient differentiable optimization without Monte Carlo gradient estimation.

D.2 Detailed Pseudo Code

We provide the pseudo-code for the training and inference procedures of ReSet in Algorithms 2 and 3, respectively. The training process consists of three key steps: i) generating the initial variational parameters $\boldsymbol{\psi}^{(0)}$ via RecNet_ϕ; ii) iteratively updating $\boldsymbol{\psi}^{(k)}$ for K steps using the entropy-regularized surrogate update in Eq. (22); and iii) computing the cross-entropy loss based on the optimized $\boldsymbol{\psi}^{(K)}$ in Eq. (5) and updating the network parameters through backpropagation. During inference, only steps i) and ii) are executed. Notably, we incorporate Langevin dynamics exclusively during training to facilitate exploration, while opting for a deterministic approach during inference to ensure stability and finer convergence.

Algorithm 2 ReSet (Training)

Require: $\{V_i, S_i^*\}_{i=1}^N$: training dataset; η: learning rate; K: number of iterations; α: step size for gradient ascent

Ensure: Optimal parameters θ^*, ϕ^*

$\theta, \phi \leftarrow$ Initialize parameters
while θ, ϕ not converged **do**
  Sample training data point $(V, S^*) \sim \{V_i, S_i^*\}_{i=1}^N$
  Obtain the initial variational parameter $\boldsymbol{\psi}^{(0)}$ via the auxiliary recognition network $\boldsymbol{\psi}^{(0)} \leftarrow \mathrm{RecNet}_\phi(V)$
  for $k = 0$ to $K-1$ **do**
    Calculate the surrogate $g_\theta(\boldsymbol{\psi}^{(k)})$ using Eq. (24)
    Update the parameter $\boldsymbol{\psi}$ by gradient ascent in Eq. (22) $\boldsymbol{\psi}^{(k+1)} \leftarrow \boldsymbol{\psi}^{(k)} + \alpha(\nabla g_\theta(\boldsymbol{\psi}^{(k)}) + \log \frac{\mathbf{1}-\boldsymbol{\psi}^{(k)}}{\boldsymbol{\psi}^{(k)}}) + \boldsymbol{\epsilon}_k$
  end for
  Set $\boldsymbol{\psi}^* \leftarrow \boldsymbol{\psi}^{(K)}$
  Update θ and ϕ by minimizing the cross-entropy loss in Eq. (5) $(\theta, \phi) \leftarrow (\theta, \phi) - \eta \nabla_{(\theta,\phi)} \Big(-\sum_{i \in S^*} \log \psi_i^* - \sum_{i \in V \setminus S^*} \log(1-\psi_i^*) \Big)$
end while
Set $\theta^*, \phi^* \leftarrow \theta, \phi$
return θ^*, ϕ^*

E Experimental Details

E.1 The Architecture of ReSet

Similar to the framework of Ou et al. (2022); Xie et al. (2024a), ReSet consists of two neural components: (i) a set function that is permutation invariant, and (ii) a recognition network that is permutation equivariant. Both components are implemented using the DeepSets architecture, with detailed architectural specifications provided in Table 5. Specifically, $\mathrm{InitLayer}(d_{in}, d_{out})$ encodes

Algorithm 3 ReSet (Inference)

Require: V: test ground set; θ^*, ϕ^*: trained parameters; K: number of iterations; α: step size for gradient ascent
Ensure: S^*: Predicted optimal subset
1: Obtain the initial variational parameter via the recognition network
2: $\boldsymbol{\psi}^{(0)} \leftarrow \text{RecNet}_{\phi^*}(V)$
3: **for** $k = 0$ to $K - 1$ **do**
4: Calculate the surrogate $g_{\theta^*}(\boldsymbol{\psi}^{(k)})$ using Eq. (24)
5: Update $\boldsymbol{\psi}$ by gradient ascent:
6: $\boldsymbol{\psi}^{(k+1)} \leftarrow \boldsymbol{\psi}^{(k)} + \alpha(\nabla g_{\theta^*}(\boldsymbol{\psi}^{(k)}) + \log \frac{\mathbf{1} - \boldsymbol{\psi}^{(k)}}{\boldsymbol{\psi}^{(k)}})$
7: **end for**
8: Set $\boldsymbol{\psi}^* \leftarrow \boldsymbol{\psi}^{(K)}$
9: Discretize the continuous solution
10: $S^* \leftarrow \text{Rounding}(\boldsymbol{\psi}^*)$
11: **return** S^*

Table 5: Detailed architectures of ReSet.

Set (Surrogate) Function	Recognition Network
$\text{InitLayer}(V, 256)$	$\text{InitLayer}(V, 256)$
$\text{MHA}(256, 4)$	$\text{FC}(256, 512, \text{ReLU})$
$\text{LayerNorm}(256)$	$\text{FC}(512, 1, \text{Sigmoid})$
SumPooling	-
$\text{FC}(256, 512, \text{ReLU})$	-
$\text{FC}(512, 1, -)$	-

the set objects into vector representations, $\text{FC}(d_{in}, d_{out}, f)$ is a fully-connected layer with activation function f, $\text{MHA}(d, h)$ denotes a multi-head attention layer with h heads.

Synthetic datasets. We consider two synthetic benchmarks, namely Two-Moons and Gaussian Mixture. Each set element is represented by a two-dimensional vector encoding its Cartesian coordinates. For both datasets, the InitLayer is instantiated as a single-layer feed-forward network with architecture $\text{FC}(2, 256, -)$.

Amazon Baby Registry. The Amazon Baby Registry dataset consists of sets of products, where each product is associated with a short textual description. We encode these descriptions using a pre-trained BERT model (Devlin et al., 2019), resulting in a 768-dimensional embedding for each element. The corresponding InitLayer is implemented as $\text{FC}(768, 256, -)$.

Double MNIST. The Double MNIST dataset contains digit images spanning all combinations from 00 to 99. Each image has a resolution of $(64, 64)$ and is reshaped into a vector of dimension 4096. Accordingly, we employ a fully connected initialization layer with architecture $\text{FC}(4096, 256, -)$.

F-MNIST. The Fashion-MNIST dataset consists of grayscale images of fashion items from 10 categories. Each image has a spatial resolution of $(28, 28)$ and is flattened into a 784-dimensional vector. Similar to Double MNIST, the InitLayer is implemented as a feed-forward neural network with architecture $\text{FC}(784, 256, -)$.

CelebA. The CelebA dataset comprises $202{,}599$ face images, each of size $(3, 64, 64)$. For this dataset, the InitLayer is realized using a convolutional neural network. Specifically, its architecture is given by

$$\text{ModuleList}([\text{Conv}(32, 3, 2, \text{ReLU}),\ \text{Conv}(64, 4, 2, \text{ReLU}),\\ \text{Conv}(128, 5, 2, \text{ReLU}),\ \text{MaxPooling},\ \text{FC}(128, 256, -)]),$$

where $\text{Conv}(d, k, s, f)$ denotes a convolutional layer parameterized by d output channels, kernel size k, stride s, and activation function f.

