DEEP ASSORTMENT OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Assortment optimization involves selecting a subset of items that maximizes expected reward under a choice model, with applications in online platforms and revenue management systems. We propose a differentiable, stochastic optimization framework that applies the Lovász extension to embed the discrete objective into the unit hypercube. The resulting continuous problem is solved efficiently with stochastic gradient descent and converted back to a near-optimal discrete assortment via a rounding scheme. Our method is scalable, model-agnostic, offers theoretical guarantees, and naturally extends to capacity-constrained settings.

1 INTRODUCTION

The goal of assortment optimization is to determine an optimal subset of items to present to users, thereby maximizing the expected revenue, engagement, or welfare. It has become an important problem in modern computational settings (Désir et al., 2016; Udwani, 2023; Berbeglia & Joret, 2020; Agrawal et al., 2019; Chen et al., 2020), with wide-ranging applications in online retail platforms, digital advertising, and recommendation systems (Qi et al., 2020; Heger & Klein, 2024). This problem essentially involves the algorithmic design for discrete and combinatorial optimization, where a discrete choice model (Train, 2009) is commonly used to capture the user preference for better alignment with human values, for example, the customized assortment for different consumer types. As digital platforms increasingly rely on data-driven decision-making, effective assortment optimization is crucial for both improving user experience and driving business outcomes.

Despite its importance, assortment optimization poses significant computational challenges. At its core, the problem is discrete in nature: selecting a subset of items from a potentially massive choice set. This discreteness renders the optimization problem non-differentiable, which prevents the direct use of gradient-based optimization techniques that have proven effective in continuous settings. Designing methods that are both computationally efficient and broadly applicable remains an open challenge.

Existing approaches have two celebrated streams of research. The first focuses on specialized algorithms tailored to particular choice models. The examples are not limited to Multinomial Logit (MNL) (Rusmevichientong et al., 2014), Nested Logit (Gallego & Topaloglu, 2014), Mallows model (Désir et al., 2016), and a neural-network choice model that accounts for assortment effects Wang et al. (2023). These methods either encounter the curse of dimensionality with scalability issues or do not come with performance guarantees. Moreover, they cannot be easily portable to other newly proposed choice models like Akchen & Mitrofanov (2025); Yang et al. (2025).

A second stream of work like Udwani (2023) casts assortment optimization as a subset selection problem. This perspective has motivated the use of approximation and heuristic algorithms, which offer computational tractability for fixed-size k-subset selection. However, such exact k-subset selection algorithms have some gaps with the assortment problem in non-capacitated or capacitated settings, where k is upper bounded by the cardinality of the whole choice set or the capacity, respectively. Then the subset selection approaches require exhaustive computation across all possible subset sizes, and thus are not directly applicable to very large item choice sets for non-capacitated assortment optimization. This limits its applicability in real-world platforms with vast item catalogs.

In this paper, we propose a new approach that bridges the gap between discrete assortment optimization and continuous gradient-based methods. We introduce a differentiable, stochastic optimization framework that is model-agnostic and thus enables applications to diverse choice models. Our contributions are threefold. First, we leverage the Lovász extension that embeds the discrete objective onto the continuous unit hypercube. Despite the non-submodularity of the objective which makes the Lovász non-convex, along with chain rules, this embedding enables the use of unbiased stochastic (sub)gradient for the relaxed distributive objective. Second, we provide a scalable algorithmic imple-

mentation with an efficient generative rounding procedure for discrete assortment recovery. Third, we establish near-optimal theoretical guarantees on the quality of the returned assortment with mild assumptions and show that our method naturally extends to cardinality-constrained settings, which was validated via extensive experiments. Collectively, these advancements yield a novel, scalable framework for assortment optimization, free from restrictive parametric assumptions, and highly applicable to practical recommendation and decision-making contexts.

1.1 Literature Review

Assortment optimization with specific choice model: A large body of research has focused on developing efficient algorithms under particular parametric choice models. For instance, assortment optimization under the MNL model has been extensively studied, leading to an optimal revenue-order policy (Talluri & Van Ryzin, 2004) with its robustness being justified in Rusmevichientong & Topaloglu (2012). Extensions to variants such as nested logit models (Davis et al., 2014; Gallego & Topaloglu, 2014) and Markov Chain (MC) choice model without constraints (Blanchet et al., 2016) have also been proposed. However, assortment optimization becomes computationally intractable in many other common settings. Notably, even the unconstrained case of a mixture of just two MNL models (Bront et al., 2009) or the constrained problem under the MC model (Désir et al., 2020) is NP-hard. In such cases, the literature has largely resorted to heuristic methods specific to each choice model structure.

Assortment optimization with general choice model: Beyond model-specific approaches, another line of work seeks to design algorithms that apply under general or even arbitrary choice models, which is the setting our work addresses. Heuristic methods such as revenue-ordered assortments (Berbeglia & Joret, 2020) or local search strategies (Jagabathula, 2014) provide tractable solutions but has rationality assumptions or lack rigorous global convergence guarantees. Recent research has also drawn connections between assortment optimization and subset selection which closely connected with our proposed algorithm, and we will review the subset selection (and sampling) algorithms in the next paragraph. Complementing these efforts, emerging machine learning–based approaches directly optimize assortments (Li et al., 2025), demonstrating strong empirical scalability, though such methods generally lack theoretical guarantees regarding solution quality.

Subset Selection and Sampling: The subset selection or sampling approaches have three promising directions as summarized in (Wijk et al., 2025): score function estimator (Williams, 1992), pathwise gradient estimator (Bengio et al., 2013), and relaxed sampling (Xie & Ermon, 2019; Yamada et al., 2020). Our approach is most relevant to the third direction. These existing designs of approximation algorithms can cause estimation biases. A notable exception is the SFESS method (Wijk et al., 2025) for k-subset sampling, which achieves unbiased estimation without reliance on sampling. However, as mentioned in the previous introduction, solving the assortment problem without the cardinality constraint becomes computationally inefficient as the consideration set grows large. Notably, the submodular maximization for subset selection is also considered for enabling provable performance bounds (Udwani, 2023; Ito, 2019; Zhang et al., 2023), yet the submodularity assumption does not apply to the general revenue function (Udwani, 2023). We refer the readers to Section 4 for a more detailed comparison of our algorithm with the literature.

Compared to the above literature, our work contributes significantly to a general-purpose and near-optimal solution that can scale well to large consideration sets and flexibility across any given choice model. Note that there are literature working on online and dynamic assortment optimization (e.g., Agrawal et al. (2019); Chen et al. (2020); Li et al. (2024)), whereas this paper focuses on the offline setting with one horizon in the assortment planning.

2 Preliminaries

2.1 Assortment Optimization

The assortment optimization problem is: given a ground set of n items $V = \{1, \dots, n\}$, select a subset $S \subseteq V$ to maximize the total reward that depends on the user's choice behavior. Formally, we study

$$\max_{S \subseteq V} r(S), \text{ where } r(S) := \sum_{j \in S} r_j \cdot \mathbb{P}(j \mid S).$$

Here, $r(S)$ denotes the expected reward when presenting the assortment S, where $r_j \in [0, \bar{r}]$ is the marginal reward obtained if item j is chosen, and $\mathbb{P}(j \mid S)$ is the choice probability of item j given

that the assortment S is offered. The choice probabilities $\mathbb{P}(j \mid S)$ depend on an underlying discrete choice model. Prominent examples include:

[leftmargin=11pt]*Basic Attraction Model (BAM).* Each product i has attraction value $v_j > 0$ and the no-purchase option has $v_0 > 0$. The choice probability of item $j \in S$ is

$$\mathbb{P}(i \mid S) = \frac{v_j}{v_0 + \sum_{k \in S} v_k}.$$

A special case of BAM is *Multinomial Logit (MNL)*, when $v_j = e^{u_j}$ for utilities $u_j \in \mathbb{R}$. *Nested Logit (NL).* Partition products into G nests $\{S_g\}_{g=1}^{G}$, each with a dissimilarity parameter $\gamma_g \in [0, 1]$. The choice probability of item j in nest g is given by

$$\mathbb{P}(j \mid S) = \frac{v_{gj}}{V_g} \frac{V_g^{\gamma_g}}{v_0 + \sum_{h=1}^{G} V_h^{\gamma_h}}$$

with $V_g := \sum_{k \in S_g} v_{gk}$. *Mixed Multinomial Logit (MMNL).* There are M customer types, each with a mixture probability α_m. Customers of type m associate the attraction value v_j^m. The choice probability of item $j \in S$ is

$$\mathbb{P}(j \mid S) = \sum_{m=1}^{M} \alpha_m \frac{v_j^m}{v_0^m + \sum_{k \in S} v_k^m}.$$

To align with the minimization convention in optimization theory, we define $f(S) := -r(S)$, so that the assortment optimization problem can be equivalently expressed as

$$\min_{S \subseteq V} f(S). \tag{1}$$

In many applications, the assortment is subject to an additional *capacity constraint*, which limits the number of items that can be offered. In this case, the optimization problem becomes

$$\min_{S \subseteq V,\ |S| \leq K} f(S), \tag{2}$$

where K denotes the maximum allowable assortment size. e remark that a related but different problem is K-subset selection, where the capacity constraint is specified as equality.

2.2 Lovász Extension

The minimization problem (1) is discrete in nature. In polyhedral theory, a powerful approach to extending discrete set functions to a continuous domain is through the Lovász extension (Lovász, 1983). While it has been primarily studied in the context of submodular functions, its definition applies more broadly and is not restricted to them. For any set function $f : 2^V \to \mathbb{R}$, its *Lovász extension* $\phi : [0, 1]^n \to \mathbb{R}$ admits two equivalent constructions. The first, known as the sorting formula, is constructed as follows. For any vector $x \in [0, 1]^n$, let π^x be a permutation that sorts the components in non-increasing order $x_{\pi^x(1)} \geq x_{\pi^x(2)} \geq \cdots \geq x_{\pi^x(n)} \geq x_{\pi^x(n+1)} := 0$. Define a chain of nested sets

$$S_k(x) := \{\pi^x(1), \ldots, \pi^x(k)\} \quad \text{for } k = 1, \ldots, n, \quad \text{with } S_0(x) = \varnothing. \tag{3}$$

The Lovász extension is given by

$$\phi(x) \triangleq \sum_{k=1}^{n} \left(x_{\pi^x(k)} - x_{\pi^x(k+1)}\right) f\left(S_k(x)\right). \tag{4}$$

An equivalent construction is provided by the threshold integral representation. For threshold $t \in [0, 1]$, define the level set $S_t(x) := \{i \in V : x_i \geq t\}$. The Lovász extension can then be expressed as:

$$\phi(x) \triangleq \int_0^1 f\left(S_t(x)\right) dt. \tag{5}$$

Both constructions will be useful in our framework.

