

---

# Optimal Continuous DR-Submodular Maximization and Applications to Provable Mean Field Inference

---

Yatao A. Bian<sup>†1</sup> Joachim M. Buhmann<sup>1</sup> Andreas Krause<sup>1</sup>

## Abstract

Mean field inference in probabilistic models is generally a highly nonconvex problem. Existing optimization methods, e.g., coordinate ascent algorithms, typically only find local optima.

In this work we propose provable mean field methods for probabilistic log-submodular models and its posterior agreement (PA) with strong approximation guarantees. The main algorithmic technique is a new Double Greedy scheme, termed DR-DoubleGreedy, for *continuous* DR-submodular maximization with box-constraints. This one-pass algorithm achieves the *optimal*  $1/2$  approximation ratio, which may be of independent interest. We validate the superior performance of our algorithms with baseline results on both synthetic and real-world datasets.

## 1. Introduction

Consider the following scenario: You want to build a recommender system for  $n$  products to sell. Let  $\mathcal{V}$  contain all the products. The system is expected to recommend a subset of products  $S \subseteq \mathcal{V}$  to the user. This recommendation should reflect relevance and diversity of the user’s choice, such that it will raise the readiness to buy. The two most important components in building such a system are (1) learning a utility function  $F(S)$ , which measures the utility of any subset of products, and (2) inference, i.e., finding the subset  $\Omega^*$  with the highest utility given the learnt function  $F(S)$ . The above task can be achieved using a class of probabilistic graphical models that devise a distribution on all subsets of  $\mathcal{V}$ . Such a distribution is known as a point process. Specifically, it defines  $p(S) \propto \exp(F(S))$ , which renders subsets of products  $S$  with high utility to be very likely sug-

gested. In general, inference in point processes is #P-hard. One resorts to approximate inference methods via either variational techniques such as mean field approximations (Wainwright et al., 2008) or sampling.

In this paper we develop mean field methods with provable guarantees. Both of the two components in the recommender system example above can be achieved via provable mean field methods since (i) the latter provide approximate inference given a utility function and, (ii) by using proper differentiation techniques, the iterative process of mean field approximation can be unrolled to serve as a differentiable layer (Zheng et al., 2015), thus enabling the backpropagation of the training error to parameters of  $F(S)$ . Thereby, learning  $F(S)$  in an end-to-end fashion can utilize modern deep learning and stochastic optimization techniques.

The most important property which we require on  $F(S)$  is *submodularity*, which naturally models relevance and diversity. Djolonga & Krause (2014) have used submodular functions  $F(S)$  to define two classes of point processes:  $p(S) \propto \exp(F(S))$  is termed probabilistic log-submodular models, while  $p(S) \propto \exp(-F(S))$  is called probabilistic log-supermodular models. They are strict generalizations of classical point processes, such as DPPs (Kulesza et al., 2012). The variational techniques from Djolonga & Krause (2014) and Djolonga et al. (2016) focus on giving tractable upper bounds of the log-partition functions. Our work provides provable *lower* bounds through mean field approximation, which also completes the picture of variational inference for probabilistic submodular models (PSMs).

The most frequently used algorithm for mean field approximation is the `CoordinateAscent` algorithm<sup>1</sup>. It maximizes the ELBO objective in a coordinate-wise manner. `CoordinateAscent` has been shown to reach stationary points/local optima. However, local optima may be arbitrarily poor, as we demonstrate in §A, and `CoordinateAscent` can get stuck in these poor local optima without extra techniques, which motivates our pursuit to develop provable methods.

We first investigate the properties of mean field approxi-

---

<sup>†</sup>Also known as An Bian. ORCID: [orcid.org/0000-0002-2368-4084](https://orcid.org/0000-0002-2368-4084). <sup>1</sup>Department of Computer Science, ETH Zurich, Zurich, Switzerland. Correspondence to: Yatao A. Bian <ybian@inf.ethz.ch>.

---

<sup>1</sup>It is known under various names in the literature, e.g., iterated conditional modes (ICM), naive mean field algorithm, etc.

mation for probabilistic log-submodular models, and show that it falls into a general class of nonconvex problems, called *continuous DR-submodular maximization with box-constraints*. Continuous submodular optimization is a class of well-behaved nonconvex programs, which has attracted considerable attention recently. Then we propose a new one-epoch algorithm for this general class of nonconvex problem, called DR-DoubleGreedy. It achieves the *optimal*  $1/2$  approximation ratio in linear time. Lastly, we extend one-epoch algorithms to multiple epochs, resulting in provable mean field algorithms, termed DG-MeanField.

**Typical Application Domains.** Recommender systems are just one illustrating example. There are numerous scenarios that can benefit from the mean field method in this work. These settings include, but not limited to, existing applications of submodular models, such as diversity models (Tschitschek et al., 2016; Djolonga et al., 2016), experimental design using approximate submodular objectives (Bian et al., 2017b), variable selection (Krause & Guestrin, 2005), data summarization (Lin & Bilmes, 2011a), dictionary learning (Krause & Cevher, 2010) etc. Another category of applications is conducting model validation using information-theoretic criteria. In order to infer the hyperparameters in the model  $F(S)$ , practitioners do validation by splitting the training data into multiple folds, and then train models on them. *Posterior-Agreement* (PA, Buhmann (2010); Bian et al. (2016)) provides an information-theoretic criterion for the models trained on these folds, to measure the fitness of one specific hyperparameter configuration. We show in §2.2 that PA can be efficiently approximated by the techniques developed in this work.

**Contributions.** We contribute in the following respects: i) We propose an *optimal* algorithm for the general problem of continuous DR-submodular maximization with box-constraints, which runs in linear time. ii) Based on the optimal algorithm, we propose provable mean field approaches for probabilistic log-submodular models and its PA. iii) We also present efficient polynomial methods to evaluate the multilinear extensions for a large category of practical objectives, which are used for optimizing the mean field objectives. iv) Extensive experimental evaluations on real-world and synthetic data support our theory.

### 1.1. Problem Statement and Related Work

**Notation.** Boldface letters, e.g.  $\mathbf{x}$ , represent vectors. Boldface capital letters, e.g.  $\mathbf{A}$ , denote matrices.  $x_i$  is the  $i^{\text{th}}$  entry of  $\mathbf{x}$ ,  $A_{ij}$  the  $(ij)^{\text{th}}$  entry of  $\mathbf{A}$ . We use  $\mathbf{e}_i$  to denote the standard  $i^{\text{th}}$  basis vector.  $f(\cdot)$  is used to specify a continuous function, and  $F(\cdot)$  to represent a set function.  $[n] := \{1, \dots, n\}$ . Given two vectors  $\mathbf{x}, \mathbf{y}$ ,  $\mathbf{x} \leq \mathbf{y}$  means  $\forall i, x_i \leq y_i$ .  $\mathbf{x} \vee \mathbf{y}$  and  $\mathbf{x} \wedge \mathbf{y}$  is defined as coordinate-wise maximum and coordinate-wise minimum, respectively.

Finally,  $\mathbf{x}|_i(k)$  is the operation of setting the  $i^{\text{th}}$  entry of  $\mathbf{x}$  to  $k$ , while keeping all the others unchanged, i.e.,  $\mathbf{x}|_i(k) = \mathbf{x} - x_i \mathbf{e}_i + k \mathbf{e}_i$ .

All of the mean field approximation problems investigated in this work fall into the following nonconvex maximization problem:

$$\underset{\mathbf{x} \in [\mathbf{a}, \mathbf{b}]}{\text{maximize}} \quad f(\mathbf{x}), \quad (\text{P})$$

where  $f : \mathcal{X} \rightarrow \mathbb{R}$  is continuous DR-submodular,  $\mathcal{X} = \prod_{i=1}^n \mathcal{X}_i$ , each  $\mathcal{X}_i$  is an interval (Bach, 2015; Bian et al., 2017c). Continuous DR-submodular functions define a subclass of continuous submodular functions with the additional diminishing returns (DR) property:  $\forall \mathbf{a} \leq \mathbf{b} \in \mathcal{X}$ ,  $\forall i \in [n], \forall k \in \mathbb{R}_+$  it holds  $f(k\mathbf{e}_i + \mathbf{a}) - f(\mathbf{a}) \geq f(k\mathbf{e}_i + \mathbf{b}) - f(\mathbf{b})$ . If  $f$  is differentiable, DR-submodularity is equivalent to  $\nabla f$  being an *antitone* mapping from  $\mathbb{R}^n$  to  $\mathbb{R}^n$ . If  $f$  is twice-differentiable, DR-submodularity is equivalent to all of the entries of  $\nabla^2 f(\mathbf{x})$  being non-positive. A function  $f$  is DR-supermodular iff  $-f$  is DR-submodular.

**Background & Related Work.** Submodularity is one of the most important properties in combinatorial optimization and has many applications for machine learning, with strong implications for both guaranteed minimization and approximate maximization in polynomial time (Krause & Golovin, 2012). Continuous extensions of submodular set functions play an important role in submodular optimization. Representative instances include the Lovász extension (Lovász, 1983), multilinear extension (Calinescu et al., 2007; Vondrák, 2008; Chekuri et al., 2014; 2015) and the softmax extension for DPPs (Gillenwater et al., 2012). Results on guaranteed optimizations have been extended to continuous domains recently, for both minimization (Bach, 2015; Staib & Jegelka, 2017) and maximization (Bian et al., 2017c;a; Wilder, 2017; Chen et al., 2018; Mokhtari et al., 2018). Specifically, Bach (2015) studies continuous submodular minimization without constraints. He also discusses the possibility of using the technique for mean field inference of probabilistic log-supermodular models. Bian et al. (2017c;a) characterize continuous submodularity using the DR property and propose provable algorithms for maximization.

Most related to this work is the classical problem of unconstrained submodular maximization (USMs), which has been studied in binary (Buchbinder et al., 2012), integer (Soma & Yoshida, 2017) and continuous domains (Bian et al., 2017c). For the general problem (P), at first glance one may consider discretization-based methods: Discretizing the continuous domain and transform (P) to be an integer optimization problem, then solve it using the reduction (Ene & Nguyen, 2016) or the integer Double Greedy algorithm (Soma & Yoshida, 2017). However, discretization-based methods are not appropriate for (P): Firstly, discretization

will inevitably introduce errors for the original continuous problem (P); Secondly, the computational cost is often high<sup>2</sup>. Thus we turn to continuous methods.

The shrunken Frank-Wolfe algorithm (Bian et al., 2017a) provides a  $1/e$  approximation guarantee and sublinear rate of convergence for (P), but it is still computationally too expensive: In each iteration it calculates the full gradient, which costs  $n$  times as much as computing a partial derivative.

Based on the above analysis, the most promising algorithm to consider would be the Double Greedy algorithm (Bian et al., 2017c), which needs to solve  $\mathcal{O}(n)$  1-D subproblems, and achieves a  $1/3$  guarantee for continuous *submodular* maximization. Since it only needs  $f(\mathbf{x})$  to be continuous submodular, we call it `Submodular-DoubleGreedy` in the sequel. In this work we propose a new Double Greedy scheme, achieving the optimal  $1/2$  approximation ratio (Bian et al., 2017c, Proposition 5) of (P).

Posterior-Agreement (PA) is developed as an information-theoretic criterion for model selection (Gorbach et al., 2017) and algorithmic validation (Gronskiy & Buhmann, 2014; Bian et al., 2016). It originates from the approximation set coding framework of Buhmann (2010). Recently, Buhmann et al. (2018) prove rigorous asymptotics of PA on two typical combinatorial problems: Sparse minimum bisection and Lawler’s quadratic assignment problem. Djolonga & Krause (2014; 2015) study variational inference for PSMs. They propose L-Field to give upper bounds for log-supermodular models through optimizing the subdifferentials.

Contemporary with the development of this work<sup>3</sup>, Niazadeh et al. (2018) proposed an optimal algorithm for DR-submodular maximization, which is based on a zero-sum game analysis. Their algorithm (Algorithm 4 of Niazadeh et al. (2018), termed BSCB: Binary-Search Continuous Bi-greedy) needs to estimate the partial derivative of the objective, which is not needed in our algorithm. Furthermore, our algorithm is arguably easier to interpret and implement than BSCB. For further comparison of these two algorithms, we provide a simple example in §C. We did extensive experiments (see §5 for details on experimental statistics) to compare them, the results show that both algorithms generate better promising solutions, however, our algorithm produces better solutions than BSCB in most of the experiments.

<sup>2</sup>e.g., the method from Soma & Yoshida (2017) reaches  $\frac{1}{2+\epsilon}$ -approximation in  $O(\frac{|V|}{\epsilon}) \log(\frac{\Delta}{\delta}) \log(B)(\theta + \log(B))$  time,  $B$ : #grids of discretization,  $\Delta$ : the maximal positive marginal gain,  $\delta$ : minimum positive marginal gain. We conducted a study of discretization-based method and defer the results to §G.2

<sup>3</sup>This work is extended from Bian et al. (2018).

## 2. Applications to Mean Field Approximation

### 2.1. Classical Mean Field Inference

Mean field inference aims to approximate the intractable distribution  $p(S) \propto \exp(F(S))$  by a fully factorized surrogate distribution  $q(S|\mathbf{x}) := \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j)$ ,  $\mathbf{x} \in [0, 1]^n$ . This can be achieved by maximizing the (ELBO) objective, which provides a lower bound for the log-partition function,  $(\text{ELBO}) \leq \log Z = \log \sum_{S \subseteq V} \exp(F(S))$ . Specifically, the optimization problem is,

$$\begin{aligned} \max_{\mathbf{x} \in [0, 1]} f(\mathbf{x}) &:= \overbrace{\mathbb{E}_{q(S|\mathbf{x})}[F(S)]}^{\text{multilinear extension of } F(S): f_{\text{mt}}(\mathbf{x})} \\ &= \sum_{i=1}^n [x_i \log x_i + (1 - x_i) \log(1 - x_i)] \\ &= f_{\text{mt}}(\mathbf{x}) + \sum_{i \in V} H(x_i), \quad (\text{ELBO}) \quad (1) \end{aligned}$$

where  $H(x_i) := -[x_i \log x_i + (1 - x_i) \log(1 - x_i)]$  is the binary entropy function and by default  $0 \log 0 = 0$ .  $f_{\text{mt}}(\mathbf{x}) := \mathbb{E}_{q(S|\mathbf{x})}[F(S)]$  is the multilinear extension of  $F(S)$ . The above (ELBO) is continuous DR-submodular w.r.t.  $\mathbf{x}$ , thus falling into the general problem class (P). At first glance,  $f_{\text{mt}}(\mathbf{x})$  seems to require an exponential number of operations for evaluation; we show in §4 that  $f_{\text{mt}}(\mathbf{x})$  and its gradients can be computed precisely in polynomial time for many classes of practical objectives, such as facility location, FLID (Tschitschek et al., 2016), set cover (Lin & Bilmes, 2011b) and graph cuts. Maximizing (ELBO) to optimality provides the tightest lower bound of  $\log Z$  in terms of the KL divergence  $\mathbb{KL}(q||p)$ . We put details in §D.