CIFAR-10. The CIFAR-10 dataset consists of $60,000$ color images of size 32×32 pixels. For this dataset, the feature extraction network is implemented as a deep convolutional neural network

consisting of three cascaded functional blocks. Specifically, the architecture is formulated as:

$$\text{ModuleList}\left([\text{Block}(32), \text{Block}(64), \text{Block}(128), \text{AvgPool}, \text{FC}(128, 256, -)]\right),$$

where each $\text{Block}(d)$ is a sub-network designed to capture hierarchical spatial features, defined as the sequence:

$$[\text{Conv}(d, 3, 2, \text{ReLU}), \text{BN}, \text{Conv}(d, 3, 1, \text{ReLU}), \text{BN}].$$

To maintain spatial resolution during convolution, zero-padding is applied such that $p = \lfloor k/2 \rfloor$. Each BN denotes a Batch Normalization layer applied after the convolution. It should be noted that the code related to this part was not publicly released in Ou et al. (2022), so we constructed the InitLayer architecture ourselves.

PDBBind. The PDBBind database provides experimentally determined binding affinities for biomolecular complexes (Liu et al., 2015a), together with high-resolution three-dimensional Cartesian coordinates of ligands and their corresponding target proteins, typically obtained via techniques such as X-ray crystallography. To extract informative representations, we leverage the atomic convolutional neural network (ACNN) (Gomes et al., 2017), which constructs nearest-neighbor graphs based on atomic spatial coordinates. In our implementation, the representation is obtained from the penultimate layer of ACNN, followed by additional feed-forward layers. Formally, the InitLayer is defined as

$$\text{ModuleList}([\text{ACNN}[:-1],\ \text{FC}(1922, 2048, \text{ReLU}),\ \text{FC}(2048, 256, -)]),$$

where $\text{ACNN}[:-1]$ denotes the ACNN architecture without its final prediction layer, yielding an output dimension of 1,922.

BindingDB. The BindingDB dataset consists of 52,273 drug–target interaction pairs. We adopt the DeepDTA framework (Öztürk et al., 2018) to encode each drug–target pair into a fixed-dimensional vector. DeepDTA represents drug compounds and target proteins as one-hot encoded sequences, which are subsequently processed by convolutional neural networks. The full specification of the InitLayer used for this dataset is reported in Table 6.

Table 6: Detailed architectures of InitLayer in the BindingDB dataset.

Drug	**Target**
$\text{Conv}(32, 4, 1, \text{ReLU})$	$\text{Conv}(32, 4, 1, \text{ReLU})$
$\text{Conv}(64, 6, 1, \text{ReLU})$	$\text{Conv}(64, 8, 1, \text{ReLU})$
$\text{Conv}(96, 8, 1, \text{ReLU})$	$\text{Conv}(96, 12, 1, \text{ReLU})$
MaxPooling	MaxPooling
$\text{FC}(96, 256, \text{ReLU})$	$\text{FC}(96, 256, \text{ReLU})$
Concat	
$\text{FC}(512, 256, -)$	

E.2 Implementation Details

We provide the training details and hyperparameter settings for ReSet. The proposed model is trained using the Adam optimizer (Kingma & Ba, 2017) with a fixed weight decay of $1e-5$ and a batch size of 128. During training, we employ a cosine annealing learning rate scheduler (Loshchilov & Hutter, 2016) together with an early stopping strategy. The initial learning rate is selected via grid search over the range $[1e-5, 1e-3]$ using half-decade intervals, with the minimum learning rate fixed at one-tenth of the chosen initial value. Training is conducted for up to 500 epochs, and early stopping is triggered if no performance improvement is observed for 20 consecutive epochs. For datasets that are more difficult to converge, we allow a longer patience. After training, the saved models are evaluated on the test set. All experiments are repeated 5 times with different random seeds, and the mean performance along with their standard deviations are reported.

Within the ReSet framework, the gradient ascent step size α and the noise magnitude ϵ in the Langevin dynamics constitute two critical hyperparameters governing model performance. To enhance the adaptivity of the learning process, we allow both parameters to be updated during training. Specifically, the initial value of α is selected empirically. In contrast, although ϵ is treated as a learnable parameter, it is constrained to remain at a small scale of $1e-5$. This strategy aims to ensure numerical stability during training while enabling precise control over the determinacy of the model.

E.3 BASELINES

Throughout our experiments, we evaluate the proposed models against four conventional approaches: Random Guess, Probabilistic Greedy Model (PGM) (Tschiatschek et al., 2018), DeepSet (Zaheer et al., 2017), and Set Transformer (Lee et al., 2019). Additionally, we compare our method with two state-of-the-art optimal subset oracles: EquiVSet (Ou et al., 2022) and INSET (Xie et al., 2024a). Detailed descriptions of these benchmarks are provided below.

- **Random.** We report the expected Jaccard coefficient obtained by random guessing. This baseline serves as a reference for assessing the intrinsic difficulty of the task.
- **PGM** (Tschiatschek et al., 2018). The probabilistic greedy model tackles Problem (2) by introducing a differentiable relaxation of the greedy maximization procedure, which relies on enumerating all possible element permutations. For a detailed exposition, we refer readers to Appendix A of Ou et al. (2022).
- **DeepSet** (Zaheer et al., 2017). By learning a permutation-invariant mapping from $2^{|V|}$ to $[0,1]^{|V|}$, the DeepSet architecture provides a generic backbone for modeling set functions in optimal subset oracle frameworks (Ou et al., 2022; Xie et al., 2024a;b). We therefore include it as a baseline in our evaluation. To adapt DeepSets for point-wise tasks, we shift from an invariant to a permutation equivariant design. Instead of global reduction, we broadcast the mean-pooled global descriptor to each element, allowing for an output of size $|V|$ that preserves the identity of individual members.
- **Set Transformer** (Lee et al., 2019). Extending the DeepSet framework, Set Transformer incorporates self-attention mechanisms to model pairwise interactions among set elements. Similar to DeepSet, it can be adapted for subset selection tasks and is included as a competitive baseline. Similar to our adaptation of DeepSet, we bypass the original global pooling (PMA) and invariant decoder, utilizing only the encoder blocks. This preserves the permutation equivariance of the self-attention mechanism, allowing for element-wise prediction while maintaining an output dimensionality of $|V|$.
- **EquiVSet** (Ou et al., 2022). EquiVSet aims to learn an optimal subset oracle without explicitly specifying the underlying set function. Instead, the set function is modeled as an energy-based formulation and approximated through a variational distribution. Optimization via mean-field variational inference requires iterative unrolling with Monte Carlo sampling to estimate gradients of the multilinear extension, which introduces substantial computational overhead.
- **INSET** (Xie et al., 2024a). Building upon EquiVSet, INSET augments the set function $F_\theta(\cdot)$ by injecting background information from the ground set V. By explicitly accounting for the symmetry of the ground set, this design introduces a stronger inductive bias and more effectively exploits contextual relationships within the global set.

E.4 DETAILED EXPERIMENTAL SETTINGS FOR PRODUCT RECOMMENDATION

The Amazon Baby Registry dataset (Gillenwater et al., 2014) is collected from Amazon and organized into multiple subsets based on product categories, such as toys and furniture. Within each category, which serves as a product universe, Amazon records multiple product subsets curated by different customers. These customer-selected subsets naturally correspond to optimal subset (OS) oracles. To ensure that each ground set V is associated with a single OS oracle S^*, we construct data samples in the form of (V, S^*) as follows. We first discard customer-selected subsets whose cardinality is equal to 1 or exceeds 30. For each remaining OS oracle S^*, we randomly sample $30 - |S^*|$ additional products from the same category to form the complement set $V \setminus S^*$. Statistics of the resulting datasets across different categories are reported in Table 7.

