Collaborative Prediction: To Join or To Disjoin Datasets

Kyung Rok Kim*[1] Yansong Wang[2] Xiaocheng Li[3] Guanting Chen*[1]

[1]University of North Carolina at Chapel Hill
[2]University of Science and Technology of China
[3]Imperial College London

Abstract

With the recent rise of generative Artificial Intelligence (AI), the need of selecting high-quality dataset to improve machine learning models has garnered increasing attention. However, some part of this topic remains underexplored, even for simple prediction models. In this work, we study the problem of developing practical algorithms that select appropriate dataset to minimize population loss of our prediction model with high probability. Broadly speaking, we investigate when datasets from different sources can be effectively merged to enhance the predictive model's performance, and propose a practical algorithm with theoretical guarantees. By leveraging an oracle inequality and data-driven estimators, the algorithm reduces population loss with high probability. Numerical experiments demonstrate its effectiveness in both standard linear regression and broader machine learning applications. Code is available at `https://github.com/kkrokii/collaborative_prediction`.

1 INTRODUCTION

In the big-data and AI era, it is widely believed that more data will always help the downstream performance even if the data could potentially introduce noise or even adversarial effect. However, such approach is applicable to machine learning models that are not susceptible to over-fitting as in Breiman [2001], Chen and Guestrin [2016]. Meanwhile, classical methods such as linear regression and classification still serve as alternatives for prediction tasks, and are still widely used in AI-related pipelines. For instance, linear probing is frequently used for large language models and other machine learning models [Alain and Bengio, 2016, Zhang et al., 2022]. With the rise of generative AI, there has been a shift in the approach to better utilize data. For example, the fine-tuning and alignment stage [Ouyang et al., 2022] increasingly emphasize the need for higher-quality datasets to achieve better performance.

In this work, we investigate methods for selecting higher-quality data to enhance predictive performance. As modern AI architectures become increasingly complex, systematically studying the impact of improved data selection on performance remains a challenging task. Henceforth, we aim to study the data refinement problem for linear regression and classification problems, which are well-studied in the theoretical statistics/machine learning community, while also relevant with current development of AI.

Combining multiple datasets to train machine learning models is a common practice as increasing the number of samples usually leads to an improvement of performance of the model. However, it is not always granted that a model trained on diverse datasets would perform better, compared to separate models fitted on each dataset respectively. As datasets may or may not be similar to each other, such practice gives rise to the question of when they can be safely merged.

Given a collection of datasets, a natural way to deal with the problem would be to combine only the datasets that are relevant. By sharing information across several datasets, a model could benefit from a larger sample size. Meanwhile, the model can enhance its overall performance and provide more accurate forecast on each task if the underlying datasets are related. This procedure of joint training among selectively chosen datasets is referred to as collaborative prediction. In collaborative prediction, the collection of assorted datasets is divided into several partitions and the same model is trained on each partition.

Research on multi-task learning [Evgeniou and Pontil, 2004] and data integration [Merkouris, 2010] also aims to address this problem by utilizing information from other datasets. The majority of the studies on multi-task learning is cen-

*Corresponding to: kkrok@unc.edu, guanting@unc.edu

Table 1: Out-of-Sample Error Reduction of Our Algorithm vs Other Regression Based Multi-task Algorithms [Murugesan et al., 2017]. Notice that these multi-task settings are not designed to specifically reduce the out-of-sample error.

Datasets	RSS	SSF	CFGSF
Multi-task Learning	12%	37%	18%
Our Algorithm	87%	64%	71%

tered around regularization of tasks. Various studies present different objective functions, optimization methods, and convergence analyses. Although the optimization methods converge and have been empirically demonstrated to perform well on numerical experiments, the formulation of the objectives or regularization is less grounded by theory. Table 1 showcases the performance advantage of our algorithm over the regression-based multi-task learning algorithms. We also note that our problem setting differs from federated learning [Kairouz et al., 2021], as we do not consider issues of data heterogeneity or exchangeability constraints. For a discussion of related work, we refer readers to Appendix A.

In this context, we propose an algorithm with provable guarantees for determining whether to combine datasets for predictive models. Our approach leverages data-driven estimations and high-probability concentration bounds. The proposed algorithm is highly versatile, applicable to both simple predictive models, such as linear regression, and more complex deep neural network architectures.

The contributions of this study are threefold:

- First, we investigate the problem of combining multiple datasets for predictive modeling. We characterize the limitations of different models and provide theoretical insights.
- Second, we establish theoretical conditions for determining whether to combine datasets, ensuring a high-probability reduction in population loss, and develop practical algorithms applicable to predictive models.
- Third, we empirically demonstrate the effectiveness of our algorithm across various real-world datasets. Additionally, we highlight its applicability to both classical and modern predictive models, such as neural networks, and show its effectiveness for improving performance.

2 COLLABORATIVE PREDICTION

Let us begin with a motivating example. Denote by $\mathcal{X}$ the space of features and $\mathcal{Y}$ the space of predictors. Suppose we are given two datasets, $D_1 = \{(x_i^{(1)}, y_i^{(1)})\}_{i=1}^{n_1}$ and $D_2 = \{(x_i^{(2)}, y_i^{(2)})\}_{i=1}^{n_2}$, drawn from distributions P_1 and P_2, respectively. Our goal is to train a predictive model $f_\theta(\cdot)$ from the hypothesis class $\mathcal{H} = \{f_\theta \mid f_\theta(\cdot) : \mathcal{X} \to \mathcal{Y}, \theta \in \Theta\}$, which may range from simple linear functions to complex deep neural networks. One approach is to train different models from data set, that is, we get f_{θ_1} after training on dataset D_1, and f_{θ_2} after training on dataset D_2; Another approach is to train the model f_{θ_c} with combined dataset $D_1 \cup D_2$. Given a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, the decision to combine the datasets or use them separately depends on comparing the population loss of the combined model to that of models trained on individual datasets. That is, we need to compare

Table 2: Accuracy of Algorithm 3 and Direct Comparison.

d	Merge?	Our Algorithm	Direct Comparison
0	Yes	82.7%	17.3%
0.1	Yes	79.4%	20.5%
0.3	No	73.1%	80.5%

$$\begin{aligned} &R(f_{\theta_c}, f_{\theta_c}) \\ &:= \mathbb{E}_{(x,y)\sim P_1}[\ell(f_{\theta_c}(x), y)] + \mathbb{E}_{(x,y)\sim P_2}[\ell(f_{\theta_c}(x), y)] \end{aligned}$$

with

$$\begin{aligned} &R(f_{\theta_1}, f_{\theta_2}) \\ &:= \mathbb{E}_{(x,y)\sim P_1}[\ell(f_{\theta_1}(x), y)] + \mathbb{E}_{(x,y)\sim P_2}[\ell(f_{\theta_2}(x), y)]. \end{aligned}$$

In practice, we only have access to the empirical loss. One simple approach is to compare the sample loss defined as

$$\begin{aligned} \hat{R}(f_{\theta_1}, f_{\theta_2}) = &\frac{1}{n_1}\sum_{i=1}^{n_1} \ell(f_{\theta_1}(x_i^{(1)}), y_i^{(1)}) \\ &+ \frac{1}{n_2}\sum_{i=1}^{n_2} \ell(f_{\theta_2}(x_i^{(2)}), y_i^{(2)}) \end{aligned}$$

However, this could fall short in some simple settings. In Table 2, we draw datasets from P_1 and P_2 where $x \sim N(0, I_p)$, $\epsilon \sim N(0,1)$, $y = x^\top \beta + \epsilon$ for some β for each P_1 and P_2, and we control the distance between the two distribution by setting different values of $d = \frac{1}{\sqrt{p}}||\beta_1 - \beta_2||$. The finding is that directly comparing the empirical loss (the column Direct in Table 3) would result in false decisions 80% of the time, especially under the scenario that d is relatively small. However, our proposed approach performs more consistently.