### 2.2. Mean Field Inference of Posterior-Agreement (PA)

In addition to the traditional mean field objective (ELBO) in (1), here we further formulate a second class of mean field objectives. They come from Posterior-Agreement (PA) for probabilistic log-submodular models, which is an information-theoretic criterion to conduct model and algorithmic validation (Buhmann, 2010; Buhmann et al., 2018; Bian et al., 2016).

Let us again consider the recommender system example: usually there are some hyperparameters in the model/utility function  $F(S)$  that require adaptation to the input data. One natural way to do so is through model validation: Split the training data into multiple folds, train a model on each fold  $D$  and infer a “noisy” posterior distribution  $p(S|D)$ . PA measures the agreement between these “noisy” posterior distributions.

Assume w.l.o.g. that there are two folds of data  $D'$ ,  $D''$  in the sequel. In the PA framework, we have two consecutive targets: 1) Direct inference based on the two posterior distributions  $p(S|D')$  and  $p(S|D'')$ . This task amounts to find the MAP solution of the PA distribution (which is discussed

in the next paragraph), it can be approximated by standard mean field inference. 2) Use the PA objective (3) as a criterion for model validation/selection. Since in general the PA objective (3) is intractable, we will still use mean field lower bounds and some upper bounds of Djolonga & Krause (2014) to obtain estimates for it.

### Mean Field Approximation of the Posterior-Agreement Distribution.

A probabilistic log-submodular model is a special case of a Gibbs random field with unit temperature and  $-F(S)$  as the energy function. In PA framework, we explicitly keep  $\beta$  as the inverse temperature,  $p_\beta(S|D) := \frac{\exp(\beta F(S|D))}{\sum_{\tilde{S} \subseteq V} \exp(\beta F(\tilde{S}|D))}$ ,  $\forall S \subseteq V$ , where  $D$  is the dataset used to train the model  $F(S|D)$ . The PA distribution is defined as,

$$p^{\text{PA}}(S) \propto p_\beta(S|D') p_\beta(S|D'') \propto \exp[\beta(F(S|D') + F(S|D''))].$$

Note that its log partition function is still intractable. In order to approximate  $p^{\text{PA}}(S)$ , we use mean field approximation with a surrogate distribution  $q(S|\mathbf{x}) := \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j)$ ,

$$\begin{aligned} \log Z^{\text{PA}} &= \log \sum_{S \subseteq V} \exp[\beta(F(S|D') + F(S|D''))] \\ &\geq \beta \mathbb{E}_{q(S|\mathbf{x})}[F(S|D')] + \beta \mathbb{E}_{q(S|\mathbf{x})}[F(S|D'')] \quad (2) \\ &\quad + \sum_{i \in V} H(x_i). \quad (\text{PA-ELBO}) \end{aligned}$$

Maximizing (PA-ELBO) in (2) still falls into the general problem class (P) (see § D for details). Maximizing (PA-ELBO) also serves as a building block for the second target below.

### Lower Bounds for the Posterior-Agreement Objective.

The PA objective is used to measure the agreement between two posterior distributions motivated by an information-theoretic analogy (Buhmann et al., 2018; Bian et al., 2016). By introducing the same surrogate distribution  $q(S|\mathbf{x})$ , one can easily derive that,

$$\begin{aligned} \log \sum_{S \subseteq V} p_\beta(S|D') p_\beta(S|D'') \quad (\text{PA objective}) \quad (3) \\ \geq \underbrace{\mathbb{H}(q) + \beta \mathbb{E}_q F(S|D') + \beta \mathbb{E}_q F(S|D'')}_{(\text{PA-ELBO}) \text{ in (2)}} \\ - \log Z(\beta; D') - \log Z(\beta; D'') \end{aligned}$$

where  $\mathbb{H}(q)$  is the entropy of  $q$ ,  $Z(\beta; D')$  and  $Z(\beta; D'')$  are the partition functions of the two noisy distributions, respectively. In order to find the best lower bound for PA objective, one need to maximize w.r.t.  $q(S|\mathbf{x})$  the (PA-ELBO) objective, at the same time, find the upper bounds for  $\log Z(\beta; D') + \log Z(\beta; D'')$ . The latter can be achieved using techniques from Djolonga & Krause (2014). We summarize the details in § E to make it self-contained.

## 3. An Optimal Algorithm for Continuous DR-Submodular Maximization

Unfortunately, problem (P) is generally hard: The 1/2 hardness result (Bian et al., 2017c, Proposition 5) can be easily translated to (P) with details deferred to § B.1. The following question arises naturally: Is it possible to achieve the optimal 1/2 approximation ratio (unless  $\text{RP}=\text{NP}$ ) by properly utilizing the extra DR property in (P)? To affirmatively answer this question, we propose a new Double Greedy algorithm for continuous DR-submodular maximization called DR-DoubleGreedy and prove a 1/2 approximation ratio.

### 3.1. A Deterministic 1/2-Approximation for Continuous DR-Submodular Maximization

---

#### Algorithm 1: DR-DoubleGreedy( $f, \mathbf{a}, \mathbf{b}$ )

---

**Input:**  $\max_{\mathbf{x} \in [\mathbf{a}, \mathbf{b}]} f(\mathbf{x})$ ,  $f(\mathbf{x})$  is DR-submodular,  $[\mathbf{a}, \mathbf{b}] \subseteq \mathcal{X}$ ; # coordinates  $n$ ; additive error level  $\delta \geq 0$ .

```

1  $\mathbf{x}^0 \leftarrow \mathbf{a}, \mathbf{y}^0 \leftarrow \mathbf{b}$ ;
2 for  $k = 1 \rightarrow n$  do
3   let  $v_k$  be the coordinate being operated;
4   find  $u_a$  such that
       $f(\mathbf{x}^{k-1}|_{v_k}(u_a)) \geq \max_{u'} f(\mathbf{x}^{k-1}|_{v_k}(u')) - \frac{\delta}{n}$ ,
5    $\delta_a \leftarrow f(\mathbf{x}^{k-1}|_{v_k}(u_a)) - f(\mathbf{x}^{k-1})$ ;
6   find  $u_b$  such that
       $f(\mathbf{y}^{k-1}|_{v_k}(u_b)) \geq \max_{u'} f(\mathbf{y}^{k-1}|_{v_k}(u')) - \frac{\delta}{n}$ ,
7    $\delta_b \leftarrow f(\mathbf{y}^{k-1}|_{v_k}(u_b)) - f(\mathbf{y}^{k-1})$ ;
8    $\mathbf{x}^k \leftarrow \mathbf{x}^{k-1}|_{v_k}(\frac{\delta_a}{\delta_a + \delta_b} u_a + \frac{\delta_b}{\delta_a + \delta_b} u_b)$ ; //update  $\mathbf{x}_k^{\text{th}}$ 
      coordinate to be a convex combination of  $u_a$  &  $u_b$ 
9    $\mathbf{y}^k \leftarrow \mathbf{y}^{k-1}|_{v_k}(\frac{\delta_a}{\delta_a + \delta_b} u_a + \frac{\delta_b}{\delta_a + \delta_b} u_b)$ ;

```

**Output:**  $\mathbf{x}^n$  or  $\mathbf{y}^n$  ( $\mathbf{x}^n = \mathbf{y}^n$ )

---

The pseudocode of DR-DoubleGreedy as summarized in Alg. 1 describes a one-epoch algorithm, sweeping over the  $n$  coordinates in one pass. Like the previous Double Greedy algorithms, the procedure maintains two solutions  $\mathbf{x}, \mathbf{y}$ , that are initialized as the lower bound  $\mathbf{a}$  and the upper bound  $\mathbf{b}$ , respectively. In iteration  $k$ , it operates on coordinate  $v_k$ , and solves the two 1-D subproblems  $\max_{u'} f(\mathbf{x}^{k-1}|_{v_k}(u'))$  and  $\max_{u'} f(\mathbf{y}^{k-1}|_{v_k}(u'))$ , based on  $\mathbf{x}^{k-1}$  and  $\mathbf{y}^{k-1}$ , respectively. It also allows solving 1-D subproblems approximately with additive error  $\delta \geq 0$  ( $\delta = 0$  recovers the error-free case). Let  $u_a$  and  $u_b$  be the solutions of these 1-D subproblems.

Unlike previous Double Greedy algorithms, we change coordinate  $v_k$  of  $\mathbf{x}^{k-1}$  and  $\mathbf{y}^{k-1}$  to be a *convex* combination of  $u_a$  and  $u_b$ , weighted by respective gains  $\delta_a, \delta_b$ . This convex combination is the key step that utilizes the DR property of  $f$ , and it also plays a crucial role in the proof.

Note that the 1-D subproblem has a closed-form solu-



tion for ELBO (1) (and similarly for PA-ELBO (2)). For coordinate  $i$ , the partial derivative of the multilinear extension is  $\nabla_i f_{\text{mt}}(\mathbf{x})$ , and for the entropy term, it is  $\nabla H(x_i) = \log \frac{1-x_i}{x_i}$ . Then  $x_i$  should be updated as  $x_i \leftarrow \sigma(\nabla_i f_{\text{mt}}(\mathbf{x})) = (1 + \exp(-\nabla_i f_{\text{mt}}(\mathbf{x})))^{-1}$ , where  $\sigma$  is the logistic sigmoid function.

**Theorem 1.** *Assume the optimal solution of  $\max_{\mathbf{x} \in [a, b]} f(\mathbf{x})$  is  $\mathbf{x}^*$ , then for Alg. 1 it holds,*

$$f(\mathbf{x}^n) \geq \frac{1}{2}f(\mathbf{x}^*) + \frac{1}{4}[f(\mathbf{a}) + f(\mathbf{b})] - \frac{5\delta}{4}. \quad (4)$$

**Proof Sketch.** The high level proof strategy is to bound the change of an intermediate variable  $\mathbf{o}^k := (\mathbf{x}^* \vee \mathbf{x}^k) \wedge \mathbf{y}^k$  through the course of Alg. 1, which is the common framework in the analysis of existing Double Greedy variants (Buchbinder et al., 2012; Gottschalk & Peis, 2015; Bian et al., 2017c; Soma & Yoshida, 2017). The novelty of our method results from the update of  $\mathbf{x}$ ,  $\mathbf{y}$ , which plays a key role in achieving the optimal 1/2 approximation ratio. Furthermore, in the analysis we find a way to utilize the DR property directly, resulting in a succinct proof. We document the details in §B.2, and summarize a sketch here. Firstly, using DR-submodularity, we prove that in each iteration, if we were to flip the 1-D subproblem solutions of  $\mathbf{x}$  and  $\mathbf{y}$ , it still does not decrease the function value (in the error-free case  $\delta = 0$ ).

**Lemma 1.** *For all  $k = 1, \dots, n$ , it holds that,*

$$\begin{aligned} f(\mathbf{x}^{k-1}|_{v_k(u_b)}) - f(\mathbf{x}^{k-1}) &\geq -\delta/n, \\ f(\mathbf{y}^{k-1}|_{v_k(u_a)}) - f(\mathbf{y}^{k-1}) &\geq -\delta/n. \end{aligned} \quad (5)$$

Then using the new update rule and the DR property, we show that the loss on intermediate variables  $f(\mathbf{o}^{k-1}) - f(\mathbf{o}^k)$  can be upper bounded by the increase of the objective value in  $\mathbf{x}$  and  $\mathbf{y}$  times 1/2.

**Lemma 2.** *For all  $k = 1, \dots, n$ , it holds that,*

$$\begin{aligned} f(\mathbf{o}^{k-1}) - f(\mathbf{o}^k) \\ \leq \frac{1}{2} [f(\mathbf{x}^k) - f(\mathbf{x}^{k-1}) + f(\mathbf{y}^k) - f(\mathbf{y}^{k-1})] + \frac{2.5\delta}{n}. \end{aligned} \quad (6)$$

Given Lemma 2, let us sum for  $k = 1, \dots, n$ . After rearrangement we reach the final conclusion.

### 3.2. Multi-epoch Extensions

Though DR-DoubleGreedy reaches the optimal 1/2 guarantee with one epoch, in practice it usually helps to use its output as an initializer, and continue coordinate-wise optimization for additional epochs. Since each step of coordinate update will never decrease the function value, the approximation guarantees will hold. We call this class of algorithms DoubleGreedy-MeanField, abbreviated as

DG-MeanField, and summarize the pseudocode in Alg. 2.

