
Probabilistic Multi-Source Data Selection via Learnable Categorical Priors: Variance Reduction and Guaranteed Derandomization

Shizhan Cai Qixin Zhang Siyuan Ma
Siyuan Liang Yongcheng Jing Dacheng Tao

College of Computing and Data Science
Nanyang Technological University

shizhan001@e.ntu.edu.sg qixin.zhang2026@gmail.com

masi0004@e.ntu.edu.sg

{siyuan.liang,yongcheng.jing,dacheng.tao}@ntu.edu.sg

Abstract

Bilevel data selection is a fundamental task with numerous applications in machine learning and statistics. A common strategy for tackling this problem is to relax the original bilevel combinatorial optimization into a continuous one. However, existing methods, including 0-1 relaxation and probabilistic relaxation, still suffer from several limitations, such as unstable gradient estimation and infeasible rounding. To address these issues, this paper proposes a new probabilistic relaxation for the multi-source bilevel data selection problem. Our key idea is to introduce multiple learnable categorical priors to parameterize a probability distribution supported only on feasible data subsets, thus naturally yielding a lossless rounding scheme. We further show that our proposed relaxation admits an effective two-point gradient estimator, which can simultaneously avoid the inverse-Hessian computation required by existing 0-1 relaxation and the unbounded score term appearing in prior probabilistic relaxation. Furthermore, we develop a reducible representation of this estimator, which substantially reduces the number of lower-level retrains required for gradient computation. Built upon this reducible two-point estimator, we then design a novel stochastic gradient algorithm, termed ReSGD, for multi-source bilevel data selection problem, and establish a rigorous approximation guarantee for the subset returned by ReSGD under weak submodularity. Finally, extensive experiments are conducted to validate the effectiveness of our proposed ReSGD.

1 Introduction

Data selection aims to identify a small subset of training samples such that training solely on this subset can achieve performance comparable to that trained on the full dataset. As modern learning systems continue to grow in scale, this task has become increasingly crucial in a wide range of applications, including efficient training [15, 33, 34, 55, 78, 79], in-context learning [63, 64, 113] and continual learning [8, 9, 48, 104, 116]. For instance, in continual learning, an effective way to counter *catastrophic forgetting* [39, 77] is to maintain a small rehearsal memory that stores a subset of previously seen data and then replay them during training on new tasks [12, 18, 72, 111].

From an optimization viewpoint, many of data selection problems [8, 23, 60, 67] can be formulated as *bilevel combinatorial optimization*, where the upper level evaluates the selected subset according to the performance of the model induced by the corresponding lower-level training problem. However, the *discrete* nature of data selection makes the underlying bilevel problem highly challenging. To circumvent this, recent works have sought to bring the original *bilevel combinatorial optimization*

into the world of continuous optimization, thereby leveraging the rich toolbox of gradient-based methods [7, 65, 83]. Existing approaches mainly follow two routes. The first is *0-1 relaxation* [8, 48], which directly relaxes the binary selective mask into a continuous variable in $[0, 1]$ and then applies implicit differentiation [73, 89] to solve the resulting bilevel continuous optimization. The second is *probabilistic relaxation* [116], which treats each binary mask as a Bernoulli random variable with learnable parameters and then optimizes these parameters via policy gradient descent [107, 101, 58].

Although these two aforementioned continuous-relaxation reformulations have made data selection tasks more tractable, their practical implementation still faces several fundamental limitations:

- ❶ **Problematic Gradient Estimations.** In 0-1 relaxation, estimating the upper-level gradient typically requires approximating the *inverse Hessian* of the lower-level objective, which is often *computationally prohibitive* [8, 48, 114]. In contrast, probabilistic relaxation avoids such second-order computation via policy-gradient estimators [101, 107], but these estimators are *unbounded* and often suffer from *instability* and *high variance* during training [46, 108];
- ❷ **Limited to Single-Source Scenarios.** Both 0-1 and probabilistic relaxation primarily focus on subset selection over a *single homogeneous dataset*. However, in many real-world applications, we often encounter *multiple heterogeneous data sources* rather than a single-source dataset. A representative example arises in continual learning, where one often needs to retain a small data subset *separately* from each past task for future replay [12, 72, 104]. Such settings cannot be trivially decomposed into multiple *independent* single-source data selection problems, since this treatment will ignore the *cross-source correlations* among datasets and lead to *redundant selections*;
- ❸ **Heuristic and Infeasible Rounding.** In continuous relaxation, *rounding* or *derandomization* refers to the conversion of the obtained continuous solution into a discrete solution [1, 11]. However, *satisfactory* rounding schemes are still *lacking* for both 0-1 and probabilistic relaxation. In 0-1 relaxation, current rounding methods are *heuristic*, so there is *no principled way* to quantify how the obtained subset relates to the original continuous solution. In probabilistic relaxation, sampling from the learned Bernoulli distributions may return a subset *outside* the target constraint [13, 19, 20];
- ❹ **Incomplete Convergence Analysis.** Existing theoretical results of both 0-1 and probabilistic relaxation mainly focus on convergence to a *continuous stationary point* [116, 48, 23, 9]. However, this is *insufficient* for data selection, since our primary interest lies in the quality of the resulting subset rather than the intermediate continuous solution. To date, how to establish *rigorous* theoretical guarantees for the obtained subset remains an *uncharted territory* in bilevel data selection.

In view of all this, a natural question arises:

Can we develop a *provable* continuous-relaxation framework, together with *guaranteed* rounding scheme and *effective* gradient estimator, for *multi-source* data selection problems?

In this paper, we will provide an affirmative answer to this question by presenting a novel probabilistic relaxation for multi-source data selection problems. Our key idea is to introduce multiple *learnable categorical priors* to parameterize a probability distribution supported only on *feasible* data subsets. Compared with prior relaxations, this constrained construction immediately yields a *lossless* rounding scheme for multi-source data selection problem. Moreover, we derive an *effective* two-point gradient estimator for our proposed probabilistic relaxation, which can simultaneously avoid the costly inverse-Hessian computation in *0-1 relaxation* [8, 48] and the unbounded score in prior *probabilistic relaxation* [116], thereby achieving a significant *variance-reduction* effect. Based on this two-point estimator, we further develop an SGD algorithm, termed ReSGD, for the multi-source data selection problem, and establish a *rigorous* approximation guarantee for the subset returned by ReSGD under *weak submodularity* [24, 102]. In summary, we make the following contributions:

- We present a novel probabilistic relaxation for *multi-source* data selection problem, which only assigns probability mass to *feasible subsets* and thus naturally induces a *lossless* rounding scheme.
- We prove that our proposed probabilistic relaxation admits an *effective* two-point gradient estimator, which avoids both the inverse-Hessian computation in *0-1 relaxation* and the unbounded score in prior *probabilistic relaxation*, thereby improving stability and reducing variance. Furthermore, we develop a reducible representation of this two-point gradient estimator under a Bayesian perspective, which can further reduce the gradient-computation cost from $\mathcal{O}(h + \sum_{i=1}^h n_i)$ retrainings to $\mathcal{O}(h)$. Here, h denotes the number of data sources, and $\sum_{i=1}^h n_i$ is the total sample size.

- Based on the proposed reducible two-point representation, we then design a novel SGD algorithm, termed ReSGD. Furthermore, under (γ, β) -weak submodularity [102], we show that the subset returned by ReSGD achieves a $\left(\frac{\gamma^2}{\beta + \beta(1-\gamma) + \gamma^2}\right)$ -approximation guarantee with respect to the optimal data subset $\mathcal{S}^*$. To the best of our knowledge, this provides *the first* rigorous theoretical guarantee for multi-source bilevel data selection problem.
- We demonstrate the practical efficacy of our ReSGD on data summarization and continual learning.

2 Bilevel Data Selection: From Single Dataset to Multi-Source Scenarios

This section first reviews the single-source bilevel data selection problem together with its two representative continuous relaxations, namely, *0-1 relaxation* [8, 9, 48] and *probabilistic relaxation* [116]. We then move on to the multi-source setting, which is the main focus of this paper.

2.1 Single-Source Bilevel Data Selection: Background and Relaxations

Given a fixed dataset $\mathcal{D} \triangleq \{(x_i, y_i)\}_{i=1}^n$, the goal of single-source data selection is to identify a subset of at most K samples from $\mathcal{D}$ such that training on this subset can achieve performance comparable to that trained on $\mathcal{D}$. Let $\ell(x_i, y_i; \theta)$ be the training loss on the i -th sample (x_i, y_i) , where θ denotes the model parameters. Then, the single-source data selection problem can be formulated as the following *bilevel combinatorial optimization* [8, 9, 27, 48, 104, 115, 116]:

$$\begin{aligned} \min_{\mathbf{m} \in \{0,1\}^n, \|\mathbf{m}\|_0 \leq K} \mathcal{L}(\theta^*(\mathbf{m})) &\triangleq \sum_{i=1}^n \ell(x_i, y_i; \theta^*(\mathbf{m})) \\ \text{s.t. } \theta^*(\mathbf{m}) &\in \arg \min_{\theta} \hat{\mathcal{L}}(\theta, \mathbf{m}) \triangleq \sum_{i=1}^n m_i \cdot \ell(x_i, y_i; \theta), \end{aligned} \quad (1)$$

where $\mathbf{m} \in \{0,1\}^n$ denotes the selective mask vector with $m_i = 1$ indicating that the i -th sample is selected and $m_i = 0$ otherwise, $\theta^*(\mathbf{m})$ is the optimal solution to the lower-level problem under the selective mask vector $\mathbf{m}$ and the symbol $\|\cdot\|_0$ represents the ℓ_0 norm.

The bilevel problem in Eq. (1) is generally known to be **NP-hard** [35, 82], implying that no polynomial-time algorithms can solve it optimally. To tackle this, recent works have relaxed Eq. (1) into a continuous one, so that gradient-based methods can be used to *approximately* solve it. To date, two representative relaxations have been developed for Eq. (1), namely, *0-1* and *probabilistic relaxation*.

0-1 Relaxation. The first line of work [8, 9, 48] directly relaxes each mask m_i as a continuous variable over $[0, 1]$. In this way, Eq. (1) can be rewritten as a *bilevel continuous optimization*, that is,

$$\begin{aligned} \min_{\mathbf{m} \in [0,1]^n, \|\mathbf{m}\|_1 \leq K} \mathcal{L}(\theta^*(\mathbf{m})) &\triangleq \sum_{i=1}^n \ell(x_i, y_i; \theta^*(\mathbf{m})), \\ \text{s.t. } \theta^*(\mathbf{m}) &\in \arg \min_{\theta} \hat{\mathcal{L}}(\theta, \mathbf{m}) \triangleq \sum_{i=1}^n m_i \cdot \ell(x_i, y_i; \theta). \end{aligned} \quad (2)$$

Compared with Eq. (1), Eq. (2) enables the use of well-established gradient-based methods from *bilevel continuous optimization* [21, 53, 56] to search for a high-quality *fractional* mask vector $\mathbf{m}$. In particular, under suitable conditions, the hypergradient w.r.t. each $m_i \in [0, 1]$ can be written as:

$$\frac{\partial \mathcal{L}(\theta^*(\mathbf{m}))}{\partial m_i} \triangleq -\nabla_{\theta} \mathcal{L}(\theta^*(\mathbf{m}))^\top \left[\frac{\partial^2 \hat{\mathcal{L}}(\theta^*(\mathbf{m}), \mathbf{m})}{\partial \theta \partial \theta^\top} \right]^{-1} \nabla_{\theta} \ell(x_i, y_i; \theta^*(\mathbf{m})). \quad (3)$$

It is worth noting that Eq. (3) requires computing the *inverse Hessian matrix* of the lower-level objective function $\hat{\mathcal{L}}$, which often incurs *substantial time and memory overhead*.

Probabilistic Relaxation. To avoid the costly inverse Hessian computation in Eq. (3), [116] recently proposed a probabilistic relaxation, which models each mask m_i as an *independent* Bernoulli random variable, namely, $m_i \sim \text{Bern}(\pi_i)$ where $\pi_i \in [0, 1]$ is a learnable parameter. Let $\boldsymbol{\pi} \triangleq (\pi_1, \dots, \pi_n)$ denote the corresponding parameter vector. Under the independence assumption, the induced distribution over binary vector $\mathbf{m}$ is $\mathbf{p}(\mathbf{m} \mid \boldsymbol{\pi}) \triangleq \prod_{i=1}^n \pi_i^{m_i} (1 - \pi_i)^{1-m_i}$. Accordingly, Eq. (1) can

be relaxed into an expected loss minimization problem with respect to the continuous parameter vector $\boldsymbol{\pi} \triangleq (\pi_1, \dots, \pi_n)$, that is to say,

$$\begin{aligned} \min_{\boldsymbol{\pi} \in [0,1]^n, \|\boldsymbol{\pi}\|_1 \leq K} \Phi(\boldsymbol{\pi}) &\triangleq \mathbb{E}_{\mathbf{p}(\mathbf{m}|\boldsymbol{\pi})} [\mathcal{L}(\boldsymbol{\theta}^*(\mathbf{m}))] \triangleq \sum_{\mathbf{m} \in \{0,1\}^n} (\mathcal{L}(\boldsymbol{\theta}^*(\mathbf{m})) \cdot \mathbf{p}(\mathbf{m}|\boldsymbol{\pi})) \\ \text{s.t.} \quad \boldsymbol{\theta}^*(\mathbf{m}) &\in \arg \min_{\boldsymbol{\theta}} \hat{\mathcal{L}}(\boldsymbol{\theta}, \mathbf{m}) \triangleq \sum_{i=1}^n m_i \cdot \ell(x_i, y_i; \boldsymbol{\theta}). \end{aligned} \quad (4)$$

A key advantage of (4) is that its gradient admits an unbiased policy gradient estimator [107, 101], i.e.,

$$\frac{\partial \Phi(\boldsymbol{\pi})}{\partial \pi_i} \triangleq \mathbb{E}_{\mathbf{p}(\mathbf{m}|\boldsymbol{\pi})} [\mathcal{L}(\boldsymbol{\theta}^*(\mathbf{m})) \cdot \nabla_{\pi_i} \ln(\mathbf{p}(\mathbf{m}|\boldsymbol{\pi}))] \triangleq \mathbb{E}_{\mathbf{p}(\mathbf{m}|\boldsymbol{\pi})} \left[\mathcal{L}(\boldsymbol{\theta}^*(\mathbf{m})) \cdot \frac{m_i - \pi_i}{\pi_i(1 - \pi_i)} \right]. \quad (5)$$

Eq.(5) implies that $\mathcal{L}(\boldsymbol{\theta}^*(\mathbf{m})) \cdot \frac{m_i - \pi_i}{\pi_i(1 - \pi_i)}$ is an unbiased stochastic gradient estimator of the relaxed objective function $\Phi(\boldsymbol{\pi})$. In sharp contrast with Eq.(3), this estimator only requires evaluating the function value $\mathcal{L}(\boldsymbol{\theta}^*(\mathbf{m}))$ and is thus *easier to implement*. However, it is worth noting that when $m_i = 1, \pi_i \rightarrow 0$ or $m_i = 0, \pi_i \rightarrow 1$, the factor $\frac{m_i - \pi_i}{\pi_i(1 - \pi_i)}$ will become *unbounded*, which can lead to *high variance, unstable updates*, and even *divergence* in practice [46, 108].

Furthermore, the Bernoulli product distribution $\mathbf{p}(\mathbf{m} | \boldsymbol{\pi})$ in Eq. (4) poses a significant challenge for rounding, since it assigns *non-zero* probability mass to subsets outside the K -size cardinality constraint. As a consequence, for an arbitrary parameter vector $\boldsymbol{\pi}$, sampling according to $\mathbf{p}(\mathbf{m} | \boldsymbol{\pi})$ may yield an *infeasible* subset whose cardinality exceeds K .

2.2 Multi-Source Bilevel Data Selection

We now turn to the multi-source scenarios. Specifically, we first suppose that the full dataset $\mathcal{D}$ consists of h heterogeneous sources, that is to say, $\mathcal{D} \triangleq \cup_{i \in [h]} \mathcal{D}_i$, where each source is given by $\mathcal{D}_i \triangleq \{(x_j^{(i)}, y_j^{(i)}) \mid j \in [n_i]\}$ and $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset, \forall i \neq j$. Here, different sources may correspond to different tasks, domains or data providers, depending on the applications.

Given a set of predefined budgets $\{K_1, \dots, K_h\}$, the goal of multi-source data selection is to select at most K_i from each source $\mathcal{D}_i$ such that the model trained on the *jointly selected subset* can achieve comparable performance as the model trained on $\mathcal{D}$. Formally, we consider the following problem:

$$\begin{aligned} \min_{\mathbf{m}_i \in \{0,1\}^{n_i}, \|\mathbf{m}_i\|_0 \leq K_i, \forall i \in [h]} \mathcal{L}(\boldsymbol{\theta}^*(\mathbf{m})) &\triangleq \sum_{i=1}^h \sum_{j=1}^{n_i} \ell(x_j^{(i)}, y_j^{(i)}; \boldsymbol{\theta}^*(\mathbf{m})) \\ \text{s.t.} \quad \boldsymbol{\theta}^*(\mathbf{m}) &\in \arg \min_{\boldsymbol{\theta}} \hat{\mathcal{L}}(\boldsymbol{\theta}, \mathbf{m}) \triangleq \sum_{i=1}^h \sum_{j=1}^{n_i} m_{i,j} \cdot \ell(x_j^{(i)}, y_j^{(i)}; \boldsymbol{\theta}), \end{aligned} \quad (6)$$

where the variable $m_{i,j} \in \{0, 1\}$ indicates whether the j -th sample from the i -th source $\mathcal{D}_i$ is selected, $\mathbf{m}_i \triangleq (m_{i,1}, \dots, m_{i,n_i})$ and $\mathbf{m} \triangleq (\mathbf{m}_1, \dots, \mathbf{m}_h)$.

Previously, when facing the multi-source data selection problem, many works [8, 9, 48, 116] artificially decomposed it into multiple *independent* single-source selection problems and then *repeatedly* applied cardinality-constrained algorithms to each source. In contrast, the upper-level objective in our Eq. (6) evaluates the model trained on the *jointly selected subset across all sources*, so the benefit of selecting a sample from one source depends *crucially* on what has been selected in other sources.

3 A New Probabilistic Relaxation via Learnable Categorical Priors

This section mainly introduces a new probabilistic relaxation for the multi-source data selection problem (6). As discussed in Section 2.1, a critical drawback of the current *probabilistic relaxation* [116] is that the induced Bernoulli product distribution will assign *non-zero* probability to subsets *outside* the target constraint, which renders the subsequent rounding operation intrinsically *infeasible*. To overcome this, a natural idea is to design a new sampling distribution whose support is *entirely* restricted to the *feasible region*. In other words, for any combinatorial constraint $\mathcal{C}$, we seek to construct

a parameterized distribution $p(\mathbf{m} \mid \boldsymbol{\pi})$ such that $p(\mathbf{m} \mid \boldsymbol{\pi}) \equiv 0$ for any $\mathbf{m} \notin \mathcal{C}$. An immediate advantage of such a construction is that every sample drawn from $p(\mathbf{m} \mid \boldsymbol{\pi})$ is automatically *feasible*.

Next, we show how to instantiate this idea for the multi-source data selection problem (6). Recall that Eq. (6) requires selecting at most K_i samples from each source $\mathcal{D}_i$. From a high-level perspective, this process closely resembles an election procedure, where each source $\mathcal{D}_i$ is viewed as a local candidate pool and K_i samples with the highest vote counts are chosen. Motivated by this observation, we introduce a learnable categorical prior $\boldsymbol{\pi}_i \triangleq (\pi_{i,1}, \dots, \pi_{i,n_i})$ for every source $\mathcal{D}_i \triangleq \{(x_j^{(i)}, y_j^{(i)}) \mid j \in [n_i]\}$. Here, each $\pi_{i,j}$ represents the probability of selecting the j -th sample from source $\mathcal{D}_i$, while the residual probability $1 - \sum_{j=1}^{n_i} \pi_{i,j}$ corresponds to selecting a dummy symbol $\emptyset$, meaning that *no sample* from $\mathcal{D}_i$ is chosen.

With these categorical priors $\boldsymbol{\pi} \triangleq (\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_h)$, we then *independently* draw K_i samples from each source $\mathcal{D}_i$ according to the categorical distribution specified by $\boldsymbol{\pi}_i$. This operation will yield a random subset $\bigcup_{i=1}^h \bigcup_{k=1}^{K_i} d_{i,k}$, where $d_{i,k}$ denotes the k -th random draw from source $\mathcal{D}_i$. Note that this subset $\bigcup_{i=1}^h \bigcup_{k=1}^{K_i} d_{i,k}$ contains at most K_i samples from each source $\mathcal{D}_i$. Therefore, this sampling rule induces a probability distribution supported only in the *feasible set* of Eq. (6). Based on this constrained sampling scheme, we naturally can relax Eq. (6) into the following continuous optimization problem over the categorical priors $\boldsymbol{\pi}$, that is to say,

$$\begin{aligned} \min_{\boldsymbol{\pi}_i \in \Delta_{n_i}, \forall i \in [h]} \Phi(\boldsymbol{\pi}) &\triangleq \sum_{d_{i,k} \in \mathcal{D}_i \cup \{\emptyset\}, \forall i,k} \left(\mathcal{L}(\boldsymbol{\theta}^*(\bigcup_{i=1}^h \bigcup_{k=1}^{K_i} d_{i,k})) \cdot \prod_{i=1}^h \prod_{k=1}^{K_i} p(d_{i,k} \mid \boldsymbol{\pi}_i) \right), \\ \text{s.t.} \quad \boldsymbol{\theta}^*(\mathcal{S}) &\in \arg \min_{\boldsymbol{\theta}} \hat{\mathcal{L}}(\boldsymbol{\theta}, \mathcal{S}) \triangleq \sum_{(x,y) \in \mathcal{S}} \ell(x, y; \boldsymbol{\theta}), \quad \forall \mathcal{S} \subseteq \mathcal{D}, \end{aligned} \quad (7)$$

where $d_{i,k}$ denotes the k -th random draw from source $\mathcal{D}_i$ with $p((x_j^{(i)}, y_j^{(i)}) \mid \boldsymbol{\pi}_i) \triangleq \pi_{i,j}$, $\Delta_{n_i} \triangleq \{\mathbf{p} \in [0, 1]^{n_i} : \|\mathbf{p}\|_1 \leq 1\}$ denotes the sub-probability simplex and $p(\emptyset \mid \boldsymbol{\pi}_i) \triangleq 1 - \sum_{j=1}^{n_i} \pi_{i,j}$.