It is also theoretically challenging to provide a tight theoretical guarantee backing the comparison of the empirical loss directly. From generalization bound result, the excess risk is bounded by

$$|\hat{R}(f_{\theta_1}, f_{\theta_2}) - R(f_{\theta_1}, f_{\theta_2})| \le O\left(R_S(\mathcal{H})\right) + O\left(\sqrt{\frac{\log(1/\delta)}{n}}\right),$$

where $R_S(\mathcal{H})$ is the empirical Rademacher complexity. By re-arranging the terms and use the upper bound for

$R(f_{\theta_c}, f_{\theta_c})$ and lower bound for $R(f_{\theta_1}, f_{\theta_2})$, we have $R(f_{\theta_c}, f_{\theta_c}) \leq R(f_{\theta_1}, f_{\theta_2})$ if

$$\hat{R}(f_{\theta_c}, f_{\theta_c}) + O\left(R_S(\mathcal{H})\right) + O\left(\sqrt{\frac{\log(1/\delta)}{n}}\right)$$
$$\leq \hat{R}(f_{\theta_1}, f_{\theta_2}) - O\left(R_S(\mathcal{H})\right) - O\left(\sqrt{\frac{\log(1/\delta)}{n}}\right).$$

The inequality above implies that to estimate the population loss from the empirical loss, we must quantify $R_S(\mathcal{H})$ (or VC dimension), which is challenging for many predictive models. Moreover, the constants (denoted by the big-O notation) in the standard generalization bounds often rely on additional assumptions on the loss function and distributions, such as boundedness or sub-Gaussian tail behavior, imposing further restrictions.

To summarize, the reasons not to compare the empirical loss directly are first, it sometimes performs badly in practice; second, it is non-trivial to give tight theoretical guarantees backing this approach. Therefore, we narrow our focus to linear regression and classification for the following reasons:

- First, these models are relatively simple and well-studied in the statistical learning literature, allowing us to derive insights that may generalize to practical scenarios.
- Second, they serve as fundamental building blocks in modern machine learning pipelines. We anticipate that some of the inherent properties of the proposed algorithm may extend to more complex machine learning models.

Motivated by this, we develop theoretical guarantees for population loss comparison in simpler models and propose algorithms applicable to both fundamental tasks and modern machine learning applications.

2.1 FORMULATION

We now formally introduce the collaborative prediction problem. Let $\{D_k\}_{k=1}^K$ be a collection of datasets, where each dataset consists of pairs $(x^{(k)}, y^{(k)})$ sampled from $\mathcal{X} \times \mathcal{Y}$ according to some probability distribution $\{P_k\}_{k=1}^K$ over the joint space. Let $\mathcal{A}(\cdot)$ denote an algorithm that takes a dataset as input and outputs a predictive model. Specifically, when trained on a dataset D, the algorithm produces a function $\mathcal{A}(D) : \mathcal{X} \to \mathcal{Y}$.

To evaluate the model's performance, we employ a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$. The population loss of the trained model is defined as

$$L(D, P) = \mathbb{E}_{(x,y)\sim P}[\ell(\mathcal{A}(D)(x), y)], \tag{1}$$

where P represents the true underlying distribution. Suppose the algorithm is trained on each dataset D_k and predicts models separately. The loss incurred by fitting individual models is the sum of losses on each dataset $\sum_{k=1}^K L(D_k, P_k)$. If the algorithm is instead trained on all the datasets $D_{all} = \cup_{k=1}^K D_k$, the loss incurred by fitting a unified model is $\sum_{k=1}^K L(D_{all}, P_k)$. Generally speaking, merging all the data would improve the *estimation error* (or variance) as the number of samples on which the algorithm is trained has increased, but it would deteriorate the *approximation error* (or bias) as the same model is applied to all the datasets that are possibly different. The problem of interest is to find a partition $\mathcal{S}$ of the collection of datasets $\{D_k\}_{k=1}^K$ such that the algorithm, when trained on each partition, minimizes the total loss. Formally, a partition S is defined over the index set $\{1, 2, \ldots, K\}$ as a collection of disjoint subsets $\{I_i\}_{i=1}^N$, where each I_i satisfies $I_i \subset \{1, 2, \ldots, K\}$ and $\cup_{i=1}^N I_i = \{1, 2, \ldots, K\}$. We denote the partition cell to which an index k belongs by $S(k)$, meaning $S(k) = i$ if $k \in I_i$. Our objective is to find a partition $S \in \mathcal{S}$ that minimizes the total population loss:

$$\min_{S\in\mathcal{S}} \sum_{k=1}^K L(\cup_{j:S(j)=S(k)} D_j, P_k).$$

2.2 COLLABORATIVE PREDICTION FOR REGRESSION

In this section, we establish theoretical results and data-driven algorithms on the collaborative prediction problem for regression. We first propose theoretical results on combining *two* datasets, then we progressively develop practical algorithms, extending from combinging two datasets to multiple datasets.

2.2.1 Criterion for Combining Datasets for Linear Regression

Suppose a dataset $D_k = \{(x_i^{(k)}, y_i^{(k)})\}_{i=1}^{n_k}$ is generated from a distribution P_k on $\mathcal{X} \times \mathcal{Y}$ for $k = 1, \cdots, K$, where $\mathcal{X} = \mathbb{R}^p$, $\mathcal{Y} = \mathbb{R}$, and $K = 2$ temporarily. We have the following standard assumption.

Assumption 1. *For all $k \leq K$, we assume that $x_i^{(k)}$ follows the marginal distribution $P_{\mathcal{X}}^{(k)}$, and there exists a parameter vector $\beta^{(k)}$ such that, for all $i \leq n_k$, the response variable satisfies $y_i^{(k)} = x_i^{(k)\top}\beta^{(k)} + \epsilon_i^{(k)}$, where $\epsilon_i^{(k)} \sim N(0, \sigma^2)$ with $\sigma^2 > 0$. Additionally, the covariates $x_i^{(k)}$ and the noise $\epsilon_i^{(k)}$ are independent.*

With this assumption, for cleanness of the notation, let us condense D_k more compactly as $(X^{(k)}, y^{(k)}) = ([x_1^{(k)}, \cdots, x_{n_k}^{(k)}]^\top, [y_1^{(k)}, \cdots, y_{n_k}^{(k)}]^\top) \in \mathbb{R}^{n_k\times p} \times \mathbb{R}^{n_k}$, and a merged dataset $D_1 \cup D_2$ as $(X^{(c)}, y^{(c)}) =$

$([X^{(1)\top}, X^{(2)\top}]^\top, [y^{(1)\top}, y^{(2)\top}]^\top)$. With this setting, we fit linear regression models on D_1 and D_2.