E.5 DETAILED EXPERIMENTAL SETTINGS FOR SET ANOMALY DETECTION

CelebA: The CelebA dataset consists of 202,599 images annotated with 40 binary attributes. As illustrated in Fig. 4, we randomly choose two attributes and form sets of fixed cardinality 8. For each ground set V, an OS oracle subset S^* is constructed by randomly sampling $n \in \{2, 3\}$ images

Table 7: The statistics of Amazon product dataset.

Categories	#products	$\lvert\mathcal{D}\rvert$	$\lvert V\rvert$	$\sum \lvert S^*\rvert$	$\mathbb{E}[\lvert S^*\rvert]$	$\min_{S^*} \lvert S^*\rvert$	$\max_{S^*} \lvert S^*\rvert$
Toys	62	2,421	30	9,924	4.09	3	14
Furniture	32	280	30	892	3.18	3	6
Gear	100	4,277	30	16,288	3.80	3	10
Carseats	34	483	30	1,576	3.26	3	6
Bath	100	3,195	30	12,147	3.80	3	11
Health	62	2,995	30	11,053	3.69	3	9
Diaper	100	6,108	30	25,333	4.14	3	15
Bedding	100	4,524	30	17,509	3.87	3	12
Safety	36	267	30	846	3.16	3	5
Feeding	100	8,202	30	37,901	4.62	3	23
Apparel	100	4,675	30	21,176	4.52	3	21
Media	58	1,485	30	6,723	4.52	3	19

such that neither of the selected attributes is exhibited. Based on this procedure, we generate training, validation, and test splits containing 10,000, 1,000, and 1,000 samples, respectively.



Figure 4: Example data points from the CelebA dataset adapted from Ou et al. (2022). Each row corresponds to one sample and contains $|S^*|$ anomalous images (highlighted in red) along with $8 - |S^|$ normal images. Within each sample, normal images exhibit both selected attributes (shown in the rightmost column), whereas anomalous images exhibit neither attribute.

Comparing with the Setting of Ou et al. (2022). Although we aim to align our experimental setup as closely as possible with that of Ou et al. (2022), the experimental protocols for the anomaly detection task on Double MNIST (Sun, 2019), F-MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky, 2009) are not publicly available. Moreover, the corresponding descriptions in the original work are relatively ambiguous. For clarity and reproducibility, we therefore re-establish and standardize the experimental settings for these three datasets. Specifically, for Double MNIST, we generate a total of $1,000$ images distributed across all digit pairs from 00 to 99.[12] For F-MNIST and CIFAR-10, we use the official datasets obtained via online sources. During sample construction, we randomly select $n \in \{2, 3, 4\}$ images from the same class to form the optimal subset oracle S^* for all three datasets. For the Double MNIST dataset, $V \backslash S^*$ is constructed by selecting $20 - |S^*|$ images, whereas for F-MNIST and CIFAR-10, it is formed by drawing $10 - |S^*|$ images from different classes. Notably, the images in $V \backslash S^*$ are drawn from distinct classes, with at most one image per class. Finally, we generate samples on the fly, producing $10,000$, $1,000$, and $1,000$ instances for the training, validation, and test sets, respectively. Visual illustrations are provided in Figures 5 and 6.

[12]The source code for dataset generation is publicly available at `https://github.com/shaohua0116/MultiDigitMNIST`.

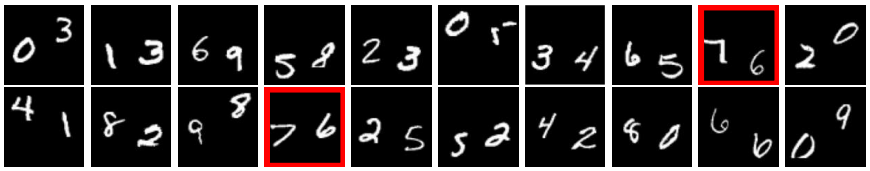


Figure 5: A sampled data for the Double MNIST dataset, which consists of $|S^*|$ images with the same digit (red box, 76 in this case) and $20 - |S^*|$ images with different digits.

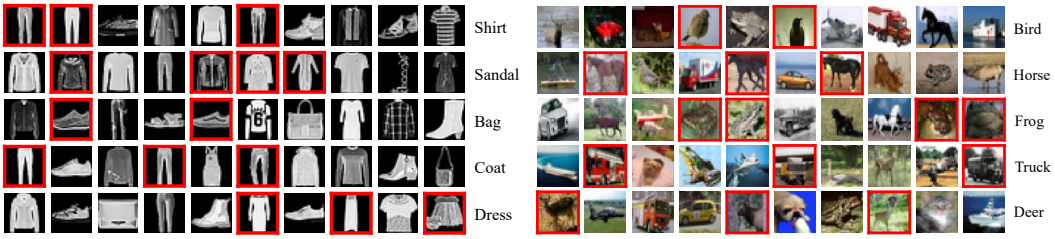


Figure 6: Illustrative samples from the F-MNIST (left) and CIFAR-10 (right) datasets. Each row represents a single sample composed of $|S^*|$ images sharing the same class label (highlighted in red) and $10 - |S^*|$ additional images drawn from distinct classes. The corresponding labels are shown in the rightmost column.

E.6 Detailed Experimental Settings for Compound Selection

In the drug discovery pipeline, identifying compounds that exhibit strong bioactivity (Wallach et al., 2015; Li et al., 2021; Ji et al., 2023), sufficient structural diversity, and favorable ADME (absorption, distribution, metabolism, and excretion) profiles (Gimeno et al., 2019) constitutes a fundamental screening stage. In practice, virtual screening is commonly implemented as a sequence of filtering operations. A typical workflow first prioritizes candidates based on predicted bioactivity, subsequently enforces diversity constraints, and finally excludes compounds with undesirable ADME characteristics, yielding a refined candidate subset. However, owing to privacy considerations and cost constraints in pharmaceutical development, intermediate supervision signals corresponding to these individual filtering stages are often inaccessible or prohibitively expensive to obtain. Consequently, learning the entire screening procedure in an end-to-end fashion becomes desirable, a setting that naturally aligns with the formulation of optimal subset (OS) oracles. Despite this alignment, fully end-to-end modeling remains challenging and often demands substantial domain knowledge. In this work, we therefore adopt a simplified yet representative setting by focusing on the prediction of high-bioactivity compounds, following the experimental protocol of Xie et al. (2024a). Experiments are conducted on the PDBBind (Liu et al., 2015a) and BindingDB (Liu et al., 2007) datasets using a unified bioactivity-based filtering criterion. Algorithm 4 illustrates the data generation procedure used to simulate the OS oracle for the compound selection task. Specifically, $\mathrm{random_choose}(\mathcal{C}, n)$ denotes uniformly sampling n compounds from a database $\mathcal{C}$ (either PDBBind or BindingDB) to construct the ground set V, while $\mathrm{topK_bioactivity}(V, m)$ selects the top-m compounds exhibiting the highest biological activity from V. These two operations together form a bioactivity-based filtering process, through which the OS oracle S^* is obtained. In our experiments, we set $(n, m) = (30, 5)$ for PDBBind and $(300, 15)$ for BindingDB, respectively. This procedure yields a data point (V, S^*), where V represents a candidate compound set and S^* corresponds to the subset of compounds with the highest bioactivity.