Remark 1. Note that, in Eq. (7), we switch from the **binary notation** used in Eq. (6) to an equivalent **set-based formulation**. Specifically, for any subset $\mathcal{S} \subseteq \mathcal{D}$, let $\boldsymbol{\theta}^*(\mathcal{S})$ denote the optimal lower-level solution trained on the data subset $\mathcal{S}$. Equivalently, $\boldsymbol{\theta}^*(\mathcal{S})$ corresponds to $\boldsymbol{\theta}^*(\mathbf{1}_{\mathcal{S}})$ of Eq. (6).

Compared with the continuous relaxation in Eq. (4), a major advantage of Eq. (7) is that it only assigns probability mass to *feasible* subsets. Consequently, for any given categorical priors $\boldsymbol{\pi} \triangleq (\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_h)$, we can directly sample a subset satisfying the multi-source budget constraints. Moreover, this sampling step is *lossless in expectation* w.r.t. the relaxed objective value $\Phi(\boldsymbol{\pi})$. Specifically, if the subset $\mathcal{S}$ is sampled via categorical priors $\boldsymbol{\pi}$, then $\mathbb{E}_{\mathcal{S}} [\mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}))] = \Phi(\boldsymbol{\pi})$.

3.1 Variance-Reduced Two-point Gradient Estimator and Reducible SGD

To leverage gradient-based methods for optimizing Eq. (7), this subsection focuses on deriving stochastic estimators for $\nabla_{\boldsymbol{\pi}} \Phi(\boldsymbol{\pi})$. A natural starting point is to follow the same route as in Eq. (5) and compute a policy gradient estimator for our new probabilistic relaxation $\Phi(\boldsymbol{\pi})$. However, as shown in Appendix C.1, such an estimator still involves an *unbounded* score factor analogous to that in Eq. (5), which can again lead to high variance and unstable training behavior, especially when the categorical priors approach the boundary of the feasible domain.

Fortunately, the special structure of our proposed new probabilistic relaxation $\Phi(\boldsymbol{\pi})$ admits a *variance-reduced* two-point gradient estimator, which can avoid the aforementioned *unbounded* score factor. Before going into the details, we first introduce some notations. For any realization of the random draws $\{d_{i,k}, \forall k \in [K_i], \forall i \in [h]\}$, we define $\mathcal{S}_{-(i,1)} \triangleq \bigcup_{(i',k') \neq (i,1)} d_{i',k'}$ as the subset obtained by excluding the first draw $d_{i,1}$ from source $\mathcal{D}_i$. Equivalently, $\mathcal{S}_{-(i,1)}$ is formed by taking the union of all sampled draws except $d_{i,1}$. With this notation, we can establish the following theorem:

Theorem 1 (Proof in Appendix C.2). *For any $i \in [h]$ and $j \in [n_i]$, the partial derivative of $\Phi(\boldsymbol{\pi})$ with respect to variable $\pi_{i,j}$ satisfies the following equality, namely,*

$$\frac{\partial \Phi(\boldsymbol{\pi})}{\partial \pi_{i,j}} \triangleq K_i \cdot \mathbb{E} \left[\underbrace{\mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)} \cup (x_j^{(i)}, y_j^{(i)}))) - \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)}))}_{\text{Two-point Representation}} \right], \quad (8)$$

Algorithm 1: SGD via Two-point Reducible Representation (ReSGD)

Input: Datasets $\{\mathcal{D}_i\}_{i=1}^h$, Budgets $\{K_i\}_{i=1}^h$, Learning rate η , Number of iterations T

Output: Selected subset $\mathcal{S} \triangleq \bigcup_{i=1}^h \bigcup_{k=1}^{K_i} d_{i,k}$

Initialize $\pi_i^{(0)} \in \Delta_{n_i}$ for all $i \in [h]$ and train a holdout model $\theta^*(\mathcal{D})$;

for $t = 0, 1, \dots, T-1$ **do**

 Draw K_i samples from source $\mathcal{D}_i$ according to $\pi_i^{(t)}$ for all $i \in [h]$;

 Construct $\mathcal{S}_{-(i,1)}^{(t)}$ by excluding the first draw from source $\mathcal{D}_i$ for all $i \in [h]$;

 Train $\theta^*(\mathcal{S}_{-(i,1)}^{(t)})$ for all $i \in [h]$;

 Compute $\hat{\mathbf{g}}_i^{(t)} \triangleq (\hat{g}_{i,1}^{(t)}, \dots, \hat{g}_{i,n_i}^{(t)})$ by reducible representation Eq. (10) $\forall i \in [h], j \in [n_i]$;

 Update $\pi_i^{(t+1)} \triangleq \Pi_{\Delta_{n_i}}^1 \left(\pi_i^{(t)} - \eta \cdot \hat{\mathbf{g}}_i^{(t)} \right), \forall i \in [h]$;

Sample an index τ uniformly at random from $\{0, 1, \dots, T-1\}$;

Draw¹ K_i samples $\{d_{i,1}, \dots, d_{i,K_i}\}$ from source $\mathcal{D}_i$ via $\pi_i^{(\tau)}, \forall i \in [h]$;

where the expectation is taken over all random draws except the first draw $d_{i,1}$ from source $\mathcal{D}_i$.

Theorem 1 suggests the partial derivative of $\Phi(\pi)$ w.r.t. $\pi_{i,j}$ can be estimated via the two-point difference $\mathcal{L}(\theta^*(\mathcal{S}_{-(i,1)} \cup \{(x_j^{(i)}, y_j^{(i)})\})) - \mathcal{L}(\theta^*(\mathcal{S}_{-(i,1)}))$. Compared with the previous Eq.(3) and Eq.(5), this two-point representation enjoys three major advantages. First, it only depends on function evaluations of the upper-level objective, thereby avoiding *costly backpropagation* and *the computation of the inverse Hessian matrix* required in Eq. (3). Second, compared to Eq.(5), Eq. (8) does not rely on any *unbounded* factor, thereby leading to more *stable* updates. Third, the two subsets $\mathcal{S}_{-(i,1)} \cup (x_j^{(i)}, y_j^{(i)})$ and $\mathcal{S}_{-(i,1)}$ only differ in *one data point*, so their corresponding function values are typically very close to each other. This strong coupling can significantly *reduce the fluctuation* of the resulting gradient estimator, and thus naturally yields a *variance-reduction* effect [26, 84, 98].

It is worth noting that the two-point representation in Eq. (8) requires evaluating the marginal contribution of every data point $(x_j^{(i)}, y_j^{(i)})$ with respect to its corresponding subset $\mathcal{S}_{-(i,1)}$, which will eventually lead to a total of $\mathcal{O}(h + \sum_{i=1}^h n_i)$ lower-level retrains. Here, $\sum_{i=1}^h n_i$ denotes the total number of data points across all sources and is typically *very large* in practice. As a result, directly computing stochastic gradients based on Eq. (8) is generally *impractical*. To address this issue, we next introduce a *computationally-efficient* reformulation of Eq. (8).

Reducible Representation. We now derive a *tractable* expression for the two-point difference of Eq. (8) under Bayesian statistical theory [41, 81]. Specifically, we can treat the model parameter θ as a random variable with prior $p(\theta)$, and seek to infer its corresponding posterior $p(\theta|\mathcal{S})$ from the already-seen data subset $\mathcal{S} \subseteq \mathcal{D}$. We further suppose that the model follows a predictive distribution $p(y|x; \mathcal{S}) \triangleq \int p(y|x; \theta) \cdot p(\theta|\mathcal{S}) d\theta$. Note that when a point estimate of θ is used, the posterior $p(\theta|\mathcal{S})$ can be viewed as a Dirac measure [28, 32]. Under this probabilistic framework, the loss function is commonly specified as the negative log-likelihood of the predictive distribution, namely, $\ell(x, y; \theta^*(\mathcal{S})) \triangleq -\log p(y|x; \mathcal{S})$. Then, throughout the Bayes' rule and conditional independence, numerous prior works [78, 104] have shown that the marginal contribution of any data point (x, y) with respect to a subset $\mathcal{S}$ can admit the following reducible form: (See Appendix C.3.)

$$\begin{aligned} \mathcal{L}(\theta^*(\mathcal{S} \cup (x, y))) - \mathcal{L}(\theta^*(\mathcal{S})) &= \log p(\mathbf{y}|\mathbf{x}; \mathcal{S}) - \log p(\mathbf{y}|\mathbf{x}; \mathcal{S} \cup (x, y)) \\ &= \log p(\mathbf{y}|\mathbf{x}; \mathcal{S}) - \log \frac{p(y|x; \mathcal{D}) \cdot p(\mathbf{y}|\mathbf{x}; \mathcal{S})}{p(y|x; \mathcal{S})} = \ell(x, y; \theta^*(\mathcal{D})) - \ell(x, y; \theta^*(\mathcal{S})), \end{aligned} \quad (9)$$

where the pair (x, y) denotes all samples in $\mathcal{D}$ and the second equality comes from Eq.(2) in [78].

By substituting Eq. (9) into Eq. (8), we then can obtain the following reducible representation:

$$\frac{\partial \Phi(\pi)}{\partial \pi_{i,j}} \triangleq K_i \cdot \mathbb{E} \left[\underbrace{\ell(x_j^{(i)}, y_j^{(i)}; \theta^*(\mathcal{D})) - \ell(x_j^{(i)}, y_j^{(i)}; \theta^*(\mathcal{S}_{-(i,1)}))}_{\text{Reducible Two-point Representation}} \right]. \quad (10)$$

¹We provide the details of *projection operation* and a more effective *reservoir rounding* scheme in Appendix E and F.

It is worth noting that the reducible two-point representation in Eq. (10) only requires $\mathcal{O}(h)$ lower-level trainings to compute the optimal parameters $\theta(\mathcal{S}_{-(i,1)}), \forall i \in [h]$ and $\theta(\mathcal{D})$. This is *significantly smaller* than the $\mathcal{O}(h + \sum_{i=1}^h n_i)$ retrainings required by Eq. (8). Furthermore, we emphasize that the above Bayesian probabilistic framework is well motivated, since several prior works have rigorously shown that many neural networks can be modeled via Bayesian probabilistic model [106, 70, 57, 40].

4 Stochastic Gradient Descent via Reducible Two-point Representation

Based on the reducible two-point representation in Eq. (10), we now present a novel stochastic gradient descent algorithm, termed ReSGD, for multi-source bilevel data selection problem.

As summarized in Algorithm 1, ReSGD first initializes a categorical prior $\pi_i^{(0)} \in \Delta_{n_i}$ for each source $\mathcal{D}_i$ and trains a holdout model on the full dataset $\mathcal{D}$. After that, at every iteration t , ReSGD draws K_i samples from each source $\mathcal{D}_i$ according to the current categorical prior $\pi_i^{(t)}$. Then, for any $i \in [h]$, we construct $\mathcal{S}_{-(i,1)}^{(t)}$ by excluding the first draw of $\mathcal{D}_i$, and train the corresponding model $\theta^*(\mathcal{S}_{-(i,1)}^{(t)})$. With these trained models, ReSGD computes the stochastic gradient vector $\hat{\mathbf{g}}_i^{(t)} \triangleq (\hat{g}_{i,1}^{(t)}, \dots, \hat{g}_{i,n_i}^{(t)})$ via Eq. (10), where each coordinate is given by $\hat{g}_{i,j}^{(t)} \triangleq K_i \left[\ell(x_j^{(i)}, y_j^{(i)}; \theta^*(\mathcal{D})) - \ell(x_j^{(i)}, y_j^{(i)}; \theta^*(\mathcal{S}_{-(i,1)}^{(t)})) \right]$. After obtaining $\hat{\mathbf{g}}_i^{(t)}$, we update each categorical prior by projected gradient descent. Note that the symbol $\Pi_{\Delta_{n_i}}(\cdot)$ in Algorithm 1 denotes the Euclidean projection onto Δ_{n_i} . Finally, after running the above SGD iterations for T steps, ReSGD draws K_i data samples from each source $\mathcal{D}_i$ and returns their union.

4.1 Approximation Guarantee under Weak Submodularity

This subsection aims to theoretically verify the effectiveness of our proposed ReSGD. Specifically, we will exploit an intrinsic combinatorial structure of our concerned multi-source data selection problem, known as *weak submodularity* [22, 24, 30, 102], to establish an approximation guarantee for ReSGD.

Before delving into the details, we first introduce some commonly used notations. Specifically, for any set function $f : 2^{\mathcal{D}} \rightarrow \mathbb{R}$, we denote the marginal gain of adding $\mathcal{B}$ to $\mathcal{A}$ by $f(\mathcal{B}|\mathcal{A}) \triangleq f(\mathcal{A} \cup \mathcal{B}) - f(\mathcal{A})$. Especially when $\mathcal{B} \triangleq \{e\}$ is a singleton, we simply write $f(e|\mathcal{A}) \triangleq f(\{e\}|\mathcal{A})$. In addition, we say that a set function f is *monotone* if and only if $f(\mathcal{A}) \leq f(\mathcal{B})$ for any $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{D}$. Since *weak submodularity* is typically used to characterize *utility maximization* problems, in the remainder of this subsection, we consider an equivalent utility function $U(\theta^*(\mathcal{S}))$ of the upper-level loss $\mathcal{L}(\theta^*(\mathcal{S}))$ in Eq. (6). Specifically, we define $U(\theta^*(\mathcal{S})) \triangleq \mathcal{L}(\theta_{\text{init}}) - \mathcal{L}(\theta^*(\mathcal{S}))$, which measures the reduction in the upper-level loss brought by training on $\mathcal{S}$ w.r.t. the initial parameter setup θ_{init} .

With the above notations, we now formally introduce the concept of *weak submodularity*.

Definition 1 ((γ, β) -Weak Submodularity). *A utility set function $f : 2^{\mathcal{D}} \rightarrow \mathbb{R}$ is called (γ, β) -weakly submodular if there exist $\gamma \in (0, 1]$ and $\beta \geq 1$ such that, for any $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{D}$,*

$$\sum_{e \in \mathcal{B} \setminus \mathcal{A}} f(e|\mathcal{A}) \geq \gamma \cdot f(\mathcal{B}|\mathcal{A}), \quad \sum_{e \in \mathcal{B} \setminus \mathcal{A}} f(e|\mathcal{B} \setminus \{e\}) \leq \beta \cdot f(\mathcal{B}|\mathcal{A}). \quad (11)$$

Here, γ is called the *lower submodularity ratio*, while β is called the *upper submodularity ratio*.

Intuitively, from the perspective of data selection, Eq. (11) measures how closely the accumulated marginal gains of individual data points approximate the joint utility gain of selecting the entire subset. Furthermore, weak submodularity has been widely observed in a broad range of applications, including variable selection [24, 30], video summarization [22, 86], neural network pruning [29, 103], and sparse optimal transport [76, 3]. Next, we will show that such a combinatorial property also broadly appears in bilevel data selection problems.

Warm-up: Bayesian Optimal Design [14, 49, 90]. We consider the data selection problem for Bayesian linear regression. Specifically, we assume that the data is generated from a linear model $\mathbf{y} \triangleq X\theta + \epsilon$, where $\mathbf{y} \triangleq [y_1, \dots, y_n]^\top \in \mathbb{R}^n$, $X \triangleq [x_1, \dots, x_n]^\top \in \mathbb{R}^{n \times p}$, the noise vector ϵ follows a Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$ and the model parameter θ is equipped with a Gaussian prior, i.e., $\theta \sim \mathcal{N}(\mathbf{0}, \Lambda)$. In this case, the lower-level optimization of bilevel data selection problem

can be written as $\theta^*(S) \triangleq \arg \min_{\theta} \sum_{(x_i, y_i) \in S} (y_i - x_i^\top \theta)^2 + \sigma^2 \theta^\top \Lambda^{-1} \theta$, and our goal is to select a small subset $S \subseteq \mathcal{D}$ such that the resulting posterior estimator has a small prediction error over the full dataset. Equivalently, we seek to minimize this expected upper-level loss function $\mathcal{L}(\theta^*(S)) \triangleq \frac{1}{2n} \mathbb{E}_{\epsilon, \theta} [\|y - X\theta^*(S)\|_2^2]$. Note that [8, 9] recently showed that $\mathcal{L}(\theta^*(S))$ is closely related to the Bayesian V-optimal criterion $\mathcal{L}_V(\theta^*(S)) \triangleq \frac{1}{2n} \mathbb{E}_{\epsilon, \theta} [\|X(\theta - \theta^*(S))\|_2^2]$. Specifically, when the number of data points n is sufficiently large, these two objectives differ only by an additive constant $\sigma^2/2$. Therefore, minimizing $\mathcal{L}(\theta^*(S))$ can be approximately reduced to minimizing $\mathcal{L}_V(\theta^*(S))$. More importantly, we further observe that the utility function induced by the Bayesian V-optimal criterion $\mathcal{L}_V(\theta^*(S))$ is (γ, β) -weakly submodular, as stated below.

Theorem 2 (Proof in Appendix B). *The utility function $U_V(S) \triangleq \mathcal{L}_V(\mathbf{0}) - \mathcal{L}_V(\theta^*(S))$ induced by the Bayesian V-optimal criterion $\mathcal{L}_V(\theta^*(S))$ is **monotone** and $(1/c, c)$ -**weakly submodular**, where $c \triangleq 1 + \frac{s^2}{\sigma^2} \cdot \lambda_{\max}(\Lambda)$, $s \triangleq \max_{i \in [n]} \|x_i\|_2$ and $\lambda_{\max}(\Lambda)$ is the largest eigenvalue of the matrix Λ .*

Motivated by the results of Theorem 2, we next establish an approximation guarantee for our proposed ReSGD algorithm under the *weak submodularity*.

Theorem 3 (Proof in Appendix D). *Let $n \triangleq |\mathcal{D}|$, $K_{\max} \triangleq \max_{i \in [h]} K_i$, learning rate $\eta \triangleq \sqrt{1/nK_{\max}^2 \cdot T}$ and S^* denotes the optimal solution of Eq. (6). If the utility function $U(\theta^*(S)) \triangleq \mathcal{L}(\theta_{\text{init}}) - \mathcal{L}(\theta^*(S))$ induced by Problem Eq. (6) is (γ, β) -weakly monotone submodular, then the subset S returned by Algorithm 1 satisfies*

$$\mathbb{E}[U(\theta^*(S))] \geq \left(\frac{\gamma^2}{\beta + \beta(1-\gamma) + \gamma^2} \right) \cdot U(\theta^*(S^*)) - \mathcal{O}\left(\sqrt{\frac{n \cdot K_{\max}^2}{T}}\right).$$

Remark 2. *Theorem 3 implies that, when the induced utility function $U(\theta^*(S))$ is (γ, β) -weakly monotone submodular, if we set $T \triangleq \mathcal{O}(nK_{\max}^2/\epsilon^2)$, the subset yielded by our proposed Algorithm 1 can secure a utility value of $(\rho \cdot U(\theta^*(S^*)) - \epsilon)$ where $\rho \triangleq \frac{\gamma^2}{\beta + \beta(1-\gamma) + \gamma^2}$. Equivalently, in terms of the original loss function, we have $\mathbb{E}[\mathcal{L}(\theta^*(S))] \leq \rho \cdot \mathcal{L}(\theta^*(S^*)) + (1-\rho) \cdot \mathcal{L}(\theta_{\text{init}}) + \epsilon$.*

5 Experiments

This section empirically validates the effectiveness of our proposed ReSGD algorithm via two different applications: Data summarization [9, 80, 95] and Replay-based continual learning [8, 48, 104, 116].

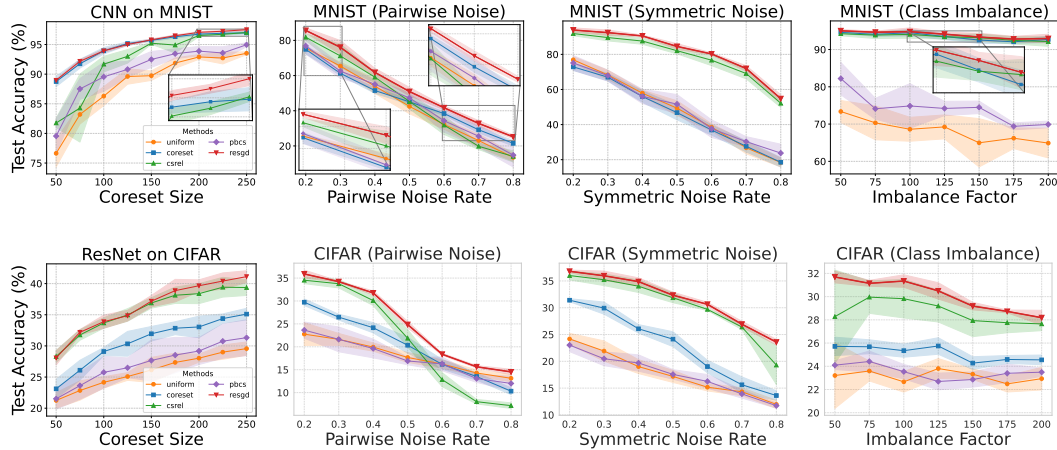


Figure 1: Performance comparison between ReSGD and other baselines on data summarization, illustrating the superior performance and stability of ReSGD under various data corruption scenarios.