A key property linking discrete and continuous optimization is the equivalence:

$$\min_{S \subseteq V} f(S) = \min_{x \in [0,1]^n} \phi(x). \tag{6}$$

The equivalence (6), established by Lovász (1983) and elaborated in Bach (2013), follows from two key observations. First, for any set $S \subseteq V$ with characteristic vector $\mathbf{1}_S$, we have $\phi\left(\mathbf{1}_S\right) = f(S)$, ensuring consistency at the vertices. Second, for any $x \in [0, 1]^n$, the value $\phi(x)$ constitutes a convex combination of the function values $\{f\left(S_k(x)\right)\}_{k=1}^{n}$ by (5), guaranteeing that the continuous minimum cannot fall below the discrete minimum. Note that the Lovász extension ϕ is convex if and only if the set function f is submodular. However, the cost function f is mostly not submod-

ular. Therefore, (6) is generally a non-convex optimization. We will tackle this challenge through distributional reparameterization in Section 4.1.

The Lovász extension ϕ possesses the following properties:

(1) (Piecewise linearity) For each permutation π, define the strict-order region:
$$\mathcal{C}_\pi := \left\{x \in [0,1]^n : x_{\pi(1)} > x_{\pi(2)} > \cdots > x_{\pi(n)}\right\}.$$
On each $\mathcal{C}_\pi$, ϕ is affine, and for all $x \in \mathcal{C}_\pi$, we have $\nabla\phi(x) = g^\pi$.

(2) (Lipschitz continuity) The function ϕ is Lipschitz continuous on $[0,1]^n$ with respect to the ℓ_1-norm. Specifically, defining
$$G := \max_{S\subseteq V, j\notin S} |f(S\cup\{j\}) - f(S)|,$$
we have for all $x, y \in [0,1]^n$:
$$|\phi(x) - \phi(y)| \le G\|x - y\|_1.$$

3 Clarke Subdifferential and Stochastic Subgradient

In general, the Lovász extension is non-convex. In the special case where it is convex, its stochastic subgradient has been well-characterized in Ito (2019). In the non-convex setting, Lemma 2.2 shows that the extension is piecewise linear and Lipschitz continuous. This allows us to define its Clarke subdifferential and construct an unbiased stochastic subgradient estimator.

By Rademacher's theorem, any locally Lipschitz function is differentiable almost everywhere. This property allows us to define a generalized notion of subdifferential suitable for this class of functions.

[Clarke Subdifferential] For a locally Lipschitz function $h : \mathbb{R}^n \to \mathbb{R}$, the Clarke subdifferential at x, denoted $\partial^\circ h(x)$, is the convex hull of all possible limit points of gradients from nearby differentiable points:
$$\partial^\circ h(x) := \operatorname{conv}\left\{\lim_{j\to\infty} \nabla h(x^j) : x^j \to x, \text{ and } h \text{ is differentiable at } x^j\right\}.$$

where conv denotes the convex hull.

For any $x \in [0,1]^n$ and any permutation π that satisfies $x_{\pi(1)} \ge x_{\pi(2)} \ge \cdots \ge x_{\pi(n)}$, we define the vector $g^\pi(x) \in \mathbb{R}^n$ by its (permutated) components:
$$g^\pi_{\pi(k)}(x) := f\left(S_k(x)\right) - f\left(S_{k-1}(x)\right), \quad k = 1, \ldots, n,$$
where the chain of sets $S_k(x)$ defined in the sorting formula of the Lovász extension (see (4)). This vector $g^\pi(x)$ represents the discrete differences of f along the chain induced by π.

The following proposition provides an explicit characterization of the Clarke subdifferential of ϕ.

[Clarke subdifferential] For every $x \in [0,1]^n$, the Clarke subdifferential of ϕ is given by:
$$\partial^\circ\phi(x) = \operatorname{conv}\left\{g^\pi : \pi \in \Pi(x)\right\},$$
where $\Pi(x)$ is the set of permutations π such that $x_{\pi(1)} \ge x_{\pi(2)} \ge \cdots \ge x_{\pi(n)}$.

When the number of items n is large, the following construction provides a stochastic estimate of a subgradient.

[Stochastic Subgradient] Fix $x \in [0,1]^n$. First, sample a permutation $\pi \in \Pi(x)$ by breaking any ties in the components of x randomly. Then, sample a rank $K \in \{1, \ldots, n\}$ with probability $1/n$. Let $S_k = \{\pi(1), \ldots, \pi(k)\}$ for $k = 0, \ldots, n$ ($S_0 = \varnothing$). The stochastic subgradient $\widehat{g}$ is defined as:
$$\widehat{g} := n\left(f(S_K) - f(S_{K-1})\right)\mathbf{e}_{\pi(K)},$$
where $\mathbf{e}_{\pi(K)}$ is the standard basis vector for the element at rank K. This estimator requires only two function evaluations, $f(S_K)$ and $f(S_{K-1})$ *per sample*, making it highly efficient.

The unbiasedness of this stochastic subgradient is established as follows. [Unbiasedness] The stochastic subgradient $\widehat{g}$ is an unbiased estimator of the specific subgradient g^π corresponding to the sampled permutation π:
$$\mathbb{E}[\widehat{g} \mid \pi] = g^\pi.$$

Consequently, its unconditional expectation lies in the Clarke subdifferential:
$$\mathbb{E}[\widehat{g}] = \mathbb{E}[g^\pi] \in \operatorname{conv}\{g^\pi : \pi \in \Pi(x)\} = \partial^\circ\phi(x).$$
That is, $\widehat{g}$ is an unbiased estimator of a vector in $\partial^\circ\phi(x)$. This unbiasedness property enables the use of gradient-based methods for optimizing the Lovász extension ϕ in the next section.

4 OUR ALGORITHM

Building on the stochastic subgradient for the Lovász extension, we now describe our main algorithmic framework.

4.1 DISTRIBUTIONAL REFORMULATION AND REPARAMETERIZATION

We first reparameterize the problem using an implicit generative model. Let $\mathcal{P}([0,1]^n)$ denote the set of probability distributions on $[0,1]^n$. We cast problem (6) as an equivalent distributional optimization, which can be interpreted as a randomized assortment optimization (Wang et al., 2024):

$$\min_{p \in \mathcal{P}([0,1]^n)} \ \mathbb{E}_{x \sim p}\big[\phi(x)\big]. \tag{7}$$

To parameterize p, we use an implicit generative model: draw a random sample z from an easy-to-sample continuous distribution ν, which is chosen to be standard Gaussian $\mathcal{N}(0, I_d)$ in our paper, and set $x = g_\theta(z)$, where $g_\theta : \mathbb{R}^d \to [0,1]^n$ is a neural network parameterized by θ, then p is parameterized as the push-forward distribution $g_{\theta\#}\nu$. Our distributional reformulation (7) becomes

$$\min_{\theta} \ \mathbb{E}_{z \sim \nu}\big[\phi\big(g_\theta(z)\big)\big]. \tag{8}$$

Note that as long as g_θ is piecewise smooth (as is true for most neural network activations), the smoothness of the Gaussian distribution ν ensures the objective in (8) is differentiable with respect to θ, despite the Lovasz extension ϕ being piecewise linear. Using the chain rule, a stochastic gradient estimator is given by

$$J_\theta g_\theta(z)^\mathsf{T} \widehat{g}, \qquad z \sim \nu, \tag{9}$$

where $J_\theta g_\theta(z) \in \mathbb{R}^{n \times d}$ is the Jacobian matrix of the mapping $\theta \mapsto g_\theta(z)$.

The estimator given by (9) is an unbiased estimator of the gradient $\nabla_\theta \mathbb{E}_{z \sim \nu}\big[\phi\big(g_\theta(z)\big)\big]$.

4.2 ROUNDING THE SOLUTION

To obtain a discrete assortment from a continuous solution $x \in [0,1]^n$, we apply a rounding procedure based on the chain of sets $S_0(x) \subset S_1(x) \subset \cdots \subset S_n(x)$ induced by sorting the components of x, defined in (3). We choose an assortment by minimizing f over the chain:

$$\hat{S}(x) \in \operatorname*{arg\,min}_{k=0,\ldots,n} f(S_k(x)). \tag{10}$$

Our rounding procedure is deterministic, selecting the optimal assortment from a sorted chain of candidate sets. This contrasts with the approach in (Ito, 2019), which employs a stochastic rounding scheme that forms a set by applying a random threshold to the continuous solution. We will give a guarantee for this rounded solution in Section 5.2.

4.3 ALGORITHM

Motivated by the preceding analysis, we introduce Deep Assortment Optimization (DAO), a noisy stochastic gradient descent algorithm for assortment optimization.

Algorithm 1 Deep Assortment Optimization (DAO)

1: **Input:** Initialize $\theta_0 \sim \mu_0$, weight-decay factor α, noise scaling parameter $\tau > 0$, learning rate η, mini-batch size B
2: **while** not converge **do**
3: Sample a mini-batch $\{z^{(b)}\}_{b=1}^B \sim \nu$ and compute a stochastic gradient estimator G using (9)
4: Update the parameter

$$\theta \leftarrow (1-\alpha)\theta - \eta G + \sqrt{2\tau\eta}\xi, \quad \xi \sim \mathcal{N}(0, I)$$

5: **end while**
6: **Output** $\hat{S}(g_\theta(z))$ in (10), with $z \sim \nu$

Our algorithm is related to the relaxed sampling approach for subset selection (e.g., Xie & Ermon (2019); Ahmed et al. (2023); Paulus et al. (2020); Yamada et al. (2020)) in its use of the reparameterization trick to generate relaxed samples in the unit cube. However, our algorithm differs in two crucial aspects: (i) we employ a Lovász extension to define the loss, rather than a surrogate loss; and (ii) we round the continuous solution to a discrete one using a principled procedure. These distinctions allow us to establish provable performance guarantees (Section 5). Furthermore, our

algorithm accommodates both capacity inequality constraints and unconstrained problems, whereas the relaxed sampling approach is restricted to equality constraints, a limitation for many assortment optimization applications.

5 Performance Guarantees

5.1 Algorithmic Convergence

We now present a convergence guarantee for Algorithm 1. Although proving convergence for training neural networks is challenging, we here provide a sensible result utilizing the mean-field theory (Mei et al., 2018; Nitanda et al., 2022). In particular, we consider a specific neural network architecture to parameterize g_θ. The model takes the form of a one-hidden-layer network where the input layer is fed with a pre-trained feature extractor $h : \mathbb{R}^d \to \mathbb{R}^H$ (which is chosen to be a one-layer ReLU network in our experiment). The hidden layer is the trainable part and outputs

$$g_\theta(z) = \frac{1}{N}\sum_{i=1}^{N} \sigma(W^i h(z) + b^i),$$

where the trainable parameters are $\theta = \{(W^i, b^i)\}_{i=1}^N$, with $W^i \in \mathbb{R}^{n\times H}$ and $b^i \in \mathbb{R}^n$. As the number of neurons N in the hidden layer tends to infinity, the empirical distribution of the trainable parameters, $\frac{1}{N}\sum_{i=1}^N \delta_{(W^i,b^i)}$, converges to a probability distribution μ; thus, the generator is now parameterized by a distribution μ:

$$g_\mu(z) = \mathbb{E}_{\theta\sim\mu}[\sigma(Wh(z) + b)],$$

This allows us to define the objective as a functional F over the space of probability distributions:

$$F(\mu) := \mathbb{E}_{z\sim\nu}\big[\phi\big(g_\mu(z)\big)\big].$$

The evolution of the distribution μ under Algorithm 1 can be understood as a gradient flow on the Wasserstein space. Note that in contrast to prior works such as (Mei et al., 2018; Nitanda et al., 2022) that rely on flat convexity F, here we deal with a nonconvex setting, which makes our result differ.