The minimizer of the empirical loss is the ordinary least squares (OLS) estimator. The OLS estimator fitted from a single dataset D_k can be denoted by

$$\hat{\beta}^{(k)} = (X^{(k)\top} X^{(k)})^{-1} X^{(k)\top} y^{(k)},$$

and the estimator fitted from the combined dataset $D_1 \cup D_2$ is

$$\hat{\beta}^{(c)} = (X^{(c)\top} X^{(c)})^{-1} X^{(c)\top} y^{(c)}.$$

Now we characterize the population loss. In the setting of linear regression, the loss ℓ is the L^2 loss and the population loss is referred to as the out-of-sample error (OSE). When (x, y) is sampled from P, we can denote the OSE of the regressor $\hat{f}(x) = x^\top \hat{\beta}$ by

$$\text{OSE}(\hat{\beta}, P) = \mathbb{E}_{(x,y)\sim P}\left[\left(y - x^\top \hat{\beta}\right)^2\right].$$

With this setting, the comparison of population loss can be denoted by

$$\sum_{k=1}^{2} \text{OSE}(\hat{\beta}^{(k)}, P_k) > \sum_{k=1}^{2} \text{OSE}(\hat{\beta}^{(c)}, P_k). \tag{2}$$

However, in practice, we only observe the sample loss. Again, it is necessary to establish theoretical results that allow us to infer population loss comparisons from sample loss observations.

To have an alternative form that is easier to compute, we define

$$\begin{aligned} &h : \mathbb{R} \to \mathbb{R}, && h(x) := A_0 x \\ &g : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}, && g(y, z) := ||y - z||_{B_0}^2 \end{aligned}$$

where A_0 and B_0 are constants determined by the underlying distributions. Their precise definitions are in Theorem 1 in Appendix B.1.1. $|| \cdot ||_{B_0}$ is the Mahalanobis norm so that $||v||_{B_0} = \sqrt{v^\top B_0 v}$. The following theorem establishes an equivalent condition for equation (2), and the proof is deferred to Appendix B.1.1.

Theorem 1. *Equation* (2) *holds if and only if* $h(\sigma^2) > g(\beta^{(1)}, \beta^{(2)})$.

Theorem 1 agrees with what one would have expected regarding the true parameters $\beta^{(1)}$ and $\beta^{(2)}$. If the two models were close so that $\|\beta^{(1)} - \beta^{(2)}\|$ for some norm $|| \cdot ||$ is small, then we can have similar condition for $||\beta^{(1)} - \beta^{(2)}||_{B_0}$ by norm equivalence. The condition in the theorem indicates that merging the datasets indeed reduces the error.

2.2.2 Provable Algorithm for Combining Datasets

The true parameters $\beta^{(k)}$ are unknown in practice, and we need to develop data-driven algorithms based on Theorem 1. By concentration and anti-concentration inequalities, with high probability we can control the deviation of the estimation of σ^2, $\beta^{(1)}$, and $\beta^{(2)}$ from their true value.

More specifically, the estimators can be bounded in the following way with a confidence level δ. They involve some distributional constants $\tilde{A}_1(\delta)$ and $\tilde{A}_2(\delta)$ which depend on δ, and $D(\cdot)$, a function of the input data, all defined in detail in Appendix B.1.2. Next, we define

$$\phi_\delta(u, v, w) = \tilde{A}_1(\delta) w + \tilde{A}_2(\delta) \|D(u - v)\|^2$$

and

$$\psi_\delta(u, v, w) = \left\{\sqrt{g(u, v)} + T_\delta(w, \|D(u - v)\|)\right\}^2$$

where $T_\delta(x, y)$ is at the scale of $O(max(x, y))$, and also depends on the confidence level δ. Under Assumption 1, the OLS estimators $\hat{\beta}^{(1)}$ and $\hat{\beta}^{(2)}$ are normally distributed. Therefore, if the parameters $\beta^{(k)}$ are replaced by their estimators $\hat{\beta}^{(k)}$ in Theorem 1, then $g(\hat{\beta}^{(1)}, \hat{\beta}^{(2)})$ would be close to its true value $g(\beta^{(1)}, \beta^{(2)})$. ψ_δ acts as a bound of $g(\hat{\beta}^{(1)}, \hat{\beta}^{(2)})$ with a margin depending on the properties of the underlying distribution and δ as defined above. Similarly, the sample variance of the noise on the combined dataset $\hat{\sigma}^{(c)2}$, having the form $\hat{\sigma}^{(c)2} = \|y^{(c)} - X^{(c)}\hat{\beta}^{(c)}\|^2/(n_1 + n_2 - p)$, would lie near its true value σ^2. Hence ϕ_δ, as a linear function of w, behaves as a surrogate of the linear function h. The following lemma formalizes this result, and we leave the proof in Appendix B.

Lemma 1. *Under Assumption 1, with probability at least* $1 - 5\delta$*, we have equation* (2) *holds if*

$$\phi_\delta(\hat{\beta}^{(1)}, \hat{\beta}^{(2)}, \hat{\sigma}^{(c)2}) \;\geq\; \psi_\delta(\hat{\beta}^{(1)}, \hat{\beta}^{(2)}, \hat{\sigma}^{(c)2}). \tag{3}$$

Lemma 1 provides a computationally feasible approach to verify (2). It utilizes the computationally feasible terms $\phi_\delta(\hat{\beta}^{(1)}, \hat{\beta}^{(2)}, \hat{\sigma}^{(c)2})$ and $\psi_\delta(\hat{\beta}^{(1)}, \hat{\beta}^{(2)}, \hat{\sigma}^{(c)2})$ as proxies of the terms $h(\sigma^2)$ and $g(\beta^{(1)}, \beta^{(2)})$ in the oracle inequality, which are computationally infeasible. This result implies a directly implementable algorithm, detailed in Algorithm 1.

Proposition 1 guarantees that with high probability, the decision made by Algorithm 1 is valid.

Proposition 1. *Under Assumption 1, with probability at least* $1 - 5\delta$*, the output of Algorithm 1 is a decision rule with minimum out-of-sample error being*

$$\min\left\{\sum_{k=1}^{2} OSE(\hat{\beta}^{(k)}, P_k), \sum_{k=1}^{2} OSE(\hat{\beta}^{(c)}, P_k)\right\}.$$

Algorithm 1 High Probability Criterion to Combine Two Datasets

```
Input: Datasets $\{D_1, D_2\}$, confidence level $\delta$
Output: A decision **Merge** whether to merge the datasets or not
Fit linear models $\hat{\beta}^{(1)}$, $\hat{\beta}^{(2)}$, and $\hat{\beta}^{(c)}$ on $D_1$, $D_2$, and $D_1 \cup D_2$, respectively
Compute $\left(\phi_\delta(\hat{\beta}^{(1)}, \hat{\beta}^{(2)}, \hat{\sigma}^{(c)2}), \psi_\delta(\hat{\beta}^{(1)}, \hat{\beta}^{(2)}, \hat{\sigma}^{(c)2})\right)$
Return **Merge** $= \mathbb{1}_{\{\phi_\delta > \psi_\delta\}}$
```

Algorithm 1's high probability guarantee comes at the price of limited applicability. Computing the functions ϕ_δ and ψ_δ requires some knowledge of the marginal distribution of x. Moreover, even when ϕ_δ and ψ_δ can theoretically be computed under distributional assumptions, it will still pose strong limitation because real data is very heterogeneously distributed. This motivate us to develop Algorithm 2, a computationally feasible algorithm.

2.2.3 Approximation Algorithm

We develop Algorithm 2, which takes Algorithm 1 as a subroutine and try to approximate Algorithm 1's decision boundary, and is computationally feasible. We apply Algorithm 1 multiple times, and with a sufficient number of repetitions, Theorem 4 in Appendix B.1.4 justifies the consistency of the approximation.