Data Summarization. We conduct experiments on three benchmarks: MNIST [25], CIFAR-10 [62], and CIFAR-100 [62]. To ensure a fair evaluation, we follow the model architecture and settings of [8, 104, 116]. We compare our ReSGD against a set of competitive baselines, including Greedy Coresets [8], PBCS [116], and CSRel [104]. Additionally, uniform sampling is included to provide

Table 1: Performance comparison on continual learning benchmarks. We mark the Best results in bold. Our memory construction method performs on par with Greedy Coreset on the MNIST dataset and consistently outperforms other baselines on more complex datasets.

Method	PermMNIST	SplitMNIST	SplitCIFAR-10	SplitCIFAR-100
Uniform	91.82 $\pm$ 0.94	78.09 $\pm$ 0.80	33.66 $\pm$ 1.11	7.06 $\pm$ 0.40
k -Means Features	93.22 $\pm$ 1.47	78.29 $\pm$ 0.59	34.38 $\pm$ 1.26	8.05 $\pm$ 0.54
k -Center Embedding	93.69 $\pm$ 0.59	78.70 $\pm$ 0.56	34.07 $\pm$ 2.14	8.97 $\pm$ 0.47
Hardest	88.17 $\pm$ 1.44	77.04 $\pm$ 0.84	25.35 $\pm$ 1.48	6.43 $\pm$ 0.37
iCaRL	93.68 $\pm$ 0.45	79.62 $\pm$ 0.50	33.26 $\pm$ 0.82	8.31 $\pm$ 0.22
OCS	84.86 $\pm$ 2.69	75.07 $\pm$ 0.91	34.12 $\pm$ 1.94	9.27 $\pm$ 0.45
BCSR	93.81 $\pm$ 0.91	78.30 $\pm$ 0.81	35.14 $\pm$ 3.64	12.11 $\pm$ 1.24
Greedy Coreset	95.67 $\pm$ 0.21	79.31 $\pm$ 0.57	34.88 $\pm$ 2.76	8.41 $\pm$ 0.25
PBCS	94.22 $\pm$ 0.61	76.30 $\pm$ 0.81	34.07 $\pm$ 2.14	13.02 $\pm$ 0.29
CSReL-CL-Prv	95.02 $\pm$ 0.23	78.64 $\pm$ 0.36	36.82 $\pm$ 0.95	15.49 $\pm$ 0.84
ReSGD	95.72 $\pm$ 0.12	80.03 $\pm$ 0.17	37.68 $\pm$ 0.13	16.70 $\pm$ 0.12

a lower-bound performance reference. The experimental details are provided in the Appendix K. Furthermore, to showcase the effectiveness of ReSGD in more practical and challenging scenarios, we follow the data corruptions via label noise (pairwise and symmetric) and class imbalance in [116]. The details are deferred to Appendix J. As shown in Figure 1, ReSGD consistently outperforms all baselines in terms of test accuracy across different coreset sizes, while exhibiting stronger *robustness* to data corruptions, including label noise and class imbalance. Similar trends are observed on CIFAR-100, with detailed results provided in Appendix G.

Continual Learning. We evaluate ReSGD on Split MNIST [112], Perm MNIST [43], Split CIFAR-10, and Split CIFAR-100. To ensure a fair evaluation, we follow the model architecture and settings of [8, 104, 116]. We set the memory size to 100 for Split MNIST and Perm MNIST, and 200 for Split CIFAR-10 and Split CIFAR-100. For the Split CIFAR-100 dataset, we split a total of 100 classes into 10 disjoint tasks and use ResNet-18 [51] as the backbone. Detailed experiment settings and hyperparameters are shown in Appendix K. The baselines include Uniform, k -Means Features [85], k -Center Embedding [96], Hardest [2], iCaRL [92], OCS [111], BCSR [48], Greedy Coreset [8], PBCS [116], and CSReL-CL-Prv [104]. The average accuracies in Table 1 show that our method performs on par with Greedy Coreset on the MNIST datasets and consistently outperforms all other baselines on the more complex datasets (Split CIFAR-10 and Split CIFAR-100), highlighting the superiority of our approach. Additionally, the experimental results indicate that considering previous tasks can further enhance CL performance. Further demonstrations of the effectiveness and compatibility of our method with other CL approaches are deferred to Appendix H.

Further Analysis. We conduct effectiveness study in Appendix I and meanwhile provide a detailed sensitivity analysis in Appendix K.

6 Conclusion

In this paper, we present a novel probabilistic relaxation for the multi-source data selection problem, which only assigns probability mass to *feasible* subsets and thus naturally induces a *lossless* rounding scheme. Most importantly, we show that this relaxation admits an effective two-point gradient estimator, avoiding both the inverse-Hessian computation in *0-1 relaxation* and the unbounded score in prior *probabilistic relaxation*. From a Bayesian perspective, we further derive a reducible representation that reduces the gradient-computation cost from $\mathcal{O}(h + \sum_{i=1}^h n_i)$ retrains to $\mathcal{O}(h)$, where h is the number of sources and $\sum_{i=1}^h n_i$ is the total sample size. Building upon this reducible representation, we develop a stochastic algorithm, termed ReSGD and prove that, under (γ, β) -weak submodularity, its returned subset achieves a $(\frac{\gamma^2}{\beta + \beta(1-\gamma) + \gamma^2})$ -approximation guarantee with respect to the optimal subset S^* . Finally, extensive experiments on data summarization and continual learning validate the effectiveness of our ReSGD.

References

- [1] Alexander A Ageev and Maxim I Sviridenko. Pipe rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8:307–328, 2004.
- [2] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019.
- [3] David Alvarez-Melis, Tommi Jaakkola, and Stefanie Jegelka. Structured optimal transport. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pages 1771–1780. PMLR, 2018.
- [4] Ahmed Attia, Sven Leyffer, and Todd S Munson. Stochastic learning approach for binary optimization: Application to bayesian optimal design of experiments. *SIAM Journal on Scientific Computing*, 44(2):B395–B427, 2022.
- [5] Amit Attia, Matan Schliserman, Uri Sherman, and Tomer Koren. Fast last-iterate convergence of SGD in the smooth interpolation regime. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.
- [6] Ari S Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. *arXiv preprint arXiv:1805.08289*, 2018.
- [7] Dimitri Bertsekas. *Convex optimization algorithms*. Athena Scientific, 2015.
- [8] Zalán Borsos, Mojmír Mutný, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in neural information processing systems*, 33:14879–14890, 2020.
- [9] Zalán Borsos, Mojmír Mutný, Marco Tagliasacchi, and Andreas Krause. Data summarization via bilevel optimization. *Journal of Machine Learning Research*, 25(73):1–53, 2024.
- [10] Jerome Bracken and James T McGill. Mathematical programs with optimization problems in the constraints. *Operations research*, 21(1):37–44, 1973.
- [11] Fanchen Bu, Hyeonsoo Jo, Soo Yong Lee, Sungsoo Ahn, and Kijung Shin. Tackling prevalent conditions in unsupervised combinatorial optimization: Cardinality, minimum, covering, and more. In *Forty-first International Conference on Machine Learning*, 2024.
- [12] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.
- [13] Grigori Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.
- [14] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical science*, pages 273–304, 1995.
- [15] Powei Chang, Jinpeng Zhang, Bowen Chen, Chenyu Wang, Chenlu Guo, Yixing Zhang, Yukang Gao, JianXiang Xiang, Yue Gao, Chaoqun Sun, Yiyi Chen, and Dongying kong. SPICE: Submodular penalized information–conflict selection for efficient large language model training. In *The Fourteenth International Conference on Learning Representations*, 2026.
- [16] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018.
- [17] Arslan Chaudhry, Marc’Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

- [18] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, P Dokuania, P Torr, and M Ranzato. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019.
- [19] Chandra Chekuri, Jan Vondrak, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 575–584, 2010.
- [20] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.
- [21] Lesi Chen, Yaohua Ma, and Jingzhao Zhang. Near-optimal nonconvex-strongly-convex bilevel optimization with fully first-order oracles. *Journal of Machine Learning Research*, 26(109):1–56, 2025.
- [22] Lin Chen, Moran Feldman, and Amin Karbasi. Weakly submodular maximization beyond cardinality constraints: Does randomization help greedy? In *International Conference on Machine Learning*, pages 804–813. PMLR, 2018.
- [23] Ziliang Chen, Yongsan Zheng, Zhao-Rong Lai, Zhanfu Yang, Cuixi Li, Yang Liu, and Liang Lin. Quadratic coreset selection: Certifying and reconciling sequence and token mining for efficient instruction tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- [24] Abhimanyu Das and David Kempe. Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection. *Journal of Machine Learning Research*, 19(3):1–34, 2018.
- [25] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- [26] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.
- [27] Justin Dumouchelle, Esther Julien, Jannis Kurtz, and Elias Boutros Khalil. Neur2biLO: Neural bilevel optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [28] Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.
- [29] Marwa El Halabi, Suraj Srinivas, and Simon Lacoste-Julien. Data-efficient structured pruning via submodular optimization. *Advances in Neural Information Processing Systems*, 35:36613–36626, 2022.
- [30] Ethan R Elenberg, Rajiv Khanna, Alexandros G Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *The Annals of Statistics*, 46(6B):3539–3568, 2018.
- [31] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.
- [32] Lawrence C Evans. *Measure theory and fine properties of functions*. Chapman and Hall/CRC, 2025.
- [33] Ziqing Fan, Siyuan Du, Shengchao Hu, Pingjie Wang, Li Shen, Zhang Ya, Dacheng Tao, and Yanfeng Wang. Combatting dimensional collapse in llm pre-training data via diversified file selection. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [34] Ziqing Fan, Yuqiao Xian, Yan Sun, Ke Shen, and Li Shen. Joint selection for large-scale pre-training data via policy gradient-based mask learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

- [35] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.
- [36] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.
- [37] Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *International conference on machine learning*, pages 1165–1173. PMLR, 2017.
- [38] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning*, pages 1568–1577. PMLR, 2018.
- [39] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.
- [40] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.
- [41] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.
- [42] Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.
- [43] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.
- [44] Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.
- [45] Riccardo Grazi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pages 3748–3758. PMLR, 2020.
- [46] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.
- [47] William W Hager. Updating the inverse of a matrix. *SIAM review*, 31(2):221–239, 1989.
- [48] Jie Hao, Kaiyi Ji, and Mingrui Liu. Bilevel coreset selection in continual learning: A new formulation and algorithm. *Advances in Neural Information Processing Systems*, 36, 2024.
- [49] Abolfazl Hashemi, Mahsa Ghasemi, Haris Vikalo, and Ufuk Topcu. Submodular observation selection and information gathering for quadratic models. In *International Conference on Machine Learning*, pages 2653–2662. PMLR, 2019.
- [50] Chaoyang He, Haishan Ye, Li Shen, and Tong Zhang. Milenas: Efficient neural architecture search via mixed-level reformulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11993–12002, 2020.
- [51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [52] Xingkang He, Lang Chen, Difei Cheng, Vijay Gupta, et al. Revisit last-iterate convergence of msgd under milder requirement on step size. *Advances in Neural Information Processing Systems*, 35:36559–36570, 2022.

- [53] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023.
- [54] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.
- [55] Xijie Huang, Zechun Liu, Shih-Yang Liu, and Kwang-Ting Cheng. Robust and efficient quantization-aware training via coreset selection. *Transactions on Machine Learning Research*, 2024.
- [56] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pages 4882–4892. PMLR, 2021.
- [57] Laurent Valentin Jospin, Hamid Laga, Farid Boussaid, Wray Buntine, and Mohammed Benamoun. Hands-on bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, 17(2):29–48, 2022.
- [58] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.
- [59] G. Khashayar and G. R. Manuel. Non-submodular function maximization subject to a matroid constraint, with applications, 2019.
- [60] Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in neural information processing systems*, 34:14488–14501, 2021.
- [61] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
- [62] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [63] Lilly Kumari, Shengjie Wang, Arnav Das, Tianyi Zhou, and Jeff Bilmes. An end-to-end submodular framework for data-efficient in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3293–3308, 2024.
- [64] Lilly Kumari, Shengjie Wang, Tianyi Zhou, Nikhil Sarda, Anthony Rowe, and Jeff Bilmes. Bumblebee: Dynamic KV-cache streaming submodular summarization for infinite-context transformers. In *First Conference on Language Modeling*, 2024.
- [65] Guanghui Lan. *First-order and stochastic optimization methods for machine learning*, volume 1. Springer, 2020.
- [66] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.
- [67] Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. Data-efficient fine-tuning for llm-based recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 365–374, 2024.
- [68] Bo Liu, Mao Ye, Stephen Wright, Peter Stone, and Qiang Liu. Bome! bilevel optimization made easy: A simple first-order approach. *Advances in neural information processing systems*, 35:17248–17262, 2022.
- [69] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

- [70] Yuhang Liu, Wenyong Dong, Lei Zhang, Dong Gong, and Qinfeng Shi. Variational bayesian dropout with a hierarchical prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7124–7133, 2019.
- [71] Zijian Liu and Zhengyuan Zhou. Revisiting the last-iterate convergence of stochastic gradient methods. In *The Twelfth International Conference on Learning Representations*, 2024.
- [72] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.
- [73] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International conference on artificial intelligence and statistics*, pages 1540–1552. PMLR, 2020.
- [74] Zhaosong Lu and Sanyou Mei. First-order penalty methods for bilevel optimization. *SIAM Journal on Optimization*, 34(2):1937–1969, 2024.
- [75] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR, 2015.
- [76] Piyushi Manupriya, Pratik Jawanpuria, Karthik S. Gurumoorthy, SakethaNath Jagarlapudi, and Bamdev Mishra. Submodular framework for structured-sparse optimal transport. In *Forty-first International Conference on Machine Learning*, 2024.
- [77] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.
- [78] Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltingen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR, 2022.
- [79] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.
- [80] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization: Identifying representative elements in massive data. *Advances in Neural Information Processing Systems*, 26, 2013.
- [81] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.
- [82] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.
- [83] Y Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013.
- [84] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.
- [85] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.
- [86] Lan N. Nguyen and My T. Thai. Efficient algorithms for monotone non-submodular maximization with partition matroid constraint. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4807–4813. International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.
- [87] Oleksiy Ostapenko, Pau Rodriguez, Massimo Caccia, and Laurent Charlin. Continual learning via local module composition. *Advances in Neural Information Processing Systems*, 34:30298–30312, 2021.

- [88] Panos M Pardalos and Naina Kuvoro. An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds. *Mathematical Programming*, 46:321–328, 1990.
- [89] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pages 737–746. PMLR, 2016.
- [90] Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. Modern bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024.
- [91] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32, 2019.
- [92] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.
- [93] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.
- [94] Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *Advances in Neural Information Processing Systems*, 31, 2018.
- [95] Jacob Schreiber, Jeffrey Bilmes, and William Stafford Noble. apricot: Submodular selection for data summarization in python. *Journal of Machine Learning Research*, 21(161):1–6, 2020.
- [96] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.
- [97] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated back-propagation for bilevel optimization. In *The 22nd international conference on artificial intelligence and statistics*, pages 1723–1732. PMLR, 2019.
- [98] Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18(52):1–11, 2017.
- [99] Daouda Sow, Kaiyi Ji, and Yingbin Liang. On the convergence theory for hessian-free bilevel algorithms. *Advances in Neural Information Processing Systems*, 35:4136–4149, 2022.
- [100] Shivakanth Sujit, Somjit Nath, Pedro Braga, and Samira Ebrahimi Kahou. Prioritizing samples in reinforcement learning with reducible loss. *Advances in Neural Information Processing Systems*, 36:23237–23258, 2023.
- [101] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.
- [102] Theophile Thiery and Justin Ward. Two-sided weak submodularity for matroid constrained optimization and regression. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178, pages 3605–3634. PMLR, 02–05 Jul 2022.
- [103] Sadegh Tofigh, Mohammad Askarizadeh, M Omair Ahmad, MNS Swamy, and Kim Khoa Nguyen. Acli: A cnn pruning framework leveraging adjacent convolutional layer interdependence and γ -weakly submodularity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [104] Ruilin Tong, Yuhang Liu, Javen Qinfeng Shi, and Dong Gong. Coreset selection via reducible loss in continual learning. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [105] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*, pages 398–414. Springer, 2022.

- [106] Veit David Wild, Sahra Ghalebikesabi, Dino Sejdinovic, and Jeremias Knoblauch. A rigorous link between deep ensembles and (variational) bayesian methods. *Advances in Neural Information Processing Systems*, 36:39782–39811, 2023.
- [107] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.
- [108] Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*, 2018.
- [109] Qingsen Yan, Dong Gong, Yuhang Liu, Anton Van Den Hengel, and Javen Qinfeng Shi. Learning bayesian sparse networks with full experience replay for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 109–118, 2022.
- [110] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3014–3023, 2021.
- [111] Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. In *International Conference on Learning Representations*, 2022.
- [112] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. Pmlr, 2017.
- [113] Jipeng Zhang, Yaxuan Qin, Renjie Pi, Weizhong Zhang, Rui Pan, and Tong Zhang. Tagcos: Task-agnostic gradient clustered coreset selection for instruction tuning data. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4671–4686, 2025.
- [114] Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bilevel optimization: Foundations and applications in signal processing and machine learning. *IEEE Signal Processing Magazine*, 41(1):38–59, 2024.
- [115] Bo Zhou, Ruiwei Jiang, and Siqian Shen. Learning to solve bilevel programs with binary tender. In *The Twelfth International Conference on Learning Representations*, 2024.
- [116] Xiao Zhou, Renjie Pi, Weizhong Zhang, Yong Lin, Zonghao Chen, and Tong Zhang. Probabilistic bilevel coreset selection. In *International Conference on Machine Learning*, pages 27287–27302. PMLR, 2022.

Appendix

Contents

A	Related Work	18
B	Proof of Theorem 2 and the Equivalence of the V-Optimal Criterion	18
B.1	Proof of Theorem 2	18
B.2	Equivalence of the V-Optimal Criterion	18
C	Gradient Estimators for $\Phi(\pi)$	22
C.1	Policy Gradient Estimator	22
C.2	Two-point Gradient Estimator	22
C.3	Reducible Two-point Representation	24
D	Proof of Theorem 3	24
E	Projection Calculation	29
F	Effective Reservoir Rounding	30
G	Data Summarization on Cifar-100 Dataset	32
H	Coreset Selection for Existing Continual Learning Methods	32
I	Experiment with Adversarial Initialization	33
J	Data Corruption	34
K	Experimental Details	35
K.1	Data Summarization Experiment Details	36
K.2	Continual Learning Experiment Details	37
L	Limitations and Broader Impact	39

A Related Work

Bilevel Continuous Optimization. Bilevel continuous optimization (BCO) provides a natural framework for problems whose objective depends on the solution of another optimization problem, and has been widely used in hyperparameter optimization [37, 38, 97], meta-learning [36, 91], and neural architecture search [31, 69, 50]. Since its early formulation in mathematical programming [10], BCO has attracted extensive attention in both optimization and machine learning [114]. A major line of work focuses on hypergradient-based methods, which can be broadly categorized into approximate implicit differentiation (AID) methods [42, 44, 45, 56, 89] and iterative differentiation (ITD) methods [37, 56, 75]. AID methods compute the hypergradient by differentiating the optimality conditions of the lower-level problem, typically involving Hessian inverse-vector products or their approximations, while ITD methods differentiate through the trajectory of the lower-level optimization procedure. Another line of work adopts penalty-based reformulations, which transform the bilevel problem into a constrained or penalized single-level problem by replacing the lower-level solution requirement with its optimality conditions [21, 74, 99, 68].

Continual Learning. Continual learning aims to train a model on a sequence of tasks while preserving the knowledge acquired from previously seen data. A central challenge in this setting is *catastrophic forgetting* [39, 77], where the model quickly loses its performance on old tasks after being updated on new ones. Existing methods address this issue from different perspectives, including regularization-based methods [16, 61, 66, 94, 109, 112], parameter isolation based methods [87, 105, 110], and replay-based methods [8, 9, 12, 72, 104, 111, 116]. Among them, replay-based continual learning combats forgetting by maintaining a small memory buffer of representative samples from previous tasks and replaying them during future training. Since the memory budget is typically much smaller than the accumulated training data, the quality of the retained samples becomes critical to the effectiveness of rehearsal. This has motivated a series of coreset selection methods for continual learning [8, 9, 116, 48, 104]. For example, [8] extends coreset selection to deep neural networks by formulating it as a bilevel optimization problem. After that, [116] further introduces probabilistic masks to perform global subset selection. Next, [48] revisits bilevel coreset selection in continual learning and proposes a new 0-1 relaxation formulation. More recently, [104] design a greedy method according to the reducible loss proposed by [78].

Weakly Submodular Maximization. The notion of γ -weak submodularity was originally introduced by [24], who showed that the standard greedy algorithm achieves a $(1 - e^{-\gamma})$ -approximation for maximizing a monotone γ -weakly submodular function under a *cardinality* constraint. Subsequently, [59] studied weakly submodular maximization beyond simple cardinality constraints and proved that the standard greedy algorithm achieves a $\frac{0.4\gamma^2}{\sqrt{r\gamma}+1}$ -approximation for maximizing a monotone γ -weakly submodular function under a matroid constraint, where r denotes the rank of the matroid. Unfortunately, this approximation guarantee explicitly depends on r and therefore deteriorates as the matroid rank increases. Note that the partition constraints considered in this paper are a special case of matroid constraints. Later, [102] introduced the upper submodularity ratio β and developed a distorted local search algorithm for monotone (γ, β) -weakly submodular maximization. This distorted local search typically requires repeated swap operations and will incur substantial computational overhead, making it difficult to apply to large-scale selection problems.