In the mean-field limit, the distributional dynamics induced by Algorithm 1 is given by

$$\mu_{t+1} = \big(\mathrm{id} - \eta\nabla\tfrac{\delta F}{\delta\mu}[\mu_t]\big)_{\#}\mu_t * \mathcal{N}(0, 2\eta\tau I), \tag{11}$$

where $\frac{\delta F}{\delta\mu}[\mu_t]$ denotes the first variation of F evaluated at μ_t (see formal definition in Appendix A.5), where $_\#$ denotes the push-forward of measures and $*$ denotes the convolution of probability measures. To characterize the convergence properties, we introduce the following two definitions. [Fixed Point] A distribution μ^* is a fixed point (stationary point) of the dynamics (11) if it satisfies the self-consistency equation $\mu^* = p_{\mu^*}$, where p_μ is the proximal Gibbs distribution associated with μ:

$$p_\mu(\theta) \propto \exp\Big(-\frac{1}{\tau}\frac{\delta F}{\delta\mu}[\mu](\theta)\Big).$$

[Fisher Information] The Fisher information of a distribution q with respect to its associated Gibbs distribution p_q is defined as:

$$\mathcal{I}(q) := \int q(\theta)\Big\|\nabla\log\frac{q(\theta)}{p_q(\theta)}\Big\|^2 d\theta.$$

The Fisher information is non-negative and equals zero if and only if $q = p_q$. It measures the rate of change of the distribution along the flow, and thus can be viewed as a measure of stationarity.

Assume the following conditions hold:

[label=(),leftmargin=*]g_θ is Lipschitz continuous and bounded with respect to θ for each z. There exist constants $C > 0$ and $k \geq 0$ such that for all θ, the norm of the Jacobian $\|J_\theta g_\theta(z)\|$ is bounded by $C\left(1 + \|z\|^k\right)$ for almost every z. There exists a integrable function $h : \mathcal{Z} \to \mathbb{R}_+$ such that for all $\theta \in \Theta$ and for $\nu-$almost every z,

$$|g_\theta(z)| \leq h(z)\left(1 + \|\theta\|_2^2\right).$$

and

$$\|\nabla_\theta g_\theta(z)\| \leq h(z),$$

These conditions are standard in the analysis of mean-field models for neural networks. They ensure that the objective functional $F(\mu)$ is well-behaved and that all regularity conditions in Assumption 1 of Nitanda et al. (2022) are satisfied except for convexity, since $\phi(x)$ is not convex in our setting. To describe our convergence result, we borrow a notion of random stopping from non-convex stochastic optimization in Euclidean space (Ghadimi & Lan, 2013). The randomized time $\tilde{t}$ is obtained by

first sampling $\tau \in \{0, \ldots, T-1\}$ with probability proportional to the step size η_τ, then sampling $U \sim \mathrm{Unif}[0, \eta_\tau]$, and finally setting $\tilde{t} = s_\tau + U$, where $s_\tau = \sum_{k=0}^{\tau-1} \eta_k$.

Under Assumption 5.1, the sequence $\{\mu_t\}$ is tight, and any convergent subsequence has a limit distribution $\mu^\star$ that is a fixed point. In addition, for a total of T iterations with a constant step size $\eta = 1/\sqrt{T}$, the expected Fisher information at a randomly chosen iteration $\tilde{t}$ satisfies

$$\mathbb{E}[\mathcal{I}(\mu_{\tilde{t}})] = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right).$$

This rate matches the optimal rate for non-convex stochastic optimization in the Euclidean space (Ghadimi & Lan, 2013). Our result is different from Nitanda et al. (2022), as we do not have a flat convex functional $F(\mu)$ as they have.

5.2 Optimality Gap

A key property of this rounding scheme is that, by construction, the reward of the rounded assortment satisfies the key inequality that

$$r(\hat{S}(x)) \geq -\phi(x).$$

This follows from the Lovász extension's integral form: $\phi(x) = \int_0^1 f(S_t(x))dt$, and since $f = -r$, we have $-\phi(x) = \int_0^1 r(S_t(x))dt$. The rounding selects the best set in the chain $\{S_t(x)\}$, dominating this average.

In our framework, the continuous solution $x = g_\theta(z)$ is generated by a neural network trained to minimize $\Phi(\theta) = \mathbb{E}_{z\sim\mathcal{N}(0,I_d)}[\phi(g_\theta(z))]$. The universal approximation theorem implies that $\phi \circ g_\theta$ can approximate $\phi(x_{S^*})$ for an optimal assortment S^*, leading to approximation error $\varepsilon_{\mathrm{approx}}$. Training may not reach the global optimum due to computational limits, yielding optimization error $\varepsilon_{\mathrm{opt}}$.

Our theoretical guarantee explicitly accounts for both errors, as formalized in the following theorem. Let $g_\theta : \mathbb{R}^d \to [0,1]^n$ be a generative neural network model. Let $\widehat{\theta}$ be the parameters output by our training algorithm, which satisfy the following ε-optimality condition:

$$\Phi(\widehat{\theta}) \leq \min_\theta \Phi(\theta) + \varepsilon_{\mathrm{opt}}.$$

Let S^* be an optimal assortment with $r(S^*) = \max_{S\subseteq V} r(S)$ and characteristic vector x_{S^*}. Assume the expressive power of the generative model yields that

$$\min_\theta \Phi(\theta) \geq \phi(x_{S^*}) - \varepsilon_{\mathrm{approx}}.$$

Additionally, assume that there exists $M > 0$ such that $|r(S)| \leq M$ for all $S \subset V$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the random seed $z \sim \mathcal{N}(0, I_d)$, the continuous solution $x = g_{\widehat{\theta}}(z)$ satisfies:

$$\phi(x) \leq \phi(x_{S^*}) + \varepsilon_{\mathrm{opt}} + \varepsilon_{\mathrm{approx}} + \kappa(\delta),$$

and consequently, the rounded assortment $\hat{S}(x)$ achieves a reward:

$$r(\hat{S}(x)) \geq r(S^*) - \varepsilon_{\mathrm{opt}} - \varepsilon_{\mathrm{approx}} - \kappa(\delta),$$

where the concentration term is explicitly given by $\kappa(\delta) = M\sqrt{2\ln(1/\delta)}$.

This result provides a rigorous foundation for our approach. The error decomposition reveals how different components contribute to the final performance: the approximation error $\varepsilon_{\mathrm{approx}}$ can be reduced by using more expressive network architectures, the optimization error $\varepsilon_{\mathrm{opt}}$ can be controlled through better training algorithms, and the concentration term $\kappa(\delta)$ diminishes as we allow higher confidence levels (smaller δ).

Notably, the theorem does not require convexity of ϕ (which would imply submodularity of f) or any specific structure like submodularity of r, making it applicable to the broad class of assortment optimization problems.

6 Extension: Handling Capacity Constraints

In this section, we extend our framework to the constrained optimization (2) through an exact penalty method. By augmenting the Lovász extension $\phi(x)$ with a penalty term, we obtain:

$$\phi_\lambda(x) = \phi(x) + \lambda\rho_K^{\mathrm{L}}(x),$$

where $\rho_K^{\mathrm{L}}(x) = \sum_{i=K+1}^n x_{(i)}$ is the Lovász extension of the cardinality violation penalty $\rho_K(S) = (|S| - K)_+$, and $x_{(1)} \geq \cdots \geq x_{(n)}$ denote the sorted components of x. The penalty function

$\rho_K^{\mathrm{L}}(x)$ admits a geometric interpretation that connects constraint violation to distance from sparsity, established by the following lemma. [Best K-sparse truncation] Let $P_K(x)$ denote the projection that keeps the top K entries of x (by magnitude) and sets the rest to zero. Then,

$$\|x - P_K(x)\|_1 = \sum_{i=K+1}^{n} x_{(i)} = \rho_K^{\mathrm{L}}(x).$$

This lemma establishes that $\rho_K^{\mathrm{L}}(x)$ measures the ℓ_1-distance from x to the set of K-sparse vectors, encouraging solutions in $\{x : \|x\|_0 \leq K\}$. The following theorem confirms that the penalized objective exactly recovers the constrained problem for a sufficiently large penalty parameter. [Exact penalty for cardinality constraint] Let G be the Lipschitz constant of the Lovász extension ϕ as defined in Lemma 2.2. For every $\lambda \geq G$,

$$\min_{x \in [0,1]^n} \phi_\lambda(x) = \min_{\substack{S \subseteq V \\ |S| \leq K}} f(S).$$

Moreover, any minimizer x_λ of the penalized problem satisfies $\rho_K^{\mathrm{L}}(x_\lambda) = 0$, implying $\|x_\lambda\|_0 \leq K$. Consequently, every threshold set $S_t(x_\lambda)$ has size at most K, and at least one chain set $S_k(x_\lambda)$ is an optimal cardinality-K assortment. By simply replacing ϕ with ϕ_λ in the distributional objective (8), we can directly apply our DAO algorithm (Algorithm 1) to solve the constrained problem. The generative rounding procedure (Theorem 5.2) ensures that when $\lambda \geq G$, the resulting assortment $\hat{S}(x)$ is not only feasible but also near-optimal for the cardinality-constrained problem.

7 Numerical Experiments

In this section, we validate our proposed algorithm DAO by comparing it with the state-of-the-art SFESS (Wijk et al., 2025) for the mixed logit (ML) choice model under both capacity-constrained and unconstrained assortment optimization settings. We also refer the reader to the experiment result for other choice model specification including nested logit (NL) and and basic attraction model (BAM) in the Appendix C. We construct a comprehensive benchmark with problem sizes $n \in \{10, 100, 1000\}$ and capacity limits $K \in \{3, 10, 30, 100\}$ subject to $K < n$. For all instances, item revenues are drawn i.i.d. from Unif$[10, 100]$ and attractions from Unif$[1, 10]$. Each parameter combination is replicated multiple times with independent random seeds to enable statistical significance testing. The mixed logit framework models customer heterogeneity through discrete mixing over $m \in \{10, 100, 1000\}$ customer segments. Each segment represents a distinct preference pattern, with the outside option parameter drawn from $v_0 \sim$ Unif$[5.0, 20.0]$ to introduce baseline utility variation. We report numerical results for both the capacity-constrained case, with capacity limits $K \in \{3, 10, 30, 100\}$, and the unconstrained case.