---

**Algorithm 2:** DG-MeanField-1/2 & DG-MeanField-1/3

---

**Input:**  $\max_{\mathbf{x} \in [a, b]} f(\mathbf{x})$ , e.g., from the ELBO (1) or PA-ELBO (2) objective

---

- 1 Option I: DG-MeanField-1/3: run Submodular-DoubleGreedy (Bian et al., 2017c) to get a 1/3 initializer  $\hat{\mathbf{x}}$ ;
  - 2 Option II: DG-MeanField-1/2: run DR-DoubleGreedy to get a 1/2 initializer  $\hat{\mathbf{x}}$ ;
  - 3 beginning with  $\hat{\mathbf{x}}$ , optimize  $f(\mathbf{x})$  coordinate by coordinate for  $T$  epochs;
- 

## 4. Efficient Methods for Calculating Multilinear Extension & its Gradients

In this section we present guaranteed methods to efficiently calculate the multilinear extension  $f_{\text{mt}}(\mathbf{x})$  and its gradients in polynomial time<sup>4</sup>. Remember that the multilinear extension is the expected value of  $F(S)$  under the surrogate distribution:  $f_{\text{mt}}(\mathbf{x}) := \mathbb{E}_{q(S|\mathbf{x})}[F(S)] = \sum_{S \subseteq \mathcal{V}} F(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j)$ . One can verify that the partial derivative of  $f_{\text{mt}}(\mathbf{x})$  is,

$$\begin{aligned} \nabla_i f_{\text{mt}}(\mathbf{x}) &= \mathbb{E}_{q(S|\mathbf{x}, x_i=1)}[F(S)] - \mathbb{E}_{q(S|\mathbf{x}, x_i=0)}[F(S)] \\ &= f_{\text{mt}}(\mathbf{x}|_i(1)) - f_{\text{mt}}(\mathbf{x}|_i(0)) \\ &= \sum_{S \subseteq \mathcal{V}, S \ni i} F(S) \prod_{j \in S \setminus \{i\}} x_j \prod_{j' \notin S} (1 - x_{j'}) \\ &\quad - \sum_{S \subseteq \mathcal{V} \setminus \{i\}} F(S) \prod_{j \in S} x_j \prod_{j' \notin S, j' \neq i} (1 - x_{j'}). \end{aligned}$$

### 4.1. Gibbs Random Fields with Finite Order of Interactions

Let us use  $\mathbf{v} \in \{0, 1\}^{\mathcal{V}}$  to equivalently denote the  $n$  binary random variables.  $F(\mathbf{v})$  corresponds to the negative energy function in Gibbs random fields. If the energy function is parameterized with a finite order of interactions, i.e.,  $F(\mathbf{v}) = \sum_{s \in \mathcal{V}} \theta_s v_s + \sum_{(s,t) \in \mathcal{V} \times \mathcal{V}} \theta_{s,t} v_s v_t + \dots + \sum_{(s_1, s_2, \dots, s_d)} \theta_{s_1, s_2, \dots, s_d} v_{s_1} \dots v_{s_d}$ ,  $d < \infty$ , then one can verify that its multilinear extension has the following closed form,

$$\begin{aligned} f_{\text{mt}}(\mathbf{x}) &= \sum_{s \in \mathcal{V}} \theta_s x_s + \sum_{(s,t) \in \mathcal{V} \times \mathcal{V}} \theta_{s,t} x_s x_t + \dots \\ &\quad + \sum_{(s_1, s_2, \dots, s_d)} \theta_{s_1, s_2, \dots, s_d} x_{s_1} \dots x_{s_d}. \end{aligned} \quad (7)$$

<sup>4</sup>Iyer & Bilmes (2015) give closed-form expressions for the partition functions of submodular point processes for several classes of objectives, which can be treated as the multilinear extensions evaluated at  $0.5 * \mathbf{1}$  with proper scaling.

The gradient of this expression can also be easily derived. Given this observation, one can quickly derive the multilinear extensions of a large category of energy functions of Gibbs random fields, e.g., graph cut, hypergraph cut, Ising models, etc. Details are in §F.

#### 4.2. Facility Location & FLID (Facility Location Diversity)

FLID is a diversity model (Tschitschek et al., 2016) that has been designed as a computationally efficient alternative to DPPs. It is in a more general form than facility location. Let  $\mathbf{W} \in \mathbb{R}_+^{|\mathcal{V}| \times D}$  be the weights, each row corresponds to the latent representation of an item, with  $D$  as the dimensionality. Then

$$\begin{aligned} F(S) &:= \sum_{i \in S} u_i + \sum_{d=1}^D (\max_{i \in S} W_{i,d} - \sum_{i \in S} W_{i,d}) \\ &= \sum_{i \in S} u'_i + \sum_{d=1}^D \max_{i \in S} W_{i,d}, \end{aligned} \quad (8)$$

which models both coverage and diversity, and  $u'_i = u_i - \sum_{d=1}^D W_{i,d}$ . If  $u'_i = 0$ , one recovers the facility location objective. The computational complexity of evaluating its partition function is  $\mathcal{O}(|\mathcal{V}|^{D+1})$  (Tschitschek et al., 2016), which is exponential in terms of  $D$ .

We now show the technique such that  $f_{\text{mt}}(\mathbf{x})$  and  $\nabla_i f_{\text{mt}}(\mathbf{x})$  can be evaluated in  $\mathcal{O}(Dn^2)$  time. Firstly, for one  $d \in [D]$ , let us sort  $W_{i,d}$  such that  $W_{i_d(1),d} \leq W_{i_d(2),d} \leq \dots \leq W_{i_d(n),d}$ . After this sorting, there are  $D$  permutations to record:  $i_d(l), l = 1, \dots, n, \forall d \in [D]$ . Now, one can verify that,

$$\begin{aligned} f_{\text{mt}}(\mathbf{x}) &= \sum_{i \in [n]} u'_i x_i + \sum_d \sum_{S \subseteq \mathcal{V}} \max_{i \in S} W_{i,d} \prod_{m \in S} x_m \prod_{m' \notin S} (1 - x_{m'}) \\ &= \sum_{i \in [n]} u'_i x_i + \sum_d \sum_{l=1}^n W_{i_d(l),d} x_{i_d(l)} \prod_{m=l+1}^n [1 - x_{i_d(m)}]. \end{aligned}$$

Sorting costs  $\mathcal{O}(Dn \log n)$ , and from the above expression, one can see that the cost of evaluating  $f_{\text{mt}}(\mathbf{x})$  is  $\mathcal{O}(Dn^2)$ . By the relation that  $\nabla_i f_{\text{mt}}(\mathbf{x}) = f_{\text{mt}}(\mathbf{x}|_i(1)) - f_{\text{mt}}(\mathbf{x}|_i(0))$ , the cost is also  $\mathcal{O}(Dn^2)$ . For  $\nabla_i f_{\text{mt}}(\mathbf{x})$ , there exists a refined way to calculate this derivative, which we explain in §F.

#### 4.3. Set Cover Functions

Suppose there are  $|C| = \{c_1, \dots, c_{|C|}\}$  concepts, and  $n$  items in  $\mathcal{V}$ . Give a set  $S \subseteq \mathcal{V}$ ,  $\Gamma(S)$  denotes the set of concepts covered by  $S$ . Given a modular function  $\mathbf{m} : 2^C \mapsto \mathbb{R}_+$ , the set cover function is defined as  $F(S) = \mathbf{m}(\Gamma(S))$ . This function models coverage in maximization, and also the notion of complexity in minimization problems

(Lin & Bilmes, 2011b). Let us define an inverse map  $\Gamma^{-1}$ , such that for each concept  $c$ ,  $\Gamma^{-1}(c)$  denotes the set of items  $v$  such that  $\Gamma^{-1}(c) \ni v$ . So the multilinear extension is,

$$\begin{aligned} f_{\text{mt}}(\mathbf{x}) &= \sum_{i \in \mathcal{V}} \mathbf{m}(\Gamma(S)) \prod_{m \in S} x_m \prod_{m' \notin S} (1 - x_{m'}) \\ &= \sum_{c \in C} m_c \left[ 1 - \prod_{i \in \Gamma^{-1}(c)} (1 - x_i) \right]. \end{aligned} \quad (9)$$

The last equality is achieved by considering the situations where a concept  $c$  is covered. One can observe that both  $f_{\text{mt}}(\mathbf{x})$  and  $\nabla_i f_{\text{mt}}(\mathbf{x})$  can be evaluated in  $\mathcal{O}(n|C|)$  time.

#### 4.4. General Case: Approximation by Sampling

In the most general case, one may only have access to the function values of  $F(S)$ . In this scenario, one can use a polynomial number of sample steps to estimate  $f_{\text{mt}}(\mathbf{x})$  and its gradients. Specifically: 1) Sample  $k$  times  $S \sim q(S|\mathbf{x})$  and evaluate function values for them, resulting in  $F(S_1), \dots, F(S_k)$ . 2) Return the average  $\frac{1}{k} \sum_{i=1}^k F(S_i)$ . According to the Hoeffding bound (Hoeffding, 1963), one can easily derive that  $\frac{1}{k} \sum_{i=1}^k F(S_i)$  is arbitrarily close to  $f_{\text{mt}}(\mathbf{x})$  with increasingly more samples: With probability at least  $1 - \exp(-k\epsilon^2/2)$ , it holds that  $|\frac{1}{k} \sum_{i=1}^k F(S_i) - f_{\text{mt}}(\mathbf{x})| \leq \epsilon \max_S |F(S)|$ , for all  $\epsilon > 0$ .

### 5. Experiments

The objectives under investigation are ELBO (1) and PA-ELBO (2). We set  $\beta = 1$  in PA-ELBO. We tested on the representative FLID model the following algorithms and baselines: The first category is one-epoch algorithms, including ① Submodular-DoubleGreedy (Bian et al., 2017c) with 1/3 approximation guarantee, ② BSCB (Alg. 4 of Niazadeh et al. (2018), where we chose  $\epsilon = 10^{-3}$ ) with 1/2 guarantee and ③ DR-DoubleGreedy (Alg. 1) with 1/2 guarantee.

The second category contain multiple-epoch algorithms: ④ CoordinateAscent-0: initialized as  $\mathbf{0}$  and coordinate-wisely improving the solution; CoordinateAscent-1: initialized as  $\mathbf{1}$ ; CoordinateAscent-Random: initialized as a uniform vector  $U(\mathbf{0}, \mathbf{1})$ . ⑤ DG-MeanField-1/3. ⑥ DG-MeanField-1/2 from Alg. 2. ⑦ BSCB-Multiepoche, which is the multi-epoch extension of BSCB: After the first epoch, it continues to improve the solution coordinate-wisely.

For all algorithms, we use the same random order to process the coordinates within each epoch. We are trying to understand: 1) In terms of DR-submodular maximization, how good are the solutions returned by one-epoch algorithms? 2) For multi-epoch algorithms, how good are the realized lower bounds? For small scale problems we can calculate the true log-partitions exhaustively, which serves as a

Table 1: Summary of results on ELBO objective (1) and PA-ELBO objective (2). Sub-DG stands for Submodular-DoubleGreedy, DR-DG stands for DR-DoubleGreedy. Boldface numbers indicate the best mean of function values returned. For ELBO, the mean and standard deviation were calculated for 10 FLID models trained on 10 folds of the data. For PA-ELBO, the mean and standard deviation were calculated for models trained over 45 pairs of folds. More details are in the experimental section.