B Proof of Theorem 2 and the Equivalence of the V-Optimal Criterion

B.1 Proof of Theorem 2

In this subsection, we show the detailed verification of Theorem 2. Before that, we recall the bilevel data selection problem for Bayesian linear regression. Specifically, the data is generated from a linear model $\mathbf{y} \triangleq X\boldsymbol{\theta} + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon}$, where $\mathbf{y} \triangleq [y_1, \dots, y_n]^\top \in \mathbb{R}^n$, $X \triangleq [x_1, \dots, x_n]^\top \in \mathbb{R}^{n \times p}$, the noise vector $\boldsymbol{\epsilon}$ follows a Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_n)$ and the model parameter $\boldsymbol{\theta}$ is equipped with a Gaussian prior, i.e., $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \Lambda)$. In this case, the lower-level optimization of bilevel data selection problem can be written as $\boldsymbol{\theta}^*(\mathcal{S}) \triangleq \arg \min_{\boldsymbol{\theta}} \sum_{(x_i, y_i) \in \mathcal{S}} (y_i - x_i^\top \boldsymbol{\theta})^2 + \sigma^2 \boldsymbol{\theta}^\top \Lambda^{-1} \boldsymbol{\theta}$, and our goal is to select a small subset $\mathcal{S} \subseteq \mathcal{D} \triangleq \{x_1, \dots, x_n\}$ such that the resulting posterior estimator $\boldsymbol{\theta}^*(\mathcal{S})$ has a small prediction error over the full dataset. More importantly, from Bayesian statistical

theory [14, 49, 90], we can show that $\theta^*(\mathcal{S})$ has the following closed form:

$$\theta^*(\mathcal{S}) \triangleq (X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X_{\mathcal{S}}^T \mathbf{y}_{\mathcal{S}},$$

where $X_{\mathcal{S}}$ and $\mathbf{y}_{\mathcal{S}}$ denotes the matrix X and the vector $\mathbf{y}$ restricted to rows induced by $\mathcal{S}$, respectively.

Since the proof of Theorem 2 will repeatedly involve matrix inverses, we begin by recalling the classical Sherman-Morrisson-Woodbury formula, that is to say,

Lemma 1 (Sherman-Morrisson-Woodbury formula [54, 47]). *Let A , U , C and V be matrices of conformable sizes. Then, $(A + UCV)^{-1} \triangleq A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$.*

With all above concepts and formula, we next prove Theorem 2.

Proof. We first simplify the Bayesian V-optimal criterion $\mathcal{L}_V(\theta^*(\mathcal{S})) \triangleq \frac{1}{2n} \mathbb{E}_{\epsilon, \theta} [\|X(\theta - \theta^*(\mathcal{S}))\|_2^2]$, namely,

$$\begin{aligned} \mathcal{L}_V(\theta^*(\mathcal{S})) &\triangleq \frac{1}{2n} \mathbb{E}_{\epsilon, \theta} [\|X(\theta^*(\mathcal{S}) - \theta)\|_2^2] \\ &= \frac{1}{2n} \mathbb{E}_{\epsilon, \theta} [\|X((X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X_{\mathcal{S}}^T \mathbf{y}_{\mathcal{S}} - \theta)\|_2^2] \\ &= \frac{1}{2n} \mathbb{E}_{\epsilon, \theta} [\|X((X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X_{\mathcal{S}}^T (X_{\mathcal{S}} \theta + \epsilon_{\mathcal{S}}) - \theta)\|_2^2] \\ &= \frac{1}{2n} \mathbb{E}_{\epsilon, \theta} [\|X((X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X_{\mathcal{S}}^T X_{\mathcal{S}} \theta - \theta + X((X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X_{\mathcal{S}}^T \epsilon_{\mathcal{S}}))\|_2^2] \\ &= \frac{1}{2n} \mathbb{E}_{\epsilon, \theta} [\|X((X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X_{\mathcal{S}}^T \epsilon_{\mathcal{S}} - \sigma^2 (X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} \Lambda^{-1} \theta)\|_2^2] \\ &= \frac{1}{2n} \mathbb{E}_{\epsilon, \theta} [\|X(X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X_{\mathcal{S}}^T \epsilon_{\mathcal{S}}\|_2^2] + \frac{1}{2n} \mathbb{E}_{\epsilon, \theta} [\|\sigma^2 \cdot X(X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} \Lambda^{-1} \theta\|_2^2] \\ &= \frac{1}{2n} \text{Tr}[X(X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X_{\mathcal{S}}^T \mathbb{E}_{\epsilon} [\epsilon_{\mathcal{S}} \epsilon_{\mathcal{S}}^T] X_{\mathcal{S}} (X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X^T] \\ &\quad + \frac{1}{2n} \text{Tr}[\sigma^2 \cdot X(X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} \Lambda^{-1} \mathbb{E}_{\theta} [\theta \theta^T] \Lambda^{-1} (X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X^T \cdot \sigma^2] \\ &= \frac{\sigma^2}{2n} \text{Tr}[X_{\mathcal{S}} (X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X^T X (X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X_{\mathcal{S}}^T] \\ &\quad + \frac{\sigma^2}{2n} \text{Tr}[\sigma^2 \cdot X(X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} \Lambda^{-1} (X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X^T] \\ &= \frac{\sigma^2}{2n} \text{Tr}[(X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X^T X (X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X_{\mathcal{S}}^T X_{\mathcal{S}}] \\ &\quad + \frac{\sigma^2}{2n} \text{Tr}[(X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X^T X (X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} (\sigma^2 \cdot \Lambda^{-1})] \\ &= \frac{\sigma^2}{2n} \text{Tr}[(X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X^T X] = \frac{\sigma^2}{2n} \text{Tr}[X(X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X^T], \end{aligned}$$

where $\mathbb{E}_{\epsilon} [\epsilon_{\mathcal{S}} \epsilon_{\mathcal{S}}^T] = \sigma^2 \mathbf{I}_{|\mathcal{S}|}$ and $\mathbb{E}_{\theta} [\theta \theta^T] = \Lambda$.

With this equality, we can rewrite the utility function $U_V(\mathcal{S})$ as

$$\begin{aligned} U_V(\mathcal{S}) &\triangleq \mathcal{L}_V(\mathbf{0}) - \mathcal{L}_V(\theta^*(\mathcal{S})) \\ &= \frac{\sigma^2}{2n} \left(\text{Tr}[\sigma^{-2} \cdot X \Lambda X^T] - \text{Tr}[X(X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1})^{-1} X^T] \right). \end{aligned}$$

For simplicity, in the remainder of this proof, for any subset $\mathcal{S}$, we set $\mathcal{M}_{\mathcal{S}} \triangleq X_{\mathcal{S}}^T X_{\mathcal{S}} + \sigma^2 \Lambda^{-1}$.

Next, we show this utility function $U_V(\mathcal{S})$ satisfies the (γ, β) -weak submodularity.

At first, for any subset $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{R} \triangleq \mathcal{B} \setminus \mathcal{A}$, we have that

$$\begin{aligned}
U_V(\mathcal{B}) - U_V(\mathcal{A}) &= \frac{\sigma^2}{2n} (\text{Tr}[X\mathcal{M}_{\mathcal{A}}^{-1}]X^T] - \text{Tr}[X\mathcal{M}_{\mathcal{B}}^{-1}]X^T]) \\
&= \frac{\sigma^2}{2n} (\text{Tr}[X\mathcal{M}_{\mathcal{A}}^{-1}]X^T] - \text{Tr}[X(\mathcal{M}_{\mathcal{A}} + X_{\mathcal{R}}^T X_{\mathcal{R}})^{-1} X^T]) \\
&= \frac{\sigma^2}{2n} (\text{Tr}[X\mathcal{M}_{\mathcal{A}}^{-1}]X^T] - \text{Tr}[X(\mathcal{M}_{\mathcal{A}}^{-1} - \mathcal{M}_{\mathcal{A}}^{-1} X_{\mathcal{R}}^T (I + X_{\mathcal{R}} \mathcal{M}_{\mathcal{A}}^{-1} X_{\mathcal{R}}^T)^{-1} X_{\mathcal{R}} \mathcal{M}_{\mathcal{A}}^{-1}) X^T]) \\
&= \frac{\sigma^2}{2n} (\text{Tr}[X\mathcal{M}_{\mathcal{A}}^{-1} X_{\mathcal{R}}^T (I + X_{\mathcal{R}} \mathcal{M}_{\mathcal{A}}^{-1} X_{\mathcal{R}}^T)^{-1} X_{\mathcal{R}} \mathcal{M}_{\mathcal{A}}^{-1} X^T]) \\
&= \frac{\sigma^2}{2n} (\text{Tr}[(I + X_{\mathcal{R}} \mathcal{M}_{\mathcal{A}}^{-1} X_{\mathcal{R}}^T)^{-1} X_{\mathcal{R}} \mathcal{M}_{\mathcal{A}}^{-1} X^T X \mathcal{M}_{\mathcal{A}}^{-1} X_{\mathcal{R}}^T])
\end{aligned}$$

Note that $\lambda_{\min}(I + X_{\mathcal{R}} \mathcal{M}_{\mathcal{A}}^{-1} X_{\mathcal{R}}^T) \geq 1$, so we have that

$$U_V(\mathcal{B}) - U_V(\mathcal{A}) \leq \frac{\sigma^2}{2n} \text{Tr}[X_{\mathcal{R}} \mathcal{M}_{\mathcal{A}}^{-1} X^T X \mathcal{M}_{\mathcal{A}}^{-1} X_{\mathcal{R}}^T].$$

Furthermore, we also have that (Note that $x_e \in \mathbb{R}^{p \times 1}$)

$$\begin{aligned}
U_V(\mathcal{A} \cup \{e\}) - U_V(\mathcal{A}) &= \frac{\sigma^2}{2n} (\text{Tr}[(1 + x_e^T \mathcal{M}_{\mathcal{A}}^{-1} x_e)^{-1} x_e^T \mathcal{M}_{\mathcal{A}}^{-1} X^T X \mathcal{M}_{\mathcal{A}}^{-1} x_e]) \\
&= \frac{\sigma^2}{2n} \cdot \frac{\text{Tr}[x_e^T \mathcal{M}_{\mathcal{A}}^{-1} X^T X \mathcal{M}_{\mathcal{A}}^{-1} x_e]}{1 + x_e^T \mathcal{M}_{\mathcal{A}}^{-1} x_e} \\
&\geq \frac{\sigma^2}{2n} \cdot \frac{\text{Tr}[x_e^T \mathcal{M}_{\mathcal{A}}^{-1} X^T X \mathcal{M}_{\mathcal{A}}^{-1} x_e]}{1 + \frac{s^2}{\sigma^2} \cdot \lambda_{\max}(\Lambda)},
\end{aligned}$$

where the final inequality follows from $\mathcal{M}_{\mathcal{A}} = X_S^T X_S + \sigma^2 \Lambda^{-1}$ such that $x_e^T \mathcal{M}_{\mathcal{A}}^{-1} x_e \leq x_e^T (\sigma^2 \Lambda^{-1})^{-1} x_e \leq \frac{x_e^T \Lambda x_e}{\sigma^2} \leq \frac{s^2}{\sigma^2} \lambda_{\max}(\Lambda)$.

Here, $s \triangleq \max_{i \in [n]} \|x_i\|_2$ and $\lambda_{\max}(\Lambda)$ is the largest eigenvalue of the matrix Λ .

As a result, we have that

$$\begin{aligned}
\sum_{e \in \mathcal{B} \setminus \mathcal{A}} (U_V(\mathcal{A} \cup \{e\}) - U_V(\mathcal{A})) &\geq \frac{\sigma^2}{2n} \cdot \frac{\sum_{e \in \mathcal{B} \setminus \mathcal{A}} \text{Tr}[x_e^T \mathcal{M}_{\mathcal{A}}^{-1} X^T X \mathcal{M}_{\mathcal{A}}^{-1} x_e]}{1 + \frac{s^2}{\sigma^2} \cdot \lambda_{\max}(\Lambda)} \\
&= \frac{\sigma^2}{2n} \cdot \frac{\text{Tr}[X_{\mathcal{R}} \mathcal{M}_{\mathcal{A}}^{-1} X^T X \mathcal{M}_{\mathcal{A}}^{-1} X_{\mathcal{R}}^T]}{1 + \frac{s^2}{\sigma^2} \cdot \lambda_{\max}(\Lambda)} \geq \frac{U_V(\mathcal{B}) - U_V(\mathcal{A})}{1 + \frac{s^2}{\sigma^2} \cdot \lambda_{\max}(\Lambda)},
\end{aligned}$$

where $\mathcal{R} \triangleq \mathcal{B} \setminus \mathcal{A}$.

Secondly, for any subset $\mathcal{A} \subseteq \mathcal{B}$ and $\mathcal{R} \triangleq \mathcal{B} \setminus \mathcal{A}$, we have that

$$\begin{aligned}
U_V(\mathcal{B}) - U_V(\mathcal{A}) &= \frac{\sigma^2}{2n} (\text{Tr}[X\mathcal{M}_{\mathcal{A}}^{-1}]X^T] - \text{Tr}[X\mathcal{M}_{\mathcal{B}}^{-1}]X^T]) \\
&= \frac{\sigma^2}{2n} (\text{Tr}[X(\mathcal{M}_{\mathcal{B}} - X_{\mathcal{R}}^T X_{\mathcal{R}})^{-1} X^T] - \text{Tr}[X\mathcal{M}_{\mathcal{B}}^{-1}]X^T]) \\
&= \frac{\sigma^2}{2n} (\text{Tr}[X(\mathcal{M}_{\mathcal{B}}^{-1} + \mathcal{M}_{\mathcal{B}}^{-1} X_{\mathcal{R}}^T (I - X_{\mathcal{R}} \mathcal{M}_{\mathcal{B}}^{-1} X_{\mathcal{R}}^T)^{-1} X_{\mathcal{R}} \mathcal{M}_{\mathcal{B}}^{-1}) X^T - \text{Tr}[X\mathcal{M}_{\mathcal{B}}^{-1}]X^T]) \\
&= \frac{\sigma^2}{2n} (\text{Tr}[X\mathcal{M}_{\mathcal{B}}^{-1} X_{\mathcal{R}}^T (I - X_{\mathcal{R}} \mathcal{M}_{\mathcal{B}}^{-1} X_{\mathcal{R}}^T)^{-1} X_{\mathcal{R}} \mathcal{M}_{\mathcal{B}}^{-1} X^T]) \\
&= \frac{\sigma^2}{2n} (\text{Tr}[(I - X_{\mathcal{R}} \mathcal{M}_{\mathcal{B}}^{-1} X_{\mathcal{R}}^T)^{-1} X_{\mathcal{R}} \mathcal{M}_{\mathcal{B}}^{-1} X^T X \mathcal{M}_{\mathcal{B}}^{-1} X_{\mathcal{R}}^T])
\end{aligned}$$

Note that $\lambda_{\max}(I - X_{\mathcal{R}}\mathcal{M}_{\mathcal{B}}^{-1}X_{\mathcal{R}}^T) \leq 1$, so we have that

$$U_V(\mathcal{B}) - U_V(\mathcal{A}) \geq \frac{\sigma^2}{2n} \text{Tr}[X_{\mathcal{R}}\mathcal{M}_{\mathcal{B}}^{-1}X^T X \mathcal{M}_{\mathcal{B}}^{-1}X_{\mathcal{R}}^T].$$

Furthermore, we also have that

$$U_V(\mathcal{B}) - U_V(\mathcal{B} \setminus \{e\}) = \frac{\sigma^2}{2n} \left(\text{Tr}[(1 - x_e^T \mathcal{M}_{\mathcal{B}}^{-1}x_e)^{-1} x_e^T \mathcal{M}_{\mathcal{B}}^{-1}X^T X \mathcal{M}_{\mathcal{B}}^{-1}x_e] \right).$$

From Sherman-Morrisson-Woodbury formula, we also can show that

$$(1 - x_e^T \mathcal{M}_{\mathcal{B}}^{-1}x_e)^{-1} = 1 + x_e^T (\mathcal{M}_{\mathcal{B}} - x_e x_e^T)^{-1} x_e = 1 + x_e^T \mathcal{M}_{\mathcal{B} \setminus \{e\}}^{-1} x_e \leq 1 + \frac{s^2}{\sigma} \lambda_{\max}(\Lambda),$$

where $\mathcal{M}_{\mathcal{B} \setminus \{e\}} \triangleq X_{\mathcal{B} \setminus \{e\}}^T X_{\mathcal{B} \setminus \{e\}} + \sigma^2 \Lambda^{-1}$ such that $x_e^T \mathcal{M}_{\mathcal{B} \setminus \{e\}}^{-1} x_e \leq x_e^T (\sigma^2 \Lambda^{-1})^{-1} x_e \leq \frac{x_e^T \Lambda x_e}{\sigma^2} \leq \frac{s^2}{\sigma} \lambda_{\max}(\Lambda)$.

As a result, we have that

$$U_V(\mathcal{B}) - U_V(\mathcal{B} \setminus \{e\}) \leq \frac{\sigma^2}{2n} \left(1 + \frac{s^2}{\sigma} \lambda_{\max}(\Lambda) \right) \cdot \text{Tr}[x_e^T \mathcal{M}_{\mathcal{B}}^{-1}X^T X \mathcal{M}_{\mathcal{B}}^{-1}x_e].$$

So we can have that

$$\begin{aligned} \sum_{e \in \mathcal{B} \setminus \mathcal{A}} (U_V(\mathcal{B}) - U_V(\mathcal{B} \setminus \{e\})) &\leq \frac{\sigma^2}{2n} \left(1 + \frac{s^2}{\sigma} \lambda_{\max}(\Lambda) \right) \cdot \sum_{e \in \mathcal{B} \setminus \mathcal{A}} \text{Tr}[x_e^T \mathcal{M}_{\mathcal{B}}^{-1}X^T X \mathcal{M}_{\mathcal{B}}^{-1}x_e] \\ &= \frac{\sigma^2}{2n} \left(1 + \frac{s^2}{\sigma} \lambda_{\max}(\Lambda) \right) \cdot \text{Tr}[X_{\mathcal{R}}\mathcal{M}_{\mathcal{B}}^{-1}X^T X \mathcal{M}_{\mathcal{B}}^{-1}X_{\mathcal{R}}^T] \\ &\leq \left(1 + \frac{s^2}{\sigma} \lambda_{\max}(\Lambda) \right) \cdot (U_V(\mathcal{B}) - U_V(\mathcal{A})). \end{aligned}$$

In summary, The utility function $U_V(\mathcal{S}) \triangleq \mathcal{L}_V(\mathbf{0}) - \mathcal{L}_V(\boldsymbol{\theta}^*(\mathcal{S}))$ be the induced by the Bayesian V-optimal criterion $\mathcal{L}_V(\boldsymbol{\theta}^*(\mathcal{S}))$ is monotone and $(1/c, c)$ -**weakly submodular**, where $c \triangleq 1 + \frac{s^2}{\sigma^2} \cdot \lambda_{\max}(\Lambda)$, $s \triangleq \max_{i \in [n]} \|x_i\|_2$ and $\lambda_{\max}(\Lambda)$ is the largest eigenvalue of the matrix Λ . $\square$

B.2 Equivalence of the V-Optimal Criterion

In this subsection, we show the equivalence between the total prediction error $\mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S})) \triangleq \frac{1}{2n} \mathbb{E}_{\epsilon, \boldsymbol{\theta}} [\|\mathbf{y} - X\boldsymbol{\theta}^*(\mathcal{S})\|_2^2]$ and the Bayesian V-optimal criterion $\mathcal{L}_V(\boldsymbol{\theta}^*(\mathcal{S})) \triangleq \frac{1}{2n} \mathbb{E}_{\epsilon, \boldsymbol{\theta}} [\|X(\boldsymbol{\theta} - \boldsymbol{\theta}^*(\mathcal{S}))\|_2^2]$. Specifically, when the number of data points n is sufficiently large, these two objectives differ only by an additive constant $\sigma^2/2$. Consequently, the weak submodularity of $\mathcal{L}_V(\boldsymbol{\theta}^*(\mathcal{S}))$ established above, to some extent, also implies the weak submodularity of the total prediction error $\mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}))$.