In our setting, the number of learnable parameters in the SFESS model is significantly larger than in the DAO model, making it evident that SFESS can capture more information and generally performs better. However, as shown in Table 1, while SFESS outperforms DAO for small-scale problems (e.g., $n = 10$, achieving near-optimal solutions with minimal gaps (0.00 ± 0.00), its performance begins to fluctuate significantly as the problem size increases, especially when $n = 100$ and $K = 30$. Particularly, for $n = 100$ and $K = 30$, the gap for SFESS increases substantially, with an error of 1.60 ± 0.69, indicating that as the problem complexity increases, SFESS's advantage is limited due to its larger number of learnable parameters. In contrast, DAO shows significantly smaller errors, close to 0, in this setting. This suggests that, for medium-scale problems, especially when K is large, DAO demonstrates better stability in maintaining low error levels compared to SFESS.

Regarding computational efficiency, DAO also outperforms SFESS. The runtime comparison in Table 2 shows that SFESS requires significantly more computational time than DAO. This indicates that as the problem size grows, SFESS's computational demands and time consumption increase, while DAO maintains relatively lower computational times across all configurations. In large-scale problems, DAO consistently completes tasks within the time limits, while SFESS often fails due to time constraints.

For the unconstrained case, we observe that when the number of products is relatively small, our algorithm can achieve the same results as the traditional Mixed Logit (ML) model, but it requires more computation time. However, as the number of products increases, our computation time decreases significantly. Although for $n = 1000$, the gap increases slightly for different m values, we believe this result is acceptable considering the substantial reduction in computation time. In fact, this demonstrates that within an acceptable gap range, our algorithm can significantly improve computational efficiency and produce results close to the optimal in a much shorter time.

		m = 10		m = 100		m = 1000	
n	K	DAO	SFESS	DAO	SFESS	DAO	SFESS
10	3	4.5 ± 4.7	0.00 ± 0.00	2.3 ± 3.3	0.00 ± 0.00	2.3 ± 3.5	-0.00 ± 0.00
100	10	2.0 ± 0.6	0.00 ± 0.00	1.5 ± 0.6	-0.00 ± 0.00	2.0 ± 0.6	-0.01 ± 0.02
100	30	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	0.00 ± 0.00	0.0 ± 0.0	1.60 ± 0.69
1000	30	3.11 ± 0.41	0.00 ± 0.00	2.9 ± 0.7	-0.00 ± 0.01	2.1 ± 0.5	-
1000	100	2.55 ± 0.61	0.46 ± 0.19	2.4 ± 0.6	0.39 ± 0.15	2.2 ± 0.8	-

Table 1: Comparison of DAO and SFESS under the Mixed Logit (ML) model (constrained). Values report the percentage gap $\pm$ standard deviation with respect to the optimal solutions. Each entry is averaged over 10 independent runs, except for $n = 1000$ where only 5 runs were performed due to computational limits. "–" indicates that the solver did not finish within the 5-minute time limit.

		m = 10		m = 100		m = 1000	
n	K	DAO	SFESS	DAO	SFESS	DAO	SFESS
10	3	72.60 ± 45.56	122.76 ± 59.26	8.52 ± 4.18	31.82 ± 16.45	2.08 ± 1.10	17.31 ± 8.54
100	10	1.52 ± 0.39	3.53 ± 1.51	0.27 ± 0.08	1.14 ± 0.70	0.01 ± 0.00	0.18 ± 0.02
100	30	11.41 ± 8.34	17.31 ± 13.27	0.56 ± 0.27	303 ± 2.19	0.04 ± 0.00	0.65 ± 0.09
1000	30	0.07 ± 0.00	0.21 ± 0.00	0.10 ± 0.01	0.52 ± 0.29	0.32 ± 0.00	-
1000	100	0.38 ± 0.19	2.06 ± 0.99	0.05 ± 0.00	0.74 ± 0.00	0.18 ± 0.00	-

Table 2: Comparison of DAO and SFESS under the Mixed Logit (ML) model (constrained). Values report the runtime ratio (DAO or SFESS time / optimal solver time) expressed as mean $\pm$ standard deviation across different (n, K, m). "–" indicates that the solver did not finish within the 5-minute time limit.

Metric	n	$m = 10$	$m = 100$	$m = 1000$
Gap (%)	10	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	100	0.02 ± 0.03	0.02 ± 0.03	0.02 ± 0.03
	1000	1.47 ± 0.33	1.70 ± 0.22	1.10 ± 0.23
Runtime ratio	10	104.76 ± 38.29	19.53 ± 6.29	4.17 ± 0.49
	100	5.64 ± 2.80	0.67 ± 0.50	0.03 ± 0.01
	1000	0.51 ± 0.40	0.06 ± 0.02	0.13 ± 0.00

Table 3: Mixed Logit (ML), unconstrained case. The table reports (top block) the percentage gap $\pm$ standard deviation with respect to the optimal solutions, and (bottom block) the runtime ratio (DAO time / optimal solver time), both across (n, m). Each entry is averaged over 10 independent runs, except for $n = 1000$ where only 5 runs were performed due to computational limits.

8 Reproducibility Statement

To ensure the reproducibility of our work, we have made comprehensive efforts to document all theoretical and experimental components. For the theoretical results, including the guarantees in Theorem 5.1 and 5.2, complete proofs with detailed explanations of assumptions (e.g., Lipschitz continuity and boundedness) are provided in the appendix. For the experimental setup, we include a full description of the datasets, preprocessing steps, hyperparameters, and evaluation metrics in the supplementary materials. Additionally, an anonymous link to downloadable source code, implementing the generative model and rounding procedure, is submitted as supplementary material to facilitate replication of our results. We encourage readers to refer to these resources for further details.

References

Shipra Agrawal, Vashist Avadhanula, Vineet Goyal, and Assaf Zeevi. Mnl-bandit: A dynamic learning approach to assortment selection. *Operations Research*, 67(5):1453–1485, 2019.

Kareem Ahmed, Zhe Zeng, Mathias Niepert, and Guy Van den Broeck. SIMPLE: A gradient estimator for k-subset sampling. In *International Conference on Learning Representations*, 2023.

Yi-Chun Akchen and Dmitry Mitrofanov. Consider or choose? the role and power of consideration sets. *arXiv preprint arXiv:2302.04354*, 2025.

F. Bach. *Learning with Submodular Functions: A Convex Optimization Perspective*, volume 6. 2013.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Gerardo Berbeglia and Gwenaël Joret. Assortment optimisation under a general discrete choice model: A tight analysis of revenue-ordered assortments. *Algorithmica*, 82(4):681–720, 2020.

Jose Blanchet, Guillermo Gallego, and Vineet Goyal. A markov chain approximation to choice modeling. *Operations Research*, 64(4):886–905, 2016.

J. J. M. Bront, I. Méndez-Díaz, and G. Vulcano. A column generation algorithm for choice-based network revenue management. *Operations Research*, 57(3):769–784, 2009.

Xi Chen, Yining Wang, and Yuan Zhou. Dynamic assortment optimization with changing contextual information. *Journal of machine learning research*, 21(216):1–44, 2020.

J. M. Davis, G. Gallego, and H. Topaloglu. Assortment optimization under variants of the nested logit model. *Operations Research*, 62(2):250–273, 2014.

Antoine Désir, Vineet Goyal, Srikanth Jagabathula, and Danny Segev. Assortment optimization under the mallows model. *Advances in Neural Information Processing Systems*, 29, 2016.

Antoine Désir, Vineet Goyal, Danny Segev, and Chun Ye. Constrained assortment optimization under the markov chain–based choice model. *Management Science*, 66(2):698–721, 2020.

Guillermo Gallego and Huseyin Topaloglu. Constrained assortment optimization for the nested logit model. *Management Science*, 60(10):2583–2601, 2014.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

Julia Heger and Robert Klein. Assortment optimization: A systematic literature review. *OR Spectrum*, 46(4):1099–1161, 2024.

Shinji Ito. Submodular function minimization with noisy evaluation oracle. *Advances in Neural Information Processing Systems*, 32, 2019.

Srikanth Jagabathula. Assortment optimization under general choice. *Available at SSRN 2512831*, 2014.

Guokai Li, Pin Gao, Stefanus Jasin, and Zizhuo Wang. From small to large: A graph convolutional network approach for solving assortment optimization problems. *arXiv preprint arXiv:2507.10834*, 2025.

Tao Li, Chenhao Wang, Yao Wang, Shaojie Tang, and Ningyuan Chen. Deep reinforcement learning for online assortment customization: A data-driven approach. *Available at SSRN 4870298*, 2024.

L. Lovász. Submodular functions and convexity. In *Mathematical Programming: The State of the Art*, pp. 235–257. Springer, 1983.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layers neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7673, Aug 2018. doi: 10.1073/pnas.1806579115. URL `https://www.pnas.org/doi/10.1073/pnas.1806579115`. Also available as arXiv:1804.06561.

Atsushi Nitanda, Denny Wu, and Taiji Suzuki. Convex analysis of the mean field langevin dynamics. In *International Conference on Artificial Intelligence and Statistics*, pp. 9741–9757. PMLR, 2022.

Max Paulus, Dami Choi, Daniel Tarlow, Andreas Krause, and Chris J Maddison. Gradient estimation with stochastic softmax tricks. *Advances in Neural Information Processing Systems*, 33:5691–5704, 2020.

Meng Qi, Ho-Yin Mak, and Zuo-Jun Max Shen. Data-driven research in retail operations—a review. *Naval Research Logistics (NRL)*, 67(8):595–616, 2020.

Paat Rusmevichientong and Huseyin Topaloglu. Robust assortment optimization in revenue management under the multinomial logit choice model. *Operations research*, 60(4):865–882, 2012.

Paat Rusmevichientong, David Shmoys, Chaoxu Tong, and Huseyin Topaloglu. Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management*, 23(11):2023–2039, 2014.

Kalyan Talluri and Garrett Van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33, 2004.

K. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2 edition, 2009.

Rajan Udwani. Submodular order functions and assortment optimization. In *International Conference on Machine Learning*, pp. 34584–34614. PMLR, 2023.

Hanzhao Wang, Zhongze Cai, Xiaocheng Li, and Kalyan Talluri. A neural network based choice model for assortment optimization. *arXiv preprint arXiv:2308.05617*, 2023.

Zhengchao Wang, Heikki Peura, and Wolfram Wiesemann. Randomized assortment optimization. *Operations Research*, 72(5):2042–2060, 2024.

Klas Wijk, Ricardo Vinuesa, and Hossein Azizpour. SFESS: Score function estimators for k-subset sampling. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Sang Michael Xie and Stefano Ermon. Reparameterizable subset sampling via continuous relaxations. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 3919–3925, 2019.

Yutaro Yamada, Ofir Lindenbaum, Sahand Negahban, and Yuval Kluger. Feature selection using stochastic gates. In *International conference on machine learning*, pp. 10648–10659. PMLR, 2020.