Category	$D$	ELBO objective (1)			PA-ELBO objective (2)		
		Sub-DG	BSCB	DR-DG	Sub-DG	BSCB	DR-DG
furniture $n=32$	2	2.078 $\pm$ 0.091	2.771 $\pm$ 0.123	<b>3.035</b> $\pm$ 0.059	0.918 $\pm$ 0.768	2.287 $\pm$ 0.399	<b>2.402</b> $\pm$ 0.159
	3	1.835 $\pm$ 0.156	2.842 $\pm$ 0.128	<b>3.026</b> $\pm$ 0.099	1.296 $\pm$ 1.176	2.536 $\pm$ 0.439	<b>2.693</b> $\pm$ 0.181
	10	1.375 $\pm$ 0.194	<b>2.951</b> $\pm$ 0.161	2.917 $\pm$ 0.103	1.504 $\pm$ 1.110	2.764 $\pm$ 0.405	<b>2.882</b> $\pm$ 0.248
carseats $n=34$	2	2.089 $\pm$ 0.166	2.863 $\pm$ 0.090	<b>3.045</b> $\pm$ 0.069	1.218 $\pm$ 1.189	2.106 $\pm$ 0.308	<b>2.383</b> $\pm$ 0.173
	3	1.890 $\pm$ 0.146	3.003 $\pm$ 0.110	<b>3.138</b> $\pm$ 0.082	1.598 $\pm$ 1.422	2.413 $\pm$ 0.397	<b>2.718</b> $\pm$ 0.194
	10	1.390 $\pm$ 0.232	<b>3.100</b> $\pm$ 0.140	3.003 $\pm$ 0.157	1.975 $\pm$ 1.587	2.670 $\pm$ 0.332	<b>2.957</b> $\pm$ 0.244
safety $n=36$	2	1.934 $\pm$ 0.402	2.727 $\pm$ 0.212	<b>2.896</b> $\pm$ 0.098	1.263 $\pm$ 1.187	2.088 $\pm$ 0.290	<b>2.335</b> $\pm$ 0.121
	3	1.867 $\pm$ 0.453	2.830 $\pm$ 0.191	<b>2.970</b> $\pm$ 0.110	1.477 $\pm$ 1.150	2.359 $\pm$ 0.354	<b>2.636</b> $\pm$ 0.158
	10	1.546 $\pm$ 0.606	2.916 $\pm$ 0.191	<b>2.920</b> $\pm$ 0.149	1.716 $\pm$ 1.209	2.507 $\pm$ 0.287	<b>2.723</b> $\pm$ 0.138
strollers $n=40$	2	2.042 $\pm$ 0.181	2.829 $\pm$ 0.144	<b>2.928</b> $\pm$ 0.060	0.587 $\pm$ 0.673	1.883 $\pm$ 0.238	<b>2.171</b> $\pm$ 0.171
	3	1.814 $\pm$ 0.264	2.958 $\pm$ 0.146	<b>2.978</b> $\pm$ 0.077	0.837 $\pm$ 0.835	2.119 $\pm$ 0.243	<b>2.496</b> $\pm$ 0.165
	10	1.328 $\pm$ 0.544	<b>3.065</b> $\pm$ 0.162	2.910 $\pm$ 0.140	1.225 $\pm$ 1.023	2.412 $\pm$ 0.230	<b>2.706</b> $\pm$ 0.252
media $n=58$	2	3.221 $\pm$ 0.066	3.309 $\pm$ 0.055	<b>3.493</b> $\pm$ 0.051	0.372 $\pm$ 0.286	<b>1.477</b> $\pm$ 0.128	1.336 $\pm$ 0.101
	3	3.276 $\pm$ 0.082	3.492 $\pm$ 0.083	<b>3.712</b> $\pm$ 0.079	0.418 $\pm$ 0.366	1.736 $\pm$ 0.177	<b>1.762</b> $\pm$ 0.095
	10	2.840 $\pm$ 0.183	3.894 $\pm$ 0.122	<b>3.924</b> $\pm$ 0.114	0.653 $\pm$ 0.727	2.309 $\pm$ 0.244	<b>2.524</b> $\pm$ 0.130
health $n=62$	2	3.197 $\pm$ 0.067	3.174 $\pm$ 0.074	<b>3.516</b> $\pm$ 0.043	0.528 $\pm$ 0.264	<b>1.679</b> $\pm$ 0.112	1.654 $\pm$ 0.066
	3	3.231 $\pm$ 0.055	3.306 $\pm$ 0.108	<b>3.707</b> $\pm$ 0.064	0.621 $\pm$ 0.405	1.918 $\pm$ 0.144	<b>2.023</b> $\pm$ 0.085
	10	2.633 $\pm$ 0.115	3.508 $\pm$ 0.120	<b>3.675</b> $\pm$ 0.110	0.808 $\pm$ 0.755	2.256 $\pm$ 0.175	<b>2.384</b> $\pm$ 0.110
toys $n=62$	2	3.543 $\pm$ 0.047	3.454 $\pm$ 0.091	<b>3.856</b> $\pm$ 0.044	0.491 $\pm$ 0.138	<b>1.789</b> $\pm$ 0.187	1.735 $\pm$ 0.058
	3	3.362 $\pm$ 0.055	3.412 $\pm$ 0.070	<b>3.736</b> $\pm$ 0.051	0.456 $\pm$ 0.149	<b>1.794</b> $\pm$ 0.197	1.763 $\pm$ 0.065
	10	3.037 $\pm$ 0.138	3.706 $\pm$ 0.108	<b>3.859</b> $\pm$ 0.119	0.547 $\pm$ 0.301	2.218 $\pm$ 0.269	<b>2.287</b> $\pm$ 0.105
diaper $n=100$	2	3.500 $\pm$ 0.058	3.517 $\pm$ 0.058	<b>3.636</b> $\pm$ 0.043	0.295 $\pm$ 0.158	<b>1.119</b> $\pm$ 0.063	0.665 $\pm$ 0.116
	3	3.739 $\pm$ 0.080	3.753 $\pm$ 0.065	<b>3.974</b> $\pm$ 0.065	0.337 $\pm$ 0.240	<b>1.429</b> $\pm$ 0.111	1.141 $\pm$ 0.120
	10	3.423 $\pm$ 0.110	4.150 $\pm$ 0.120	<b>4.203</b> $\pm$ 0.086	0.386 $\pm$ 0.504	1.969 $\pm$ 0.201	<b>2.009</b> $\pm$ 0.199
feeding $n=100$	2	3.942 $\pm$ 0.041	3.808 $\pm$ 0.024	<b>3.970</b> $\pm$ 0.036	0.393 $\pm$ 0.034	<b>0.894</b> $\pm$ 0.022	0.501 $\pm$ 0.029
	3	4.333 $\pm$ 0.031	4.095 $\pm$ 0.032	<b>4.390</b> $\pm$ 0.031	0.503 $\pm$ 0.072	<b>1.232</b> $\pm$ 0.041	0.893 $\pm$ 0.046
	10	4.611 $\pm$ 0.053	4.553 $\pm$ 0.079	<b>4.860</b> $\pm$ 0.056	0.608 $\pm$ 0.239	1.808 $\pm$ 0.087	<b>1.820</b> $\pm$ 0.078
gear $n=100$	2	3.311 $\pm$ 0.046	3.150 $\pm$ 0.037	<b>3.430</b> $\pm$ 0.040	0.232 $\pm$ 0.068	<b>1.019</b> $\pm$ 0.048	0.590 $\pm$ 0.043
	3	3.538 $\pm$ 0.048	3.347 $\pm$ 0.045	<b>3.721</b> $\pm$ 0.050	0.303 $\pm$ 0.132	<b>1.257</b> $\pm$ 0.085	1.020 $\pm$ 0.064
	10	3.065 $\pm$ 0.083	3.550 $\pm$ 0.050	<b>3.670</b> $\pm$ 0.067	0.312 $\pm$ 0.232	<b>1.566</b> $\pm$ 0.130	1.514 $\pm$ 0.072
bedding $n=100$	2	3.406 $\pm$ 0.080	3.374 $\pm$ 0.088	<b>3.620</b> $\pm$ 0.062	0.525 $\pm$ 0.121	1.932 $\pm$ 0.194	<b>2.001</b> $\pm$ 0.080
	3	3.648 $\pm$ 0.106	3.564 $\pm$ 0.083	<b>3.876</b> $\pm$ 0.081	2.499 $\pm$ 0.972	2.250 $\pm$ 0.269	<b>2.624</b> $\pm$ 0.066
	10	3.355 $\pm$ 0.161	3.799 $\pm$ 0.144	<b>3.912</b> $\pm$ 0.082	<b>3.919</b> $\pm$ 0.045	2.578 $\pm$ 0.358	3.157 $\pm$ 0.091
apparel $n=100$	2	3.560 $\pm$ 0.094	3.527 $\pm$ 0.046	<b>3.784</b> $\pm$ 0.059	0.332 $\pm$ 0.519	<b>1.557</b> $\pm$ 0.140	1.492 $\pm$ 0.189
	3	3.878 $\pm$ 0.092	3.755 $\pm$ 0.062	<b>4.140</b> $\pm$ 0.063	0.445 $\pm$ 0.675	1.922 $\pm$ 0.235	<b>2.231</b> $\pm$ 0.202
	10	3.751 $\pm$ 0.087	4.084 $\pm$ 0.075	<b>4.425</b> $\pm$ 0.066	0.505 $\pm$ 1.005	2.368 $\pm$ 0.333	<b>2.953</b> $\pm$ 0.238
bath $n=100$	2	2.957 $\pm$ 0.087	3.024 $\pm$ 0.032	<b>3.198</b> $\pm$ 0.056	0.266 $\pm$ 0.306	<b>1.091</b> $\pm$ 0.071	0.827 $\pm$ 0.184
	3	3.062 $\pm$ 0.085	3.195 $\pm$ 0.058	<b>3.448</b> $\pm$ 0.058	0.347 $\pm$ 0.436	<b>1.358</b> $\pm$ 0.117	1.307 $\pm$ 0.177
	10	2.497 $\pm$ 0.135	3.426 $\pm$ 0.076	<b>3.438</b> $\pm$ 0.089	0.502 $\pm$ 0.644	1.715 $\pm$ 0.163	<b>1.777</b> $\pm$ 0.214

natural upper bound of ELBO. All algorithms are implemented in Python3, and source code will be public on the author’s homepage.

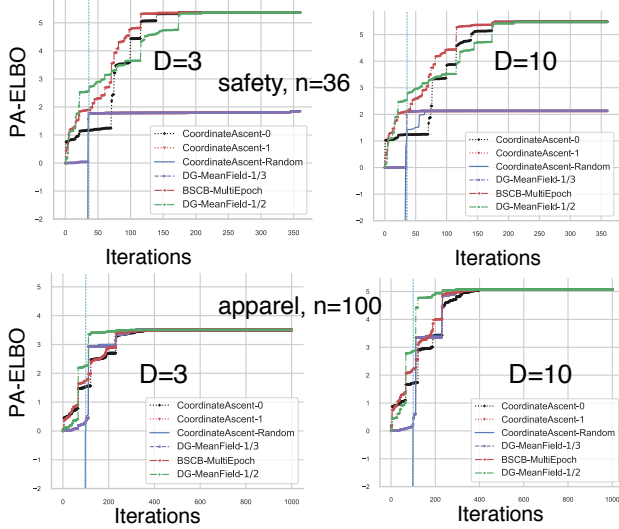


Figure 1: PA-ELBO on Amazon data. The figures trace trajectories of multi-epoch algorithms. Cyan vertical line shows the one-epoch point.

**Real-world Dataset.** We tested the mean field methods on the trained FLID models from Tschitschek et al. (2016) on Amazon Baby Registries dataset. After preprocessing, this dataset has 13 categories, e.g., “feeding” & “furniture”. One category contains a certain number of registries over the ground set of this category, e.g., “strollers” has 5,175 registries with  $n = 40$ . One can refer to Table 1 for specific dimensionalities on each of the category<sup>5</sup>. For each category, three classes of models were trained, with latent dimensions  $D = 2, 3, 10$ , respectively, on 10 folds of the data.

### 5.1. Results on One-epoch Algorithms

Table 1 summarizes the outputs of one-epoch algorithms for both ELBO and PA-ELBO objectives. For each category, the results of FLID models with three dimensionalities ( $D = 2, 3, 10$ ) are reported.

**ELBO Objective.** The results are summarized in columns 3 to 5 in Table 1. The mean and standard deviation are calculated for 10 FLID models trained on 10 folds of the data. One can observe that both DR-DoubleGreedy and BSCB improve over the baseline Submodular-DoubleGreedy, which has only a  $1/3$  approximation guarantee. Furthermore, DR-DoubleGreedy generates better solutions than

BSCB for almost all of the cases, though they have the same approximation guarantees.

**PA-ELBO objective.** The results are summarized in columns 6 to 8 in Table 1. For each category, out of the 10 folds of data, we have  $\binom{10}{2} = 45$  pairs of folds. For each pair of folds, one PA-ELBO objective is defined. The mean and standard deviation are computed for these 45 pairs for each category and each latent dimensionality  $D$ . One can still observe that DR-DoubleGreedy and BSCB significantly improve over Submodular-DoubleGreedy. Moreover, DR-DoubleGreedy usually produces better solutions than BSCB.

### 5.2. Results on Multi-epoch Algorithms

**PA-ELBO Objective.** Figure 1 shows representative results on PA-ELBO objectives. One can see that after one epoch, DG-MeanField-1/2 usually returns the best solution. However, CoordinateAscent is quite sensitive to initializations. After sufficiently many iterations, most multi-epoch algorithms converge to similar function values. This is consistent with the intuition since after one epoch, all algorithms are using the same strategy: conducting coordinate-wise maximization. However, for CoordinateAscent with unlucky initializations, e.g., for category “safety” (row 1), it may get stuck in poor local optima.

**More Experiments.** Due to space limit, the results on ELBO objectives along with the true log-partition values are put into §G.1, the experimental study of the discretization-based algorithm (Soma & Yoshida, 2017) are put in to §G.2, and synthetic results are deferred to §G.4.

## 6. Conclusions

Probabilistic structured models play an important role in machine learning today, especially models with submodular costs. We have proposed a novel Double Greedy scheme with optimal  $1/2$  approximation ratio for the general problem of box-constrained continuous DR-submodular maximization. Based on the optimal scheme, we present provable mean field algorithms for probabilistic log-submodular models and their posterior agreement score.

## Acknowledgements

Yatao Bian would like to thank Nico Gorbach and Josip Djolonga for fruitful discussions, and to thank Sebastian Tschitschek for sharing the source code and data. This research was partially supported by the Max Planck ETH Center for Learning System, and by SNSF NFP 75 grant no. 407540-167212.

<sup>5</sup>More details on this dataset can be found in Gillenwater et al. (2014).



## References

- Bach, F. Submodular functions: from discrete to continuous domains. *arXiv preprint arXiv:1511.00394*, 2015.
- Bian, A., Levy, K. Y., Krause, A., and Buhmann, J. M. Continuous dr-submodular maximization: Structure and algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 486–496, 2017a.
- Bian, A., Buhmann, J. M., and Krause, A. Optimal dr-submodular maximization and applications to provable mean field inference. *arXiv preprint arXiv:1805.07482*, 2018.
- Bian, A. A., Buhmann, J. M., Krause, A., and Tschitschek, S. Guarantees for greedy maximization of non-submodular functions with applications. In *International Conference on Machine Learning (ICML)*, 2017b.
- Bian, A. A., Mirzasoleiman, B., Buhmann, J. M., and Krause, A. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 111–120, 2017c.
- Bian, Y., Gonskiy, A., and Buhmann, J. M. Information-theoretic analysis of maxcut algorithms. In *IEEE Information Theory and Applications Workshop (ITA)*, pp. 1–5, 2016. URL <http://people.inf.ethz.ch/ybian/docs/pa.pdf>.
- Buchbinder, N., Feldman, M., Naor, J., and Schwartz, R. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pp. 649–658. IEEE, 2012.
- Buhmann, J. M. Information theoretic model validation for clustering. In *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, pp. 1398–1402. IEEE, 2010.
- Buhmann, J. M., Dumazert, J., Gonskiy, A., and Szpankowski, W. Posterior agreement for large parameter-rich optimization problems. *TCS*, 2018.
- Calinescu, G., Chekuri, C., Pál, M., and Vondrák, J. Maximizing a submodular set function subject to a matroid constraint. In *International Conference on Integer Programming and Combinatorial Optimization*, pp. 182–196. Springer, 2007.
- Chekuri, C., Vondrák, J., and Zenklusen, R. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.
- Chekuri, C., Jayram, T., and Vondrák, J. On multiplicative weight updates for concave and submodular function maximization. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pp. 201–210. ACM, 2015.
- Chen, L., Hassani, H., and Karbasi, A. Online continuous submodular maximization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1896–1905, 2018.
- Djolonga, J. and Krause, A. From map to marginals: Variational inference in bayesian submodular models. In *Neural Information Processing Systems (NIPS)*, 2014.
- Djolonga, J. and Krause, A. Scalable variational inference in log-supermodular models. In *International Conference on Machine Learning (ICML)*, 2015.
- Djolonga, J., Tschitschek, S., and Krause, A. Variational inference in mixed probabilistic submodular models. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1759–1767, 2016.
- Ene, A. and Nguyen, H. L. A reduction for optimizing lattice submodular functions with diminishing returns. *arXiv preprint arXiv:1606.08362*, 2016.
- Feige, U., Mirrokni, V. S., and Vondrak, J. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 40(4):1133–1153, 2011.
- Gillenwater, J., Kulesza, A., and Taskar, B. Near-optimal map inference for determinantal point processes. In *Advances in Neural Information Processing Systems*, pp. 2735–2743, 2012.
- Gillenwater, J. A., Kulesza, A., Fox, E., and Taskar, B. Expectation-maximization for learning determinantal point processes. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3149–3157, 2014.
- Gorbach, N. S., Bian, A. A., Fischer, B., Bauer, S., and Buhmann, J. M. Model selection for Gaussian process regression. In *German Conference on Pattern Recognition*, pp. 306–318, 2017.
- Gottschalk, C. and Peis, B. Submodular function maximization on the bounded integer lattice. In *Approximation and Online Algorithms*, pp. 133–144. Springer, 2015.
- Gonskiy, A. and Buhmann, J. M. How informative are minimum spanning tree algorithms? In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pp. 2277–2281. IEEE, 2014.
- Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