Because Algorithm 2 is an approximation algorithm, there is no high probability guarantee. Therefore, Algorithm 2 aim to estimate and maximize the *success rate* (SR), meaning the output is correct compared to the estimated ground truth. From (3), for each δ the condition $\phi_\delta > \psi_\delta$ implies that our algorithm suggests a merge, and the success of the merge could be determined by whether OSE_dif $:= \sum_{k=1}^{2} \mathrm{OSE}(\hat{\beta}^{(k)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) - \sum_{k=1}^{2} \mathrm{OSE}(\hat{\beta}^{(c)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) > 0$. Henceforth, the success rate (SR) is formally defined as

$$\mathrm{SR} = \mathbb{P}\left((\phi_\delta - \psi_\delta) \times \text{OSE\_dif} > 0\right). \tag{4}$$

As we do not know the exact OSE_dif due to the lack of information of the distribution, we approximate it based on given data. More specifically, suppose our available dataset D_k is generated from a distribution P_k for $k = 1, 2$. A subset D_k^{train} of D_k is reserved for training and the rest $D_k^{out} = D_k - D_k^{train}$ is used to generate out-of-samples $\{(\tilde{x}_i^{(k)}, \tilde{y}_i^{(k)})\}_{i=1}^{\tilde{n}_k}$ for some out-of-sample size $\tilde{n}_k$, independent from the training data. Under these settings, OSE_dif is estimated by $\widehat{\text{OSE}}\text{\_dif} := \sum_{k=1}^{2} \widehat{\mathrm{OSE}}\left(\hat{\beta}^{(k)}, D_k^{out}\right) - \sum_{k=1}^{2} \widehat{\mathrm{OSE}}\left(\hat{\beta}^{(c)}, D_k^{out}\right)$ where $\widehat{\mathrm{OSE}}$ approximates the true out-of-sample error of an estimator $\hat{\beta}$ by

$$\widehat{\mathrm{OSE}}\left(\hat{\beta}, D_k^{out}\right) = \frac{1}{\tilde{n}_k} \sum_{i=1}^{\tilde{n}_k} \left(\tilde{y}^{(k)} - \tilde{x}^{(k)\top}\hat{\beta}\right)^2.$$

Finally, we repeatedly apply the algorithm several times and estimate SR by its average success rate $\widehat{\mathrm{SR}}$.

We then present Algorithm 2 with SR as the optimization target. Algorithm 2 treats δ as a hyperparameter, whose optimal value can be finetuned by maximizing the estimated success rate $\widehat{\mathrm{SR}}$. We follow Arlot and Bach [2009] and employ grid search to find the optimal value. As Proposition 1 holds in probability, we evaluate the SR by repeating Algorithm 1 several times. The full procedure is described in detail in Algorithm 2.

Algorithm 2 Merging Two Datasets with Optimal Confidence Level

```
    Input: Datasets $\{D_1, D_2\}$, hyperparameter range $[\alpha_{min}, \alpha_{max}]$, grid search window size $\eta$, accuracy threshold $\lambda$
    Output: Decision $\mathbb{1}_{merge}$ whether to merge the datasets or not, proxy accuracy $proxy\_acc$
    **Part1: Tuning $\alpha$**
 1: **for** $\alpha = \alpha_{min}, \alpha_{min} + \eta, \cdots, \alpha_{max}$ **do**
 2:     Initialize $correct_\alpha$ = 0
 3:     **for** $j = 1, 2, \cdots, max\_iterations$ **do**
 4:         Sample $(X^{(k)}, y^{(k)})$ from $D_k$ $(k = 1, 2)$
 5:         Run Algorithm 1 on $\{(X^{(k)}, y^{(k)})\}_{k=1}^{2}$ with $\log\frac{1}{\delta}$ replaced by $\alpha$ to obtain $(\phi_\alpha, \psi_\alpha)$
 6:         Generate out-of-samples $(\tilde{X}^{(k)}, \tilde{y}^{(k)})$ from $D_k$ for $k = 1, 2$
 7:         Compute $\widehat{\text{OSE}}$_dif
 8:         $correct_\alpha \leftarrow correct_\alpha + \mathbb{1}_{(\phi_\alpha - \psi_\alpha) \times OSE\_\text{dif} > 0}$
 9:     **end for**
10: **end for**
11: Choose $\alpha_{opt} = \text{argmax}_\alpha correct_\alpha$
    **Part2: Analysis of the data**
12: Repeat 2-9 with $\alpha = \alpha_{opt}$
13: Return $proxy\_acc = correct_{\alpha_{opt}} / max\_iterations$ and **Merge** $= \mathbb{1}_{proxy\_acc > \lambda}$
```

Algorithm 2 provides a self-contained process of making a decision in practice. It uses only a portion of the given dataset to train models. The rest are used as out-of-samples to compute $\widehat{OSE}$. In Algorithm 2, Algorithm 1 is applied with different values of α, and the optimal value is chosen based on its performance. After α is tuned, the algorithm can be employed to make the actual decision. Due to the page limit, we put Theorem 4 that justifies the consistency of the approximation, in Appendix B.1.4.

2.2.4 Extension to Multiple Datasets

Algorithm 2 is based on Proposition 1 which provides provable guarantee for a pairwise decision. To generalize to the cases where multiple datasets are given, Algorithm 2 is applied iteratively to choose which datasets to merge. Here we compare multiple datasets and form clusters by adopting a greedy algorithm. Specifically, given a dataset that has not been assigned any cluster yet, we compare it with other datasets that also have not been clustered. We choose the one with the largest boost in performance when merged with the given dataset, and decide to combine the two if the elevated performance exceeds a certain threshold. The comparison is repeated until no other dataset can be merged to the current cluster. Details of the algorithm are presented in Algorithm 3, with complexity analysis in Appendix C.4.

Algorithm 3 Merging Multiple Datasets with Greedy Algorithm

```
   Input: A collection of datasets $\{D_k\}_{k=1}^K$, accuracy threshold $\lambda$
   Output: Cluster of each dataset $\{c_k\}_{k=1}^K$
1: while $\exists D_k$ without a cluster do
2:     Assign a new cluster $c_k$ to $D_k$
3:     For every $D_j$ without a cluster, choose $j$ that maximizes $proxy\_acc_{(k,j)}$ by Algorithm 2
4:     while $proxy\_acc_{(k,j)} > \lambda$ do
5:         Assign $c_j$ the same cluster as $c_k$, and merge $D_k$ and $D_j$
6:         Repeat 3
7:     end while
8: end while
9: Return $\{c_k\}_{k=1}^K$
```

We note that other clustering methods may be exploited and could possibly enhance the performance. The greedy algorithm for Algorithm 3 is chosen merely for simplicity. The main point of the algorithms is that information is shared across datasets with theoretical guarantees.