Yiqi Yang, Zhi Wang, Rui Gao, and Shuang Li. Reproducing kernel hilbert space choice model. In *Proceedings of the 26th ACM Conference on Economics and Computation*, pp. 819–819, 2025.

Haixiang Zhang, Zeyu Zheng, and Javad Lavaei. Gradient-based algorithms for convex discrete optimization via simulation. *Operations research*, 71(5):1815–1834, 2023.

A PROOFS

A.1 PROOFS FOR SECTION 2

A.1.1 PROOF OF LEMMA 2.2

We prove each property separately.

Proof of (1).

For any fixed permutation π, consider the region $\mathcal{C}_\pi$. For any $x \in \mathcal{C}_\pi$, the components have a strict ordering. By Lemma A.2.1, ϕ is differentiable at x and its gradient is given by $\nabla\phi(x) = g^\pi$, where g^π is the constant vector with components $g^\pi_{\pi(k)} = f(S_k) - f(S_{k-1})$ for $k = 1, \ldots, n$ and $S_k = \{\pi(1), \ldots, \pi(k)\}$ with $S_0 = \emptyset$. Moreover, from the sorting formula of the Lovász extension, we have that

$$\phi(x) = \sum_{k=1}^{n} x_{\pi(k)} \left(f(S_k(x)) - f(S_{k-1}(x)) \right) = \langle g^\pi, x \rangle,$$

where g^π is the constant vector with components $g^\pi_{\pi(k)} = f(S_k) - f(S_{k-1})$ for $k = 1, \ldots, n$. This shows that ϕ is an affine function on $\mathcal{C}_\pi$ with constant gradient $\nabla\phi(x) = g^\pi$.

Since the hypercube $[0, 1]^n$ can be partitioned into finitely many such regions $\mathcal{C}_\pi$ (and their boundaries), ϕ is piecewise affine.

Proof of (2).

By Lemma A.2.1, on each region $\mathcal{C}_\pi$, the gradient $\nabla\phi(x) = g^\pi$ is constant. We have:

$$\|\nabla\phi(x)\|_\infty = \|g^\pi\|_\infty = \max_{1 \le k \le n} |f(S_k) - f(S_{k-1})| \le G,$$

where the inequality follows from the definition of G as the maximum marginal change when adding any single element to any subset.

Now, for any $x, y \in [0, 1]^n$, consider the straight line segment connecting them: $\gamma(t) = x + t(y - x)$ for $t \in [0, 1]$. The function ϕ is piecewise affine and therefore absolutely continuous along this path. By the fundamental theorem of calculus for absolutely continuous functions, we have that

$$\phi(y) - \phi(x) = \int_0^1 \langle \nabla\phi(\gamma(t)), y - x \rangle dt,$$

where the gradient exists almost everywhere along the path.

Taking absolute values and using the Cauchy-Schwarz inequality yields

$$|\phi(y) - \phi(x)| \le \int_0^1 |\langle \nabla\phi(\gamma(t)), y - x \rangle| dt \le \int_0^1 \|\nabla\phi(\gamma(t))\|_\infty \|y - x\|_1 dt.$$

Since $\|\nabla\phi(\gamma(t))\|_\infty \le G$ almost everywhere, we obtain that

$$|\phi(y) - \phi(x)| \le G\|y - x\|_1 \int_0^1 dt = G\|y - x\|_1,$$

which completes the proof of the Lipschitz continuity. □

A.2 Proofs for Section 3

A.2.1 Proof of Proposition 3

To prove Proposition 3, we introduce a preliminary lemma. When the components of x have a strict ordering, the gradient of ϕ is well-defined and has a simple form. If x has a strict order, i.e., $x_{\pi(1)} > x_{\pi(2)} > \cdots > x_{\pi(n)}$ for some unique permutation π, then the Lovász extension ϕ is differentiable at x, and its gradient is given by:

$$\nabla\phi(x) = g^\pi(x).$$

This result holds regardless of whether ϕ is convex, as it is a fundamental property of the Lovász extension.

Proof. Since x has a strict order, there exists an open ball $B(x) \subset \mathbb{R}^n$ centered at x such that for all $y \in B(x)$, the ordering of components remains unchanged, i.e.,

$$y_{\pi(1)} > y_{\pi(2)} > \cdots > y_{\pi(n)}.$$

This follows from the continuity of the coordinate functions and the strict inequalities at x.

For any $y \in B(x)$, using the sorting formula (4), the Lovász extension $\phi(y)$ can be expressed as

$$\phi(y) = \sum_{k=1}^{n} y_{\pi(k)} \left(f(S_k(x)) - f(S_{k-1}(x)) \right),$$

where the sets $S_k(x)$ are fixed for all $y \in B(x)$ since the permutation π is invariant in $B(x)$.

Thus, $\phi(y)$ is a linear function of y on $B(x)$, since the coefficients $a_k = f(S_k(x)) - f(S_{k-1}(x))$ are constants. Specifically,

$$\phi(y) = \langle g^\pi(x), y \rangle,$$

where $g^\pi(x)$ is the vector with components $g^\pi(x)_{\pi(k)} = a_k$ for $k = 1, \ldots, n$.

Since ϕ is linear on $B(x)$, it is differentiable at x, and its gradient is the constant vector of coefficients:

$$\nabla\phi(x) = g^\pi(x).$$

This completes the proof. □

Now, we begin the proof of Proposition 3.

By Lemma 2.2(2), ϕ is Lipschitz continuous on $[0,1]^n$, so the Clarke subdifferential $\partial^\circ\phi(x)$ is well-defined for every $x \in [0,1]^n$. By definition, $\partial^\circ\phi(x)$ is the convex hull of all limits of sequences $\nabla\phi(x^j)$ where $x^j \to x$ and ϕ is differentiable at x^j.

From Lemma 2.2(1), ϕ is piecewise affine on the strict-order regions $\mathcal{C}_\pi$. Moreover, by Lemma A.2.1, at any point x with strict order, $\nabla\phi(x) = g^\pi$ for the corresponding permutation π.

We now prove the two inclusions.

(i) Inclusion $\partial^\circ\phi(x) \supseteq \operatorname{conv}\{g^\pi : \pi \in \Pi(x)\}$. Fix any $\pi \in \Pi(x)$. We need to show that $g^\pi \in \partial^\circ\phi(x)$. To do this, we construct a sequence $x^{(\varepsilon)} \to x$ such that for sufficiently small $\varepsilon > 0$, $x^{(\varepsilon)} \in \mathcal{C}_\pi$ and hence by Lemma A.2.1, $\nabla\phi(x^{(\varepsilon)}) = g^\pi$.

Let the distinct values of x be $\bar{x}^1 > \bar{x}^2 > \cdots > \bar{x}^m$, and define the tie blocks $B_r = \{j \in V : x_j = \bar{x}^r\}$ for $r = 1, \ldots, m$. Since $\pi \in \Pi(x)$, it respects the order of the blocks: for any $i \in B_r$ and $j \in B_s$ with $r < s$, we have $x_i > x_j$, and π places all elements of B_r before those of B_s. Within each block B_r, the order given by π may be arbitrary.

Now, construct a perturbation $\eta^{(\varepsilon)}$ as follows: for each block B_r, choose distinct numbers $b_i^{(r)}$ for $i \in B_r$ such that they are strictly decreasing in the order prescribed by π within B_r. For example, set $b_i^{(r)} = -k$ for the k-th element in B_r according to π. Then, we define

$$\eta_i^{(\varepsilon)} = \varepsilon^r b_i^{(r)} \quad \text{for } i \in B_r, \quad \text{and} \quad x^{(\varepsilon)} = x + \eta^{(\varepsilon)}.$$

For small ε, the perturbations are small. Since for $i \in B_r$ and $j \in B_s$ with $r < s$, we have $x_i - x_j = \bar{x}^r - \bar{x}^s > 0$, and the perturbations are of order ε^r and ε^s with $r < s$, so for small ε, $x_i^{(\varepsilon)} > x_j^{(\varepsilon)}$ because the perturbation does not change the order between blocks. Within each block, the distinct $b_i^{(r)}$ ensure strict order as prescribed by π. Thus, for small ε, $x^{(\varepsilon)} \in \mathcal{C}_\pi$.

Therefore, $\nabla\phi(x^{(\varepsilon)}) = g^\pi$ for all small ε, and since $x^{(\varepsilon)} \to x$ as $\varepsilon \to 0$, we have g^π as a limit of gradients, so $g^\pi \in \partial^\circ\phi(x)$. Since this holds for every $\pi \in \Pi(x)$, and $\partial^\circ\phi(x)$ is convex, we have $\partial^\circ\phi(x) \supseteq \operatorname{conv}\{g^\pi : \pi \in \Pi(x)\}$.

(ii) Inclusion $\partial^\circ \phi(x) \subseteq \operatorname{conv}\{g^\pi : \pi \in \Pi(x)\}$.

Take any $v \in \partial^\circ \phi(x)$. By definition, there exists a sequence $x^j \to x$ such that ϕ is differentiable at each x^j and $\nabla\phi(x^j) \to v$. Since ϕ is piecewise affine (Lemma 2.2(1)), each x^j lies in some strict-order region $\mathcal{C}_{\pi^j}$, and thus $\nabla\phi(x^j) = g^{\pi^j}$. There are only finitely many permutations, so by passing to a subsequence, we may assume that $\pi^j = \pi$ for all j for some fixed permutation π. Then $v = \lim_{j\to\infty} \nabla\phi(x^j) = g^\pi$.

We now show that $\pi \in \Pi(x)$. That is, for any i, j such that $x_i > x_j$, we must have that π places i before j.

Since $x^j \to x$, for large j, we have $x_i^j > x_j^j$ due to that $x_i > x_j$. Since x^j has strict order consistent with π, it must be that π places i before j. This holds for all such pairs, so π respects the order of the values of x. Specifically, if we consider the tie blocks, for any $i \in B_r$ and $j \in B_s$ with $r < s$, we have $x_i > x_j$, so π places i before j. Within blocks, the order may be arbitrary, so indeed $\pi \in \Pi(x)$. Therefore, $v = g^\pi$ for some $\pi \in \Pi(x)$. Since every point in $\partial^\circ\phi(x)$ is a limit of such gradients, and the set $\{g^\pi : \pi \in \Pi(x)\}$ is finite, the convex hull of these limits is exactly $\operatorname{conv}\{g^\pi : \pi \in \Pi(x)\}$. Thus, $\partial^\circ \phi(x) \subseteq \operatorname{conv}\{g^\pi : \pi \in \Pi(x)\}$.