- Iyer, R. and Bilmes, J. Submodular point processes with applications to machine learning. In *Artificial Intelligence and Statistics*, pp. 388–397, 2015.
- Krause, A. and Cevher, V. Submodular dictionary selection for sparse representation. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 567–574, 2010.
- Krause, A. and Golovin, D. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3:19, 2012.
- Krause, A. and Guestrin, C. Near-optimal nonmyopic value of information in graphical models. In *21st conference on uncertainty in artificial intelligence (UAI)*, 2005.
- Kulesza, A., Taskar, B., et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012.
- Lin, H. and Bilmes, J. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 510–520. Association for Computational Linguistics, 2011a.
- Lin, H. and Bilmes, J. Optimal selection of limited vocabulary speech corpora. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011b.
- Lovász, L. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pp. 235–257. Springer, 1983.
- Mokhtari, A., Hassani, H., and Karbasi, A. Decentralized submodular maximization: Bridging discrete and continuous settings. *arXiv preprint arXiv:1802.03825*, 2018.
- Niazadeh, R., Roughgarden, T., and Wang, J. Optimal algorithms for continuous non-monotone submodular and dr-submodular maximization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 9617–9627. 2018.
- Soma, T. and Yoshida, Y. Non-monotone dr-submodular function maximization. In *AAAI*, volume 17, pp. 898–904, 2017.
- Staib, M. and Jegelka, S. Robust budget allocation via continuous submodular functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.
- Tschiatschek, S., Djolonga, J., and Krause, A. Learning probabilistic submodular diversity models via noise contrastive estimation. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.
- Vondrák, J. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pp. 67–74, 2008.
- Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.
- Wilder, B. Risk-sensitive submodular optimization. In *AAAI*, 2017.
- Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1529–1537, 2015.

## Appendix

### A. There Exist Poor Local Optima

If one only assume the objective function  $f(\mathbf{x})$  to be continuous DR-submodular, and considering that the multilinear extension of a submodular set function is continuous DR-submodular, we can take the examples from literatures on combinatorial optimization, e.g., [Feige et al. \(2011\)](#), to show that bad local optima exist.

Here we provide a *stronger* example, where we assume that the objective function  $f(\mathbf{x})$  has the same structure as the ELBO objective (1). And still there exist bad local optima. These local optima have arbitrarily small objective value compared to the global optimum. And `CoordinateAscent` will get stuck in this local optimum without extra techniques.

Suppose that we have a directed graph  $G = (\mathcal{V}, A)$  with four vertices,  $\mathcal{V} = \{1, 2, 3, 4\}$  and four arcs,  $A = \{(1, 2), (2, 3), (3, 2), (3, 4)\}$ . The weights of the arcs are (let  $b, c$  be large positive numbers):  $w_{1,2} = c$ ,  $w_{2,3} = c$ ,  $w_{3,4} = c$ ,  $w_{3,2} = bc$ . Let  $F(S)$  denote the sum of weights of arcs leaving  $S$ . Consider its ELBO (using techniques from §4.1),

$$f(\mathbf{x}) = f_{\text{mt}}(\mathbf{x}) + \sum_{i \in \mathcal{V}} H(x_i) \quad (10)$$

$$= \sum_{(i,j) \in A} w_{ij} x_i (1 - x_j) + \sum_{i \in \mathcal{V}} H(x_i) \quad (11)$$

$$= cx_1(1 - x_2) + cx_2(1 - x_3) + cx_3(1 - x_4) + bcx_3(1 - x_2) + \sum_{i \in \mathcal{V}} H(x_i). \quad (12)$$

Consider the point  $\mathbf{y} = [0.5, 1, 0, 0.5]^\top$ , it has function value  $f(\mathbf{y}) = c + 2 \log 2$ . Consider a second point  $\bar{\mathbf{x}} = [1, 0, 1, 0]^\top$ , while the global optimum  $f(\mathbf{x}^*)$  must be greater than  $f(\bar{\mathbf{x}}) = (2 + b)c$ . When  $b$  becomes large, the ratio  $\frac{f(\mathbf{y})}{f(\mathbf{x}^*)} \leq \frac{c+2 \log 2}{(2+b)c}$  can be arbitrarily small.

`CoordinateAscent` may get stuck on the point  $\mathbf{y} = [0.5, 1, 0, 0.5]^\top$ . This can be illustrated by considering the course of `CoordinateAscent`. Suppose wlog. that `CoordinateAscent` processes coordinates in the order of  $1 \rightarrow 4$  (actually it is the same with any orders).

For coordinate 1,  $\nabla_1 f_{\text{mt}}(\mathbf{x}) = c(1 - x_2)$ , so  $\nabla_1 f_{\text{mt}}(\mathbf{y}) = 0$ , after applying  $\sigma(\nabla_1 f_{\text{mt}}(\mathbf{y}))$ ,  $y_1$  remains to be 0.5.

For coordinate 2,  $\nabla_2 f_{\text{mt}}(\mathbf{x}) = c(1 - x_3) - bcx_3$ , so  $\nabla_2 f_{\text{mt}}(\mathbf{y}) = c$ . When  $c$  is sufficiently large (approaching infinity), after applying  $\sigma(\nabla_2 f_{\text{mt}}(\mathbf{y}))$ ,  $y_2$  will still be 1.

For coordinate 3,  $\nabla_3 f_{\text{mt}}(\mathbf{x}) = -cx_2 + c(1 - x_4) + bc(1 - x_2)$ , so  $\nabla_3 f_{\text{mt}}(\mathbf{y}) = -0.5c$ . When  $c$  is sufficiently large (approaching infinity), after applying  $\sigma(\nabla_3 f_{\text{mt}}(\mathbf{y}))$ ,  $y_3$  will still be 0.

For coordinate 4,  $\nabla_4 f_{\text{mt}}(\mathbf{x}) = -cx_3$ , so  $\nabla_4 f_{\text{mt}}(\mathbf{y}) = 0$ , after applying  $\sigma(\nabla_4 f_{\text{mt}}(\mathbf{y}))$ ,  $y_4$  remains to be 0.5.

### B. Proofs for DR-DoubleGreedy

#### B.1. Hardness of Problem (P)

**Observation 1.** *The problem of maximizing a generally non-monotone DR-submodular continuous function subject to box-constraints is NP-hard. Furthermore, there is no  $(1/2 + \epsilon)$ -approximation for any  $\epsilon > 0$ , unless  $RP = NP$ .*

The proof is very similar to the that of [Bian et al. \(2017c, Proposition 5\)](#), so we just briefly explain here. One observation is that the multilinear extension of a submodular set function is also continuous DR-submodular, so we can use the same reduction as in [Bian et al. \(2017c, Proposition 5\)](#) to prove the hardness results as above.

#### B.2. Detailed Proof of Theorem 1

**Theorem 1.** *Assume the optimal solution of  $\max_{\mathbf{x} \in [\mathbf{a}, \mathbf{b}]} f(\mathbf{x})$  is  $\mathbf{x}^*$ , then for Alg. 1 it holds,*

$$f(\mathbf{x}^n) \geq \frac{1}{2}f(\mathbf{x}^*) + \frac{1}{4}[f(\mathbf{a}) + f(\mathbf{b})] - \frac{5\delta}{4}. \quad (4)$$

*Proof of Theorem 1.* Define  $\mathbf{o}^k := (\mathbf{x}^* \vee \mathbf{x}^k) \wedge \mathbf{y}^k$ . It is clear that  $\mathbf{o}^0 = \mathbf{x}^*$  and  $\mathbf{o}^n = \mathbf{x}^n = \mathbf{y}^n$ . One can notice that as Alg. 1 progresses,  $\mathbf{o}^k$  moves from  $\mathbf{x}^*$  to  $\mathbf{x}^n$  (or  $\mathbf{y}^n$ ).

Let  $r_a = \frac{\delta_a}{\delta_a + \delta_b}$ ,  $r_b = 1 - r_a$ ,  $u = r_a u_a + (1 - r_a) u_b$ .

**Lemma 1.** For all  $k = 1, \dots, n$ , it holds that,

$$\begin{aligned} f(\mathbf{x}^{k-1}|_{v_k}(u_b)) - f(\mathbf{x}^{k-1}) &\geq -\delta/n, \\ f(\mathbf{y}^{k-1}|_{v_k}(u_a)) - f(\mathbf{y}^{k-1}) &\geq -\delta/n. \end{aligned} \quad (5)$$

*Proof of Lemma 1.* One can observe that  $\mathbf{x}^{k-1} \leq \mathbf{y}^{k-1}$ , so from DR-submodularity:  $f(\mathbf{x}^{k-1}|_{v_k}(u_b)) - f(\mathbf{x}^{k-1}) \geq f(\mathbf{y}^{k-1}|_{v_k}(u_b)) - f(\mathbf{y}^{k-1}|_{v_k}(u_{v_k})) \geq -\frac{\delta}{n}$ .

Similarly, because of  $\mathbf{x}^{k-1} \leq \mathbf{y}^{k-1}$  and  $u_a \leq b_{v_k}$ , from DR-submodularity:  $f(\mathbf{y}^{k-1}|_{v_k}(u_a)) - f(\mathbf{y}^{k-1}) \geq f(\mathbf{x}^{k-1}|_{v_k}(u_a)) - f(\mathbf{x}^{k-1}|_{v_k}(b_{v_k})) \geq -\frac{\delta}{n}$ . ■

**Lemma 2.** For all  $k = 1, \dots, n$ , it holds that,

$$\begin{aligned} f(\mathbf{o}^{k-1}) - f(\mathbf{o}^k) \\ \leq \frac{1}{2} [f(\mathbf{x}^k) - f(\mathbf{x}^{k-1}) + f(\mathbf{y}^k) - f(\mathbf{y}^{k-1})] + \frac{2.5\delta}{n}. \end{aligned} \quad (6)$$

*Proof of Lemma 2.* Step I:

Let us try to lower bound the RHS of Lemma 2.

$$\begin{aligned} f(\mathbf{x}^k) - f(\mathbf{x}^{k-1}) &= f(\mathbf{x}^{k-1}|_{v_k}(r_a u_a + r_b u_b)) - f(\mathbf{x}^{k-1}) \\ &\stackrel{\textcircled{1}}{\geq} r_a f(\mathbf{x}^{k-1}|_{v_k}(u_a)) + r_b f(\mathbf{x}^{k-1}|_{v_k}(u_b)) - f(\mathbf{x}^{k-1}) \\ &= r_a [f(\mathbf{x}^{k-1}|_{v_k}(u_a)) - f(\mathbf{x}^{k-1})] + r_b [f(\mathbf{x}^{k-1}|_{v_k}(u_b)) - f(\mathbf{x}^{k-1})] \\ &\stackrel{\textcircled{2}}{\geq} r_a \delta_a - r_b \frac{\delta}{n}, \end{aligned}$$

where ① is because of that  $f$  is concave along one coordinate, ② is from Lemma 1.

Similarly,

$$\begin{aligned} f(\mathbf{y}^k) - f(\mathbf{y}^{k-1}) &= f(\mathbf{y}^{k-1}|_{v_k}(r_a u_a + r_b u_b)) - f(\mathbf{y}^{k-1}) \\ &\geq r_a f(\mathbf{y}^{k-1}|_{v_k}(u_a)) + r_b f(\mathbf{y}^{k-1}|_{v_k}(u_b)) - f(\mathbf{y}^{k-1}) \\ &= r_a [f(\mathbf{y}^{k-1}|_{v_k}(u_a)) - f(\mathbf{y}^{k-1})] + r_b [f(\mathbf{y}^{k-1}|_{v_k}(u_b)) - f(\mathbf{y}^{k-1})] \\ &\geq -r_a \frac{\delta}{n} + r_b \delta_b. \end{aligned}$$

So it holds that

$$f(\mathbf{x}^k) - f(\mathbf{x}^{k-1}) + f(\mathbf{y}^k) - f(\mathbf{y}^{k-1}) \geq r_a \delta_a + r_b \delta_b - \frac{\delta}{n} = \frac{\delta_a^2 + \delta_b^2}{\delta_a + \delta_b} - \frac{\delta}{n}. \quad (13)$$

Step II:

Now let us upper bound the LHS of Lemma 2.

Notice that  $\mathbf{o}^{k-1} := (\mathbf{x}^* \vee \mathbf{x}^{k-1}) \wedge \mathbf{y}^{k-1}$ . For  $\mathbf{o}^{k-1}$ , its  $v_k$ -th coordinate is  $x_{v_k}^*$ . From  $\mathbf{o}^{k-1}$  to  $\mathbf{o}^k$ , its  $v_k$ -th coordinate changes to be  $u$ . So,



$$f(\mathbf{o}^{k-1}) - f(\mathbf{o}^k) = f(\mathbf{o}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{o}^{k-1}|_{v_k}(u)) \quad (14)$$

Let us consider the following two situations:

1.  $x_{v_k}^* \leq u$ .