2.3 COLLABORATIVE PREDICTION FOR CLASSIFICATION

Similar arguments can be made on classification tasks as in regression. We employ logistic regression model to make decisions and assess its population error with cross entropy loss, with details in Appendix B.2. Broadly speaking, the decision is built on the same criterion as before, equation (2), with the regression error replaced by classification error. Once again, we want to represent the population loss in terms of β, the linear parameters of the covariates that generates the distribution. Under moderate assumptions, it can be proved that combining the datasets gives smaller population error bound whenever

$$\Psi(\beta^{(1)}, \beta^{(2)}) \leq \Phi(\beta^{(1)}) + \Phi(\beta^{(2)}),$$

where the precise definition of Ψ and Φ can be found in Appendix B.2.1 and B.2.2, and the statement is in Appendix B.2.3, respectively. The major difference is that Ψ contains $O\left(\left(\frac{1}{\sqrt{n_1}} + \frac{1}{\sqrt{n_2}}\right) \|\beta^{(1)} - \beta^{(2)}\|\right)$ term while Φ contains $O\left(\frac{\beta}{\sqrt{n}}\right)$. Hence the error bound on the combined dataset decreases if the distance between the true parameters $\|\beta^{(1)} - \beta^{(2)}\|$ is small, which aligns with our previous conclusion on regression. Due to the page limit, we refer readers to Appendix B.2 for precise statements and details.

3 NUMERICAL EXPERIMENTS

In this section, we empirically show the effectiveness and broad applicability of our algorithm. The structure of the section is as follows.

- In Section 3.1, we conduct experiments in synthetic environments. The results demonstrate strong performance, aligning well with our theoretical predictions.
- In Section 3.2, we evaluate our algorithm on real datasets, casting them as regression problems. It exhibits strong performance, effectively handling datasets with heterogeneous distributions.
- In Section 3.3, we extend our algorithm to neural networks. The results not only show its robustness and strong performance across different settings but also highlight its applicability to different predictive models.

Experimental details, such as the structure of neural networks or the choice of the hyperparameters, can be found in Appendix C.1 and C.2.

3.1 SYNTHETIC ENVIRONMENT

We first test our example in the synthetic environment, where we know (or can sample) the ground truth such that whether datasets should be combined can be precisely determined. Also, the extent to which the datasets are heterogeneous could be controlled by regulating the difference between the underlying distributions. By experimenting within different environment setups, we can have a comprehensive idea how and when our algorithm will be effective.

We first consider the case where two datasets are available, and further deal with general settings on subsequent sections. Each of the dataset is generated by a linear model $y_j^{(k)} = x_j^{\top(k)} \beta^{(k)} + \epsilon_j^{(k)}$ $(k = 1, 2,\ j = 1, \cdots, n_k,$ $\beta^{(k)} \in \mathbb{R}^p)$ with randomly generated covariates $x^{(k)} \sim$

Table 3: Accuracy of Algorithm 2 on the Synthetic Data: $n_1 = n_2 = 50, \mu_1 = \mu_2 = 0, \Sigma_1 = \Sigma_2 = I$

p	d	**Merge?**	**Algorithm 2**	**Direct Comparison**
10	0	Yes	82.7%	17.3%
10	0.1	Yes	79.4%	20.5%
10	0.3	No	73.1%	80.5%
20	0	Yes	87.4%	12.6%
20	0.1	Yes	80.4%	19.4%
20	0.3	No	70.1%	92.5%

Table 4: Accuracy of Algorithm 2 on the Synthetic Data: $n_1 = n_2 = 50, \mu_1 = 0, \mu_2 = \mathbf{1}_p, \Sigma_1 = \Sigma_2 = I$

p	d	**Merge?**	**Algorithm 2**	**Direct Comparison**
10	0	Yes	80.6%	19.4%
10	0.1	Yes	73.6%	22.6%
10	0.3	No	98.0%	98.2%
20	0	Yes	84.7%	15.3%
20	0.1	Yes	66.6%	32.7%
20	0.3	No	99.1%	99.9%

$N(\mu^{(k)}, \Sigma^{(k)})$ and Gaussian noise $\epsilon \sim N(0, \sigma^2 I)$. We select different combinations of p, $\mu^{(k)}$, and $\Sigma^{(k)}$, and compare the sum of the out-of-sample errors of individual models $\sum_{k=1}^{2} OSE(\hat{\beta}^{(k)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}})$ with the out-of-sample error of combined model $\sum_{k=1}^{2} OSE(\hat{\beta}^{(c)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}})$.

Because not much related algorithms are available, we compare Algorithm 2 with the algorithm called Direct Comparison, which is a naive algorithm that directly compares the sample loss (described in the beginning of Section 2). In the synthetic environment, we know the ground truth of the decision whether or not to combine the datasets. The accuracy metric is computed by comparing the ground truth with the outputs of algorithms for evaluation.

We show results for two settings. In both settings the datasets are sampled from different distributions. In the first one (in Table 3), we gradually shift $\beta^{(2)}$ by increasing d, where $\beta^{(2)} - \beta^{(1)} = d\mathbf{1}_p$ and $\mathbf{1}_p \in \mathbb{R}^p$ is a unit vector. In the second environment (in Table 4), we make additional change so that the covariates x is sampled from a distribution with shifted mean ($\mu_1 = 0$ and $\mu_2 = \mathbf{1_p}$). The performance of Algorithm 2 and Direct Comparison is summarized in Table 3 and Table 4. The accuracy metric is computed by comparing the ground truth with their outputs.

Algorithm 2 successfully reduces the out-of-sample error by making adaptive decisions on a wide variety of circumstances. Intuitively, combining the datasets for small values of d would aid the performance of a model, as the underlying distributions are similar, while the datasets should be deemed as distinct and sharing information will not improve the model if d is large. The ground truth decision for combining the datasets changes mostly around $c \in (0.1, 0.3)$. Algorithm 2 adeptly sorts out the situation. For $c \leq 0.1$, combining the two datasets indeed lowers the out-of-sample error for the most of the cases, and the algorithm also suggest to merge the datasets. For $c \geq 0.3$, information of two distributions becomes irrelevant according to the out-of-sample errors, and Algorithm 2 also captures this fact.

3.2 APPLICATION ON REAL-WORLD DATA

We now apply the proposed algorithm to real-life datasets. We demonstrate results on four datasets, which are selected from frequently cited sources in machine learning community such as the UC Irvine repository. The datasets are: 1) Demand Forecast for Optimized Inventory Planning (DFOIP) [Aguilar, 2023], which contains records of online transactions on different items, 2) Walmart Data Analysis and Forcasting (WDAF) [Sahu, 2023], which is consisted of weekly sales and factors that might impact customers, 3) Boom Bikes (BB) [Mishra, 2021], data on the number of daily bike rentals with weather information, and 4) Productivity Prediction of Garment Employees (PPGE) [Siri, 2021], data on productivity of workers in a company.

We manually divide the datasets into separate parts based on certain features. The datasets now can be viewed as sampled from different conditional distributions (conditioned on the feature), and it becomes reasonable to believe that a single model can be consistently applied to these parts. Namely, DFOIP is partitioned for each item, ten most popular of which are used for the experiment. WDAF is divided into holiday and non-holiday weeks, BB is split by weather conditions, and PPGE is divided according to day.

We find that carefully choosing which datasets to combine can greatly improve model performance. Merging all DFOIP datasets, for instance, increases data size but may reduce accuracy, as customer behavior varies across products. Separate models offer tailored predictions, but may struggle with new items due to limited data. In this case, Algorithm 3 provides a solution by merging datasets of similar products.

The performance of Algorithm 3 are illustrated in Table 5. Note that on real data, the true out-of-sample errors are inaccessible as parameters of underlying distributions are unknown. Hence out-of-sample errors are estimated via bootstrapped samples. As Table 5 exhibits, Algorithm 3 reduces out-of-sample error more than 17.48% by congregating pertinent datasets, compared to out-of-sample error of individual models trained independently on each dataset. The results suggest that even with simple clustering, our algorithm greatly reduces the error.