Combining both inclusions, we have equality. The proof is complete. □

A.2.2 Proof of Proposition 3

We first fix $x \in [0,1]^n$ and condition on a sampled permutation $\pi \in \Pi(x)$. By Definition 3, the stochastic subgradient $\widehat{g}$ is defined as that

$$\widehat{g} = n\left(f(S_K) - f(S_{K-1})\right)\mathbf{e}_{\pi(K)},$$

where K is a random rank uniformly sampled from $\{1, \ldots, n\}$ with probability $1/n$, and $S_k = \{\pi(1), \ldots, \pi(k)\}$ for $k = 0, \ldots, n$ (with $S_0 = \varnothing$). Using the law of total expectation over K, the conditional expectation of $\widehat{g}$ given x and π is computed as that

$$\mathbb{E}[\widehat{g} \mid x, \pi] = \sum_{k=1}^{n} \mathbb{P}(K = k \mid \pi) \cdot n\left(f(S_k) - f(S_{k-1})\right)\mathbf{e}_{\pi(k)}.$$

Since $\mathbb{P}(K = k \mid \pi) = 1/n$ by the uniform sampling, it simplifies to that

$$\mathbb{E}[\widehat{g} \mid x, \pi] = \sum_{k=1}^{n} \left(f(S_k) - f(S_{k-1})\right)\mathbf{e}_{\pi(k)} = g^\pi, \tag{12}$$

where the last equality follows from the definition of g^π as the subgradient associated with π. Thus, $\mathbb{E}[\widehat{g} \mid \pi] = g^\pi$ (noting that conditioning on π implies x is fixed).

Now, consider the expectation conditional only on x. Let Q be the distribution on $\Pi(x)$ induced by the tie-breaking rule (e.g., uniform within each block of tied values). Taking expectation of both sides in (12) over $\pi \sim Q$ yields that

$$\mathbb{E}[\widehat{g} \mid x] = \mathbb{E}_{\pi\sim Q}\left[\mathbb{E}[\widehat{g} \mid x, \pi]\right] = \mathbb{E}_{\pi\sim Q}[g^\pi].$$

Since $g^\pi \in \{g^\pi : \pi \in \Pi(x)\}$ for each π, the expectation $\mathbb{E}_{\pi\sim Q}[g^\pi]$ is a convex combination of these vectors. By Proposition 3, we have $\operatorname{conv}\{g^\pi : \pi \in \Pi(x)\} = \partial^\circ\phi(x)$, so we have

$$\mathbb{E}[\widehat{g} \mid x] \in \partial^\circ \phi(x).$$

If x is fixed, then the unconditional expectation $\mathbb{E}[\widehat{g}]$ equals $\mathbb{E}[\widehat{g} \mid x]$, and hence $\mathbb{E}[\widehat{g}] \in \partial^\circ\phi(x)$. This completes the proof of both equalities. □

A.3 Proofs for Section 4

The following lemma establishes the differentiability of $\Phi(\theta) = \mathbb{E}_{z\sim\nu}[\phi(g_\theta(z))]$ with respect to θ. Assume that the function $g_\theta : Z \to \mathbb{R}^d$ is piecewise smooth in θ, meaning that for each $z \in Z$, the map $\theta \mapsto g_\theta(z)$ is piecewise continuously differentiable. The probability measure ν on Z has a density, and there exists an integrable function $h : Z \to \mathbb{R}$ such that for all θ, the norm of the Jacobian $\|J_\theta g_\theta(z)\|$is bounded by $h(z)$ for almost every z. Then, the function $\Phi(\theta) = \mathbb{E}_{z\sim\nu}[\phi(g_\theta(z))]$is differentiable with respect to θ, and

$$\nabla_\theta \Phi(\theta) = \mathbb{E}_{z\sim\nu}[\nabla_\theta \phi(g_\theta(z))],$$

where $\nabla_\theta \phi(g_\theta(z))$ exists for almost every z and is given by the chain rule.

Proof. Since ϕ is piecewise linear, it is differentiable almost everywhere. Similarly, since $g_\theta(z)$ is piecewise smooth in θ, the function $\theta \mapsto \phi(g_\theta(z))$ is differentiable almost everywhere in θ for each z. The gradient is given by the chain rule:

$$\nabla_\theta \phi(g_\theta(z)) = J_\theta g_\theta(z)^\mathsf{T} \nabla_x \phi(x)\big|_{x=g_\theta(z)},$$

where $\nabla_x \phi(x)$ exists almost everywhere.

By the Lipschitz continuity of ϕ, we have $\|\nabla_x \phi(x)\| \le L_\phi$ almost everywhere, where L_ϕ is the Lipschitz constant. Therefore,

$$\|\nabla_\theta \phi(g_\theta(z))\| \le \|J_\theta g_\theta(z)^\mathsf{T}\| \cdot \|\nabla_x \phi(x)\| \le \|J_\theta g_\theta(z)\| L_\phi.$$

By assumption, $\|J_\theta g_\theta(z)\| \le h(z)$ for some integrable function h, so $\|\nabla_\theta \phi(g_\theta(z))\| \le L_\phi h(z)$, which is integrable since h is integrable.

Thus, by the dominated convergence theorem, we can interchange the gradient and the expectation:

$$\nabla_\theta \Phi(\theta) = \nabla_\theta \mathbb{E}_{z\sim\nu}[\phi(g_\theta(z))] = \mathbb{E}_{z\sim\nu}[\nabla_\theta \phi(g_\theta(z))].$$

This completes the proof. □

A.3.1 Proof of Theorem 4.1

Let $G(z) = J_\theta g_\theta(z)^\mathsf{T} \widehat{g}$, where $z \sim \nu$. We aim to show that $\mathbb{E}[G(z)] = \nabla_\theta \Phi(\theta)$, with $\Phi(\theta) = \mathbb{E}_{z\sim\nu}[\phi(g_\theta(z))]$.

By Proposition 3, $\mathbb{E}[\widehat{g} \mid x] \in \partial\phi(x)$ for $x = g_\theta(z)$, where $\partial\phi(x)$ is the subdifferential of ϕ at x.

For fixed z, conditional on z, we have that

$$\mathbb{E}[G(z) \mid z] = \mathbb{E}[J_\theta g_\theta(z)^\mathsf{T} \widehat{g} \mid z] = J_\theta g_\theta(z)^\mathsf{T} \mathbb{E}[\widehat{g} \mid z] = J_\theta g_\theta(z)^\mathsf{T} \mathbb{E}[\widehat{g} \mid x] \in J_\theta g_\theta(z)^\mathsf{T} \partial\phi(x).$$

By the chain rule for subdifferentials, since g_θ is differentiable and ϕ is convex, it follows that:

$$J_\theta g_\theta(z)^\mathsf{T} \partial\phi(x) = \partial_\theta(\phi \circ g_\theta)(z),$$

where $\partial_\theta(\phi \circ g_\theta)(z)$ denotes the subdifferential of the function $\theta \mapsto \phi(g_\theta(z))$ with respect to θ, at fixed z.

Thus, it follows that

$$\mathbb{E}[G(z) \mid z] \in \partial_\theta(\phi \circ g_\theta)(z).$$

Taking expectation over z yields that

$$\mathbb{E}[G(z)] = \mathbb{E}_z\left[\mathbb{E}[G(z) \mid z]\right] \in \mathbb{E}_z\left[\partial_\theta(\phi \circ g_\theta)(z)\right].$$

Since $\Phi(\theta)$ is differentiable with respect to θ, we can interchange the gradient and the expectation by the dominated convergence theorem:

$$\nabla_\theta \Phi(\theta) = \nabla_\theta \mathbb{E}_{z\sim\nu}[\phi(g_\theta(z))] = \mathbb{E}_{z\sim\nu}[\nabla_\theta \phi(g_\theta(z))].$$

Since $\Phi(\theta)$ is differentiable with respect to θ (as ensured by the smoothness of ν and the piecewise smoothness of g_θ), the subdifferential $\partial_\theta \Phi(\theta)$ is a singleton containing only $\nabla_\theta \Phi(\theta)$. Therefore, we obtain that

$$\mathbb{E}[G(z)] = \nabla_\theta \Phi(\theta),$$

which proves that $G(z)$ is an unbiased estimator of $\nabla_\theta \Phi(\theta)$.

□

A.4 Preliminaries and Proofs for Section 5

A.5 Preliminaries

We first give the definition of a gradient for the functional F. [First Variation] The *first variation* of the functional F at a distribution μ, denoted $\frac{\delta F}{\delta \mu}[\mu](\theta)$, is the functional derivative of F with respect to μ evaluated at a point θ. Formally,

$$\frac{\delta F}{\delta \mu}[\mu](\theta) := \lim_{\epsilon \to 0} \frac{F\big((1-\epsilon)\mu + \epsilon\nu\big) - F(\mu)}{\epsilon},$$

for any perturbation distribution ν. The first variation provides the infinitesimal direction along which the functional F decreases under perturbations of μ.

A.5.1 Proof of Theorem 5.1

To prove the theorem, we adopt the mean-field theory developed in Nitanda et al. (2022).

We first provide a brief overview of the mean-field theory and highlight several key results in Nitanda et al. (2022) that will be used in our proof. Let $\theta^i := (W^i, b^i)$. As $N \to \infty$, the empirical distribution $\frac{1}{N}\sum_{i=1}^N \delta_{\theta^i}$ converges to the a distribution μ, and $g_\theta(z)$ converges to

$$\mathbb{E}_\mu[\sigma(Wh(z) + b)].$$

Thereby, we write the objective of (8) as

$$F(\mu) := \mathbb{E}_{z\sim\nu}\big[\phi\big(\mathbb{E}_\mu[\sigma(Wh(z)+b)]\big)\big].$$

Denote by $\frac{\delta F}{\delta \mu}[\cdot]$ the first variation of F, and by $\nabla \frac{\delta F}{\delta \mu}[\mu](\cdot)$ the Wasserstein gradient of F at μ. Let $(\mu_t)_{t\geq 0}$ be the laws of the iterates

$$\theta_{t+1} = (1-\alpha)\theta_t - \nabla \frac{\delta F}{\delta \mu}[\mu_t](\theta_t) + \sqrt{2\eta_t \tau}\xi_t.$$

Then

$$\mu_{t+1} = (\mathrm{id} - \eta \nabla \tfrac{\delta F}{\delta \mu}[\mu_t])_{\#}\mu_t * \mathcal{N}(0, 2\eta_t \tau I).$$

Define a Lyapunov functional

$$\mathcal{L}(\mu) = F(\mu) + \tau H(\mu),$$

where $H(\mu) = \mathbb{E}_\mu[\log \mu]$ is the negative entropy of μ. Define the Fisher information of a distribution q as

$$\mathcal{I}(q) := \int q(\theta) \Big\| \nabla \log \frac{q}{p_q}(\theta) \Big\|^2 d\theta,$$

where p_q is the proximal Gibbs associated with q:

$$p_q(\theta) \propto \exp\Big(-\frac{1}{\tau}\frac{\delta F}{\delta q}[q](\theta)\Big).$$

For each s, let $(q_s^{(t)})_{s\in[0,\eta_t]}$ be the Langevin interpolation with $q_0^{(t)} = \mu_t$ and $q_{\eta_t}^{(t)} = \mu_{t+1}$. Then we have

$$|\dot{q}_s^{(t)}|^2 = \tau^2 \mathcal{I}\big(q_s^{(t)}\big).$$

From Nitanda et al. (2022, Theorem 2 and Lemma 1), we have the following results:

(R1) $\sup_t \mathbb{E}_{\mu_t}[\|\theta\|^2] < \infty$.