Lemma 2. *Under the aforementioned assumptions of Bayesian linear regression, we can show that*

$$\lim_{n \rightarrow \infty} \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S})) - \mathcal{L}_V(\boldsymbol{\theta}^*(\mathcal{S})) = \frac{\sigma^2}{2}.$$

Proof. **Fixed $\boldsymbol{\theta}$,** we have,

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S})) &= \frac{1}{2n} \mathbb{E}_{\epsilon} [\|\mathbf{y} - X\boldsymbol{\theta}^*(\mathcal{S})\|_2^2] \\ &= \frac{1}{2n} \mathbb{E}_{\epsilon} [\|X\boldsymbol{\theta} - X\boldsymbol{\theta}^*(\mathcal{S})\|_2^2] - \frac{1}{n} \mathbb{E}_{\epsilon} [\epsilon^\top (X\boldsymbol{\theta}^*(\mathcal{S}) - X\boldsymbol{\theta})] + \frac{1}{2n} \mathbb{E}_{\epsilon} [\|\epsilon\|_2^2] \\ &= \mathcal{L}_V(\boldsymbol{\theta}^*(\mathcal{S})) - \frac{1}{n} \mathbb{E}_{\epsilon} [\epsilon^\top (X\boldsymbol{\theta}^*(\mathcal{S}) - X\boldsymbol{\theta})] + \frac{\sigma^2}{2} \\ &= \mathcal{L}_V(\boldsymbol{\theta}^*(\mathcal{S})) - \frac{1}{n} \mathbb{E}_{\epsilon} \left[\sum_{i \in \mathcal{S}} \epsilon_i (x_i^T \boldsymbol{\theta}^*(\mathcal{S}) - x_i^T \boldsymbol{\theta}) \right] - \frac{1}{n} \mathbb{E}_{\epsilon} \left[\sum_{i \in [n] \setminus \mathcal{S}} \epsilon_i (x_i^T \boldsymbol{\theta}^*(\mathcal{S}) - x_i^T \boldsymbol{\theta}) \right] + \frac{\sigma^2}{2} \end{aligned}$$

When $n \rightarrow \infty$, we have $\lim_{n \rightarrow \infty} \frac{1}{n} \mathbb{E}_\epsilon [\sum_{i \in S} \epsilon_i (x_i^T \theta^*(S) - x_i^T \theta)] = 0$ since S is a finite set. Since $\theta^*(S)$ only depends on ϵ_S , we have

$$\mathbb{E}_\epsilon \left[\sum_{i \in [n] \setminus S} \epsilon_i (x_i^T \theta^*(S) - x_i^T \theta) \right] = \sum_{i \in [n] \setminus S} \mathbb{E}_\epsilon [\epsilon_i] \mathbb{E}_\epsilon [x_i^T \theta^*(S) - x_i^T \theta] = 0.$$

As a consequence, as $\lim_{n \rightarrow \infty} \mathcal{L}(\theta^*(S)) - \mathcal{L}_V(\theta^*(S)) = \frac{\sigma^2}{2}$.

□

C Gradient Estimators for $\Phi(\pi)$

In this section, we will show two different gradient estimators for our proposed probabilistic relaxation $\Phi(\pi)$, namely policy gradient estimator and two-point estimator in Eq.(8).

C.1 Policy Gradient Estimator

At first, from Eq.(7), we know that

$$\begin{aligned} \Phi(\pi) &\triangleq \mathbb{E}_{d_{i,k} \sim \pi_i, \forall i,k} \left[\mathcal{L}(\theta^*(\cup_{i=1}^h \cup_{k=1}^{K_i} d_{i,k})) \right] \\ &= \sum_{d_{i,k} \in \mathcal{D}_i \cup \{\emptyset\}, \forall i,k} \left(\mathcal{L}(\theta^*(\cup_{i=1}^h \cup_{k=1}^{K_i} d_{i,k})) \cdot \prod_{i=1}^h \prod_{k=1}^{K_i} p(d_{i,k} \mid \pi_i) \right), \end{aligned}$$

where the symbol $d_{i,k} \sim \pi_i$ means that we sample $d_{i,k}$ from $\mathcal{D}_i \cup \{\emptyset\}$ according to the categorical distribution π_i . Then, like the policy gradient methods [116, 101, 4], we can show that

$$\begin{aligned} \frac{\partial \Phi(\pi)}{\partial \pi_{i,j}} &= \sum_{d_{i,k} \in \mathcal{D}_i \cup \{\emptyset\}, \forall i \in [h], k \in [K_i]} \left(\mathcal{L}(\theta^*(\cup_{i=1}^h \cup_{k=1}^{K_i} d_{i,k})) \nabla_{\pi_{i,j}} \left(\prod_{i=1}^h \prod_{k=1}^{K_i} p(d_{i,k} \mid \pi_i) \right) \right) \\ &= \sum_{d_{i,k} \in \mathcal{D}_i \cup \{\emptyset\}, \forall i,k} \left(\mathcal{L}(\theta^*(\cup_{i=1}^h \cup_{k=1}^{K_i} d_{i,k})) \nabla_{\pi_{i,j}} \log \left(\prod_{i=1}^h \prod_{k=1}^{K_i} p(d_{i,k} \mid \pi_i) \right) \left(\prod_{i=1}^h \prod_{k=1}^{K_i} p(d_{i,k} \mid \pi_i) \right) \right) \\ &= \mathbb{E}_{d_{i,k} \sim \pi_i, \forall i \in [h], k \in [K_i]} \left[\mathcal{L}(\theta^*(\cup_{i=1}^h \cup_{k=1}^{K_i} d_{i,k})) \nabla_{\pi_{i,j}} \log \left(\prod_{i=1}^h \prod_{k=1}^{K_i} p(d_{i,k} \mid \pi_i) \right) \right] \\ &= \mathbb{E}_{d_{i,k} \sim \pi_i, \forall i \in [h], k \in [K_i]} \left[\mathcal{L}(\theta^*(\cup_{i=1}^h \cup_{k=1}^{K_i} d_{i,k})) \sum_{k=1}^{K_i} \left(\frac{\mathbb{I}[d_{i,k} = (x_j^{(i)}, y_j^{(i)})]}{\pi_{i,j}} - \frac{\mathbb{I}[d_{i,k} = \emptyset]}{1 - \sum_{k=1}^{K_i} \pi_{i,k}} \right) \right], \end{aligned}$$

where the second equality follows from $\frac{d \log(x)}{dx} = 1/x$ and $\mathbb{I}[\cdot]$ represents the indicator function.

Note that, like Eq. (5), the policy gradient estimator of our proposed probabilistic relaxation still involves an *unbounded* score factor $\left(\frac{\mathbb{I}[d_{i,k} = (x_j^{(i)}, y_j^{(i)})]}{\pi_{i,j}} - \frac{\mathbb{I}[d_{i,k} = \emptyset]}{1 - \sum_{k=1}^{K_i} \pi_{i,k}} \right)$. Specifically, when $\pi_{i,k} \rightarrow 0$ or $\sum_{k=1}^{K_i} \pi_{i,k} \rightarrow 1$, the factor $\left(\frac{\mathbb{I}[d_{i,k} = (x_j^{(i)}, y_j^{(i)})]}{\pi_{i,j}} - \frac{\mathbb{I}[d_{i,k} = \emptyset]}{1 - \sum_{k=1}^{K_i} \pi_{i,k}} \right)$ will also become unbounded.

C.2 Two-point Gradient Estimator

In this subsection, we verify Theorem 1.

Proof. At first, we can show that

$$\begin{aligned}
\frac{\partial \Phi(\boldsymbol{\pi})}{\partial \pi_{i,j}} &= \sum_{d_{i,\hat{k}} \in \mathcal{D}_i \cup \{\emptyset\}, \forall \hat{i} \in [h], \hat{k} \in [K_i]} \left(\mathcal{L}(\boldsymbol{\theta}^*(\cup_{\hat{i}=1}^h \cup_{\hat{k}=1}^{K_i} d_{\hat{i},\hat{k}})) \nabla_{\pi_{i,j}} \left(\prod_{\hat{i}=1}^h \prod_{\hat{k}=1}^{K_i} p(d_{\hat{i},\hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right) \right) \\
&= \sum_{d_{i,\hat{k}} \in \mathcal{D}_i \cup \{\emptyset\}, \forall \hat{i}, \hat{k}} \left(\mathcal{L}(\boldsymbol{\theta}^*(\cup_{\hat{i}=1}^h \cup_{\hat{k}=1}^{K_i} d_{\hat{i},\hat{k}})) \sum_{k=1}^{K_i} \left(\nabla_{\pi_{i,j}} p(d_{i,k} \mid \boldsymbol{\pi}_i) \cdot \prod_{(\hat{i},\hat{k}) \neq (i,k)} p(d_{\hat{i},\hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right) \right) \\
&= \sum_{k=1}^{K_i} \left(\sum_{d_{i,\hat{k}} \in \mathcal{D}_i \cup \{\emptyset\}, \forall \hat{i}, \hat{k}} \left(\mathcal{L}(\boldsymbol{\theta}^*(\cup_{\hat{i}=1}^h \cup_{\hat{k}=1}^{K_i} d_{\hat{i},\hat{k}})) \nabla_{\pi_{i,j}} p(d_{i,k} \mid \boldsymbol{\pi}_i) \cdot \prod_{(\hat{i},\hat{k}) \neq (i,k)} p(d_{\hat{i},\hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right) \right) \\
&= K_i \cdot \left(\sum_{d_{i,\hat{k}} \in \mathcal{D}_i \cup \{\emptyset\}, \forall \hat{i}, \hat{k}} \left(\mathcal{L}(\boldsymbol{\theta}^*(\cup_{\hat{i}=1}^h \cup_{\hat{k}=1}^{K_i} d_{\hat{i},\hat{k}})) \nabla_{\pi_{i,j}} p(d_{i,1} \mid \boldsymbol{\pi}_i) \cdot \prod_{(\hat{i},\hat{k}) \neq (i,1)} p(d_{\hat{i},\hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right) \right) \\
&= K_i \left(\sum_{d_{i,\hat{k}} \in \mathcal{D}_i \cup \{\emptyset\}, (\hat{i},\hat{k}) \neq (i,1)} \left(\sum_{d_{i,1} \in \mathcal{D}_i \cup \{\emptyset\}} \mathcal{L}(\boldsymbol{\theta}^*(\cup_{\hat{i},\hat{k}} d_{\hat{i},\hat{k}})) \nabla_{\pi_{i,j}} p(d_{i,1} \mid \boldsymbol{\pi}_i) \right) \prod_{(\hat{i},\hat{k}) \neq (i,1)} p(d_{\hat{i},\hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right), \tag{12}
\end{aligned}$$

where the fourth equality comes from the independence between $\{d_{i,1}, d_{i,2}, \dots, d_{i,K_i}\}$.

When $d_{i,1} = (x_j^{(i)}, y_j^{(i)})$ or $d_{i,1} = \emptyset$, we can know that $\nabla_{\pi_{i,j}} p(d_{i,1} \mid \boldsymbol{\pi}_i) = 1$ or $\nabla_{\pi_{i,j}} p(d_{i,1} \mid \boldsymbol{\pi}_i) = -1$, respectively. As for other cases, we have $\nabla_{\pi_{i,j}} p(d_{i,1} \mid \boldsymbol{\pi}_i) = 0$. Thus, if we define $\mathcal{S}_{-(i,1)} \triangleq \bigcup_{(i',k') \neq (i,1)} d_{i',k'}$ as the subset obtained by excluding the first draw $d_{i,1}$ from source $\mathcal{D}_i$, we have the following equality:

$$\sum_{d_{i,1} \in \mathcal{D}_i \cup \{\emptyset\}} \mathcal{L}(\boldsymbol{\theta}^*(\cup_{\hat{i},\hat{k}} d_{\hat{i},\hat{k}})) \nabla_{\pi_{i,j}} p(d_{i,1} \mid \boldsymbol{\pi}_i) = \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)} \cup (x_j^{(i)}, y_j^{(i)}))) - \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)})).$$

As a result, we have that

$$\frac{\partial \Phi(\boldsymbol{\pi})}{\partial \pi_{i,j}} \triangleq K_i \cdot \mathbb{E} \left[\mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)} \cup (x_j^{(i)}, y_j^{(i)}))) - \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)})) \right], \tag{13}$$

where the expectation is taken over all random draws except $d_{i,1}$. $\square$

C.3 Reducible Two-point Representation

At first, we review the Bayes rule: for any random event A , B and C , we have

$$p(A | B, C) \triangleq \frac{p(B | A, C) \cdot p(A | C)}{p(B | C)}.$$

As a result, in Eq.(9), if we set $A = \mathbf{y}$, $B = y$ and $C = \{\mathbf{x}, x, \mathcal{S}\}$, we have that

$$\begin{aligned} p(\mathbf{y} | \mathbf{x}; \mathcal{S} \cup (x, y)) &= \frac{p(y | \mathbf{x}; \mathcal{S} \cup (x, y)) \cdot p(\mathbf{y} | \mathbf{x}, x, \mathcal{S})}{p(y | \mathbf{x}, x, \mathcal{S})} \\ &= \frac{p(y | \mathbf{x}; \mathcal{S} \cup \mathcal{D}) \cdot p(\mathbf{y} | \mathbf{x}, \mathcal{S})}{p(y | \mathbf{x}, \mathcal{S})} = \frac{p(y | \mathbf{x}; \mathcal{D}) \cdot p(\mathbf{y} | \mathbf{x}, \mathcal{S})}{p(y | \mathbf{x}, \mathcal{S})}, \end{aligned}$$

where the second equality comes from the the conditional independence $p(y_i | x_i, x_j; \mathcal{S}) = p(y_i | x_i; \mathcal{S})$ (Please refer to Eq.(2) in [100]).

As a result, we have that

$$\log p(\mathbf{y} | \mathbf{x}; \mathcal{S} \cup (x, y)) - \log p(\mathbf{y} | \mathbf{x}, \mathcal{S}) = \log p(y | \mathbf{x}; \mathcal{D}) - \log p(y | \mathbf{x}, \mathcal{S}).$$

Furthermore, in Bayesian probabilistic framework, the loss function is commonly specified as the negative log-likelihood of the predictive distribution, namely, $\ell(x, y; \boldsymbol{\theta}^*(\mathcal{S})) \triangleq -\log p(y | \mathbf{x}; \mathcal{S})$, then we can have that, throughout the independence between samples:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S})) &= \sum_{i=1}^n \ell(x_i, y_i; \boldsymbol{\theta}^*(\mathcal{S})) \\ &= -\sum_{i=1}^n \log p(y_i | x_i; \mathcal{S}) = -\log \left(\prod_{i=1}^n p(y_i | x_i; \mathcal{S}) \right) = -\log p(\mathbf{y} | \mathbf{x}; \mathcal{S}). \end{aligned}$$

As a result, we have

$$\begin{aligned} &\mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S} \cup (x, y))) - \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S})) \\ &= \log p(\mathbf{y} | \mathbf{x}; \mathcal{S}) - \log p(\mathbf{y} | \mathbf{x}; \mathcal{S} \cup (x, y)) \\ &= \log p(y | \mathbf{x}; \mathcal{D}) - \log p(y | \mathbf{x}, \mathcal{S}) = \ell(x, y; \boldsymbol{\theta}^*(\mathcal{D})) - \ell(x, y; \boldsymbol{\theta}^*(\mathcal{S})). \end{aligned}$$

D Proof of Theorem 3

In this section, we prove the core theorem of this paper, namely, Theorem 3. Before that, we prove a key first-order inequality regarding the upper-level utility function $U(\boldsymbol{\theta}^*(\mathcal{S})) \triangleq \mathcal{L}(\boldsymbol{\theta}_{\text{init}}) - \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}))$.

Next, we define some commonly used notations. At first, we set the corresponding probabilistic relaxation of the upper-level utility function $U(\boldsymbol{\theta}^*(\mathcal{S}))$ as $\Psi(\boldsymbol{\pi})$, that is,

$$\begin{aligned} \Psi(\boldsymbol{\pi}) &\triangleq \sum_{d_{i,k} \in \mathcal{D}_i \cup \{\emptyset\}, \forall i, k} \left(U(\boldsymbol{\theta}^*(\cup_{i=1}^h \cup_{k=1}^{K_i} d_{i,k})) \cdot \prod_{i=1}^h \prod_{k=1}^{K_i} p(d_{i,k} | \boldsymbol{\pi}_i) \right) \\ &= \sum_{d_{i,k} \in \mathcal{D}_i \cup \{\emptyset\}, \forall i, k} \left((\mathcal{L}(\boldsymbol{\theta}_{\text{init}}) - \mathcal{L}(\boldsymbol{\theta}^*(\cup_{i=1}^h \cup_{k=1}^{K_i} d_{i,k}))) \cdot \prod_{i=1}^h \prod_{k=1}^{K_i} p(d_{i,k} | \boldsymbol{\pi}_i) \right) \\ &= \mathcal{L}(\boldsymbol{\theta}_{\text{init}}) - \Phi(\boldsymbol{\pi}), \end{aligned}$$

where $\Phi(\boldsymbol{\pi})$ is our proposed probabilistic relaxation in Eq.(7). Furthermore, for any feasible data subset $\mathcal{S}$ satisfying $|\mathcal{S} \cap \mathcal{D}_i| = K_i$ for all $i \in [h]$, we define its corresponding *uniform prior* as $\boldsymbol{\pi}_{\mathcal{S}} \triangleq \sum_{i=1}^h \frac{1}{K_i} \mathbf{1}_{\mathcal{S} \cap \mathcal{D}_i}$. In other words, for each source $\mathcal{D}_i$, $\boldsymbol{\pi}_{\mathcal{S}}$ places mass $1/K_i$ on every candidate in $\mathcal{S} \cap \mathcal{D}_i$ and mass 0 on the remaining candidates. Note that $\mathbf{1}_{\mathcal{A}}$ denotes the indicator vector of subset $\mathcal{A}$, whose entry is 1 if data candidates is included in $\mathcal{A}$ and 0 otherwise.

With the above preparations in place, we then show that

Theorem 4. When the utility function $U(\boldsymbol{\theta}^*(\mathcal{S})) \triangleq \mathcal{L}(\boldsymbol{\theta}_{\text{init}}) - \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}))$ induced by Problem Eq. (6) is (γ, β) -weakly monotone submodular, for any feasible subset $\mathcal{B}$, namely, $|\mathcal{B} \cap \mathcal{D}_i| \leq K_i, \forall i \in [h]$, the following inequality holds:

$$\gamma^2 \cdot U(\boldsymbol{\theta}^*(\mathcal{B})) - (\beta + \beta(1 - \gamma) + \gamma^2) \cdot \Psi(\boldsymbol{\pi}) \leq \langle \boldsymbol{\pi}_{\mathcal{B}} - \boldsymbol{\pi}, \nabla \Psi(\boldsymbol{\pi}) \rangle.$$

Proof. At first, we investigate $\langle \boldsymbol{\pi}_{\mathcal{B}}, \nabla \Psi(\boldsymbol{\pi}) \rangle$. To this end, for each source $\mathcal{D}_i$, we rewrite $\mathcal{B} \cap \mathcal{D}_i = \{(x_{j_1}^{(i)}, y_{j_1}^{(i)}), \dots, (x_{j_{K_i}}^{(i)}, y_{j_{K_i}}^{(i)})\}$, where $j_1, \dots, j_{K_i} \in [n_i]$ are distinct indices.

$$\begin{aligned} \langle \boldsymbol{\pi}_{\mathcal{S}}, \nabla \Psi(\boldsymbol{\pi}) \rangle &= \left\langle \sum_{i=1}^h \frac{1}{K_i} \mathbf{1}_{\mathcal{S} \cap \mathcal{D}_i}, \nabla \Psi(\boldsymbol{\pi}) \right\rangle = \sum_{i=1}^h \sum_{k=1}^{K_i} \frac{1}{K_i} \cdot \frac{\partial \Psi(\boldsymbol{\pi})}{\partial \pi_{i,j_k}} = - \sum_{i=1}^h \sum_{k=1}^{K_i} \frac{1}{K_i} \cdot \frac{\partial \Phi(\boldsymbol{\pi})}{\partial \pi_{i,j_k}} \\ &= \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[\mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)})) - \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)} \cup (x_{j_k}^{(i)}, y_{j_k}^{(i)}))) \right] \\ &= \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[U(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)} \cup (x_{j_k}^{(i)}, y_{j_k}^{(i)}))) - U(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)})) \right] \\ &= \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[U(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,k)} \cup (x_{j_k}^{(i)}, y_{j_k}^{(i)}))) - U(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,k)})) \right] \end{aligned} \quad (14)$$

where $\mathcal{S}_{-(i,k)} \triangleq \bigcup_{(i',k') \neq (i,k)} d_{i',k'}$ as the subset obtained by excluding the k -th draw $d_{i,k}$ from source $\mathcal{D}_i$ and the final inequality comes from the fact that $\mathcal{S}_{-(i,k)}$ and $\mathcal{S}_{-(i,1)}$ are identically distributed.

Furthermore, if a set function f is (γ, β) -weakly monotone submodular, we firstly can show that for any two data candidates $e_1, e_2 \in \mathcal{D}$ and the subset $\mathcal{D} \subseteq \mathcal{D}$, we have that

$$\gamma(f(e_1|\mathcal{B} \cup \{e_2\}) + f(e_2|\mathcal{B})) = \gamma(f(\mathcal{B} \cup \{e_1, e_2\}) - f(\mathcal{B})) \leq f(e_1|\mathcal{B}) + f(e_2|\mathcal{B}),$$

such that

$$f(e_1|\mathcal{B}) \geq \gamma f(e_1|\mathcal{B} \cup \{e_2\}) - (1 - \gamma)f(e_2|\mathcal{B}). \quad (15)$$

Note that, in the field of set function optimization, we often denote the marginal gain of adding $\mathcal{B}$ to $\mathcal{A}$ by $f(\mathcal{B}|\mathcal{A}) \triangleq f(\mathcal{A} \cup \mathcal{B}) - f(\mathcal{A})$. Especially when $\mathcal{B} \triangleq \{e\}$ is a singleton, we simply write $f(e|\mathcal{A}) \triangleq f(\{e\}|\mathcal{A})$.

According to Eq.(15), we subsequently can have that

$$\begin{aligned} \langle \boldsymbol{\pi}_{\mathcal{S}}, \nabla \Psi(\boldsymbol{\pi}) \rangle &= \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[U(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,k)} \cup (x_{j_k}^{(i)}, y_{j_k}^{(i)}))) - U(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,k)})) \right] \\ &= \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[U(\boldsymbol{\theta}^*((x_{j_k}^{(i)}, y_{j_k}^{(i)}) | \mathcal{S}_{-(i,k)})) \right] \\ &\geq \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[\gamma \cdot U(\boldsymbol{\theta}^*((x_{j_k}^{(i)}, y_{j_k}^{(i)}) | \mathcal{S}_{-(i,k)} \cup d_{i,k})) - (1 - \gamma) \cdot U(\boldsymbol{\theta}^*(d_{i,k} | \mathcal{S}_{-(i,k)})) \right], \end{aligned} \quad (16)$$

where $d_{i,k}$ is the k -th draw from source $\mathcal{D}_i$.