In this case:

$$\begin{aligned} & f(\mathbf{o}^{k-1}) - f(\mathbf{o}^k) \\ &= f(\mathbf{o}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{o}^{k-1}|_{v_k}(u)) \\ &\stackrel{\textcircled{3}}{\leq} f(\mathbf{y}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{y}^{k-1}|_{v_k}(u)) \\ &= f(\mathbf{y}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{y}^{k-1}|_{v_k}(r_a u_a + r_b u_b)) \\ &\stackrel{\textcircled{4}}{\leq} r_a [f(\mathbf{y}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{y}^{k-1}|_{v_k}(u_a))] + r_b [f(\mathbf{y}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{y}^{k-1}|_{v_k}(u_b))] \\ &\leq r_a [f(\mathbf{y}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{y}^{k-1}|_{v_k}(u_a))] + r_b \frac{\delta}{n} \quad (\text{selection rule of Alg. 1}) \\ &\stackrel{\textcircled{5}}{\leq} r_a [f(\mathbf{y}^{k-1}|_{v_k}(u_b)) + \frac{\delta}{n} - (f(\mathbf{y}^{k-1}) - \frac{\delta}{n})] + r_b \frac{\delta}{n} \\ &\leq r_a \delta_b + (2r_a + r_b) \frac{\delta}{n}, \end{aligned}$$

where  $\textcircled{3}$  is because  $\mathbf{o}^{k-1} \leq \mathbf{y}^{k-1}$  and DR-submodularity of  $f$ ,  $\textcircled{4}$  is from concavity of  $f$  along one coordinate,  $\textcircled{5}$  is because of the selection rule of Alg. 1 and Lemma 1.

2.  $x_{v_k}^* > u$ :

In this case:

$$\begin{aligned} & f(\mathbf{o}^{k-1}) - f(\mathbf{o}^k) \\ &= f(\mathbf{o}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{o}^{k-1}|_{v_k}(u)) \\ &\leq f(\mathbf{x}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{x}^{k-1}|_{v_k}(u)) \quad (\mathbf{o}^{k-1} \geq \mathbf{x}^{k-1} \text{ \& DR-submodularity}) \\ &= f(\mathbf{x}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{x}^{k-1}|_{v_k}(r_a u_a + r_b u_b)) \\ &\leq r_a [f(\mathbf{x}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{x}^{k-1}|_{v_k}(u_a))] + r_b [f(\mathbf{x}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{x}^{k-1}|_{v_k}(u_b))] \\ &\leq r_a \frac{\delta}{n} + r_b [f(\mathbf{x}^{k-1}|_{v_k}(x_{v_k}^*)) - f(\mathbf{x}^{k-1}|_{v_k}(u_b))] \\ &\leq r_a \frac{\delta}{n} + r_b [(f(\mathbf{x}^{k-1}|_{v_k}(u_a)) + \frac{\delta}{n}) - (f(\mathbf{x}^{k-1}) - \frac{\delta}{n})] \\ &= r_b \delta_a + (2r_b + r_a) \frac{\delta}{n} \end{aligned}$$

We can conclude that in both the above cases, it holds that

$$f(\mathbf{o}^{k-1}) - f(\mathbf{o}^k) \leq \frac{\delta_a \delta_b}{\delta_a + \delta_b} + \frac{2\delta}{n} \quad (15)$$

Combining Equation (13) and Equation (15) we can get,

$$\frac{1}{2} [f(\mathbf{x}^k) - f(\mathbf{x}^{k-1}) + f(\mathbf{y}^k) - f(\mathbf{y}^{k-1})] \geq f(\mathbf{o}^{k-1}) - f(\mathbf{o}^k) - \frac{2.5\delta}{n} \quad (16)$$

Thus we reach Lemma 2. ■

Now we can finalize the proof. For Lemma 2, let us sum for  $k = 1, \dots, n$ , we can get,

$$f(\mathbf{x}^*) - f(\mathbf{x}^n) \leq \frac{1}{2}[f(\mathbf{x}^n) - f(\mathbf{a}) + f(\mathbf{y}^n) - f(\mathbf{b})] + 2.5\delta \quad (17)$$

After rearrangement, one can show that  $f(\mathbf{x}^n) \geq \frac{1}{2}f(\mathbf{x}^*) + \frac{1}{4}[f(\mathbf{a}) + f(\mathbf{b})] - \frac{5\delta}{4}$ . ■

### C. Difference of DR-DoubleGreedy and BSCB

In this section, we show that DR-DoubleGreedy behaves fundamentally different from BSCB by a simple example. It is worth noting that DR-DoubleGreedy does not require  $f$  to be differentiable, while BSCB explicitly needs the partial derivative of  $f$ . Extensive experimental statistics (Table 1) documents that DR-DoubleGreedy performs better than BSCB in most cases.

In terms of time cost, BSCB and DR-DoubleGreedy have the similar running time asymptotically. To see this: 1) both algorithms run  $n$  iterations for the  $n$  coordinates; 2) in iteration  $i$ , the main cost of both algorithms lie in calculating the partial derivative of the multilinear extension at  $\mathbf{x}$  and  $\mathbf{y}$ . Empirically, our algorithm can be further optimized: it just needs to evaluate  $\delta_a$  and  $\delta_b$ , which can be obtained without extra cost (using the previously calculated partial derivative plus the multilinear nature of the multilinear extensions). While BSCB needs to use binary search with precision  $\epsilon$  to find the proper value  $u$ , which needs to be run for multiple iterations.

To prove the difference, we provide a counter example: consider the 2-D DR-submodular quadratic program:

$$f(x) = 0.5\mathbf{x}^\top \mathbf{H}\mathbf{x} + \mathbf{h}^\top \mathbf{x}, \mathbf{H} = [-1, -1; -1, -2], \mathbf{h} = [0.5; 1]. \quad (18)$$

Define  $g([x_1; x_2]) := \frac{\partial f(\mathbf{x})}{\partial x_1} = -x_1 - x_2 + 0.5$

Consider the box-constrained DR-submodular max problem:

$$\max f(\mathbf{x}), 0 \leq \mathbf{x} \leq 1 \quad (19)$$

Starting with coordinate 1,

- For BSCB:  $g([z; 0]) = -z + 0.5$ ,  $g([z; 1]) = -z - 0.5$ . In order to find the equilibrium, we set  $g([z; 0]) * (1 - z) + g([z; 1]) * z$  to be 0, which amounts to  $-2z + 0.5 = 0$ , so  $z = 1/4$ .

- For DR-DoubleGreedy:

Solving 1-D subproblem  $u_a = \arg \max_{x_1} f([x_1; 0])$  one gets  $u_a = 0.5$ , and  $\delta_a = f([0.5; 0]) - f([0; 0]) = 1/8$ .

Solving 1-D subproblem  $u_b = \arg \max_{x_1} f([x_1; 1])$  one gets  $u_b = 0$ , and  $\delta_b = f([0; 1]) - f([1; 1]) = 1$ .

So  $u = (1/9) * u_a + (8/9) * u_b = 1/18$ .

Q.E.D. (Every step is in closed form)

### D. Mean Field Lower Bounds for PSMs

Log-submodular models (Djolonga & Krause, 2014) are a class of probabilistic point processes over subsets of a ground set  $\mathcal{V} = [n]$ , where the log-densities are submodular set functions  $F(S)$ :  $p(S) = \frac{1}{Z} \exp(F(S))$ , where  $Z = \sum_{S \subseteq \mathcal{V}} \exp(F(S))$  is the partition function.

Mean-field inference aims to approximate  $p(S)$  by a fully factorized product distribution  $q(S|\mathbf{x}) := \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j)$ ,  $\mathbf{x} \in [0, 1]^n$ , by minimizing the distance measured w.r.t. the Kullback-Leibler divergence between  $q$  and  $p$ , i.e.,  $\mathbb{KL}(q||p) = \sum_{S \subseteq \mathcal{V}} q(S|\mathbf{x}) \log \frac{q(S|\mathbf{x})}{p(S)}$ .  $\mathbb{KL}(q||p)$  is non-negative, so

$$0 \leq \mathbb{KL}(q||p) = \sum_{S \subseteq \mathcal{V}} q(S|\mathbf{x}) \log \frac{q(S|\mathbf{x})}{p(S)}$$

$$= -\mathbb{E}_{q(S|\mathbf{x})}[\log p(S)] + \mathbb{H}(q(S|\mathbf{x})) \quad (20)$$

$$= -\sum_{S \subseteq \mathcal{V}} F(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j) + \sum_{i=1}^n [x_i \log x_i + (1 - x_i) \log(1 - x_i)] + \log Z. \quad (21)$$

where  $\mathbb{H}(\cdot)$  is the entropy. So one can get  $\log Z \geq \sum_{S \subseteq \mathcal{V}} F(S) \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j) - \sum_{i=1}^n [x_i \log x_i + (1 - x_i) \log(1 - x_i)] = (\text{ELBO})$ .

Multilinear extension  $f_{\text{mt}}(\mathbf{x})$  of a submodular set function is continuous DR-submodular (Bach, 2015), and  $-\sum_{i=1}^n [x_i \log x_i + (1 - x_i) \log(1 - x_i)]$  is separable and concave on each coordinate, so (ELBO) is DR-submodular w.r.t.  $\mathbf{x}$ . Maximizing (ELBO) amounts to minimizing the Kullback-Leibler divergence.

For (PA-ELBO) (2), it is the sum of two multilinear extensions (weighted by  $\beta > 0$ ) and the binary entropy term, since the non-negative sum of two DR-submodular functions is still DR-submodular, so (PA-ELBO) in (2) is also continuous DR-submodular. Thus it fits into the general optimization problem (P).

## E. Full Lower Bounds of PA Objective

By giving upper bounds for  $\log Z(\beta; D') + \log Z(\beta; D'')$ , we can get the full lower bounds of the PA objective.

Let us take one  $\log Z(\beta; D')$  for example. This can be achieved using techniques of Djolonga & Krause (2014), which is done by optimizing supergradients of  $F(S|D')$ . A representative supergradient is the bar supergradient, which is defined as: if  $i \in A$ ,  $\bar{s}^A = F_{\mathcal{V}-\{i\}}(\{i\}|D')$ , if  $i \notin A$ ,  $\bar{s}^A = F(\{i\}|D')$ , where  $F_B(A|D')$  is the marginal gain of  $A$  based on  $B$ . Then,

$$\log Z(\beta; D') \leq \min_A \log Z^+(\bar{s}^A, F(A|D') - \bar{s}^A(A)) = \min_A F(A|D') + \mathbf{m}(A|D'), \quad (22)$$

where  $\mathbf{m}(\{i\}|D') = \log(1 + e^{-F_{\mathcal{V}-\{i\}}(\{i\}|D')}) - \log(1 + e^{F(\{i\}|D')})$ .

So the full lower bound of PA objective in (3) is,

$$\log \sum_{S \subseteq \mathcal{V}} p_\beta(S|D') p_\beta(S|D'') \quad (\text{Posterior-Agreement objective}) \quad (23)$$

$$\begin{aligned} &= -\left[ \sum_{S \subseteq \mathcal{V}} q(S|\mathbf{x}) \right] \log \frac{\sum_{S \subseteq \mathcal{V}} q(S|\mathbf{x})}{\sum_{S \subseteq \mathcal{V}} p_\beta(S|D') p_\beta(S|D'')} \\ &\stackrel{\text{log-sum inequality}}{\geq} -\sum_{S \subseteq \mathcal{V}} q(S|\mathbf{x}) \log \frac{q(S|\mathbf{x})}{p_\beta(S|D') p_\beta(S|D'')} = \mathbb{H}(q) + \mathbb{E}_q \log p_\beta(S|D') + \mathbb{E}_q \log p_\beta(S|D'') \\ &= \underbrace{\mathbb{H}(q) + \beta \mathbb{E}_q F(S|D') + \beta \mathbb{E}_q F(S|D'')}_{(\text{PA-ELBO}) \text{ in (2)}} - \log Z(\beta; D') - \log Z(\beta; D'') \\ &\geq \max_q \underbrace{\mathbb{H}(q) + \beta \mathbb{E}_q F(S|D') + \beta \mathbb{E}_q F(S|D'')}_{(\text{PA-ELBO}) \text{ in (2)}} - \min_A [F(A|D') + \mathbf{m}(A|D')] - \min_A [F(A|D'') + \mathbf{m}(A|D'')] \quad (24) \end{aligned}$$

## F. Detailed Multilinear Extension in Closed Form

### F.1. More on Sampling

**Lemma 3** (Hoeffding Bound, Theorem 2 in Hoeffding (1963)). *Let  $X_1, \dots, X_m$  be independent random variables such that for each  $i$ ,  $a \leq X_i \leq b$ , with  $a, b \in \mathbb{R}$ . Let  $\bar{X} = \frac{1}{m} \sum_{i=1}^m X_i$ . Then*

$$\Pr[|\bar{X} - \mathbb{E}(X)| > t] \leq e^{-\frac{2t^2 m}{(b-a)^2}}. \quad (25)$$

According to the Hoeffding bound (Hoeffding, 1963), one can easily derive that  $\frac{1}{k} \sum_{i=1}^k F(S_i)$  is arbitrarily close to  $f_{\text{mt}}(\mathbf{x})$  with increasingly more samples: With probability at least  $1 - e^{-k\epsilon^2/2}$ , it holds that  $|\frac{1}{k} \sum_{i=1}^k F(S_i) - f_{\text{mt}}(\mathbf{x})| \leq \epsilon \max_S |F(S)|$ , for all  $\epsilon > 0$ .

## F.2. Some Gibbs Random Fields

**Undirected MAXCUT.** For MAXCUT, its objective is  $F(\mathbf{v}) = \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(v_i + v_j - 2v_i v_j)$ ,  $\mathbf{v} \in \{0, 1\}^V$ . Its multilinear extension is  $f_{\text{mt}}(\mathbf{x}) = \frac{1}{2} \sum_{(i,j) \in E} w_{ij}(x_i + x_j - 2x_i x_j)$ ,  $\mathbf{x} \in [0, 1]^V$ .

**Directed MAXCUT.** Its objective is  $F(\mathbf{v}) = \sum_{(i,j) \in E} w_{ij}v_i(1 - v_j)$ ,  $\mathbf{v} \in \{0, 1\}^V$ . Its multilinear extension is  $f_{\text{mt}}(\mathbf{x}) = \frac{1}{2} \sum_{(i,j) \in E} w_{ij}x_i(1 - x_j)$ ,  $\mathbf{x} \in [0, 1]^V$ .