Table 5: Performance of Algorithm 3 Reducing $\widehat{\text{OSE}}$

	Individual	**Algorithm 3**	**Reduction**
DFOIP	4.14×10^{8}	1.25×10^{6}	99.7%
WDAF	8.98×10^{11}	7.41×10^{11}	17.5%
BB	2.057	1.347	34.5%
PPGE	5.57×10^{-2}	3.95×10^{-2}	29.1%

3.3 INTEGRATION ON NEURAL NETWORKS

We further integrate our algorithm to neural networks. Neural networks are often pre-trained on massive amount of data and then fine-tuned for specific tasks by modifying the final layer. It is well-known that the representations before the final layer contain useful information of the input. Treating these representations as input, our algorithm can effectively distinguish between similar and dissimilar representations and improve the predictive performance. See Appendix C.1 for the details of applying our algorithms in the representation space of neural networks.

Five datasets are analyzed in this section: 1) Rossmann Store Sales (RSS) [FlorianKnauer and Cukierski, 2015], which is consisted of sales of different stores along with additional information on each day, 2) Store Sales Forcasting (SSF) [Cukierski, 2014], weekly sales data of Walmart stores, 3) Corporación Favorita Grocery Sales Forecasting (CFGSF) [Favorita et al., 2017], another retail sales data in Ecuador, 4) Seoul Bike Sharing Demand (SBSD) [Veerappampalayam et al., 2020], weekly bike demand data, and 5) Metro Interstate Traffic Volume (MITV) [Hogue, 2019], data of traffic volume in Minneapolis. Retail datasets are divided by store types. SSF and CFGSF is further split into department and item types. The last two datasets are divided based on season and weather.

We use two simple networks: multi-layer perceptron with one hidden layer (MLP1) and two hidden layers (MLP2). While more sophisticated and advanced models could definitely be used to extract representation, and those models would indeed return finer representation of data, our algorithm would be further highlighted if it performs well even on these simple models. After training the models, we take the penultimate layer's representations as new datasets. We then apply Algorithm 3 and check whether out-of-sample error is reduced by the algorithm.

The outcome of Algorithm 3 is summarized in Table 6 along with out-of-sample errors estimated by bootstrapping as before. The errors are measured on representation of the models rather than the original datasets. As can be confirmed from Table 6, the error decreases 64.22% in minimum. The results suggest that our algorithm is good at distinguishing and combining relevant datasets, even on these simple neural network models, suggesting the effectiveness of our algorithm for more complex models as well.

Table 6: Algorithm 3 on Representation with $\widehat{\text{OSE}}$

MLP1	Individual	Alg. 3	Reduction
RSS	2.197×10^{3}	2.326×10^{2}	89.41%
SSF	1.381×10^{4}	4.310×10^{3}	68.79%
CFGSF	1.564×10^{1}	4.308	72.46%
SBSD	1.325×10^{1}	4.830	63.55%
MITV	2.111×10^{3}	1.388×10^{3}	72.46%
MLP2	**Individual**	**Alg. 3**	**Reduction**
RSS	1.327×10^{3}	2.760×10^{2}	79.20%
SSF	3.277×10^{6}	3.816×10^{4}	98.83%
CFGSF	1.320×10^{2}	4.723×10^{1}	64.22%
SBSD	1.146×10^{1}	4.110	63.55%
MITV	1.699×10^{3}	2.028×10^{2}	88.06%

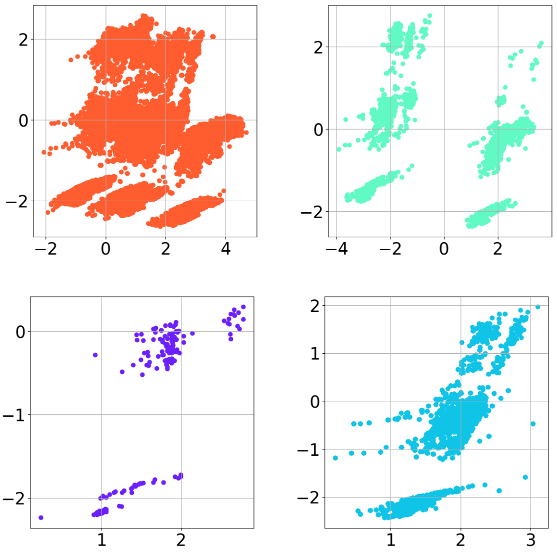

Figure 1: Representation of MLP2 on CFGSF

The effectiveness of clustering the datasets can also be confirmed by visualizing the representation. Figure 1 demonstrates MLP2's representation projected onto $\mathbb{R}^2$ by principal component analysis. We present four collections of datasets from CFGSF, with each collection containing multiple datasets that are classified as the same cluster by Algorithm 3. As illustrated, each cluster includes multiple groups of representation from different datasets. Figure 1 shows that our algorithm successfully reduces error even when the underlying distribution of the input data is heterogeneous and does not perfectly follow Gaussian distribution.

4 CONCLUSION

We present a data-driven algorithm for combining multiple datasets to share information across similar tasks with theoretical guarantee. We empirically demonstrate that our proposed algorithm smartly combine similar datasets and reduces population loss on real data. Additionally, we showed that it can be successfully integrated into neural networks. Future works include exploring other clustering strategies with provable guarantees to improve dataset selection. It is also interesting to extend this framework to other machine learning models.

References

Oscar Aguilar. Demand Forecast for Optimized Inventory Planning, 2023. URL www.kaggle.com/datasets/oscarm524/demand-forecast-for-optimized-inventory-planning.

Guillaume Alain and Yoshua Bengio. Understanding Intermediate Layers Using Linear Classifier Probes. *arXiv:1610.01644*, 2016.

Yonatan Amit, Michael Fink, Nathan Srebro, and Shimon Ullman. Uncovering Shared Structures in Multiclass Classification. In *Proceedings of the 24th international conference on Machine learning*, 2007.

Rie Kubota Ando and Tong Zhang. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine Learning Research*, 6: 1817–1853, 2005.

Sylvain Arlot and Francis Bach. Data-driven Calibration of Linear Estimators with Minimal Penalties. *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, page 46–54, 2009.

Ben Athiwaratkun and Keegan Kang. Feature Representation in Convolutional Neural Networks. *arxiv:1507.02313*, 2015.

Bart Bakker and Tom Heskes. Task Clustering and Gating for Bayesian Multitask Learning. *The Journal of Machine Learning Research*, 4:83–99, 2003.

Peter Bartlett and Shahar Mendelson. Rademacher and Gaussian Complexities: Risk Bounds and Structural Results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Aviad Barzilai and Koby Crammer. Convex Multi-Task Learning by Clustering. *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 38, 2015.

Avrim Blum, Nika Haghtalab, Ariel D. Procaccia, and Mingda Qiao. Collaborative PAC Learning. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 30:2389–2398, 2017.

Stephane Boucheron, Gabor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford Academic, 2013.

Leo Breiman. Random Forests. *Machine Learning*, 45(1): 5–32, 2001.

Tianqi Chen and Carlos Guestrin. Xgboost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794, 2016.

Will Cukierski. Walmart Recruiting - Store Sales Forecasting, 2014. URL https://www.kaggle.com/c/walmart-recruiting-store-sales-forecasting.

Sanjoy Dasgupta and Anupam Gupta. An Elementary Proof of a Theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22:60–65, 2002.

Theodoros Evgeniou and Massimiliano Pontil. Regularized Multi–Task Learning. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117, 2004.