Algorithm 4 OS Oracle Construction Procedure for the Compound Selection Task

Input: $\mathcal{C}$: compound database; n: cardinality of the ground set; m: number of highly active compounds
Output: A data sample (V, S^*)
1: **Ground Set Formation:**
2: Uniformly sample n compounds from the database $\mathcal{C}$
3: $V \leftarrow \text{random_choose}(\mathcal{C}, n)$
4: **Bioactivity-based Selection (OS Oracle):**
5: Select the top-m compounds with the highest bioactivity scores from V
6: $S^* \leftarrow \text{topK_bioactivity}(V, m)$
7: **return** (V, S^*)

F Additional Experiments

F.1 Gradient Alignment of the Learned Surrogate

We evaluate the learned surrogate g_θ by comparing its gradient directions with those of its induced multilinear extension. Small synthetic sets allow us to compute the reference gradient exactly and directly assess the surrogate's role in guiding variational optimization.

Experimental setup. We construct synthetic sets of $d \in \{8, 10, 12\}$ two-dimensional elements by perturbing three anchor points with Gaussian noise and randomly permuting the resulting elements. Optimal-subset labels are obtained by exhaustive search under a coverage utility with a cardinality penalty, encouraging compact subsets that cover the three anchor regions. Each model is trained across these set sizes using one unrolled inference step and only optimal-subset supervision, without explicit function-value or gradient-matching losses.

Gradient comparison. For each architecture, we freeze a single trained model and evaluate it on one held-out input set for each cardinality. For each fixed V, define $F_\theta(S) := g_\theta(\mathbf{1}_S)$ and write $f := f^{F_\theta}_{\text{mt}}$. We enumerate all 2^d subsets to compute f and its gradient exactly, without Monte Carlo approximation. For each input, we sample 64 probability vectors uniformly from $(0, 1)^d$ and measure

$$C(\boldsymbol{\psi}) = \frac{\langle \nabla_{\boldsymbol{\psi}} g_\theta(\boldsymbol{\psi}), \nabla_{\boldsymbol{\psi}} f(\boldsymbol{\psi}) \rangle}{\|\nabla_{\boldsymbol{\psi}} g_\theta(\boldsymbol{\psi})\|_2 \|\nabla_{\boldsymbol{\psi}} f(\boldsymbol{\psi})\|_2}. \tag{68}$$

Both models use the same input sets and probability vectors. The comparison excludes the entropy term to isolate the alignment between the two objectives.

Results. Figure 7 shows positive gradient alignment at all sampled points. For $d = 8, 10, 12$, the median cosine similarities are 0.952, 0.878, and 0.845 for EquiVSet-R, and 0.982, 0.944, and 0.973 for INSET-R. INSET-R also exhibits less variation across probability vectors, with standard deviations of 0.021–0.035, compared with 0.064–0.127 for EquiVSet-R. These results support the use of g_θ to provide useful local gradient directions for the multilinear objective on these synthetic inputs, even without explicit gradient supervision.

F.2 Ablation Study

We examine the effects of multi-head self-attention, Langevin noise, and temperature learning for both EquiVSet and INSET. We augment each baseline with multi-head self-attention, denoted by "-MHA", and remove Langevin noise from our R variants, denoted by "w/o L". In the full models, the temperature T is a learnable parameter optimized jointly with the network weights. We additionally evaluate $\text{R}_{(T=1)}$, which fixes $T = 1$ throughout training. Table 8 reports the results. Both full R variants achieve higher mean scores than their original and MHA-augmented baselines on all six datasets. Adding self-attention alone therefore does not reproduce the gains of our framework. Removing Langevin noise yields mixed results: the deterministic variants obtain the highest mean scores on Toys in both model families and on Furniture for INSET, whereas the full variants perform better on Carseats and Diaper. These observations indicate that the benefit of Langevin noise is

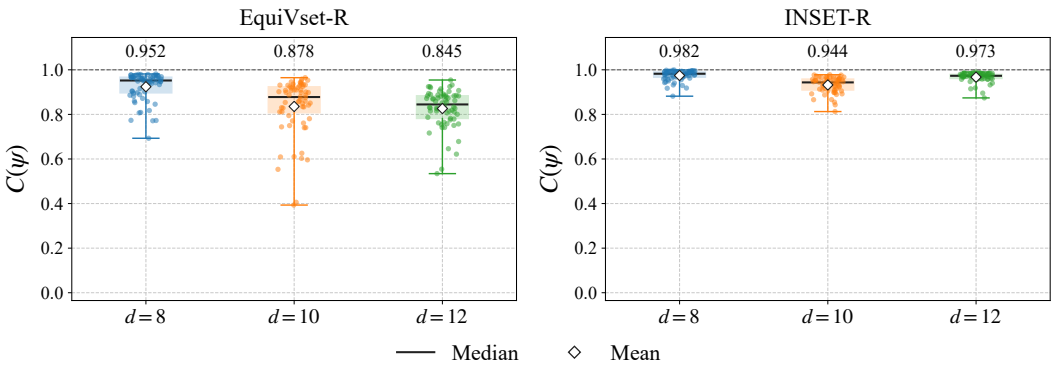


Figure 7: Gradient cosine similarity between the learned surrogate g_θ and its exact induced multilinear extension $f_{\mathrm{mt}}^{F_\theta}$. Each group uses 64 uniformly sampled probability vectors for one held-out set of size d, with one trained model per architecture. Dots denote individual samples; boxes and whiskers show the interquartile and full ranges, respectively. Black bars and labels indicate medians, and white diamonds indicate means. The dashed line marks perfect alignment.

Table 8: Ablation results on six Amazon datasets. Values are test Jaccard scores (%), reported as mean ± standard deviation. Bold indicates the highest mean within each model family.

Variants	Toys	Furniture	Carseats	Diaper	Safety	Media
EquiVSet	68.0 ± 2.0	17.2 ± 0.9	21.0 ± 1.0	83.0 ± 1.0	25.0 ± 3.0	57.0 ± 1.0
EquiVSet-MHA	75.7 ± 1.4	15.1 ± 1.8	20.6 ± 3.2	88.5 ± 0.7	13.9 ± 0.6	62.9 ± 2.8
EquiVSet-R$_{(T=1)}$	73.1 ± 11.3	35.3 ± 1.3	23.5 ± 2.3	89.1 ± 0.9	**37.9 ± 5.4**	**67.2 ± 1.1**
EquiVSet-R$_{(\text{w/o L})}$	**78.7 ± 0.7**	33.8 ± 4.8	26.5 ± 2.4	89.8 ± 1.0	37.6 ± 5.3	64.6 ± 4.0
EquiVSet-R	77.9 ± 1.0	**35.9 ± 1.3**	**26.9 ± 1.0**	**90.4 ± 0.6**	32.1 ± 9.4	67.0 ± 1.0
INSET	76.9 ± 0.5	16.9 ± 5.0	23.1 ± 3.4	88.0 ± 0.7	23.8 ± 1.5	62.0 ± 2.3
INSET-MHA	77.8 ± 1.3	17.4 ± 1.5	23.5 ± 2.0	90.2 ± 1.7	14.0 ± 0.5	65.5 ± 2.4
INSET-R$_{(T=1)}$	79.1 ± 1.4	34.7 ± 4.5	28.2 ± 3.7	90.0 ± 0.7	35.3 ± 6.4	**68.7 ± 1.1**
INSET-R$_{(\text{w/o L})}$	**79.3 ± 0.7**	**36.9 ± 1.0**	29.7 ± 2.8	89.9 ± 0.5	37.4 ± 5.0	67.8 ± 1.3
INSET-R	79.0 ± 0.6	36.5 ± 1.5	**30.2 ± 3.9**	**90.4 ± 0.5**	**38.3 ± 4.0**	68.1 ± 1.1