**Ising models.** For Ising models with non-positive pairwise interactions,  $F(\mathbf{v}) = \sum_{s \in V} \theta_s v_s + \sum_{(s,t) \in E} \theta_{st} v_s v_t$ ,  $\mathbf{v} \in \{0, 1\}^V$ , this objective can be easily verified to be submodular. Its multilinear extension is:

$$f_{\text{mt}}(\mathbf{x}) = \sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E} \theta_{st} x_s x_t \quad (26)$$

Lower bound of its log-partition function is  $f_{\text{mt}}(\mathbf{x}) + \sum_{s \in V} H_s(x_s)$ ,  $\mathbf{x} \in [0, 1]^V$ . When updating  $x_s$  and fix all other coordinates, it is easy to see that

$$x_s \leftarrow \sigma(\theta_s + \sum_{t \in N(s)} \theta_{st} x_t), \quad (27)$$

where  $N(s)$  are the neighbors of  $s$ .

## F.3. More on FLID-style Objectives

The more refined way to compute partial directives can be expressed by considering the following derivation,

$$\begin{aligned} & \nabla_i \left( f_{\text{mt}}(\mathbf{x}) - \sum_{i \in [n]} u'_i x_i \right) \\ &= \sum_{S \subseteq V, S \ni i} F(S) \prod_{j \in S \setminus \{i\}} x_j \prod_{j' \notin S} (1 - x_{j'}) - \sum_{S \subseteq V \setminus \{i\}} F(S) \prod_{j \in S} x_j \prod_{j' \notin S, j' \neq i} (1 - x_{j'}) \\ &= \sum_{d=1}^D \left[ \sum_{S \subseteq V, S \ni i} \max_{i \in S} W_{i,d} \prod_{j \in S \setminus \{i\}} x_j \prod_{j' \notin S} (1 - x_{j'}) - \sum_{S \subseteq V \setminus \{i\}} \max_{i \in S} W_{i,d} \prod_{j \in S} x_j \prod_{j' \notin S, j' \neq i} (1 - x_{j'}) \right] \\ &= \sum_{d=1}^D [W_{i_d(l_i), d} \prod_{m=l_i+1}^n (1 - x_m) + \sum_{l=l_i+1}^n W_{i_d(l), d} x_{i_d(l)} \prod_{m=l+1}^n (1 - x_m) \\ &\quad - \sum_{l=1}^{l(i)} W_{i_d(l), d} x_{i_d(l)} \prod_{m=l+1, m \neq l(i)}^n (1 - x_m) - \sum_{l=l(i)+1}^n W_{i_d(l), d} x_{i_d(l)} \prod_{m=l+1}^n (1 - x_m)] \end{aligned}$$

## F.4. Approximation for Concave Over Modular Functions

A general form is,

$$\begin{aligned} F(S) &= \sum_{j=1}^M w_j \psi(\mathbf{m}^j(S)) \\ &= \sum_{j=1}^M w_j [\mathbf{m}^j(S)]^a. \end{aligned}$$

$\psi(\cdot)$  is a concave function, a common choice is  $\psi(y) = y^a$ ,  $a \in (0, 1]$ . A simple approximation is  $\hat{F}(S) = \sum_{j=1}^M w_j \sum_{i \in S} (m_i^j)^a$ , which approximates  $F(S)$  up to a factor of  $\mathcal{O}(|S|^{1-a})$  (Iyer & Bilmes, 2015). Since  $\hat{F}(S)$  is modular, one can directly get its multilinear extension.



## G. More Experimental Results

We put more results in this section. It includes experiments on both synthetic datasets and real-world datasets.

### G.1. ELBO Objective

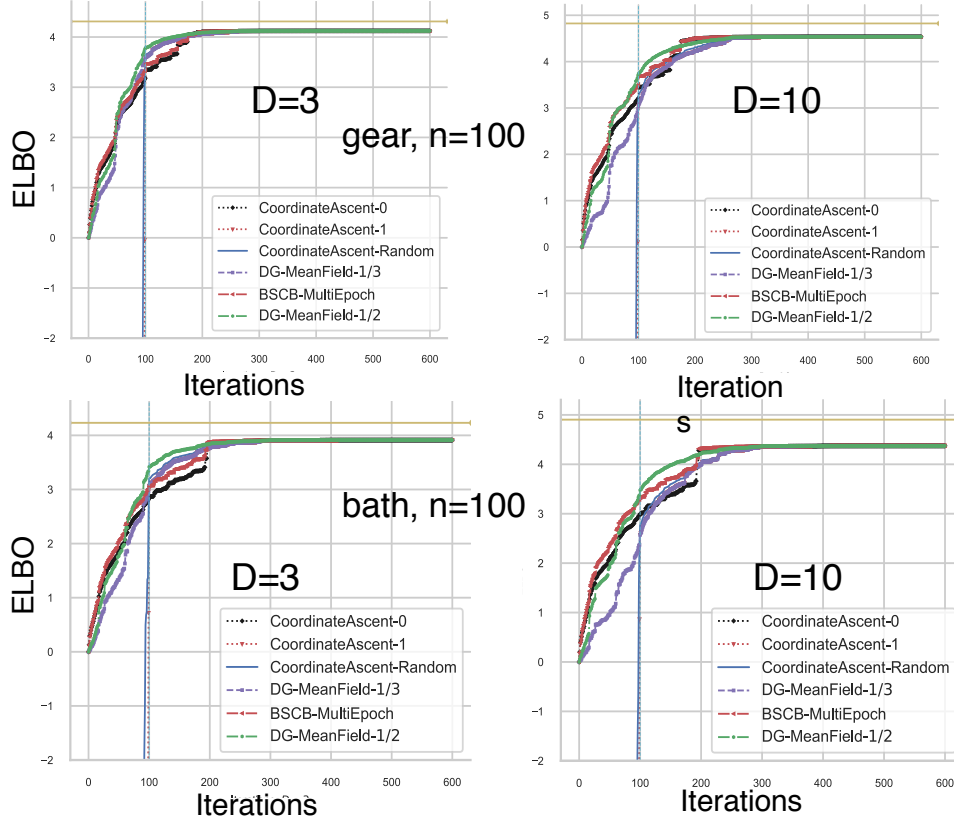


Figure 2: Typical trajectories of multi-epoch algorithms on ELBO objective for Amazon data. 1st row: “gear”; 2nd row: “bath”. Cyan vertical line shows the one-epoch point. Yellow line shows the true log-partition.

Figure 2 records typical trajectories of multi-epoch algorithms for ELBO objectives. Note that the cyan vertical lines indicate the one-epoch point. It shows that after one epoch, DG-MeanField-1/2 almost always returns the best solution, and it is also the fastest one to converge. However, CoordinateAscent is quite sensitive to initializations. After sufficiently many iterations, all multi-epoch algorithms converge to similar ELBO value. This is consistent with the intuition since after one epoch, all algorithms are using the same strategy: conducting coordinate-wise maximization. One can also observe that the obtained ELBO is close to the true log partition functions (yellow lines).

### G.2. Study of the Discretization-based Algorithm

We conducted comparisons with the exact version of the algorithm (Algorithm 1 of Soma & Yoshida (2017), termed SY-Alg1). For number of bins in the discretization, we tested with 10, 50, 100 and 500. Figure 3 shows the trajectories of SY-Alg1 in comparison with other one-epoch algorithms, where we set number of discretization bins to be 500.

From the experiments we observe that SY-Alg1 is much slower than DR-DoubleGreedy, which is apparent since SY-Alg1 needs lots of function value evaluations due to discretization. By looking at the trajectories, one can see that SY-Alg1 behaves similarly as BSCB when using sufficiently many discretization bins (greater than 50).

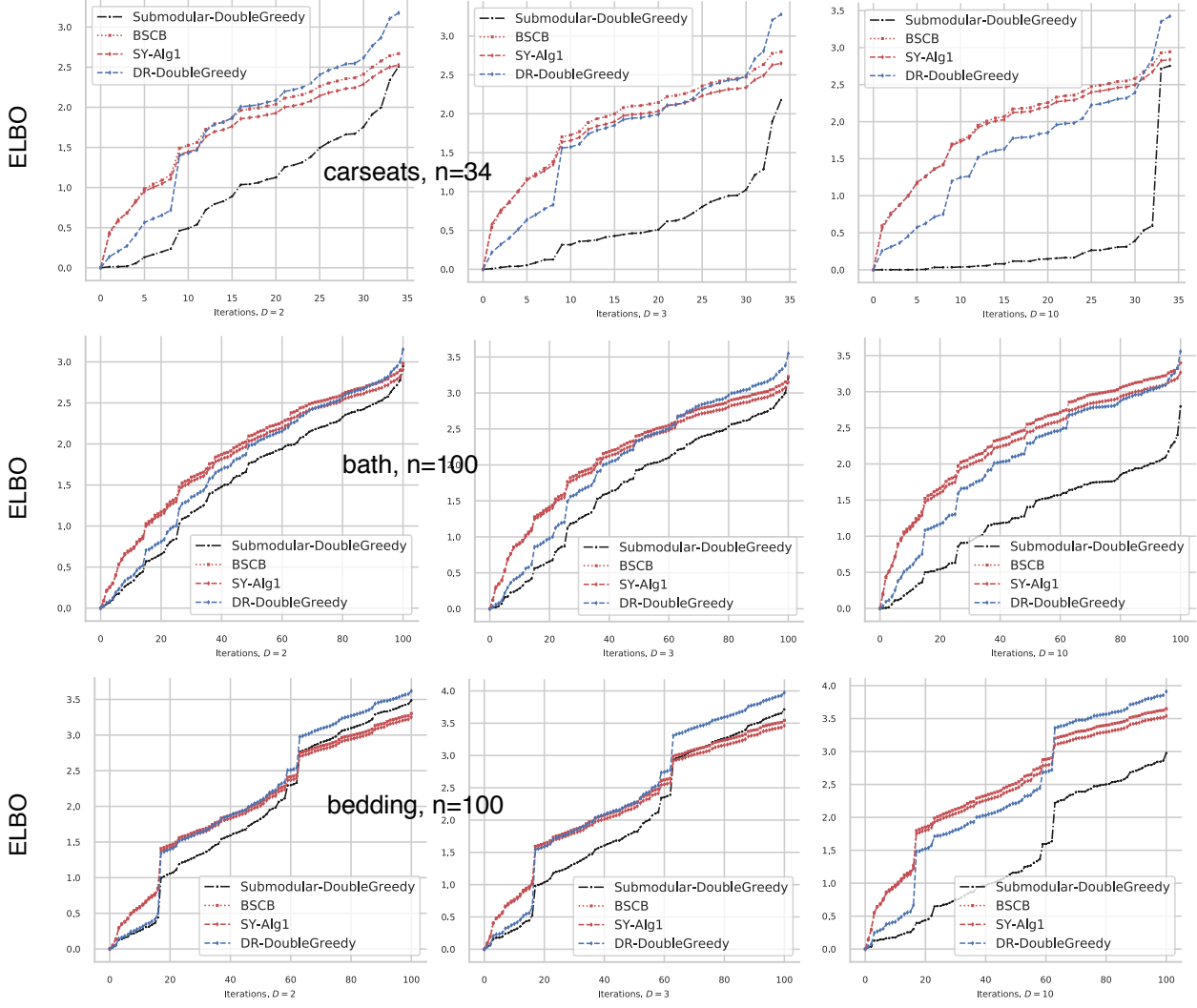


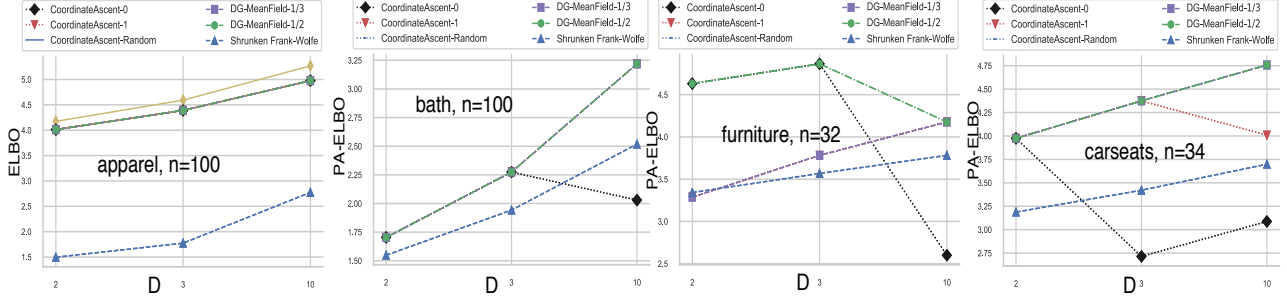
Figure 3: Typical trajectories of one-epoch algorithms on ELBO objective for Amazon data. 1st row: “carseats”; 2nd row: “bath”; 3rd row: “bedding”. Number of bins is set as 500 for SY-Alg1.

### G.3. Experiments on Shrunk Frank-Wolfe

Though `shrunk FW` method is not only computationally too expensive, but also have worse approximation guarantee, we still would like to see whether it would produces good solution with more computational resources. In order to verify this, we run all multi-epoch algorithms for 6 epochs, while run `shrunk FW` for 60 epochs, results are shown in the figure below: even with 10 times more computations, `shrunk FW` still performs worse than the proposed algorithm DG-MeanField-1/2. Sometimes `shrunk FW` has comparable performance with coordinate descent variant.

### G.4. Synthetic Results

We generate FLID models in the following manner: We firstly generate the latent representation matrix  $\mathbf{W} \in \mathbb{R}^{n \times D}$  such that each entry of  $W_{i,d} \sim U(0, 1)$ . It is clear that for FLID,  $F(\emptyset) = 0$ . We then set  $\mathbf{u}$  to be proportional to  $D$  in a random way  $\mathbf{u} = 0.1D * \mathbf{1} * U(0, 1)$  so the objective is non-monotone. Figure 4 records the results: one row corresponds to the results for a specific  $n$ . First column is the function value returned by the algorithms, which are the average of 10 repeated experiments. The other columns are trajectories of multi-epoch algorithms, since behavior is similar for different repeated



experiments, we plot the first one here. Yellow lines are the true log-partition returned by exhaustive search, cyan vertical lines shows the one-epoch point. One can see that for one-epoch algorithms, DR-DoubleGreedy returns the highest value. For multi-epoch algorithms, DG-MeanField-1/2 is the fastest one to converge. After sufficiently many epoches, the three multi-epoch algorithms converge to solutions with similar function value.