Then, from the first inequality in Eq.(11), we have that

$$\begin{aligned}
& \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[U \left(\boldsymbol{\theta}^* \left((x_{j_k}^{(i)}, y_{j_k}^{(i)}) \mid \mathcal{S}_{-(i,k)} \cup d_{i,k} \right) \right) \right] \\
& \geq \gamma \cdot \mathbb{E} \left[U \left(\boldsymbol{\theta}^* \left(\bigcup_{i=1}^h \bigcup_{k=1}^{K_i} (x_{j_k}^{(i)}, y_{j_k}^{(i)}) \mid \mathcal{S}_{-(i,k)} \cup d_{i,k} \right) \right) \right] \\
& = \gamma \cdot \mathbb{E} \left[U \left(\boldsymbol{\theta}^* (\mathcal{B} \mid \bigcup_{i=1}^h \bigcup_{k=1}^{K_i} d_{i,k}) \right) \right] \\
& = \gamma \cdot \mathbb{E} \left[U \left(\boldsymbol{\theta}^* (\mathcal{B} \cup (\bigcup_{i=1}^h \bigcup_{k=1}^{K_i} d_{i,k})) \right) - U \left(\boldsymbol{\theta}^* (\bigcup_{i=1}^h \bigcup_{k=1}^{K_i} d_{i,k}) \right) \right] \\
& \geq \gamma \cdot \left(U(\boldsymbol{\theta}^*(\mathcal{B})) - \Psi(\boldsymbol{\pi}) \right),
\end{aligned} \tag{17}$$

where the final inequality follows from the monotonicity of the utility function U .

Furthermore, from the second inequality in Eq.(11), we also have,

$$\begin{aligned}
& \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[U \left(\boldsymbol{\theta}^* (d_{i,k} \mid \mathcal{S}_{-(i,k)}) \right) \right] \\
& \leq \beta \cdot \mathbb{E} \left[U \left(\boldsymbol{\theta}^* (\bigcup_{i=1}^h \bigcup_{k=1}^{K_i} d_{i,k}) \right) - U(\boldsymbol{\theta}^*(\emptyset)) \right] \\
& \leq \beta \cdot \mathbb{E} \left[U \left(\boldsymbol{\theta}^* (\bigcup_{i=1}^h \bigcup_{k=1}^{K_i} d_{i,k}) \right) \right] = \beta \cdot \Psi(\boldsymbol{\pi}),
\end{aligned} \tag{18}$$

where the second inequality follows from Eq.(11).

Merging Eq.(18) and Eq.(17) into Eq.(16), we have that

$$\begin{aligned}
& \langle \boldsymbol{\pi}_{\mathcal{S}}, \nabla \Psi(\boldsymbol{\pi}) \rangle \\
& \geq \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[\gamma \cdot U \left(\boldsymbol{\theta}^* ((x_{j_k}^{(i)}, y_{j_k}^{(i)}) \mid \mathcal{S}_{-(i,k)} \cup d_{i,k}) \right) - (1 - \gamma) \cdot U \left(\boldsymbol{\theta}^* (d_{i,k} \mid \mathcal{S}_{-(i,k)}) \right) \right] \\
& \geq \gamma^2 \left(U(\boldsymbol{\theta}^*(\mathcal{B})) - \Psi(\boldsymbol{\pi}) \right) - (1 - \gamma) \beta \cdot \Psi(\boldsymbol{\pi}) \\
& = \gamma^2 \cdot U(\boldsymbol{\theta}^*(\mathcal{B})) - (\beta(1 - \gamma) + \gamma^2) \cdot \Psi(\boldsymbol{\pi}).
\end{aligned} \tag{19}$$

Secondly, we investigate $\langle \boldsymbol{\pi}, \nabla \Psi(\boldsymbol{\pi}) \rangle$.

At first, from the fifth equality in Eq (12) and Eq (13), we have that

$$\begin{aligned}
& \frac{\partial \Psi(\boldsymbol{\pi})}{\partial \pi_{i,j}} = - \frac{\partial \Phi(\boldsymbol{\pi})}{\partial \pi_{i,j}} \\
& = -K_i \left(\sum_{d_{i,\hat{k}} \in \mathcal{D}_i \cup \{\emptyset\}, (\hat{i}, \hat{k}) \neq (i,1)} \left(\sum_{d_{i,1} \in \mathcal{D}_i \cup \{\emptyset\}} \mathcal{L} \left(\boldsymbol{\theta}^* (\bigcup_{\hat{i}, \hat{k}} d_{i,\hat{k}}) \right) \nabla_{\pi_{i,j}} \mathbb{P}(d_{i,1} \mid \boldsymbol{\pi}_i) \right) \prod_{(\hat{i}, \hat{k}) \neq (i,1)} \mathbb{P}(d_{i,\hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right) \\
& = -K_i \left(\sum_{d_{i,\hat{k}} \in \mathcal{D}_i \cup \{\emptyset\}, (\hat{i}, \hat{k}) \neq (i,1)} \mathcal{L} \left(\boldsymbol{\theta}^* ((x_j^{(i)}, y_j^{(i)}) \mid \bigcup_{(\hat{i}, \hat{k}) \neq (k,1)} d_{i,\hat{k}}) \right) \prod_{(\hat{i}, \hat{k}) \neq (i,1)} \mathbb{P}(d_{i,\hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right) \\
& = K_i \left(\sum_{d_{i,\hat{k}} \in \mathcal{D}_i \cup \{\emptyset\}, (\hat{i}, \hat{k}) \neq (i,1)} U \left(\boldsymbol{\theta}^* ((x_j^{(i)}, y_j^{(i)}) \mid \bigcup_{(\hat{i}, \hat{k}) \neq (k,1)} d_{i,\hat{k}}) \right) \prod_{(\hat{i}, \hat{k}) \neq (i,1)} \mathbb{P}(d_{i,\hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right)
\end{aligned}$$

As a result, we have that

$$\begin{aligned}
& \langle \boldsymbol{\pi}, \nabla \Psi(\boldsymbol{\pi}) \rangle \\
&= \sum_{i=1}^h \sum_{j=1}^{n_i} \pi_{i,j} \cdot \frac{\partial \Psi(\boldsymbol{\pi})}{\partial \pi_{i,j}} \\
&= \sum_{i=1}^h \sum_{j=1}^{n_i} \pi_{i,j} K_i \left(\sum_{d_{\hat{i}, \hat{k}} \in \mathcal{D}_{\hat{i}} \cup \{\emptyset\}, (\hat{i}, \hat{k}) \neq (i, 1)} U(\boldsymbol{\theta}^*((x_j^{(i)}, y_j^{(i)}) \mid \cup_{(\hat{i}, \hat{k}) \neq (k, 1)} d_{\hat{i}, \hat{k}})) \prod_{(\hat{i}, \hat{k}) \neq (i, 1)} p(d_{\hat{i}, \hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right) \\
&= \sum_{i=1}^h K_i \sum_{j=1}^{n_i} p(d_{i,1} = (x_j^{(i)}, y_j^{(i)}) \mid \boldsymbol{\pi}_i) \left(\sum_{d_{\hat{i}, \hat{k}} \in \mathcal{D}_{\hat{i}} \cup \{\emptyset\}, (\hat{i}, \hat{k}) \neq (i, 1)} U(\boldsymbol{\theta}^*((x_j^{(i)}, y_j^{(i)}) \mid \cup_{(\hat{i}, \hat{k}) \neq (k, 1)} d_{\hat{i}, \hat{k}})) \prod_{(\hat{i}, \hat{k}) \neq (i, 1)} p(d_{\hat{i}, \hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right) \\
&= \sum_{i=1}^h K_i \sum_{j=1}^{n_i} \left(\sum_{d_{\hat{i}, \hat{k}} \in \mathcal{D}_{\hat{i}} \cup \{\emptyset\}, (\hat{i}, \hat{k}) \neq (i, 1)} U(\boldsymbol{\theta}^*((x_j^{(i)}, y_j^{(i)}) \mid \cup_{(\hat{i}, \hat{k}) \neq (k, 1)} d_{\hat{i}, \hat{k}})) \cdot p(d_{i,1} = (x_j^{(i)}, y_j^{(i)}) \mid \boldsymbol{\pi}_i) \prod_{(\hat{i}, \hat{k}) \neq (i, 1)} p(d_{\hat{i}, \hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right) \\
&= \sum_{i=1}^h K_i \sum_{j=1}^{n_i} \left(\sum_{d_{\hat{i}, \hat{k}} \in \mathcal{D}_{\hat{i}} \cup \{\emptyset\}, \forall \hat{i}, \hat{k}} U(\boldsymbol{\theta}^*(d_{k,1} \mid \cup_{(\hat{i}, \hat{k}) \neq (k, 1)} d_{\hat{i}, \hat{k}})) \prod_{\hat{i}=1}^h \prod_{\hat{k}=1}^{K_{\hat{i}}} p(d_{\hat{i}, \hat{k}} \mid \boldsymbol{\pi}_{\hat{i}}) \right) \\
&= \sum_{i=1}^h K_i \cdot \mathbb{E} \left[U(\boldsymbol{\theta}^*(d_{i,1} \mid \cup_{(\hat{i}, \hat{k}) \neq (k, 1)} d_{\hat{i}, \hat{k}})) \right] = \sum_{i=1}^h \sum_{k=1}^{K_i} \cdot \mathbb{E} \left[U(\boldsymbol{\theta}^*(d_{i,k} \mid \cup_{(\hat{i}, \hat{k}) \neq (k, 1)} d_{\hat{i}, \hat{k}})) \right] \\
&= \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[U(\boldsymbol{\theta}^*(d_{i,k} \mid \mathcal{S}_{-(i,k)})) \right].
\end{aligned}$$

Then, from the weak submodularity of U , we have

$$\langle \boldsymbol{\pi}, \nabla \Psi(\boldsymbol{\pi}) \rangle \leq \sum_{i=1}^h \sum_{k=1}^{K_i} \mathbb{E} \left[U(\boldsymbol{\theta}^*(d_{i,k} \mid \mathcal{S}_{-(i,k)})) \right] \leq \beta \cdot \Psi(\boldsymbol{\pi}), \quad (20)$$

where the final inequality comes from Eq.(11).

As a result, by merging Eq. 20 and Eq. 19, we then have that

$$\begin{aligned}
\langle \boldsymbol{\pi}_B - \boldsymbol{\pi}, \nabla \Psi(\boldsymbol{\pi}) \rangle &= \langle \boldsymbol{\pi}_B, \nabla \Psi(\boldsymbol{\pi}) \rangle - \langle \boldsymbol{\pi}, \nabla \Psi(\boldsymbol{\pi}) \rangle \\
&\geq \gamma^2 \cdot U(\boldsymbol{\theta}^*(\mathcal{B})) - (\beta(1 - \gamma) + \gamma^2) \cdot \Psi(\boldsymbol{\pi}) - \beta \cdot \Psi(\boldsymbol{\pi}) \\
&\geq \gamma^2 \cdot U(\boldsymbol{\theta}^*(\mathcal{B})) - (\beta + \beta(1 - \gamma) + \gamma^2) \cdot \Psi(\boldsymbol{\pi}).
\end{aligned}$$

□

Beside Theorem 4, we next prove the boundedness of $\nabla \Psi(\boldsymbol{\pi})$, that is to say,

Lemma 3. *if we denote the maximum marginal value of the utility function $U(\boldsymbol{\theta}^*(\mathcal{S}))$ as $M_U \triangleq \max_{\mathcal{S} \subseteq \mathcal{D}, e \in \mathcal{D} \setminus \mathcal{S}} (U(\boldsymbol{\theta}^*(\mathcal{S} \cup \{e\})) - U(\boldsymbol{\theta}^*(\mathcal{S})))$, we have that, for any $i \in [h]$, $j \in [n_i]$,*

$$\left| \frac{\partial \Psi(\boldsymbol{\pi})}{\partial \pi_{i,j}} \right| \leq M_U \cdot K_{max},$$

where $K_{max} \triangleq \max_{i \in [h]} K_i$ is the maximum budget over the h communities $\{\mathcal{D}_1, \dots, \mathcal{D}_h\}$. Similarly, we also can infer that the two-point gradient estimator in Eq.(8) also can be bounded by $M_U \cdot K_{max}$.

Next, we prove Theorem 3.

Proof. At first, let $\mathcal{S}^* \subseteq \mathcal{D}$ denote the optimal solution for Eq.(6). Then, we denote $\boldsymbol{\pi}^{(t)} \triangleq (\boldsymbol{\pi}_1^{(t+1)}, \dots, \boldsymbol{\pi}_h^{(t+1)})$ and set the corresponding two-point gradient estimator as $\widehat{\nabla \Phi}(\boldsymbol{\pi})$, namely, $\widehat{\nabla \Phi}(\boldsymbol{\pi}) \triangleq [\frac{\partial \Phi(\boldsymbol{\pi})}{\partial \pi_{i,j}}]_{i,j}$ with $\frac{\partial \Phi(\boldsymbol{\pi})}{\partial \pi_{i,j}} \triangleq K_i \left(\mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)} \cup (x_j^{(i)}, y_j^{(i)}))) - \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)})) \right)$.

Subsequently, according to the projective step in ReSGD, we can show that

$$\begin{aligned}
\left\| \pi^{(t+1)} - \pi_{S^*} \right\|_2^2 &\leq \left\| \pi^{(t)} - \eta \cdot \widehat{\nabla \Phi}(\pi^{(t)}) - \pi_{S^*} \right\|_2^2 \\
&= \left\| \pi^{(t)} - \pi_{S^*} \right\|_2^2 - 2\eta \cdot \left\langle \widehat{\nabla \Phi}(\pi^{(t)}), \pi^{(t)} - \pi_{S^*} \right\rangle + \eta^2 \left\| \widehat{\nabla \Phi}(\pi^{(t)}) \right\|_2^2 \\
&= \left\| \pi^{(t)} - \pi_{S^*} \right\|_2^2 - 2\eta \cdot \left\langle \widehat{\nabla \Phi}(\pi^{(t)}), \pi^{(t)} - \pi_{S^*} \right\rangle + \eta^2 \cdot n \cdot M_U^2 \cdot K_{max}^2
\end{aligned}$$

where the final inequality comes from Lemma 3.

As a result, we have that

$$\begin{aligned}
\mathbb{E} \left\| \pi^{(t+1)} - \pi_{S^*} \right\|_2^2 &\leq \left\| \pi^{(t)} - \eta \cdot \widehat{\nabla \Phi}(\pi^{(t)}) - \pi_{S^*} \right\|_2^2 \\
&= \mathbb{E} \left\| \pi^{(t)} - \pi_{S^*} \right\|_2^2 - 2\eta \cdot \mathbb{E} \left\langle \widehat{\nabla \Phi}(\pi^{(t)}), \pi^{(t)} - \pi_{S^*} \right\rangle + \eta^2 \cdot n \cdot M_U^2 \cdot K_{max}^2 \\
&= \mathbb{E} \left\| \pi^{(t)} - \pi_{S^*} \right\|_2^2 - 2\eta \cdot \mathbb{E} \left[\left\langle \mathbb{E}[\widehat{\nabla \Phi}(\pi^{(t)}) \mid \pi^{(t)}], \pi^{(t)} - \pi_{S^*} \right\rangle \right] + \eta^2 \cdot n \cdot M_U^2 \cdot K_{max}^2 \quad (21) \\
&= \mathbb{E} \left\| \pi^{(t)} - \pi_{S^*} \right\|_2^2 - 2\eta \cdot \mathbb{E} \left[\left\langle \nabla \Phi(\pi^{(t)}), \pi^{(t)} - \pi_{S^*} \right\rangle \right] + \eta^2 \cdot n \cdot M_U^2 \cdot K_{max}^2 \\
&= \mathbb{E} \left\| \pi^{(t)} - \pi_{S^*} \right\|_2^2 + 2\eta \cdot \mathbb{E} \left[\left\langle \nabla \Psi(\pi^{(t)}), \pi^{(t)} - \pi_{S^*} \right\rangle \right] + \eta^2 \cdot n \cdot M_U^2 \cdot K_{max}^2,
\end{aligned}$$

where the final equality follows from $\nabla \Psi = -\nabla \Phi$.

Furthermore, according to (γ, β) -weak submodularity, namely, Theorem 4, we also can show that,

$$\begin{aligned}
\mathbb{E} \left\| \pi^{(t+1)} - \pi_{S^*} \right\|_2^2 &\leq \mathbb{E} \left\| \pi^{(t)} - \pi_{S^*} \right\|_2^2 + 2\eta \cdot \mathbb{E} \left[\left\langle \nabla \Psi(\pi^{(t)}), \pi^{(t)} - \pi_{S^*} \right\rangle \right] + \eta^2 \cdot n \cdot M_U^2 \cdot K_{max}^2 \\
&\leq \mathbb{E} \left\| \pi^{(t)} - \pi_{S^*} \right\|_2^2 - 2\eta \cdot \mathbb{E} \left[\left\langle \nabla \Psi(\pi^{(t)}), \pi_{S^*} - \pi^{(t)} \right\rangle \right] + \eta^2 \cdot n \cdot M_U^2 \cdot K_{max}^2 \\
&\leq \mathbb{E} \left\| \pi^{(t)} - \pi_{S^*} \right\|_2^2 + 2\eta \cdot \left((\beta + \beta(1 - \gamma) + \gamma^2) \cdot \mathbb{E}[\Psi(\pi^{(t)})] - \gamma^2 \cdot U(\theta^*(\mathcal{B})) \right) + \eta^2 \cdot n \cdot M_U^2 \cdot K_{max}^2
\end{aligned}$$

We further can have that

$$\begin{aligned}
&\sum_{t=0}^{T-1} 2\eta \cdot \left(\gamma^2 \cdot U(\theta^*(\mathcal{B})) - (\beta + \beta(1 - \gamma) + \gamma^2) \cdot \mathbb{E}[\Psi(\pi^{(t)})] \right) \\
&\leq \sum_{t=0}^{T-1} \left(\mathbb{E} \left\| \pi^{(t)} - \pi_{S^*} \right\|_2^2 - \mathbb{E} \left\| \pi^{(t+1)} - \pi_{S^*} \right\|_2^2 \right) + nT\eta^2 \cdot M_U^2 \cdot K_{max}^2 \quad (22) \\
&= \mathbb{E} \left\| \pi^{(0)} - \pi_{S^*} \right\|_2^2 - \mathbb{E} \left\| \pi^{(T)} - \pi_{S^*} \right\|_2^2 + nT\eta^2 \cdot M_U^2 \cdot K_{max}^2.
\end{aligned}$$

Due to the randomness of the index τ , we finally can infer that the subset return by our ReSGD satisfies that

$$\begin{aligned}
\mathbb{E}[U(\theta^*(S))] &= \mathbb{E}[\Psi(\pi^{(\tau)})] = \frac{\sum_{t=0}^{T-1} \mathbb{E}[\Psi(\pi^{(t)})]}{T} \\
&\geq \left(\frac{\gamma^2}{\beta + \beta(1 - \gamma) + \gamma^2} \right) \cdot U(\theta^*(\mathcal{B})) - \frac{\mathbb{E} \left\| \pi^{(0)} - \pi_{S^*} \right\|_2^2 - \mathbb{E} \left\| \pi^{(T)} - \pi_{S^*} \right\|_2^2}{2\eta \cdot T \cdot (\beta + \beta(1 - \gamma) + \gamma^2)} - \frac{n \cdot M_U^2 \cdot K_{max}^2}{2(\beta + \beta(1 - \gamma) + \gamma^2)} \cdot \eta,
\end{aligned}$$

where the second equality follows from the randomness of the index τ and the final inequality comes from Eq.(22).

Since the Lemma B.1 in [102] show that $\beta(1 - \gamma) + \gamma^2 \geq \frac{3}{4}$, when $\eta \triangleq \sqrt{\frac{1}{nK_{max}^2 T}}$, we have

$$\mathbb{E}[U(\theta^*(S))] \geq \frac{\gamma^2}{\beta + \beta(1 - \gamma) + \gamma^2} \cdot U(\theta^*(S^*)) - \mathcal{O}\left(\sqrt{\frac{n \cdot K_{max}^2}{T}}\right).$$

□

Algorithm 2: Projection onto Δ_n

Input: Vector $\mathbf{z} \in \mathbb{R}^n$ **Output:** Vector $\boldsymbol{\pi} \triangleq \Pi_{\Delta_n}(\mathbf{z})$ Set $\mathbf{z}_+ \leftarrow \max\{\mathbf{z}, \mathbf{0}\}$;**if** $\mathbf{1}^\top \mathbf{z}_+ \leq 1$ **then** **return** $\mathbf{z}_+$;Find $\lambda^* \geq 0$ such that $\sum_{j=1}^n \max\{z_j - \lambda^*, 0\} = 1$ by bisection.Set $\pi_j \leftarrow \max\{z_j - \lambda^*, 0\}$ for all $j \in [n]$;**return** $\boldsymbol{\pi}$;

E Projection Calculation

In Algorithm 1, each prior vector is updated by projected gradient descent, namely,

$$\boldsymbol{\pi}_i^{(t+1)} \triangleq \Pi_{\Delta_{n_i}} \left(\boldsymbol{\pi}_i^{(t)} - \eta \hat{\mathbf{g}}_i^{(t)} \right), \quad \forall i \in [h],$$

where $\Delta_{n_i} \triangleq \{\mathbf{p} \in [0, 1]^{n_i} : \|\mathbf{p}\|_1 \leq 1\}$ denotes the sub-probability simplex.