Dongyang Fan, Celestine Mendler-Dünner, and Martin Jaggi. Collaborative Learning via Prediction Consensus. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023.

Corporación Favorita, inversion, Julia Elliott, and Mark McDonald. Corporación Favorita Grocery Sales Forecasting, 2017. URL https://kaggle.com/competitions/favorita-grocery-sales-forecasting.

FlorianKnauer and Will Cukierski. Rossmann Store Sales, 2015. URL https://kaggle.com/competitions/rossmann-store-sales.

Richard Gill. Non- and Semi-Parametric Maximum Likelihood Estimators and the von Mises Method. *Scandinavian Journal of Statistics*, 16:97–128, 1989.

D. L. Hanson and F. T. Wright. A Bound on Tail Probabilities for Quadratic Forms in Independent Random Variables. *The Annals of Mathematical Statistics*, 42(3): 1079–1083, 1971.

John Hogue. Metro Interstate Traffic Volume. UCI Machine Learning Repository, 2019. DOI: https://doi.org/10.24432/C5X60B.

Daniel Hsu, Sham Kakade, and Tong Zhang. A Tail Inequality for Quadratic Forms of Subgaussian Random Vectors. *Electronic Communications in Probability*, 17, 2011.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning. *Foundations and Trends in Machine Learning*, 14(1–2), 2021.

A. Kolmogorov. Sur les Propriétés des Fonctions de Concentrations de M. P. Lévy. *Annales de l'institut Henri Poincaré*, 1958.

Abhishek Kumar and Hal Daume III. Learning Task Grouping and Overlap in Multi-task Learning. *Proceedings of the 29th International Coference on Machine Learning*, 2012.

Rafal Latala and Krzysztof Oleszkiewicz. Small Ball Probability Estimatesin Terms of Width. *Studia Mathematica*, 169:305–314, 2005.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces. Classics in Mathematics.* Springer-Verlag, Berlin, 2011.

Yixuan Li, Yosinski Jason, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent Learning: Do Different Neural Networks Learn the Same Representations? *International Conference on Learning Representation*, 2016.

Percy Liang, Francis Bach, Guillaume Bouchard, and Michael I. Jordan. Asymptotically Optimal Regularizationin Smooth Parametric Models. *Advances in Neural Information Processing Systems*, 22, 2009.

Ying Lin, Kaibo Liu, Eunshin Byon, Xiaoning Qian, Shan Liu, and Shuai Huang. A Collaborative Learning Framework for Estimating Many Individualized Regression Models in a Heterogeneous Population. *IEEE Transactions on Reliability*, 67(1):328–341, 2018.

Karim Lounici, Massimiliano Pontil, Alexandre B. Tsybakov, and Sara van de Geer. Taking Advantage of Sparsity in Multi-Task Learning. *Annual Conference Computational Learning Theory*, 2009.

Shachar Lovett. An Elementary Proof of Anti-Concentration of Polynomials in Gaussian Variables. *Electronic Colloquium on Computational Complexity*, 17:182, 2010.

Colin McDiarmid. On the Method of Bounded Differences. *Invited Papers at the Twelfth British Combinatorial Conference*, pages 148–188, 1989.

Takis Merkouris. Combining Information from Multiple Surveys by Using Regression for Efficient Small Domain Estimation. *Journal of the Royal Statistical Society*, 2010.

Aditya Mishra. Boom Bikes - Linear Regression, 2021. URL `https://www.kaggle.com/datasets/kratos2597/boom-bikes-linear-regression`.

Keerthiram Murugesan, Jaime Carbonell, and Yiming Yang. Co-Clustering for Multitask Learning. *arxiv:1703.00994*, 2017.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training Language Models to Follow Instructions with Human Feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022.

Mark Rudelson and Roman Vershynin. Hanson-Wright Inequality and Sub-gaussian Concentration. *arXiv:1306.2872*, 2013.

Amit Kumar Sahu. Walmart Data Analysis and Forcasting, 2023. URL `https://www.kaggle.com/datasets/asahu40/walmart-data-analysis-and-forcasting`.

Naum Zuselevich Shor. *Minimization Methods for Non-differentiable Functions*. Springer, 1985.

Sirisha Siri. Productivity Prediction of Garment Employees, 2021. URL `https://www.kaggle.com/datasets/ishadss/productivity-prediction-of-garment-employees`.

Matthieu Solnon, Sylvain Arlot, and Francis Bach. Multi-task Regression using Minimal Penalties. *Journal of Machine Learning Research*, 13:2773–2812, 2012.

Nathan Srebro, Noga Alon, and Tommi S. Jaakkola. Generalization Error Bounds for Collaborative Prediction with

Low-Rank Matrices. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, 2004.

Sathishkumar Veerappampalayam, Jangwoo Park, and Yongyun Cho. Using Data Mining Techniques for Bike Sharing Demand Prediction in Metropolitan City. *Computer Communications*, pages 353–366, 2020.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Yaqiang Yao, Jie Cao, and Huanhuan Chen. Robust Task Grouping with Representative Tasks for Clustered Multi-Task Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 1408–1417, 2019.

Kai Yu, John Lafferty, Shenghuo Zhu, and Yihong Gong. Large-Scale Collaborative Prediction Using a Nonparametric Random Effects Model. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009.

Lining Zhang, Mengchen Wang, Liben Chen, and Wenxin Zhang. Probing gpt-3's Linguistic Knowledge on Semantic Tasks. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 297–304, 2022.

Qiyuan Zhang, Fuyuan Lyu, Xue Liu, and Chen Ma. Collaborative Performance Prediction for Large Language Models, 2024.

Qiang Zhou and Qi Zhao. Flexible Clustered Multi-Task Learning by Learning Representative Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38 (2):266–278, 2016.

Supplementary Material

Kyung Rok Kim[*1] **Yansong Wang**[2] **Xiaocheng Li**[3] **Guanting Chen**[*1]

[1]University of North Carolina at Chapel Hill
[2]University of Science and Technology of China
[3]Imperial College London

A RELATED WORK

Multi-task learning. Given a collection of datasets from multiple tasks, multi-task learning aims to enhance the learning ability by exploiting commonalities between the tasks. A large portion of the literature focuses on optimization algorithms and their convergence on risk minimization problem. Structural learning with linear predictors was studied by Ando and Zhang [2005]. Without theoretical justification, Evgeniou and Pontil [2004] took regularization into account in multi-task learning. Amit et al. [2007] proposed to decompose tasks into feature clusters and task clusters to learn common characteristics among different tasks. Lin et al. [2018], Kumar and Daume III [2012], Barzilai and Crammer [2015], and Murugesan et al. [2017] also used similar ideas with different regularization tactics. Zhou and Zhao [2016] and Yao et al. [2019] suggested to seek for representative tasks to flexibly cluster tasks by allowing a single task to be related to multiple representative tasks without specifying the number of cluster in advance.

Several studies also attempted to address the question related to the performance of the estimators. Lounici et al. [2009] worked on multi-task linear regression and provided a performance guarantee with high probability for a specific regularization coefficient, assuming sparsity of the tasks. Liang et al. [2009] devised asymptotical criterion for when a plugin estimator should be preferred to oracle or unregularized estimator. Blum et al. [2017] studied PAC framework of multi-task learning on personalized and centralized settings without making connection among similar tasks. The work that is most closely related to our study was done by Solnon et al. [2012], where they tackled multi-task kernel regression problem by estimating regularization coefficient based on the given data and providing high probability guarantees on their model.