### G.5. More Results on ELBO Objective

See Figure 5 for more results on the ELBO objective from Amazon data.

### G.6. More Results on PA-ELBO Objective

Figure 6 illustrates more results on the PA-ELBO objective from Amazon data.

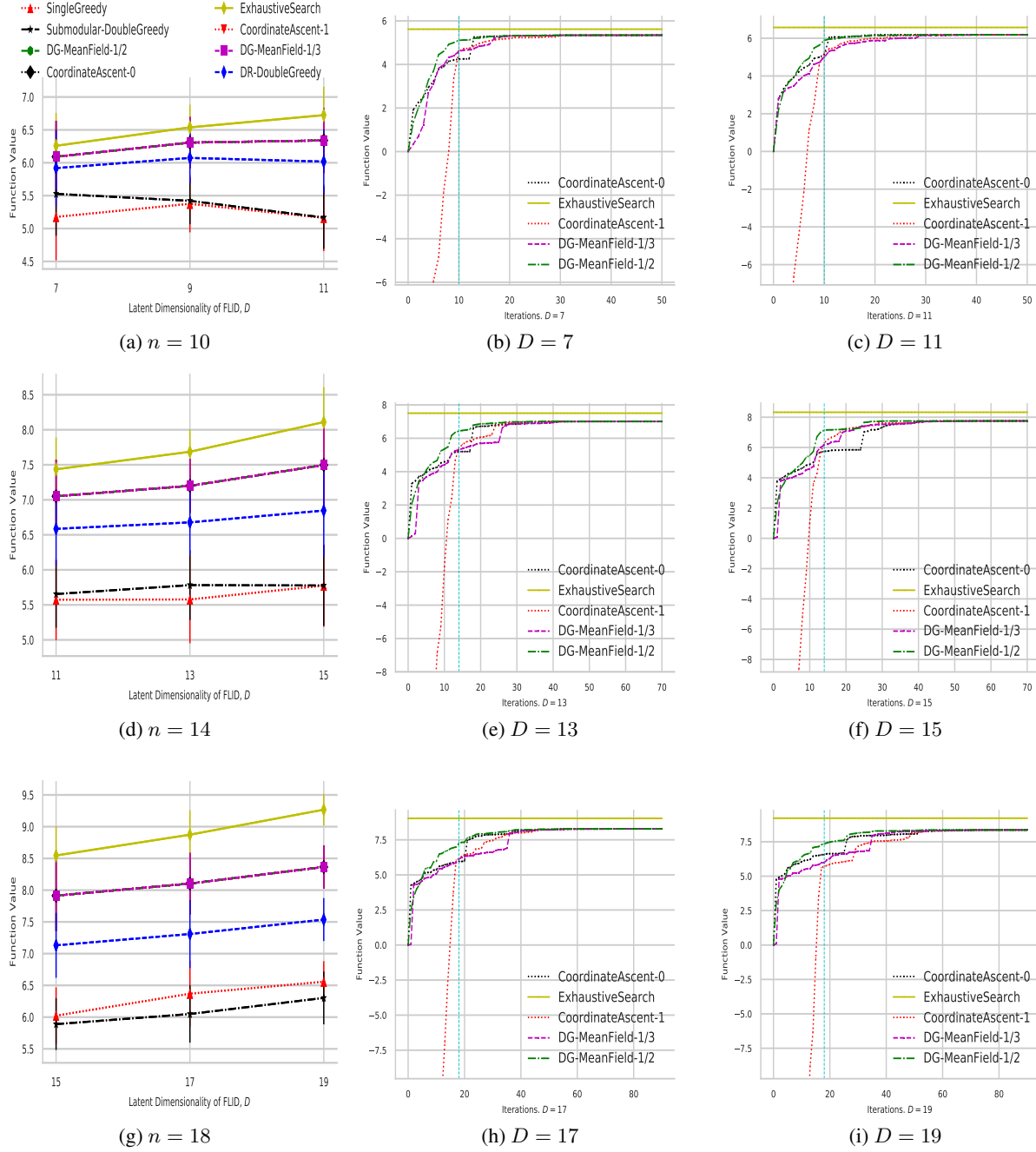
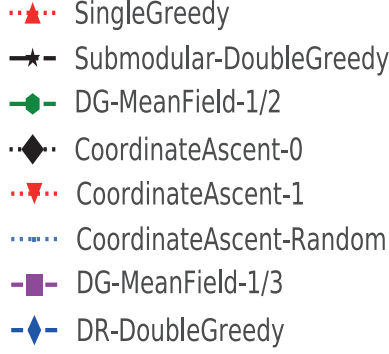


Figure 4: FLID synthetic results. First column shows the function values returned by different algorithms. The other columns show trajectories of multi-epoch algorithms. Cyan vertical line shows one-epoch. Yellow line indicates the true log-partition value generated by exhaustive search. SingleGreedy is the one-epoch version of CoordinateAscent-0.





(a) Legend for subfigs (b,d,g)

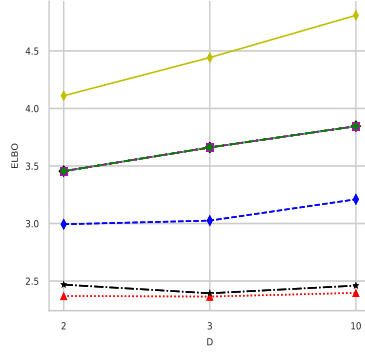
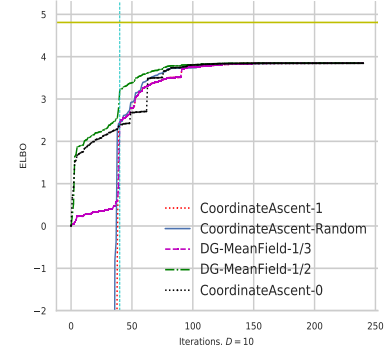
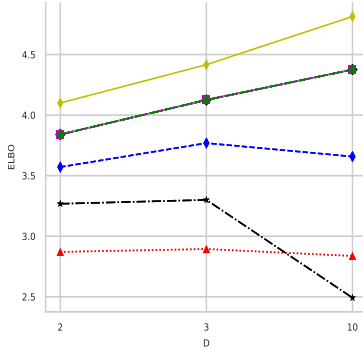
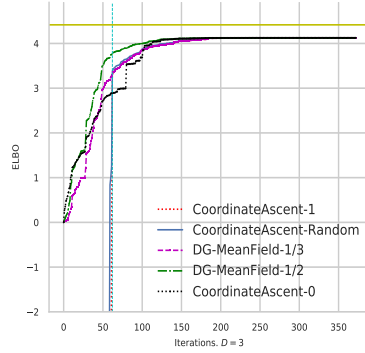
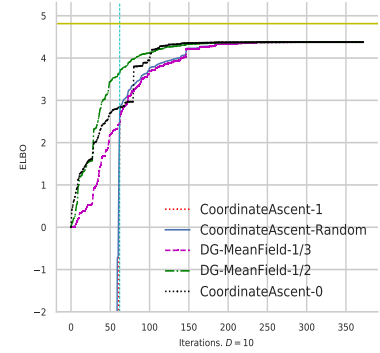
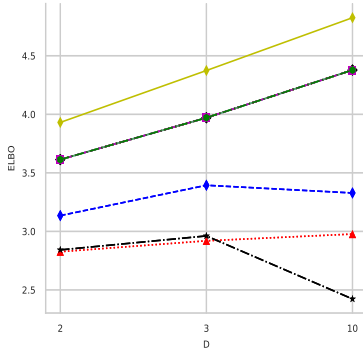
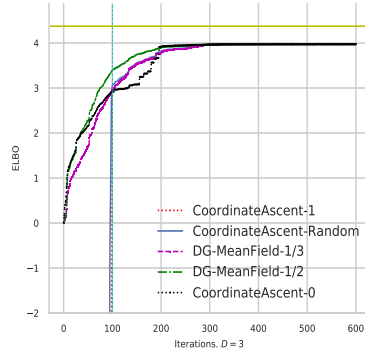
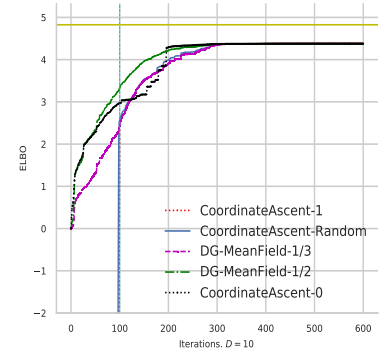
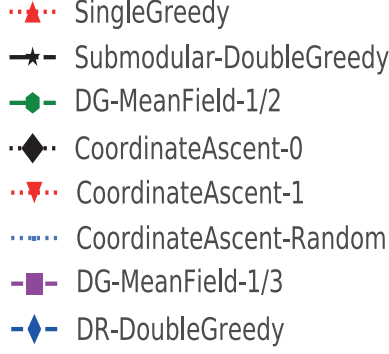

 (b) “strollers”,  $n = 40$ 

 (c)  $D = 10$ 

 (d) “health”,  $n = 62$ 

 (e)  $D = 3$ 

 (f)  $D = 10$ 

 (g) “bath”,  $n = 100$ 

 (h)  $D = 3$ 

 (i)  $D = 10$ 

Figure 5: ELBO objective on Amazon data. 1st row: “strollers”; 2nd row: “health”. 3rd row: “bath”; Subfigs (b,d,g) shows the ELBO returned by all algorithms, other columns traces trajectories of multi-epoch algorithms. Cyan vertical line show the one-epoch point. Yellow line shows the true log-partition value. SingleGreedy is the one-epoch version of CoordinateAscent-0.



(a) Legend for subfigs (b,d,g)

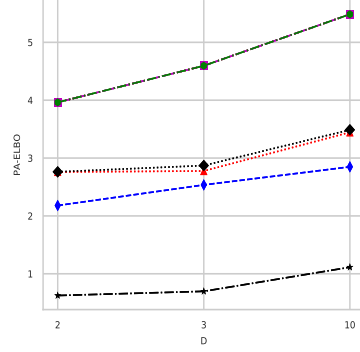
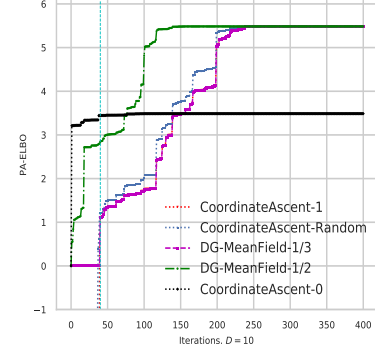
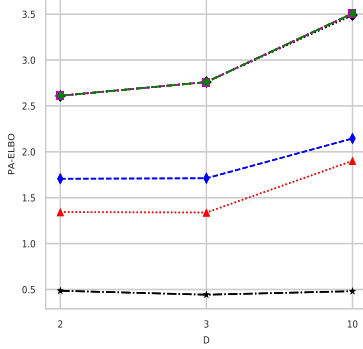
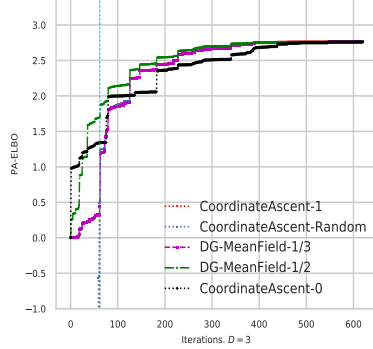
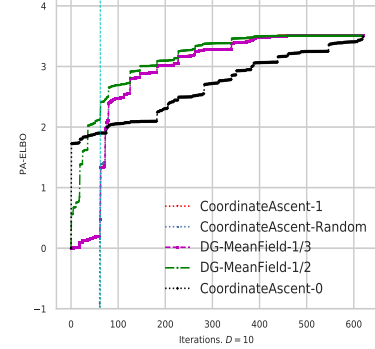
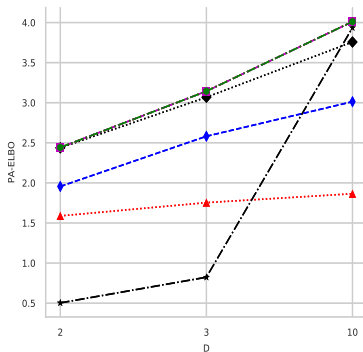
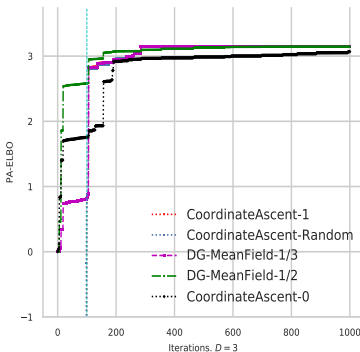
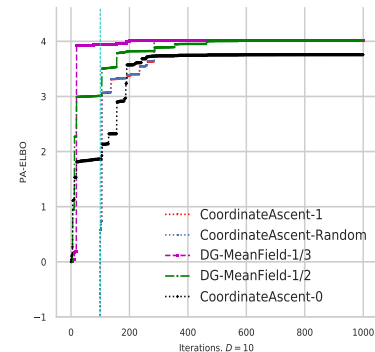

 (b) “furniture”,  $n = 40$ , folds (6,10)

 (c)  $D = 10$ 

 (d) “toys”,  $n = 62$ , folds (6,8)

 (e)  $D = 3$ 

 (f)  $D = 10$ 

 (g) “bedding”,  $n = 100$ , folds (7,9)

 (h)  $D = 3$ 

 (i)  $D = 10$ 

Figure 6: PA-ELBO objective on Amazon data. First row: “furniture”; second row: “toys”; third row: “bedding”. Subfigs (b,d,g) show the PA-ELBO returned by all algorithms, other columns traces trajectories of multi-epoch algorithms. Cyan vertical line shows the one-epoch point. SingleGreedy is the one-epoch version of CoordinateAscent-0.