dataset-dependent, with deterministic updates providing a competitive alternative in several cases. The variants with learnable temperature achieve higher mean scores than their fixed-temperature counterparts on four of the six datasets in each model family. On Toys, EquiVSet-R with learnable temperature also exhibits lower variability, with a standard deviation of 1.0 compared with 11.3 when $T = 1$. Nevertheless, fixing $T = 1$ remains competitive and yields the highest mean scores on Media in both families and on Safety for EquiVSet. These results suggest that learning the temperature can improve performance and stability, although its benefits vary across datasets.

F.3 Comparison with iDiffMF

Another closely related work is iDiffMF (Özcan et al., 2025). iDiffMF uses implicit differentiation through the mean-field fixed point to reduce the cost of backpropagation, while retaining Monte Carlo estimates of the multilinear gradient. Our approach learns a differentiable surrogate g_θ whose gradients directly guide variational optimization, replacing these sampling-based gradient estimates. Table 9 compares performance on the 16 datasets covered in their original paper. We use the DiffMF, iDiffMF$_2$, and iDiffMF$_*$ results reported in the original paper, together with our results. The entries follow the experimental protocols and reporting conventions of their respective studies. Both EquiVSet-R and INSET-R achieve higher reported mean MJC than all three baselines on 15 of the 16 datasets, with comparable results on Gaussian Mixture. The largest differences are observed on Two Moons and Furniture.

Table 9: Comparison with DiffMF and iDiffMF. Test MJC (%) is reported as mean ± standard deviation. DiffMF and iDiffMF results are quoted from Özcan et al. (2025). Bold indicates the highest reported mean in each row.

Dataset	DiffMF	iDiffMF$_2$	iDiffMF$_*$	EquiVSet-R	INSET-R
CelebA	54.42 ± 0.70	56.30 ± 0.58	56.55 ± 0.49	58.8 ± 0.6	$\mathbf{59.2 \pm 0.2}$
Gaussian Mixture	90.96 ± 0.05	90.95 ± 0.18	$\mathbf{91.03 \pm 0.09}$	91.0 ± 0.1	90.9 ± 0.1
Two Moons	58.45 ± 0.15	58.48 ± 0.15	58.97 ± 0.04	99.2 ± 0.3	$\mathbf{99.7 \pm 0.1}$
Apparel	70.60 ± 1.35	76.13 ± 4.65	73.80 ± 5.71	82.2 ± 0.9	$\mathbf{83.3 \pm 1.3}$
Bath	71.87 ± 0.27	77.68 ± 0.98	76.43 ± 0.81	84.7 ± 0.9	$\mathbf{86.7 \pm 0.6}$
Bedding	67.66 ± 0.39	77.88 ± 0.80	76.94 ± 1.05	86.8 ± 1.2	$\mathbf{87.6 \pm 1.7}$
Carseats	20.15 ± 0.65	21.94 ± 1.43	22.42 ± 1.04	26.9 ± 1.0	$\mathbf{30.2 \pm 3.9}$
Diaper	81.74 ± 1.18	82.76 ± 0.62	82.07 ± 0.90	$\mathbf{90.4 \pm 0.6}$	$\mathbf{90.4 \pm 0.5}$
Feeding	77.44 ± 0.46	81.93 ± 1.00	81.52 ± 1.84	87.2 ± 1.4	$\mathbf{88.6 \pm 0.5}$
Furniture	16.84 ± 0.05	19.93 ± 2.68	18.69 ± 0.93	35.9 ± 1.3	$\mathbf{36.5 \pm 1.5}$
Gear	66.06 ± 2.86	73.90 ± 10.29	73.57 ± 6.74	78.7 ± 2.6	$\mathbf{82.1 \pm 1.1}$
Health	59.64 ± 0.81	72.55 ± 1.10	72.32 ± 1.03	81.4 ± 0.6	$\mathbf{82.2 \pm 0.2}$
Media	51.32 ± 1.11	56.39 ± 2.68	55.58 ± 1.75	67.0 ± 1.0	$\mathbf{68.1 \pm 1.1}$
Safety	24.66 ± 5.56	26.02 ± 1.68	25.38 ± 1.88	32.1 ± 9.4	$\mathbf{38.3 \pm 4.0}$
Toys	64.39 ± 1.64	68.53 ± 1.35	68.91 ± 1.00	77.9 ± 1.0	$\mathbf{79.0 \pm 0.6}$
BindingDB	73.22 ± 1.08	76.83 ± 0.50	77.48 ± 1.04	$\mathbf{78.3 \pm 0.8}$	77.8 ± 1.2

F.4 Computational Efficiency

Table 10 compares the inference time and memory usage of different methods under increasing sample and set sizes. Across all settings, replacing Monte Carlo sampling with the proposed learned relaxation consistently improves computational efficiency. For EquiVSet, the relaxation variant achieves up to $2.6\times$ faster inference and reduces memory usage by up to $1.6\times$. The improvement is more pronounced for INSET, where INSET-R obtains up to $3.8\times$ speedup and $2.5\times$ lower memory usage. These gains become especially important as N and $|V|$ grow, suggesting that the proposed relaxation effectively alleviates the computational and memory overhead caused by sampling-based gradient estimation.

Table 10: Computational efficiency on the Two Moons dataset under varying sample and set sizes. The average inference time per epoch (ms) and peak memory usage (MB) are reported.

Method (#Params)	N=1000, V=100		N=2000, V=200		N=3000, V=300	
	Time	Memory	Time	Memory	Time	Memory
EquiVSet (0.76M)	178	936	686	1942	1575	3043
EquiVSet-R (0.26M)	**68**	**569**	**370**	**1352**	**1021**	**2468**
INSET (1.14M)	479	2554	1892	5368	4420	8505
INSET-R (0.52M)	**125**	**1021**	**732**	**2483**	**2025**	**4396**

F.5 Comparison with Low-Variance Gradient Estimators

Table 11 compares our method with low-variance gradient estimators by increasing the number of Monte Carlo samples. Although using more samples slightly improves performance, it also substantially increases inference time. In contrast, our method achieves the best performance for both EquiVSet and INSET while requiring significantly less inference time. This suggests that directly learning the relaxation provides a more favorable performance-efficiency trade-off than reducing gradient variance through additional sampling.

Table 11: Comparison with low-variance gradient estimators on the Toys dataset. Performance (%) and average inference time (ms) are reported.