(R2) There exist a constant $C < \infty$ such that

$$\mathcal{L}(\mu_{t+1}) - \mathcal{L}(\mu_t) \leq -\tau^2 \int_0^{\eta_t} \mathcal{I}\big(q_s^{(t)}\big)\, dt + C\eta_t^2. \tag{13}$$

Notably, neither of these results relies on the convexity of ϕ, an assumption that does not hold in our setting.

Proof of Theorem 5.1. Summing (13) from $t = 0$ to $T-1$ and using that $\mathcal{L}$ is bounded below along $(\mu_t)_t$ (implied from (R1)), we obtain

$$\lambda^2 \sum_{t=0}^{T-1} \int_0^{\eta_t} \mathcal{I}\big(q_s^{(t)}\big) ds \leq \mathcal{L}(\mu_0) - \mathcal{L}(\mu_T) + C\sum_{t=0}^{T-1}\eta_t^2 \leq \mathcal{L}(\mu_0) - \inf \mathcal{L} + C\sum_{t=0}^{T-1}\eta_t^2. \tag{14}$$

Letting $T \to \infty$ and using $\sum_t \eta_t^2 < \infty$ gives

$$\sum_{t=0}^{\infty} \int_0^{\eta_t} \mathcal{I}\big(q_s^{(t)}\big)\, ds \;<\; \infty. \tag{15}$$

Define $s_0 := 0$, $s_{t+1} := s_t + \eta_t$, and the concatenated curve

$$Q_s \;:=\; q^{(t)}_{s-s_t} \quad \text{for } s \in [s_t, s_{t+1}).$$

Then, by the identity $|\dot{q}^{(t)}_s|^2 = \lambda^2 \mathcal{I}(q^{(t)}_s)$ and (15), we have

$$\sum_{t=0}^{\infty} W_2^2(\mu_{t+1}, \mu_t) = \sum_{t=0}^{\infty} W_2^2\big(q^{(t)}_{\eta_t}, q^{(t)}_0\big) \le \int_0^{\infty} |\dot{Q}_s|^2 ds = \sum_{t=0}^{\infty} \int_{s_t}^{s_{t+1}} |\dot{Q}_s|^2 \, ds = \lambda^2 \sum_{t=0}^{\infty} \int_0^{\eta_t} \mathcal{I}\big(q^{(t)}_s\big) ds < \infty.$$

For each t, choose $\tau_t \in [s_t, s_{t+1}]$ so that

$$|\dot{Q}_{\tau_t}|^2 \;\le\; \frac{1}{\eta_t} \int_{s_t}^{s_{t+1}} |\dot{Q}_s|^2 \, ds,$$

which is possible by the mean value theorem. Since $\eta_t \downarrow 0$ and $\sum_k \int_{s_t}^{s_{t+1}} |\dot{Q}_s|^2 ds < \infty$, we have $|\dot{Q}_{\tau_t}| \to 0$. By (R1) the family (Q_{τ_t}) is tight in $\mathcal{P}_2$, hence (passing to a subsequence if needed) $Q_{\tau_t} \to \mu_\infty$ in W_2. We claim that $\mu_\infty = p_{\mu_\infty}$. Indeed, since the map $q \mapsto p_q$ is W_2-continuous; the lower semicontinuity of metric slopes yields that $q \mapsto \mathcal{I}(q)$ is lower semicontinuous with respect to W_2 convergence. Therefore,

$$0 \le \lambda^2 \mathcal{I}(\mu_\infty) \;\le\; \liminf_{t\to\infty} |\dot{Q}_{\tau_t}|^2 = 0,$$

so $\mathcal{I}(\mu_\infty) = 0$, i.e., $\nabla \log \frac{\mu_\infty}{p_{\mu_\infty}} = 0$ μ_∞–a.e., and hence $\mu_\infty = p_{\mu_\infty}$.

Finally, let us derive the convergence rate. Generate random vector (τ, U) with probabilities

$$\mathbb{P}(\tau = t) = \frac{\eta_t}{\sum_{t'=1}^{T-1} \eta_{t'}}, \quad U \sim \text{Unif}[0, \eta_\tau].$$

Set

$$\tilde{t} = s_\tau + U, \qquad H_T = \sum_{t'=1}^{T} \eta_{t'}$$

Then $\tilde{t} \sim \text{Unif}[0, H_T]$. Using (14),

$$\mathbb{E}[\mathcal{I}(Q_{\tilde{t}})] \le \frac{\mathcal{L}(\mu_0) - \inf \mathcal{L}}{\tau H_T} + \frac{C \sum_{t=0}^{T-1} \eta_t^2}{\tau^2 H_T}.$$

Setting $\eta_t = \eta = 1/\sqrt{T}$ yields $H_T = \sqrt{T}$ and

$$\mathbb{E}[\mathcal{I}(Q_{\tilde{t}})] \le \frac{\mathcal{L}(\mu_0) - \inf \mathcal{L}}{\tau \sqrt{T}} + \frac{C}{\tau^2 \sqrt{T}}.$$

□

A.5.2 Proof of Theorem 5.2

By the approximation error and optimization error assumptions, it follows that

$$\Phi(\widehat{\theta}) \le \phi(x_{S^*}) + \varepsilon_{\text{approx}} + \varepsilon_{\text{opt}}.$$

Consider the random variable $X = \phi(g_{\widehat{\theta}}(z))$ where $z \sim \mathcal{N}(0, I_d)$. Since ϕ is bounded by M, we have $|X| \le M$ and $\mathbb{E}[X] = \Phi(\widehat{\theta})$. By Hoeffding's inequality for bounded random variables, we have

$$\mathbb{P}(X \ge \mathbb{E}[X] + t) \le \exp\left(-\frac{2t^2}{(2M)^2}\right) = \exp\left(-\frac{t^2}{2M^2}\right).$$

Set $\delta = \exp\left(-\frac{t^2}{2M^2}\right)$, so $t = M\sqrt{2\ln(1/\delta)} = \kappa(\delta)$. Then, with probability at least $1 - \delta$, it holds that

$$\phi(x) = X \le \mathbb{E}[X] + \kappa(\delta) = \Phi(\hat{\theta}) + \kappa(\delta) \le \phi(x_{S^*}) + \varepsilon_{\text{approx}} + \varepsilon_{\text{opt}} + \kappa(\delta).$$

By the key property of the rounding scheme, we have $r(\hat{S}(x)) \ge -\phi(x)$. Since $\phi(x_{S^*}) = -r(S^*)$, it holds with probability at least $1 - \delta$ that

$$-\phi(x) \ge r(S^*) - \varepsilon_{\text{approx}} - \varepsilon_{\text{opt}} - \kappa(\delta).$$

Therefore, we have

$$r(\hat{S}(x)) \ge r(S^*) - \varepsilon_{\text{approx}} - \varepsilon_{\text{opt}} - \kappa(\delta).$$

The proof is complete. □

A.6 Proofs for Section 6

A.6.1 Proof of Theorem 6

Fix an arbitrary $x \in [0,1]^n$ and let $z = P_K(x)$, where $P_K(x)$ denotes the projection of x onto the set of vectors with at most K non-zero entries. By Lemma 2.2, the Lovász extension ϕ is Lipschitz continuous with constant G, which implies:

$$\phi(x) \geq \phi(z) - G\|x - z\|_1.$$

Since $\|x - z\|_1 = \rho_K^{\mathrm{L}}(x)$, By Lemma 6, we have that

$$\phi(x) \geq \phi(z) - G\rho_K^{\mathrm{L}}(x).$$

Rearranging terms yields that

$$\phi(x) + G\rho_K^{\mathrm{L}}(x) \geq \phi(z).$$

For any $\lambda \geq G$, it follows that

$$\phi_\lambda(x) = \phi(x) + \lambda\rho_K^{\mathrm{L}}(x) \geq \phi(x) + G\rho_K^{\mathrm{L}}(x) \geq \phi(z).$$

Taking the infimum over $x \in [0,1]^n$ on the left-hand side and over z with $\|z\|_0 \leq K$ on the right-hand side (noting that z depends on x), we obtain that

$$\inf_{x\in[0,1]^n} \phi_\lambda(x) \geq \inf_{\substack{z\in[0,1]^n \\ \|z\|_0\leq K}} \phi(z). \tag{16}$$

Conversely, for any z with $\|z\|_0 \leq K$, we have $\rho_K^{\mathrm{L}}(z) = 0$, which implies that

$$\phi_\lambda(z) = \phi(z) + \lambda\rho_K^{\mathrm{L}}(z) = \phi(z).$$

Hence, it follows that

$$\inf_{x\in[0,1]^n} \phi_\lambda(x) \leq \inf_{\substack{z\in[0,1]^n \\ \|z\|_0\leq K}} \phi(z). \tag{17}$$

Combining inequalities (16) and (17), we conclude that

$$\min_{x\in[0,1]^n} \phi_\lambda(x) = \min_{\substack{z\in[0,1]^n \\ \|z\|_0\leq K}} \phi(z).$$

Now, for any z with $\|z\|_0 \leq K$, the Lovász extension satisfies that

$$\phi(z) = \int_0^1 f(S_t(z))\, dt,$$

where $S_t(z) = \{i \in V : z_i > t\}$. Since $\|z\|_0 \leq K$, each threshold set $S_t(z)$ has cardinality at most K. Thus, $\phi(z)$ is a convex combination of $f(S)$ for sets S with $|S| \leq K$, implyiny that

$$\min_{\substack{z\in[0,1]^n \\ \|z\|_0\leq K}} \phi(z) \geq \min_{\substack{S\subseteq V \\ |S|\leq K}} f(S).$$

Moreover, for any $S \subseteq V$ with $|S| \leq K$, the characteristic vector $z = \mathbf{1}_S$ satisfies $\phi(z) = f(S)$, so we have that

$$\min_{\substack{z\in[0,1]^n \\ \|z\|_0\leq K}} \phi(z) = \min_{\substack{S\subseteq V \\ |S|\leq K}} f(S).$$

Therefore, we prove that

$$\min_{x\in[0,1]^n} \phi_\lambda(x) = \min_{\substack{S\subseteq V \\ |S|\leq K}} f(S) = -\max_{\substack{S\subseteq V \\ |S|\leq K}} r(S),$$

where the last equality follows from $f(S) = -r(S)$.

To prove the moreover part, let x_λ be a minimizer of $\phi_\lambda(x)$. Suppose, for contradiction, that $\rho_K^{\mathrm{L}}(x_\lambda) > 0$. Then, with $z = P_K(x_\lambda)$, we have that

$$\phi_\lambda(x_\lambda) = \phi(x_\lambda) + \lambda\rho_K^{\mathrm{L}}(x_\lambda) > \phi(x_\lambda) + G\rho_K^{\mathrm{L}}(x_\lambda) \geq \phi(z),$$

where the last inequality holds by Lemma 2.2. However, since $\rho_K^{\mathrm{L}}(z) = 0$, we have $\phi_\lambda(z) = \phi(z)$, and by optimality of x_λ,

$$\phi_\lambda(x_\lambda) \leq \phi_\lambda(z) = \phi(z),$$

which is a contradiction. Hence, $\rho_K^{\mathrm{L}}(x_\lambda) = 0$, which implies $\|x_\lambda\|_0 \leq K$.