If we set $\mathbf{z} \triangleq \boldsymbol{\pi}_i^{(t)} - \eta \hat{\mathbf{g}}_i^{(t)}$, the projection operation is equivalent to solving the following quadratic programming:

$$\min_{\boldsymbol{\pi} \in \mathbb{R}^n} \frac{1}{2} \|\boldsymbol{\pi} - \mathbf{z}\|_2^2, \quad \text{s.t.} \quad \boldsymbol{\pi} \geq \mathbf{0}, \quad \mathbf{1}^\top \boldsymbol{\pi} \leq 1. \quad (23)$$

The main purpose of this section is to provide an efficient method for this singly constrained quadratic programming problem Eq. (23), namely, Algorithm 2.

Specifically, Algorithm 2 separates the projection into two cases. It first computes the nonnegative part $\mathbf{z}_+ \triangleq \max\{\mathbf{z}, \mathbf{0}\}$. When $\|\mathbf{z}_+\|_1 \leq 1$, this vector already lies in Δ_n and is therefore the Euclidean projection of $\mathbf{z}$ onto Δ_n .

Specifically, Algorithm 2 proceeds in two cases. It first computes the nonnegative truncation $\mathbf{z}_+ \triangleq \max\{\mathbf{z}, \mathbf{0}\}$. If $\|\mathbf{z}_+\|_1 \leq 1$, then $\mathbf{z}_+$ is already feasible for Eq. (23). In this case, no further normalization is needed, and the projection is simply given by $\mathbf{z}_+$.

If $\|\mathbf{z}_+\|_1 > 1$, we find the optimal dual variable $\lambda^* \geq 0$ by solving the following one-dimensional equation: $\sum_{j=1}^n \max\{z_j - \lambda^*, 0\} = 1$. Since the left-hand side is monotonically decreasing in λ^* , this equation can be efficiently solved by bisection [88]. After obtaining λ^* , the projected vector is computed coordinate-wise as $\pi_j \triangleq \max\{z_j - \lambda^*, 0\}$, $\forall j \in [n]$.

We now justify Algorithm 2. If $\mathbf{z}_+ \triangleq \max\{\mathbf{z}, \mathbf{0}\}$ satisfies $\mathbf{1}^\top \mathbf{z}_+ \leq 1$, then $\mathbf{z}_+ \in \Delta_n$. Moreover, for each coordinate, the closest nonnegative point to z_j is $\max\{z_j, 0\}$. Hence, under this case, the nonnegativity projection already satisfies the ℓ_1 constraint, and therefore $\Pi_{\Delta_n}(\mathbf{z}) = \mathbf{z}_+$.

It remains to consider the case $\mathbf{1}^\top \mathbf{z}_+ > 1$. In this case, the constraint $\mathbf{1}^\top \boldsymbol{\pi} \leq 1$ must be active at the optimum; otherwise, one could still decrease the objective by moving $\boldsymbol{\pi}$ closer to $\mathbf{z}_+$. Therefore, the problem reduces to finding the closest point to $\mathbf{z}$ over the set $\{\boldsymbol{\pi} \in \mathbb{R}_+^n : \mathbf{1}^\top \boldsymbol{\pi} = 1\}$. Consider the following Lagrangian

$$\mathcal{L}(\boldsymbol{\pi}, \lambda, \boldsymbol{\mu}) = \frac{1}{2} \|\boldsymbol{\pi} - \mathbf{z}\|_2^2 + \lambda(\mathbf{1}^\top \boldsymbol{\pi} - 1) - \boldsymbol{\mu}^\top \boldsymbol{\pi},$$

where $\lambda \in \mathbb{R}$ is the multiplier for the equality constraint $\mathbf{1}^\top \boldsymbol{\pi} = 1$, and $\boldsymbol{\mu} \geq \mathbf{0}$ is the multiplier for the nonnegativity constraint $\boldsymbol{\pi} \geq \mathbf{0}$. The KKT conditions give

$$\pi_j - z_j + \lambda - \mu_j = 0, \quad \mu_j \pi_j = 0, \quad \pi_j \geq 0, \quad \mu_j \geq 0, \quad \forall j \in [n].$$

From these conditions, if $\pi_j > 0$, then $\mu_j = 0$ and thus $\pi_j = z_j - \lambda$. If $\pi_j = 0$, then $\lambda - z_j = \mu_j \geq 0$ such that $z_j - \lambda \leq 0$. Therefore, the optimal solution must take the form $\pi_j^* = \max\{z_j - \lambda^*, 0\}$, $\forall j \in [n]$, for some scalar λ^* . Since we know that the budget constraint is active in this case, λ^* is determined by $\sum_{j=1}^n \max\{z_j - \lambda^*, 0\} = 1$.

Algorithm 3: Reservoir Rounding Scheme

Input: Priors $\{\pi_i\}_{i=1}^h$, budgets $\{K_i\}_{i=1}^h$, datasets $\{\mathcal{D}_i\}_{i=1}^h$

Output: Selected subset $\mathcal{S} \triangleq \bigcup_{i=1}^h \bigcup_{k=1}^{K_i} d_{i,k}$

for $i = 1, 2, \dots, h$ **do**

 Set $K_i^{\min} \leftarrow \min\{\|\pi_i\|_0, K_i\}$;

 Draw $K_i^{\min}$ distinct samples $\{d_{i,1}, \dots, d_{i,K_i^{\min}}\}$ from $\mathcal{D}_i$ via π_i *without replacement*;

if $K_i^{\min} < K_i$ **then**

 Draw $K_i - K_i^{\min}$ additional samples $\{d_{i,K_i^{\min}+1}, \dots, d_{i,K_i}\}$ *uniformly at random* from $\mathcal{D}_i \setminus \{d_{i,1}, \dots, d_{i,K_i^{\min}}\}$;

F Effective Reservoir Rounding

A straightforward rounding strategy is to *independently* sample K_i data candidates from each source $\mathcal{D}_i$ according to the learned prior π_i . However, this operation may select the same data candidate multiple times. Consequently, the number of distinct selected data points from source $\mathcal{D}_i$ can be smaller than the prescribed budget K_i , i.e., $|\bigcup_{k=1}^{K_i} \{d_{i,k}\}| < K_i$. When the induced utility function is monotone, such duplicate selections can reduce the practical utility of the final selected subset, since part of the selection budget is effectively wasted.

To address this issue, we adopt a simple without-replacement sampling scheme, termed Reservoir Rounding. As detailed in Algorithm 3, Reservoir Rounding first computes $K_i^{\min} \triangleq \min\{\|\pi_i\|_0, K_i\}$, which is the maximum number of distinct candidates that can be sampled according to the nonzero support of π_i . It then draws $K_i^{\min}$ distinct data candidates from $\mathcal{D}_i$ according to π_i *without replacement*. If $K_i^{\min} < K_i$, the remaining $K_i - K_i^{\min}$ candidates are *uniformly* sampled from the unselected data points in $\mathcal{D}_i$. In this way, reservoir rounding scheme always returns exactly K_i distinct candidates from each source $\mathcal{D}_i$. In particular, we show that this reservoir rounding scheme not only ensures that the final selected subset contains exactly K_i distinct data candidates from each source $\mathcal{D}_i$ for all $i \in [h]$, but also guarantees that the expected utility of the resulting subset is *lossless*, that is to say,

Theorem 5. Suppose the induced utility function $U(\theta^*(\mathcal{B}))$ is monotone with respect to the selected subset $\mathcal{B}$. For any priors $\pi_i \in \Delta_{n_i}$ and $\|\pi_i\|_1 = 1, \forall i \in [h]$, let $\mathcal{S}$ be the subset returned by Algorithm 3. Then $\mathcal{S}$ satisfies the following two properties: 1) $|\mathcal{S} \cap \mathcal{D}_i| = K_i, \forall i \in [h]$ and 2) $\mathbb{E}[U(\theta^*(\mathcal{S}))] \geq \Psi(\pi_1, \dots, \pi_h)$, where Ψ denotes the probabilistic relaxation of the utility function $U(\theta^*(\mathcal{B}))$.

Remark 3. The condition $\|\pi_i\|_1 = 1$ is imposed only for simplicity. When $\|\pi_i\|_1 < 1$, one can naively normalize the prior as $\tilde{\pi}_i \triangleq \frac{\pi_i}{\|\pi_i\|_1}$. Since the induced utility function U is monotone, increasing the total sampling mass from $\|\pi_i\|_1$ to 1 can increase the value of the corresponding probabilistic relaxation.

Proof. We first verify the exact-budget property. For each source $\mathcal{D}_i$, Algorithm 3 sets $K_i^{\min} \triangleq \min\{\|\pi_i\|_0, K_i\}$. It then samples $K_i^{\min}$ distinct candidates according to π_i without replacement. If $K_i^{\min} < K_i$, the algorithm additionally samples $K_i - K_i^{\min}$ candidates from the remaining unselected data points in $\mathcal{D}_i$. Therefore, all selected candidates from $\mathcal{D}_i$ are distinct, and $|\mathcal{S} \cap \mathcal{D}_i| = K_i, \forall i \in [h]$.

It remains to prove the utility guarantee. At first, we review the definition $\Psi(\pi)$. Specifically, for each source $\mathcal{D}_i$, we independently draw K_i data candidates $\{e_{i,1}, \dots, e_{i,K_i}\}$ from $\mathcal{D}_i \cup \{\emptyset\}$. If we let $\mathcal{S}_{ind} \triangleq \bigcup_{i=1}^h \bigcup_{k=1}^{K_i} e_{i,k}$, we have $\Psi(\pi) = \mathbb{E}_{\mathcal{S}}[U(\theta^*(\mathcal{S}_{ind}))]$. Then, we denote by $\mathcal{A}_i$ the first-stage subset sampled from $\mathcal{D}_i$ without replacement, namely, $\mathcal{A}_i \triangleq \{d_{i,1}, \dots, d_{i,K_i^{\min}}\}$.

Since the second sampling step only adds extra candidates to $\bigcup_{i=1}^h \mathcal{A}_i$, we naturally have that $\mathbb{E}[U(\theta^*(\mathcal{S}))] \geq \mathbb{E}[U(\theta^*(\bigcup_{i=1}^h \mathcal{A}_i))]$, where $\mathcal{S}$ is the output of Algorithm 3. Thus, it suffices to show that $\mathbb{E}[U(\theta^*(\bigcup_{i=1}^h \mathcal{A}_i))] \geq \mathbb{E}_{\mathcal{S}}[U(\theta^*(\mathcal{S}_{ind}))] = \Psi(\pi)$.

Specifically, we will first show that, for any subset $\tilde{\mathcal{S}}$ independent of the randomness from source $\mathcal{D}_i$, the following inequality holds

$$\mathbb{E}_{\mathcal{A}_i} \left[U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \mathcal{A}_i) \right) \right] \geq \mathbb{E} \left[U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \left(\bigcup_{k=1}^{K_i} e_{i,k} \right)) \right) \right], \quad (24)$$

where $\{e_{i,1}, \dots, e_{i,K_i}\}$ are drawn independently according to π_i .

If $\|\pi_i\|_0 < K_i$, then the first-stage without-replacement sampling selects all data candidates with nonzero probability under π_i . Therefore, $\bigcup_{k=1}^{K_i} e_{i,k} \subseteq \mathcal{A}_i$. By monotonicity of U , the above inequality Eq.(24) holds.

It remains to consider the case $\|\pi_i\|_0 \geq K_i$. Let $\mathcal{A}_i = \{p_{N_1}^{(i)}, \dots, p_{N_{K_i}}^{(i)}\}$ where $p_{N_k}^{(i)}$ denotes the data pair sampled without replacement from source $\mathcal{D}_i$. We then prove Eq.(24).

It is worth noting that when conditioned on the first $K_i - 1$ selected indices $N_1, \dots, N_{K_i-1}$, the last without-replacement draw selects an unchosen candidate m with probability $\frac{\pi_{i,m}}{1-P}$, $m \notin \{N_1, \dots, N_{K_i-1}\}$, where $P \triangleq \sum_{r=1}^{K_i-1} \pi_{i,N_r}$.

Hence,

$$\begin{aligned} & \mathbb{E} \left[U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\} \cup \{v_i^{N_{K_i}}\}) \right) \middle| N_1, \dots, N_{K_i-1} \right] \\ &= \sum_{m \notin \{N_1, \dots, N_{K_i-1}\}} \frac{\pi_{i,m}}{1-P} U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\} \cup \{v_i^m\}) \right) \\ &= \sum_{m \notin \{N_1, \dots, N_{K_i-1}\}} \left(\pi_{i,m} + \frac{P \cdot \pi_{i,m}}{1-P} \right) \cdot U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\} \cup \{v_i^m\}) \right) \\ &\geq \sum_{m \notin \{N_1, \dots, N_{K_i-1}\}} \pi_{i,m} \cdot U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\} \cup \{v_i^m\}) \right) \\ &\quad + \frac{P \cdot \sum_{m \notin \{N_1, \dots, N_{K_i-1}\}} \pi_{i,m}}{1-P} \cdot U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\}) \right) \\ &= \sum_{m \notin \{N_1, \dots, N_{K_i-1}\}} \pi_{i,m} \cdot U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\} \cup \{v_i^m\}) \right) + P \cdot U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\}) \right) \\ &= \sum_{m \notin \{N_1, \dots, N_{K_i-1}\}} \pi_{i,m} \cdot U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\} \cup \{v_i^m\}) \right) \\ &\quad + \sum_{m \in \{N_1, \dots, N_{K_i-1}\}} \pi_{i,m} \cdot U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\} \cup \{v_i^m\}) \right) \\ &= \mathbb{E}_{e_{i,K_i}} \left[U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\} \cup \{e_{i,K_i}\}) \right) \right], \end{aligned}$$

where the fourth equality follows from the fact that when $m \in \{N_1, \dots, N_{K_i-1}\}$, we have

$$U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\} \cup \{v_i^m\}) \right) = U \left(\boldsymbol{\theta}^*(\tilde{\mathcal{S}} \cup \{v_i^{N_1}, \dots, v_i^{N_{K_i-1}}\}) \right).$$

This shows that the last without-replacement draw can be replaced by an independent draw without increasing the expected function value. Repeating the same argument for the $(K_i - 1)$ -th draw, the $(K_i - 2)$ -th draw, and so on, we obtain the result of Eq.(24).

Finally, applying the above source-wise comparison Eq.(24) sequentially over $i = h, h-1, \dots, 1$ and using the tower property of conditional expectation, we obtain

$$\mathbb{E} \left[U(\boldsymbol{\theta}^*(\bigcup_{i=1}^h \mathcal{A}_i)) \right] \geq \mathbb{E}_{\mathcal{S}} [U(\boldsymbol{\theta}^*(\mathcal{S}_{ind}))] = \Psi(\boldsymbol{\pi}).$$

Combining this with the fact $\mathbb{E}[U(\boldsymbol{\theta}^*(\mathcal{S}))] \geq \mathbb{E}[U(\boldsymbol{\theta}^*(\bigcup_{i=1}^h \mathcal{A}_i))]$, we have $\mathbb{E}[U(\boldsymbol{\theta}^*(\mathcal{S}))] \geq \mathbb{E}_{\mathcal{S}} [U(\boldsymbol{\theta}^*(\mathcal{S}_{ind}))] = \Psi(\boldsymbol{\pi})$. $\square$

G Data Summarization on Cifar-100 Dataset

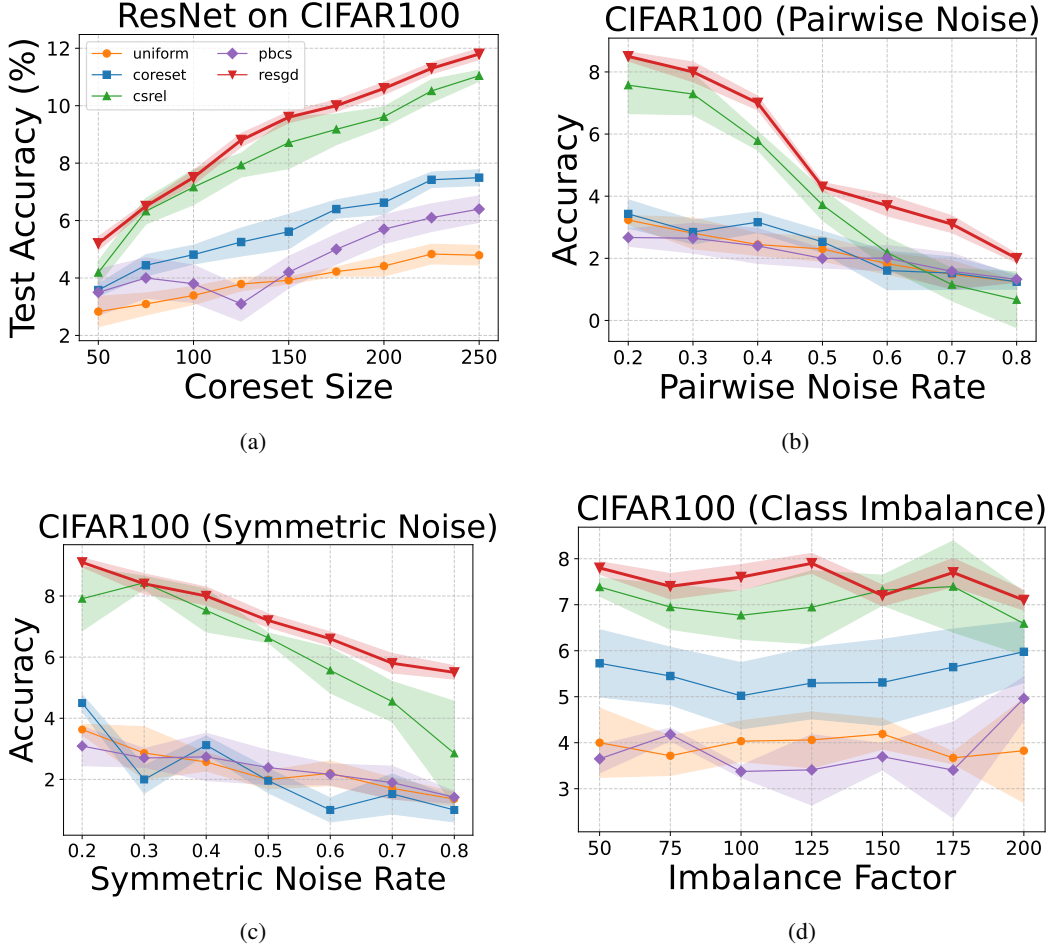


Figure 2: Performance comparison between ReSGD and baseline methods on data summarization. (a) Test accuracy across different coreset sizes for ResNet on CIFAR100. (b) and (c) evaluate the robustness of each method against varying rates of pairwise noise and symmetric noise on CIFAR100, respectively. (d) illustrates the performance under different class imbalance factors on CIFAR100. Shaded areas denote the variance across multiple runs. Overall, the results demonstrate the superior performance and stability of ReSGD under various data corruption and distribution shift scenarios.

H Coreset Selection for Existing Continual Learning Methods

We instantiate our method by replacing the reservoir sampling component in ER [93] with ReSGD and CS-Rel [104]. Following [104], we evaluate all methods on Split CIFAR-100 and Split Tiny-ImageNet under memory budgets of 200 and 500. Both benchmarks are divided into 10 class-incremental tasks, and all methods use a ResNet-18 backbone [51]. We compare against standard rehearsal-based baselines, including ER, A-GEM [17], and FDR [6]. Results are summarized in Table 2. For Split CIFAR-100, the 100 classes are evenly split into 10 tasks with 10 classes per task. For Split Tiny-ImageNet, the 200 classes are evenly split into 10 tasks with 20 classes per task. In both cases, all samples from the current task are used for training. We adopt the same data augmentation pipeline as Mammoth [12], including random crop, random horizontal flip, and normalization. We also use the same ResNet-18 implementation provided by Mammoth for both datasets. For optimization, we use SGD with batch size 32 on both benchmarks. The learning rate is set to 2×10^{-2} for Split CIFAR-100 and 3×10^{-2} for Split Tiny-ImageNet. The cross-entropy loss factor is fixed to 1.0. The number of training epochs is kept consistent with the corresponding baselines.

Table 2: Average accuracy (%) on Split CIFAR-100 and Split Tiny-ImageNet. Results are averaged over 10 incremental tasks with memory buffers of 200 and 500 samples. We use ResNet-18 as the backbone for all experiments. Bold indicates the best performance.

Methods	Split CIFAR-100		Split Tiny ImageNet	
	200	500	200	500
A-GEM	9.40 ± 0.05	9.42 ± 0.08	8.07 ± 0.08	8.06 ± 0.04
ER	14.18 ± 0.45	21.08 ± 0.16	8.49 ± 0.16	9.99 ± 0.29
FDR	15.32 ± 0.73	22.83 ± 0.73	8.70 ± 0.19	10.54 ± 0.21
CSReL-ER	15.35 ± 0.73	22.65 ± 0.81	8.66 ± 0.06	10.44 ± 0.17
ReSGD-ER	15.38 ± 0.25	23.21 ± 0.22	8.87 ± 0.05	10.67 ± 0.12

I Experiment with Adversarial Initialization

In this section, we detail the empirical setup designed to rigorously evaluate the error-correction capabilities and robustness of the ReSGD algorithm. The experiment specifically investigates the algorithm’s capacity to filter corrupted labels and recover a high-quality coreset when subjected to an extreme, adversarial initialization state.