Number of Samples		10	20	30	40	50	Ours
EquiVSet	MJC	71.6±0.7	74.0±0.6	74.2±0.5	75.3±0.2	75.9±0.6	**77.9±1.0**
	Time	98	160	223	290	360	**34**
INSET	MJC	76.8±1.0	77.2±0.5	76.9±1.3	76.9±0.6	77.2±0.4	**79.0±0.6**
	Time	165	300	423	547	669	**54**

F.6 Sensitivity Analysis of Hyperparameters

Impact of the Gradient Ascent Step Size α. In ReSet, the step size α for gradient ascent in Eq. (22) serves as a critical hyperparameter governing both convergence velocity and numerical stability, directly impacting the final performance. An excessively large step size may lead to divergence, while a disproportionately small one risks trapping the optimization in suboptimal local minima and unnecessarily prolonging inference time. We evaluate the performance of our two proposed OS Oracle variants across different datasets under various α configurations, with results illustrated in Figures 8 and 9. Furthermore, in our practical implementation, we set α as a learnable parameter to achieve an adaptive step size. This approach alleviates the burden of manual hyperparameter tuning for unseen datasets and enables the model to dynamically calibrate its optimization dynamics for diverse tasks. The performance of this self-adaptive configuration is denoted as "Adaptive" in the corresponding figures. The empirical results indicate that a fixed step size near 0.5 generally yields favorable outcomes. Notably, the adaptive strategy consistently outperforms the fixed-parameter configurations in most scenarios, demonstrating its superior capability in navigating the optimization landscape.

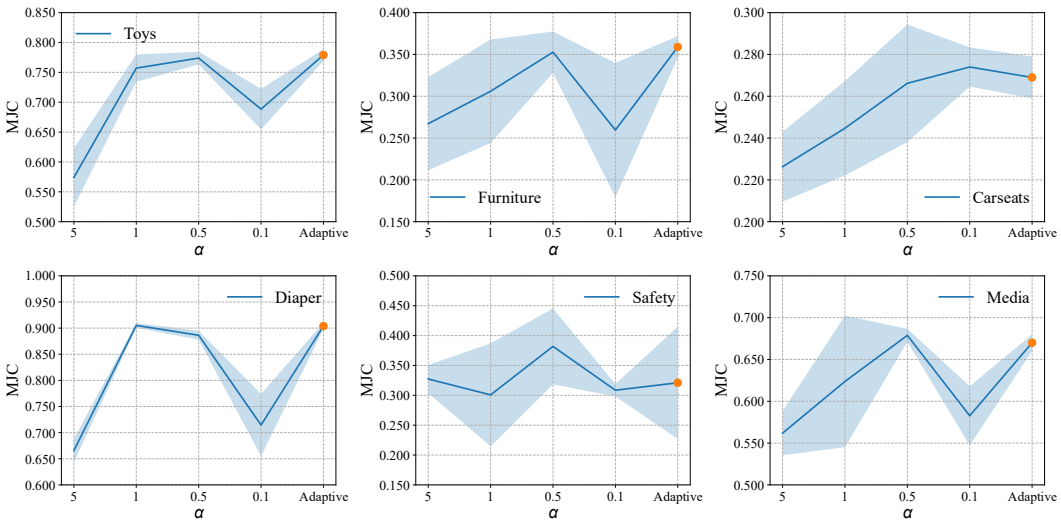


Figure 8: Sensitivity analysis of EquiVSet-R performance under different values of α, where "Adaptive" denotes the variant with α as a learnable parameter.

Impact of the Number of Gradient Ascent Steps K. In ReSet, the number of gradient ascent steps K determines the extent of iterative refinement in Eq. (22) and directly affects the computational cost. We evaluate our two proposed OS oracle variants on the six Amazon datasets with $K \in \{0, 1, 2, 3, 4, 5\}$, keeping all other settings fixed and repeating each configuration over five random seeds. The results are illustrated in Figures 10 and 11. When $K = 0$, predictions rely solely on the recognition network, yielding performance close to the DeepSet baseline on Toys, Diaper, and Media. Notably, a single refinement step guided by $\nabla_{\psi} g_{\theta}$ increases mean MJC by approximately

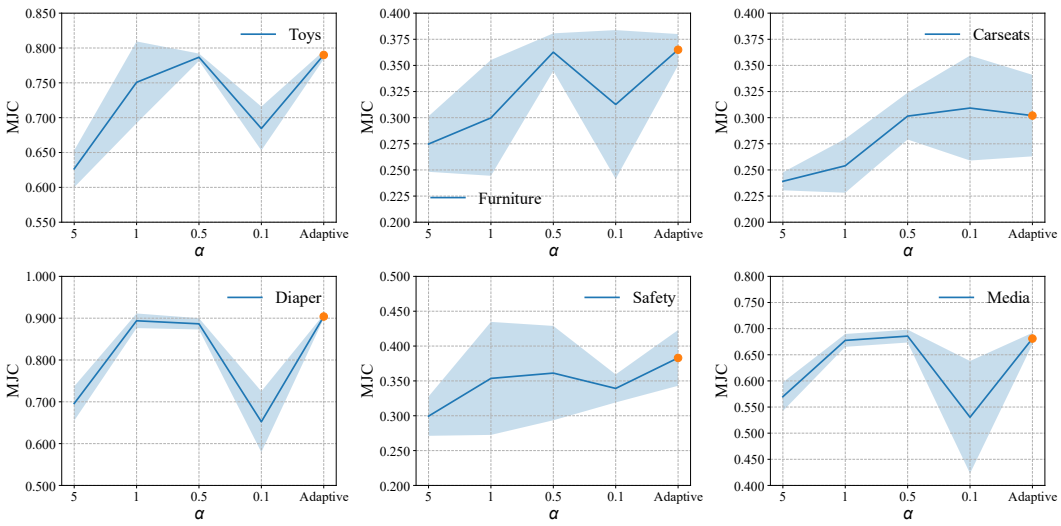


Figure 9: Sensitivity analysis of INSET-R performance under different values of α, where "Adaptive" denotes the variant with α as a learnable parameter.

24–44 percentage points on these datasets. These one-step gains exceed the improvements of both EquiVSet and INSET over DeepSet reported in Table 2, highlighting the benefit of refining the initial predictions. The empirical results indicate that $K = 1$ achieves the highest mean MJC in eight of the twelve settings, while the remaining settings favor $K = 2$ or $K = 3$. Increasing K further provides no consistent improvement and can lead to lower performance or greater variability across seeds.[13] Table 12 further reports the per-epoch runtime (ms/epoch), which increases approximately linearly with K. At $K = 5$, the runtime is 3.2–4.6 times that at $K = 1$ across the evaluated settings. In our practical implementation, we follow the $K = 1$ setting of EquiVSet (Ou et al., 2022) and INSET (Xie et al., 2024a) for a fair comparison. This choice also provides a favorable trade-off between predictive performance and computational cost.

Table 12: Per-epoch runtime (ms/epoch) under different numbers of gradient ascent steps K on six Amazon datasets.