Consequently, every threshold set $S_t(x_\lambda)$ satisfies $|S_t(x_\lambda)| \leq \|x_\lambda\|_0 \leq K$. The chain sets $S_k(x_\lambda)$ are threshold sets, and since $\phi(x_\lambda)$ is an average of $f(S_t(x_\lambda))$ over $t \in [0,1]$, and x_λ minimizes ϕ_λ, it follows that at least one chain set $S_k(x_\lambda)$ must achieve the minimum value of f, and thus is an optimal cardinality-K assortment. The proof is complete. □

B EXPERIMENT SETTING

To ensure reproducibility, random seeds are deterministically generated as $\texttt{seed} = n \cdot 10^6 + m \cdot 10^4 + K \cdot 10 + \texttt{rep}$, where $(n, m, K, \texttt{rep})$ index the experimental configuration.

B.1 DAO

Our proposed Differentiable Assortment Optimizer (DAO) is a neural generator model designed for end-to-end optimization of discrete assortment problems. DAO constructs a latent-variable generator that maps Gaussian noise vectors to probabilistic selection logits, enabling gradient-based learning despite the inherently combinatorial structure of the problem. To adapt across model classes, DAO incorporates model-specific oracles (e.g., BAM, ML, NL) that provide unbiased stochastic gradients of the expected revenue with respect to the generator output.

The neural architecture of DAO model consists of a shared multi-layer perceptron with hidden dimension $H = 256$, followed by $N = 4$ parallel output channels. Each channel applies a ReLU transformation and produces a candidate selection vector; these are averaged and passed through a sigmoid nonlinearity to yield smooth selection probabilities. The latent dimension is fixed at $z_{\text{dim}} = 64$, ensuring compact yet expressive stochastic embeddings.

Training employs the Adam optimizer with learning rate 10^{-3}, batch size 64, and temperature-controlled Gumbel-softmax relaxation for discrete rounding. We use a cosine annealing schedule for the relaxation parameter τ, initialized at 0.15, decayed to 0.02 over the course of training with a 100-step warmup. The total number of iterations scales with problem size according to $\texttt{steps} = 1200 \times \text{clip}\Big(\frac{n}{25\sqrt{K}}, 1, 10\Big)$, yielding between 1,200 and 12,000 updates per instance.

B.2 SFESS

The core methodology combines parametric neural networks to generate product selection logits with SFESS-based discrete sampling mechanisms, enabling gradient-based optimization of inherently non-differentiable combinatorial objectives. To accommodate different market structures, we construct three model-specific variants with tailored revenue and loss functions: (1) the Multi-category SFESS implementing nested logit structure with category-level dissimilarity parameters to capture within- and across-category substitution patterns, (2) the Mixed Logit SFESS incorporating customer heterogeneity through type-specific attraction parameters, and (3) the base SFESS model with multinomial logit (MNL) choice probabilities for single-segment markets.

The neural architecture across all variants consists of multi-layer perceptrons with hidden dimensions ranging from 64 to 256 neurons, incorporating batch normalization and dropout (0.1-0.2) for regularization, with Xavier initialization strategies. Product features (prices, attractions, and customer-type specific parameters) are min-max normalized and processed through feature encoders before generating selection logits, which are numerically stabilized via clamping to [-10, 10]. Training employs the AdamW optimizer (learning rate 0.001-0.005, weight decay 5×10^{-4}) with ReduceLROnPlateau scheduling (patience=50, factor=0.8), processing 8-300 SFESS samples per iteration over 200-1000 iterations depending on problem scale. The discrete product selection process uses control-variate score function estimators to reduce variance in policy gradients, with gradient trimming (max norm = 1.0) to ensure training stability.

Across all three model variants, we maintain consistent hyperparameter settings for fair comparison. The base SFESS model is configured with hidden layer dimension of 256, learning rate of 0.001, 500 training iterations, batch size of 64, and n samples per iteration (instance-dependent). The Mixed Logit SFESS variant employs identical architecture (hidden size 256) with a slightly elevated learning rate of 0.005 to accommodate the increased complexity of heterogeneous customer types, maintaining 500 iterations and batch size 64. The Multi-category SFESS model similarly uses hidden dimension 256, learning rate 0.005, 500 iterations, and batch size 64, with additional early stopping criteria (patience=150, minimum improvement threshold 10^{-6}, improvement window=20 iterations). All models utilize the control-variate score function estimator for variance reduction in gradient estimation, with the number of SFESS samples per iteration determined by instance-specific metadata. Notably, for constrained choice models, the optimal assortment size does not necessarily equal the capacity constraint K, as revenue may be maximized at smaller assortment sizes due to cannibalization effects; therefore, we enumerate all values $1 \leq k \leq K$ and select the k yielding maximum revenue. For computational consistency, the capacity parameter is set to k+1 across all experiments to account for the no-purchase option, ensuring uniform problem formulation across different choice model structures.

B.3 HARDWARD

All experiments are conducted on a single NVIDIA RTX A4000 GPU (16 GB memory, CUDA 12.4, driver version 550.144.03) with TF32 enabled, and results are averaged across $\texttt{REPS} = 10$ independent repetitions for statistical robustness.

C MORE NUMERICAL RESULTS

C.1 NESTED LOGIT

The Nested Logit (NL) model captures substitution patterns through a hierarchical choice structure. The n items are partitioned into $g \in \{2, 20, 200\}$ nests with $g < n$ and $n \bmod g = 0$, ensuring balanced allocation across nests. The dissimilarity parameter γ is sampled from $\text{Unif}[0.2, 0.9]$ to cover the spectrum from near-independence ($\gamma \to 1$) to strong within-nest correlation ($\gamma \to 0$). For consistency across instances, the outside option utility is fixed at $v_0 = 15.0$.

The detailed numerical results are presented as follows. DAO achieves competitive accuracy across both constrained (Table 4) and unconstrained (Table 6) settings. In the large-scale case with $n = 1000$ and $K = 100$ (Table 5), DAO demonstrates particularly favorable runtime performance, maintaining accuracy while completing well within the cutoff. By contrast, the optimal solver frequently fails to finish within the 5-minute limit, underscoring DAO's advantage in large-scale NL problems. Overall, these results highlight DAO's robustness and scalability across diverse Nested Logit instances

C.1.1 CAPACITY-CONSTRAINED

		$G = 2$		$G = 20$		$G = 200$	
n	K	DAO	SFESS	DAO	SFESS	DAO	SFESS
10	3	2.35 ± 4.06	–	–	–	–	–
100	10	3.90 ± 3.47	–	2.63 ± 1.69	–	–	–
100	30	0.00 ± 0.00	–	0.00 ± 0.00	–	–	–
1000	10	3.41 ± 2.75	–	4.37 ± 2.18	–	7.01 ± 2.45	–
1000	30	2.90 ± 1.79	–	2.38 ± 0.91	–	3.98 ± 0.51	–

Table 4: Nested Logit (NL) with capacity constraints. Values report the percentage gap $\pm$ standard deviation relative to the optimal solutions across (n, K, G). Entries marked "–" in the upper-right blocks correspond to settings where $n < G$, which are not meaningful for NL. Entries marked "–" in the last row correspond to cases where the classical solver required to compute the optimal solution did not finish within the 5-minute time limit.

DAO consistently delivers tighter gaps at larger problem scales. In contrast, the optimal solver often fails to finish within the 5-minute cutoff when $n = 1000$, highlighting DAO's advantage in large-scale instances.

		$G = 2$		$G = 20$		$G = 200$	
n	K	DAO	SFESS	DAO	SFESS	DAO	SFESS
1000	100	<0.04	–	<0.05	–	<0.09	–

Table 5: Nested Logit (NL) runtime comparison: ratio of DAO and SFESS runtime to that of the optimal solver (mean $\pm$ standard deviation). For large-scale settings ($n = 1000$), DAO achieves competitive accuracy while maintaining significantly lower runtime ratios.

C.1.2 UNCONSTRAINED

C.2 BASIC ATTRACTION MODEL

Our final specification employs the Basic Attraction Model (BAM) as a multinomial logit baseline. We set the attraction of the outside option to $a_0 = 10.0$ in all instances to provide a consistent calibration of the probability of choice.

For the BAM model, we can observe that DAO achieves competitive accuracy in unconstrained setting. However,when dealing with simple product choices without distinguishing between categories,

n	$G = 2$	$G = 20$	$G = 200$
10	0.00 ± 0.00	–	–
100	0.01 ± 0.02	0.02 ± 0.03	–
1000	1.53 ± 0.18	1.58 ± 0.18	1.62 ± 0.24

Table 6: Nested Logit (NL) without capacity constraints. Values report the percentage gap $\pm$ standard deviation relative to the optimal solutions across (n, G). Entries marked "–" indicate parameter combinations that are not meaningful (e.g., $n < G$).

	$n = 10$		$n = 100$		$n = 1000$	
Setting	DAO	SFESS	DAO	SFESS	DAO	SFESS
Unconstrained	0.00 ± 0.00	–	0.04 ± 0.04	-	1.57 ± 0.35	–
$K = 3$	0.61 ± 1.29	-0.00 ± 0.00	8.27 ± 4.58	-	4.52 ± 1.99	–
$K = 10$	–	–	1.49 ± 0.84	-0.00 ± 0.00	2.74 ± 0.81	-0.00 ± 0.00
$K = 30$	–	–	0.02 ± 0.02	-0.00 ± 0.00	2.20 ± 0.53	–
$K = 100$	–	–	–	–	2.26 ± 0.51	–

Table 7: Comparison of DAO and SFESS under the Basic Attraction Model (BAM). The first row shows the unconstrained version, followed by results with capacity constraints at different K. Values report the percentage gap $\pm$ standard deviation with respect to the optimal solutions across (n, K).

the difference in the number of model parameters has a significant impact on the final results. Moreover, for the DAO model, as n increases, the difference between the DAO model and the classical BAM model can only be reduced when K increases. A potential reason for this is that as K grows, it becomes closer to the number of products selected in the unconstrained BAM case.

D LLM USAGE

In the preparation of this work, the authors used a large language model (LLM) primarily for two purposes: (1) polishing and refining the writing of certain parts of the paper to improve clarity and readability, and (2) assisting in generating portions of boilerplate implementation code. The LLM was not involved in formulating research ideas, deriving theoretical results, conducting experiments, or interpreting findings. All text and code produced with the aid of the LLM were carefully reviewed, corrected as necessary, and verified by the authors. This LLM usage was limited to supporting roles and did not influence the scientific contributions or conclusions of the work.