Dataset Partitioning and Asymmetric Label Corruption. The experiment utilizes a subset of the MNIST dataset, partitioned into two disjoint subsets: Holdout Set $\mathcal{D}_{holdout}$ consists of 10,000 pristine samples and serves as the reference distribution; Training Pool $\mathcal{D}_{train}$ consists of 10,000 separate samples, acting as the active selection pool from which the coreset is sampled. To simulate severe, systematic data corruption, we introduce asymmetric label noise. We uniformly sample a subset $\mathcal{C} \subset \mathcal{D}_{train}$ containing 1,000 instances. For each instance $(x_i, y_i) \in \mathcal{C}$, the ground-truth label y_i is deterministically distorted according to the following mapping:

$$y_i^{noisy} = (y_i^{clean} + 2) \pmod{10}$$

The indices of the corrupted subset $\mathcal{C}$ are recorded for downstream evaluation of the selection purity.

Reference Model Pre-training. Prior to the ReSGD sampling loop, a baseline reference model, denoted as $\theta^*(\mathcal{D})$, is trained to convergence exclusively on the clean dataset $\mathcal{D}_{holdout}$. During the subsequent ReSGD iterations, the parameters of $\theta^*(\mathcal{D})$ are strictly frozen. This model acts as an oracle to compute the reference loss $\ell(x_i, y_i; \theta^*(\mathcal{D}))$ and does not undergo further parameter updates.

Adversarial Probability Initialization and ϵ -Exploration. To stress-test the ReSGD framework, we initialize the sampling probability vector $\mathbf{m}^{(1)} \in \mathbb{R}^{10000}$ in a highly adversarial manner. We randomly select a further subset $\mathcal{A} \subset \mathcal{C}$, where $|\mathcal{A}| = 200$. To prevent algorithmic stagnation (where clean samples would never be selected or evaluated if assigned a probability of 0), we introduce an ϵ -exploration constant. The initial probability mass is distributed as follows:

$$m_i^{(1)} = \begin{cases} \frac{0.90}{|\mathcal{A}|}, & \text{if } i \in \mathcal{A} \\ \frac{0.10}{|\mathcal{D}_{train}| - |\mathcal{A}|}, & \text{if } i \notin \mathcal{A} \end{cases}$$

This initialization places 90% of the sampling probability strictly on corrupted samples, forcing the algorithm to recover from a nearly adversarial starting state while maintaining the mathematical viability of gradient computation across the entire pool.

Evaluation Metrics To empirically validate the selection efficacy of the proposed method, we monitor three primary metrics:

- **Distribution Shift:** We plot the trajectory of the total probability mass assigned to the corrupted subset $\mathcal{C}$ over T iterations: $\sum_{i \in \mathcal{C}} m_i^{(t)}$. We hypothesize a rapid decay from the initial 0.90 towards 0.

- **Coreset Purity:** At the final iteration T , we extract the Top- K samples with the highest probability values in $\mathbf{m}^{(T+1)}$. The purity is defined as the empirical fraction of clean samples within this Top- K subset.
- **Downstream Utility:** We train a new model from scratch using the final selected Top- K coreset and report its top-1 classification accuracy on the standard MNIST test set, establishing a comparison against a uniform random sampling baseline.

Ablation Results. The result is illustrated in Figures 3 and 4.

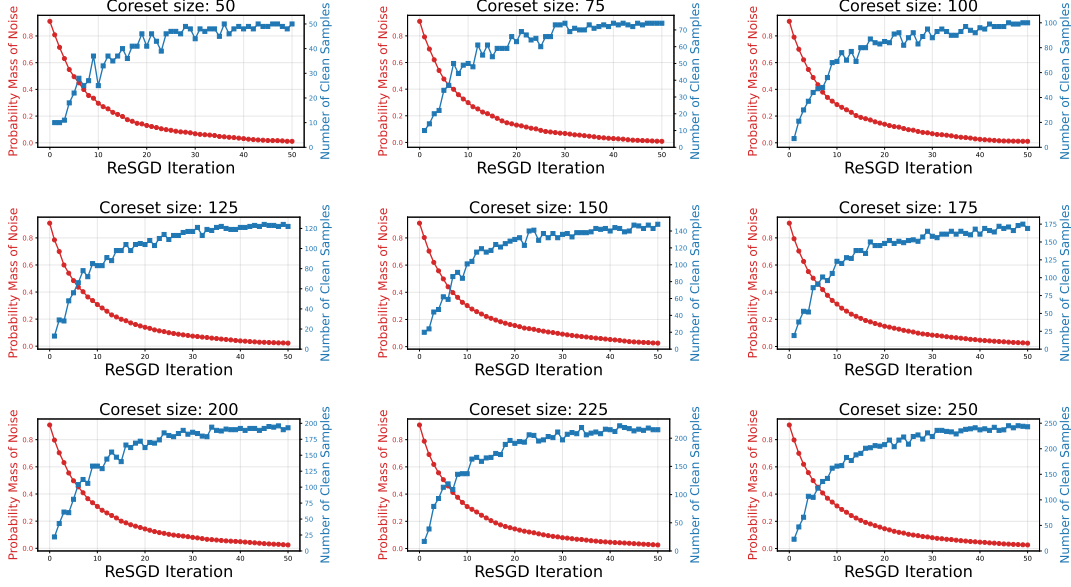


Figure 3: Performance on MNIST under Asymmetric Noise. We track the evolution of the sampling probability mass on corrupted labels (red line) and the coreset purity (blue line) across 50 iterations for various coreset sizes $K \in \{50, \dots, 250\}$. Despite an adversarial start where 90% of the initial mass is concentrated on noisy samples, ReSGD consistently reallocates probability to clean samples, achieving near-perfect purity across all tested coreset scales.

J Data Corruption

To demonstrate the robustness of selection under noisy conditions, we test data summarization performance under the label noise case on the MNIST, CIFAR-10, and CIFAR-100 datasets. For experiments with label noise and class imbalance, the coreset size is set to 200 for MNIST and CIFAR-10. Here is the detailed protocol:

- **Label Noise:** We experiment with two types of label corruption: Symmetric Noise and Pairwise Noise. For symmetric noise, the true label of a sample is uniformly flipped to any of the other $C - 1$ classes with a probability p . For pairwise noise (pairflip), the true label is flipped to an adjacent class with a probability p . We vary the noise ratio p from 0.2 to 0.8 to test the methods under varying degrees of corruption.
- **Class Imbalance:** We construct a long-tailed (imbalanced) dataset by exponentially decaying the number of samples across classes. Following standard practices, we define the imbalance factor as $\mu = \frac{n_{max}}{n_{min}}$, where n_{max} and n_{min} represent the number of training samples in the largest (head) and smallest (tail) classes, respectively. Specifically, the number of samples retained for the i -th class is calculated as $n_i = n_{max} \cdot \sigma^i$, where the decay coefficient is defined as $\sigma = \mu^{-\frac{1}{C-1}}$. In our experiments, we vary the imbalance factor μ from 50 to 200.

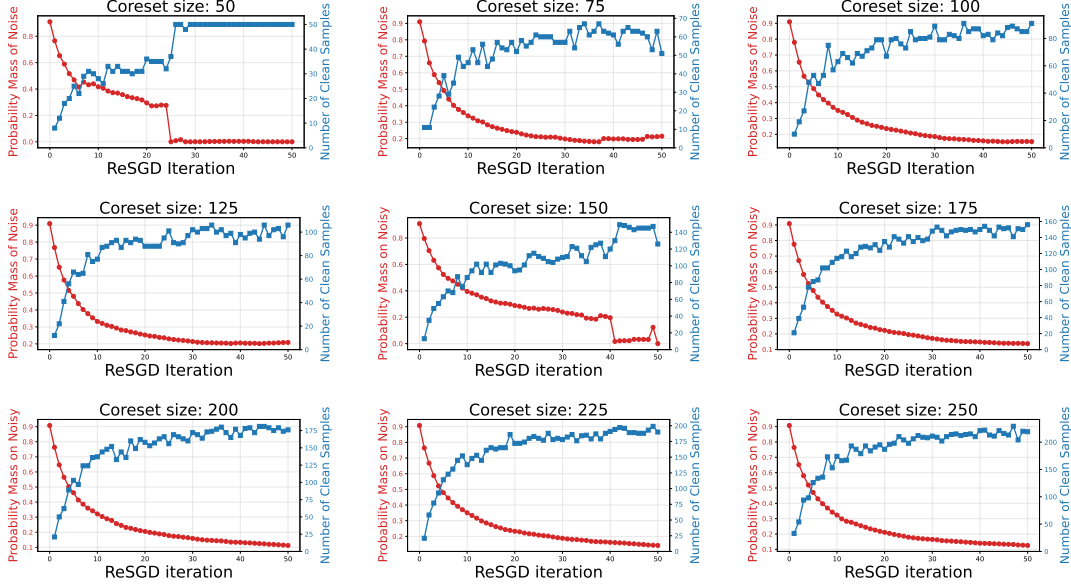


Figure 4: Performance on CIFAR-10 under Asymmetric Noise. Similar to the MNIST experiments, ReSGD demonstrates strong recovery capabilities on the more complex CIFAR-10 dataset. The rapid decay of the noise-associated probability mass (red) and the corresponding climb in the number of clean samples selected (blue) highlight the algorithm’s robustness to systematic label distortion across different target coreset budgets.

We analyze a coreset of 200 samples selected within the noise ratio range of 0.0 to 0.4 under pairwise noise, with the results illustrated in Figure 5.

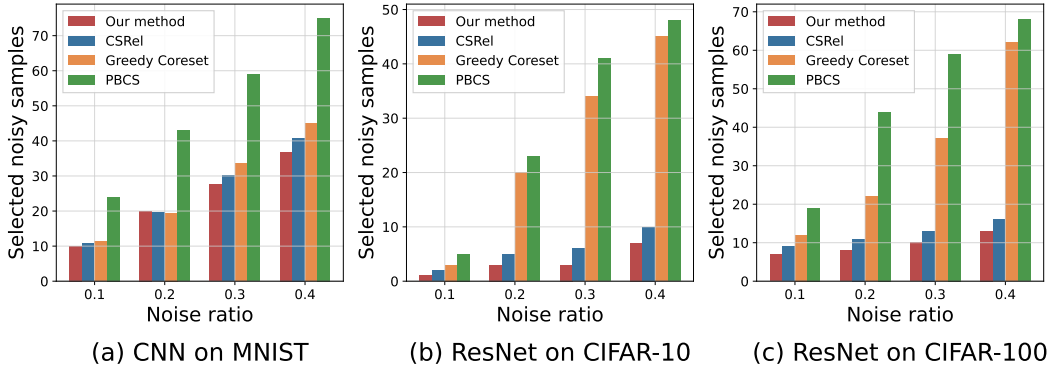


Figure 5: Comparison of the number of selected noisy samples across different noise ratios. We evaluate our method against baseline approaches (CSRel [104], Greedy Coreset [8], and PBCS [116]) on three datasets: (a) MNIST using CNN, (b) CIFAR-10 using ResNet, and (c) CIFAR-100 using ResNet. ReSGD consistently selects the fewest noisy samples across all noise levels, demonstrating its superior robustness and ability to distinguish clean data from pairwise noisy samples.

K Experimental Details

We note that different baselines in the literature are evaluated under different training and evaluation protocols (e.g., number of training epochs, data repetition strategies, and evaluation schedules). A naive unification of these protocols—for example, adopting the evaluation setting of baseline A for all methods—can lead to suboptimal or misleading performance for other baselines such as B and C, whose effectiveness is tied to their original selection regimes. Therefore, to ensure a fair and faithful

comparison, we follow the evaluation protocol used in each original work when reporting results, rather than enforcing a single unified protocol across all methods.

To ensure statistical significance, all experimental results are reported as the mean $\pm$ standard deviation across 5 independent runs with different random seeds. All experiments were conducted on a server equipped with four NVIDIA H20 GPUs. The total computational cost for reproducing the complete set of experiments is approximately 800 GPU hours.

K.1 Data Summarization Experiment Details

We conduct CNN-based experiments on MNIST and ResNet-based experiments on CIFAR-10 and CIFAR-100.

CNN on MNIST. The CNN architecture comprises two convolutional blocks, each consisting of convolution, dropout, max-pooling, and ReLU activation. The numbers of filters in the two convolutional layers are 32 and 64, respectively, both with a kernel size of (5×5) . These are followed by two fully connected layers of sizes 128 and 10, with dropout applied. The dropout probability is set to 0.5. We adopt experimental settings consistent with prior work [8, 116, 104].

- **Greedy Coreset and Uniform Sampling [8]:** The CNN is trained on the data summary using the Adam optimizer with a learning rate of $5e^{-4}$ for 4000 epochs.
- **PBCS [116]:** The projected stochastic gradient descent update for coreset scores is given by

$$\mathbf{s}^{t+1} \leftarrow \mathcal{P}_{\mathcal{C}} \left(\mathbf{s}^t - \eta \mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}^*(\mathbf{m})) \nabla_{\mathbf{s}} \ln p(\mathbf{m} | \mathbf{s}^t) \right). \quad (25)$$

The number of iterations for optimizing the coreset scores $\mathbf{s}$ is $T = 500$, where $t \in \{1, 2, \dots, T\}$, and the learning rate for score updates is $\eta = 5e^{-2}$. The inner optimization $\boldsymbol{\theta}^*(\mathbf{m})$ is performed using Adam with a learning rate of $5e^{-3}$ for 300 epochs. The CNN is then trained on the resulting data summary using Adam with a learning rate of $2e^{-3}$ for 4000 epochs.

- **CSRel [104]:** The reducible loss for coreset selection is defined as

$$G_i = \ell(x_i, y_i; \boldsymbol{\theta}_S) - \ell(x_i, y_i; \boldsymbol{\theta}_D). \quad (26)$$

The holdout model $\boldsymbol{\theta}_D$ is trained using SGD with a learning rate of $1.5e^{-3}$ for 125 epochs. The inner model $\boldsymbol{\theta}_S$ is trained using SGD with a learning rate of $2e^{-2}$ for 100 epochs. Finally, the CNN is trained on the selected data summary using SGD with a learning rate of $2e^{-2}$ for 3000 epochs.

For ReSGD, the holdout model $\boldsymbol{\theta}_D$ is trained using SGD with a learning rate of $1.5e^{-3}$ for 200 epochs. The leave-one-out model $\boldsymbol{\theta}^*(\mathcal{S}_{-(i,1)})$ is trained using SGD with a learning rate of $5e^{-4}$ for 1000 epochs. Finally, the CNN is trained on the selected data summary using SGD with a learning rate of $2e^{-2}$ for 3000 epochs. The number of iterations T is 20 and η is $1e^{-7}$.

ResNet on Cifar-10 and Cifar-100. The experiments of ResNet-18 on Cifar-10 and Cifar-100 follows the setting in [8, 116, 104]. The details of the implementation of baselines are as follows:

- **Greedy Coreset and Uniform Sampling [8]:** The ResNet is trained by Adam optimizer with a learning rate $5e^{-5}$ in 6 epochs. In each epoch, the coreset is repeatedly sampled to 60000 and the total update iterations are 937 (6000/batch size 64).
- **PBCS [116]:** The number of iterations for optimizing the coreset scores $\mathbf{s}$ is $T = 100$, where $t \in \{1, 2, \dots, T\}$, and the learning rate for score updates is $\eta = 3e^{-2}$. The inner optimization $\boldsymbol{\theta}^*(\mathbf{m})$ is performed using Adam with a learning rate of $2e^{-3}$ for 80 epochs. The CNN is then trained on the resulting data summary using Adam with a learning rate of $2e^{-3}$ for 4000 epochs.
- **CSRel [104]:** The holdout model $\boldsymbol{\theta}_D$ is trained using SGD with a learning rate of $3e^{-3}$ for 100 epochs. The inner model $\boldsymbol{\theta}_S$ is trained using SGD with a learning rate of $1e^{-2}$ for 100 epochs. Finally, the ResNet is trained on the selected data summary using Adam with a learning rate of $5e^{-4}$ for 1800 epochs. Initial coreset size is set to 0 and the selection step is 200.

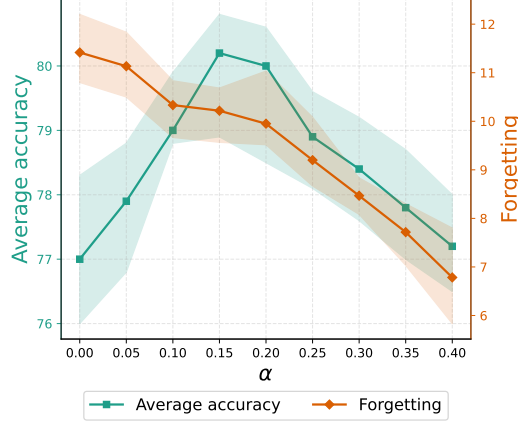


Figure 6: Impact of α on Average Accuracy and Forgetting. Comparison of performance metrics under different loss factors α on the Permuted MNIST dataset. Results show that $\alpha \in [0.15, 0.20]$ provides an optimal trade-off for mitigating forgetting while maintaining overall accuracy.

For ReSGD, the holdout model $\theta_{\mathcal{D}}$ is trained using SGD with a learning rate of $3e^{-3}$ for 100 epochs. The leave-one-out model $\theta^*(\mathcal{S}_{-(i,1)})$ is trained using SGD with a learning rate of $5e^{-4}$ for 100 epochs. Finally, the CNN is trained on the selected data summary using SGD with a learning rate of $5e^{-4}$ for 1800 epochs. The number of iterations T is 20 and η is $1e^{-8}$.

K.2 Continual Learning Experiment Details

We evaluate our data summarization approach for continual learning across four standard benchmarks: Split MNIST [112], Perm MNIST [43], Split CIFAR-10, and Split CIFAR-100.

Datasets and Augmentations. Following the evaluation protocol established by [8, 104], the dataset splits are constructed as follows: Split MNIST and Split CIFAR-10 are both divided into 5 tasks with 2 classes per task. Split CIFAR-100 is divided into 10 tasks, each containing 10 classes. Perm MNIST consists of 10 tasks, where each task corresponds to a randomly permuted version of the original MNIST images. For the training data formulation, we randomly sample 1,000 images per task for Split MNIST, Split CIFAR-10, and Perm MNIST. For Split CIFAR-100, we utilize the full dataset for each task. Regarding data augmentation, we apply standard normalization to both Split MNIST and Perm MNIST. For the more complex CIFAR-10 and CIFAR-100 datasets, our augmentation pipeline includes random cropping, random horizontal flipping, and normalization.

Network Architectures. We employ distinct backbone architectures tailored to the complexity of each dataset:

- Split MNIST: A standard CNN architecture (as detailed in section K.1).
- Split CIFAR-10: A standard ResNet-18 architecture (as detailed in section K.1).
- Split CIFAR-100: A modified ResNet-18 architecture, wherein a Batch Normalization layer is injected into each convolutional block.
- Permuted MNIST: A fully connected network consisting of two hidden layers with 100 units each, ReLU activation functions, and dropout regularization with a probability of 0.2 applied to the hidden layers.

Optimization and Hyperparameters. Our training procedures follow the methodologies of [8, 116, 104]. The optimization process involves a specific loss factor, denoted as α (refer to the setting in [104]). The hyperparameter configurations are specifically tuned for each dataset as follows:

- For Split MNIST, Split CIFAR-10, and Perm MNIST, we utilize the Adam optimizer with a batch size of 256 and a learning rate of 5×10^{-4} . Models for these three datasets are trained for 400 epochs. The loss factor α is set to 100.0 for Split MNIST, 20.0 for Split CIFAR-10, and 0.15 for Perm MNIST.

- For Split CIFAR-100, we employ the SGD optimizer with a smaller batch size of 32 and a higher learning rate of 2×10^{-2} . The model is trained for 100 epochs, and the loss factor α is set to 4.0. This SGD-based setup is consistently applied across all compared methods for the CIFAR-100 benchmark.

The choices of the loss factor Following [104], we adopt the forgetting metric to quantify the performance degradation across subsequent tasks for the experiments in Section 5.2. The forgetting for task j after training on task k is computed as:

$$f_j^k = \max_{l \in \{1, \dots, k-1\}} a_{l,j} - a_{k,j}, \quad \forall j < k, \quad (27)$$

where $a_{k,j}$ denotes the accuracy on task j upon the completion of task k . We evaluate both average accuracy and forgetting on the Permuted MNIST dataset. As illustrated in Figure 6, we analyze these metrics across varying loss factors α . Empirically, the optimal balance is achieved at $\alpha = 0.15$, though $\alpha = 0.2$ also yields robust performance in practice.

L Limitations and Broader Impact

Limitations. In this work, we introduce a new probabilistic relaxation for the multi-source bilevel data selection problem, which only assigns probability mass to feasible data subsets and thus naturally yields a lossless rounding scheme. In addition, we show that this new probabilistic relaxation admits an effective two-point gradient estimator, which can simultaneously avoid the inverse-Hessian computation required by existing 0 – 1 relaxation and the unbounded score term appearing in prior probabilistic relaxation. Built upon this reducible two-point estimator, we further design a novel stochastic gradient algorithm, termed ReSGD, for the multi-source bilevel data selection problem, and establish a rigorous approximation guarantee for the subset returned by ReSGD under weak submodularity. A limitation of the current analysis is that ReSGD follows the standard random-output scheme in stochastic optimization: it needs to sample τ uniformly from $\{0, 1, \dots, T - 1\}$ and returns a subset drawn according to the historical prior $\pi^{(\tau)}$. Although theoretically convenient, in practice, this random-output scheme may be inconvenient, since it requires storing all T historical iterates or recovering the sampled historical iterate after training. Fortunately, recent advances [5, 52, 71] in last-iterate convergence of stochastic gradient methods provide a promising way to remove this random-sampling step, so that one can directly return the subset sampled from the final prior $\pi^{(T)}$ while hopefully preserving a comparable approximation guarantee. We leave this last-iterate variant of ReSGD as an important direction for future work.

Broader Impact. This work mainly focuses on theoretically exploring the multi-source bilevel data selection problem. Therefore, we do not foresee any direct negative societal impact induced by our proposed method.