Number of Steps K		0	1	2	3	4	5
EquiVSet-R	Toys	21.76	61.46	108.07	160.98	188.88	239.91
	Furniture	1.90	8.94	12.37	17.63	23.20	28.53
	Carseats	5.24	17.43	26.47	37.37	48.60	59.86
	Diaper	48.27	127.68	235.56	337.22	442.61	547.42
	Safety	2.11	8.63	12.06	17.43	22.71	27.94
	Media	16.51	35.77	57.20	82.78	108.44	134.37
INSET-R	Toys	27.39	94.22	186.31	258.39	340.35	423.19
	Furniture	2.12	11.76	21.02	30.75	40.25	49.84
	Carseats	5.20	24.40	44.18	64.58	85.42	106.11
	Diaper	48.51	217.94	413.51	610.61	804.38	1001.57
	Safety	2.27	11.20	20.86	30.30	39.83	49.31
	Media	15.25	53.91	101.21	149.41	197.64	247.89

Evolution of the Learnable Temperature T. In ReSet, the temperature T in Eq. (19) controls the balance between expected energy and entropy. In our implementation, T is treated as a learnable

[13] Theorem 3.1 analyzes optimization of a fixed surrogate; here, the model parameters are learned jointly with the K unrolled updates.

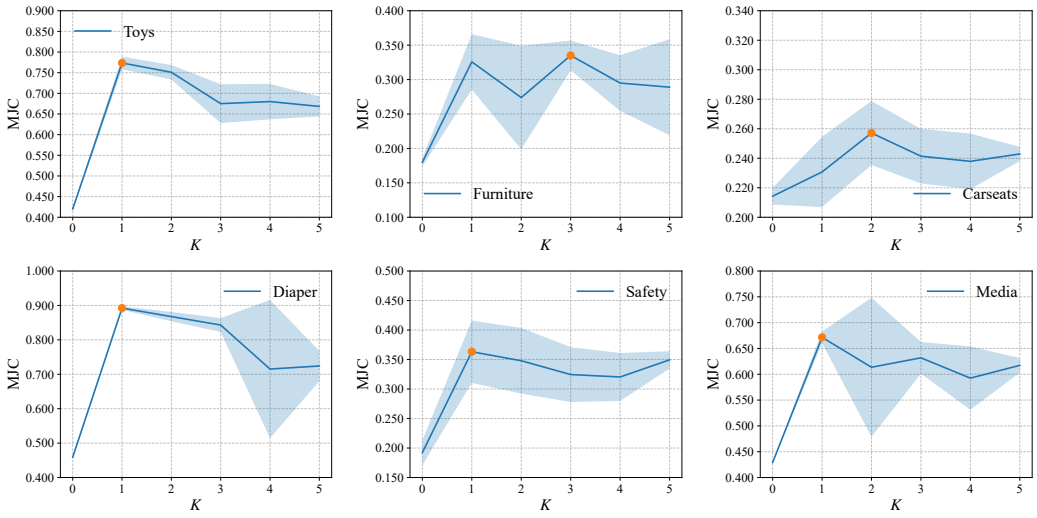


Figure 10: Sensitivity analysis of EquiVSet-R performance under different values of K. Curves and shaded regions show the mean MJC and one standard deviation over five random seeds. Orange markers indicate the highest mean in each dataset.

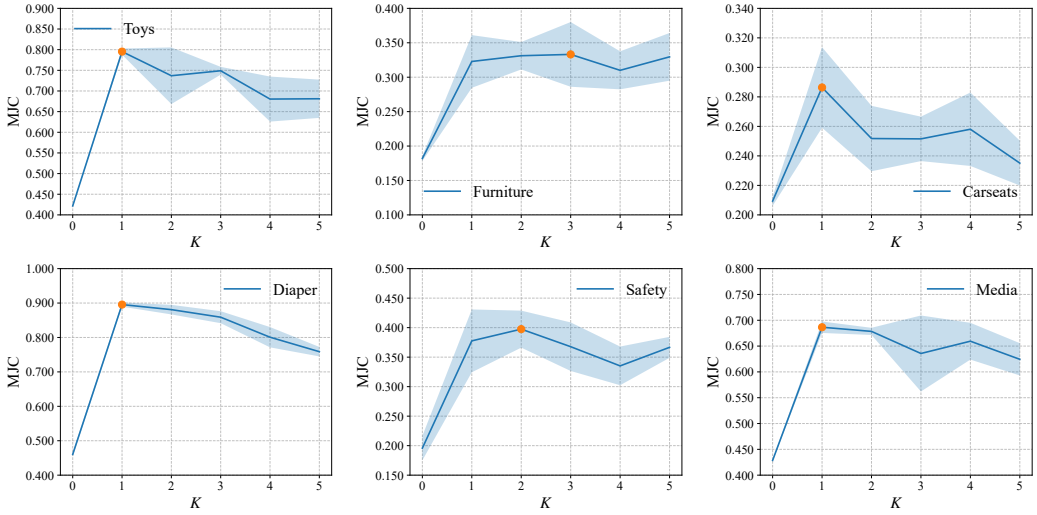


Figure 11: Sensitivity analysis of INSET-R performance under different values of K. Curves and shaded regions show the mean MJC and one standard deviation over five random seeds. Orange markers indicate the highest mean in each dataset.

parameter and optimized jointly with the network weights. We examine the evolution of the learned temperature during training for our two proposed OS oracle variants on Toys, Carseats, and Media, as shown in Figure 12. For EquiVSet-R, the temperature decreases slightly at the beginning of training across all three datasets. For INSET-R, it decreases throughout training on all three datasets. Overall, the learned temperature remains close to its initialization while exhibiting distinct trends across datasets and model variants. These modest adjustments illustrate how temperature learning fine-tunes the balance between expected energy and entropy during training.

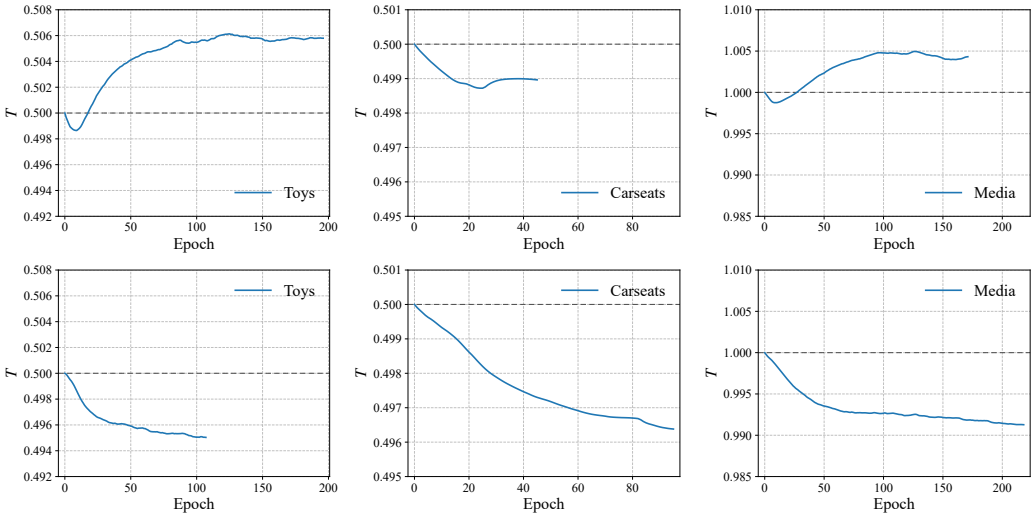


Figure 12: Evolution of the learnable temperature T during training. The top and bottom rows correspond to EquiVSet-R and INSET-R, respectively; Dashed horizontal lines indicate the initial temperature.

F.7 Set Size Transferability Analysis

Following Ou et al. (2022), we experiment to understand the pattern of set size transfer ability. In this experiment, we train the models using fixed sizes of the ground set but test the trained model on different sizes. We fix the size of OS oracle S^* to be 10, and train the model with ground set V of size 100. After training, we test it using varying sizes of ground set in the range of $\{200, 400, 600, 800, 1000\}$. The experiments with two OS Oracle variants are conducted on the Two-Moons and Gaussian-Mixture datasets, respectively, with the results shown in Figures 13 and 14. As illustrated in the results, both proposed OS Oracle variants exhibit an observable degree of set-size transferability. When trained on a fixed ground set size and evaluated on increasingly larger sets, the models experience a gradual performance degradation, which aligns with theoretical expectations regarding the increased complexity of the search space. This decline is relatively marginal on the Gaussian Mixture dataset but becomes more pronounced on the Two Moons dataset. We attribute this discrepancy to the intrinsic task complexity; the non-linear, intertwined geometry of the Two-Moons distribution likely poses a greater challenge for zero-shot size transfer compared to the more clustered Gaussian-Mixture distribution.

G Related Work

G.1 Set Function Learning

Many existing studies focus on learning an unknown function to predict the value of a given set (Wendler et al., 2019; 2021; De & Chakrabarti, 2022), commonly referred to as a function-value (FV) oracle. Zaheer et al. (2017) introduced the DeepSets architecture to model permutation-invariant and permutation-equivariant functions for set-valued prediction. Building on DeepSets, Lee et al. (2019) incorporated the Transformer architecture to capture relational dependencies among

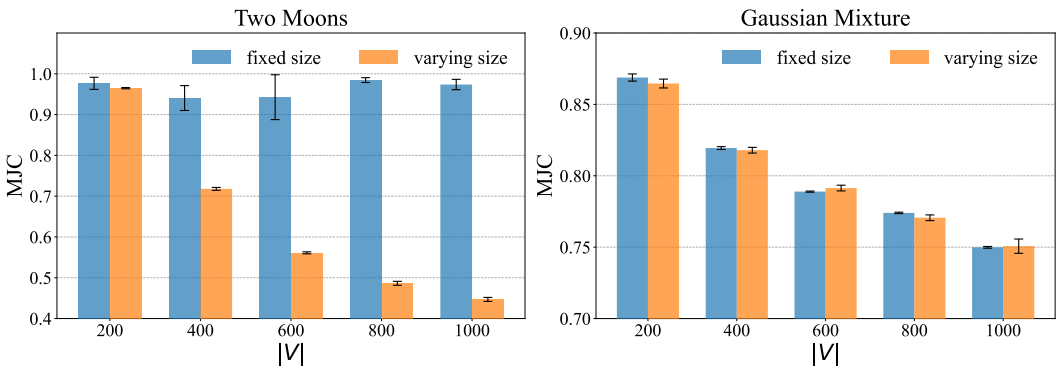


Figure 13: Synthetic results of EquiVSet-R under the set-size transferability setting are reported. The blue bars correspond to scenarios where the ground set size used at test time matches that employed during training, whereas the yellow bars indicate evaluations conducted with mismatched ground set sizes at inference.

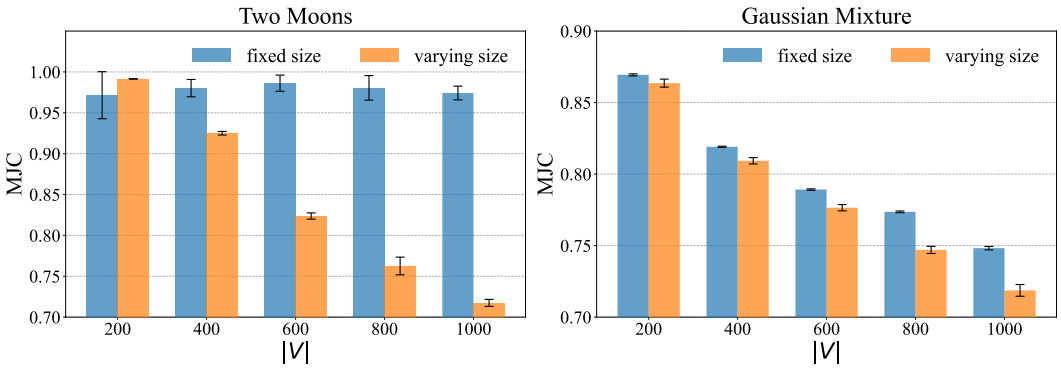


Figure 14: Synthetic results of INSET-R under the set-size transferability setting are reported. The blue bars correspond to scenarios where the ground set size used at test time matches that employed during training, whereas the yellow bars indicate evaluations conducted with mismatched ground set sizes at inference.

elements within a set. Additionally, several works have explored deep architectures for submodular set functions (Bilmes & Bai, 2017; Dolhansky & Bilmes, 2016; Balcan & Harvey, 2018). It is important to note that these methods typically require a substantial amount of supervision, which poses challenges for their practical deployment.

Optimal Subset Oracle. Ou et al. (2022) propose to learn latent set functions using optimal subsets rather than explicit function values as supervision, formulating the problem under an energy-based modeling framework with parameters updated via variational inference. Building upon this framework, Xie et al. (2024a;b) further incorporate contextual information from the superset and hierarchical attention mechanisms. Özcan et al. (2025) further improve the fixed-point iterations in variational inference through implicit differentiation. However, these methods require unrolled Monte Carlo sampling for gradient estimation during iterative optimization, rendering the training process inefficient.

G.2 Energy-based Modeling

Energy-based learning (LeCun et al., 2006) provides a classical framework for modeling underlying data distributions, in which model parameters are learned via principled techniques such as contrastive divergence (Carreira-Perpinan & Hinton, 2005; Tieleman, 2008; Du & Mordatch, 2019) and

score matching (Hyvärinen, 2005; Vincent, 2011; Song et al., 2020). In the framework of Ou et al. (2022), energy-based model is employed to parameterize the set mass function. Building upon this formulation, we reinterpret the associated variational inference procedure from the perspective of free energy estimation (He et al., 2025) and energy minimization (Gladstone et al., 2025). This viewpoint aligns closely with a broad line of work that casts learning and inference as energy minimization problems (Du et al., 2022; 2024; Wang & Du, 2025).

G.3 Submodular Function Learning

In practical applications, learning set functions typically necessitates the incorporation of structural priors. The underlying objectives are often assumed to be submodular, *i.e.*, exhibiting the diminishing returns property. Unlike prior work that predominantly relies on function value oracles for parameter estimation (Dolhansky & Bilmes, 2016; Bilmes & Bai, 2017; Djolonga & Krause, 2017; Kothawade et al., 2020; De & Chakrabarti, 2022; Bhatt et al., 2024), this work learns latent set functions characterized by more general weak submodular properties through a more practical and computationally efficient optimal subset oracle.