Concentration and anti-concentration inequality. Studies on concentration inequality aim to provide a bound on the tail probability of a distribution. A wide range of study deals with sub-Gaussian random variables, see Vershynin [2018] and Boucheron et al. [2013] for example. Hanson and Wright [1971] first proposed a tail bound for a quadratic form of independent random variables, and Rudelson and Vershynin [2013] later gave a modern proof. Hsu et al. [2011] also dealt with quadratic forms of sub-Gaussian random vectors. Meanwhile, works on anti-concentration inequality pose a bound on the probability that a random variable resides in a small region. The concept of anti-concentration is based on Levy's concentration function (Kolmogorov [1958]). Latala and Oleszkiewicz [2005] provided an upper bound on the probability that a random variable falls in a small ball. Dasgupta and Gupta [2002] presented similar type of bound on χ^2 distribution.

Representation in neural networks. It is widely accepted that features from a neural network can act as representation of data. Alain and Bengio [2016] extensively investigated features of each layer of neural networks and empirically showed that deeper features predict class labels well. Li et al. [2016] discovered that features of different models could converge to a set of features which serves as a low dimensional representation. Athiwaratkun and Kang [2015] removed the top output layer of convolutional neural networks (CNNs) and demonstrated that classification with features outperforms the original CNN. Bakker and Heskes [2003] used outputs from the penultimate layer of a multi-layer perceptron as features for multi-task

[*] Corresponding to: kkrok@unc.edu, guanting@unc.edu

[*] Corresponding to: kkrok@unc.edu, guanting@unc.edu

learning, and empirically proved that task clustering increases the model performance.

Collaborative prediction. The term collaborative prediction is used in the literature in slightly different points of view. One line of research, Fan et al. [2023] and Zhang et al. [2024] for example, focuses on generating better predictions given a fixed set of trained models. These works either demonstrate performance gain empirically or provide theory on asymptotics. Another line of research uses the term collaborative prediction and collaborative filtering interchangeably to refer to the problem of matrix completion, as in Srebro et al. [2004] and Yu et al. [2009]. On the contrary, we define collaborative prediction as training an unspecified number of models on homogeneous datasets.

B PROOFS AND DETAILS

In this section, we prove the theorems presented in the main text and delineate details of the classification problem in Section 2.3. We first review the notations and setups. Then the theorems are proved under common assumptions. Additionally, we briefly consider overparametrized regime.

B.1 REGRESSION

On regression, it is possible to precisely compute out-of-sample error of a model. We first provide the expression for the error, which contains true unknown parameters. We next replace the parameters with data-driven estimators which could be computed explicitly given data. Consistency of the error is also proved under appropriate assumptions.

B.1.1 Exact Out-of-Sample Error

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the space of features and predictors, respectively. A dataset D_k is a set of data sampled from $\mathcal{X} \times \mathcal{Y}$ by some probability distribution. Suppose $D_k = \{(x_i^{(k)}, y_i^{(k)}) \in \mathbb{R}^p \times \mathbb{R} | i = 1, \cdots, n_k\}$ is generated from a distribution $P^{(k)}_{\mathcal{X}\times\mathcal{Y}}$ for $k = 1, \cdots, K$. We concatenate all the samples in the same dataset as $(X^{(k)}, y^{(k)}) = \left(\begin{bmatrix} x_1^{(k)\top} \\ \vdots \\ x_{n_k}^{(k)\top} \end{bmatrix}, \begin{bmatrix} y_1^{(k)} \\ \vdots \\ y_{n_k}^{(k)} \end{bmatrix} \right)$, and restate Assumption 1 below.

Assumption 1. *(restated) A sample $(x^{(k)}, y^{(k)})$ is generated from the distribution $P^{(k)}_{\mathcal{X}\times\mathcal{Y}}$ such that $x^{(k)} \sim P^{(k)}_{\mathcal{X}}$, the marginal distribution w.r.t. X, and $y^{(k)} = x^{(k)\top}\beta^{(k)} + \epsilon^{(k)}$, where $\epsilon^{(k)} \sim N(0, \sigma^2)$ and $\sigma > 0$. Additionally, the covariates and the noise are independent.*

Let us consider the case where only two datasets D_1 and D_2 exist. We stack all the data as $(X^{(c)}, y^{(c)}) = \left(\begin{bmatrix} X^{(1)} \\ X^{(2)} \end{bmatrix}, \begin{bmatrix} y^{(1)} \\ y^{(2)} \end{bmatrix} \right)$. The (unbiased) OLS estimator on D_k is $\hat{\beta}^{(k)} = (X^{(k)\top}X^{(k)})^{-1}X^{(k)\top}y^{(k)}$, the (unbiased) OLS estimator on $D_1 \cup D_2$ is $\hat{\beta}^{(c)} = (X^{(c)\top}X^{(c)})^{-1}X^{(c)\top}y^{(c)}$, and the (unbiased) estimator for the variance of the noise on $D_1 \cup D_2$ is $\hat{\sigma}^{(c)2} = \frac{\|y^{(c)} - X^{(c)}\hat{\beta}^{(c)}\|^2}{n_1+n_2-p}$. It is well-known that the out-of-sample error of the OLS estimator OSE($\hat{\beta}^{(k)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}$) is of the following form

$$\text{OSE}(\hat{\beta}^{(k)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) = \sigma^2 + \sigma^2 tr(E[\tilde{x}^{(k)}\tilde{x}^{(k)\top}]E[(X^{(k)\top}X^{(k)})^{-1}]),$$

where $\tilde{x}^{(k)}$ is an out-of-sample generated from the same marginal distribution $P^{(k)}_{\mathcal{X}}$. For the following theoretical development, we present an alternative form of OSE($\hat{\beta}^{(c)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}$).

Lemma 2. *Under Assumption 1,*

$$\mathit{OSE}(\hat{\beta}^{(c)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) = \sigma^2 + \sigma^2 tr(W_k E[(X^{(c)\top}X^{(c)})^{-1}]) + (\beta^{(1)} - \beta^{(2)})^\top E[Z_{3-k}^\top W_k Z_{3-k}](\beta^{(1)} - \beta^{(2)}),$$

where $W_k = E[\tilde{x}^{(k)}\tilde{x}^{(k)\top}]$ for an out-of-sample $\tilde{x}^{(k)} \sim P^{(k)}_X$ and $Z_k = (X^{(c)\top}X^{(c)})^{-1}X^{(k)\top}X^{(k)}$.

Remark. For our two datasets case, the $(3-k)$th dataset refers to the other dataset instead of D_k. For instance, if $k = 1$ so that $\hat{\beta}^{(1)}$ is being analyzed, then Z_{3-k} refers to $(X^{(c)\top}X^{(c)})^{-1}X^{(2)\top}X^{(2)}$.

Proof. By definition,

$$\begin{aligned}
OSE(\hat{\beta}^{(c)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) &= E[(\tilde{y}^{(k)} - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})^2] \\
&= E[(\tilde{x}^{(k)\top}\beta^{(k)} + \epsilon - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})^2] \\
&= E[(\tilde{x}^{(k)\top}\beta^{(k)} - E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}])^2] + E[(E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}] - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})^2] \\
&\quad + 2E\left[(\tilde{x}^{(k)\top}\beta^{(k)} - E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}])(E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}] - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})\right] + \sigma^2.
\end{aligned}$$

Note that

$$\begin{aligned}
E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}] &= E[\tilde{x}^{(k)\top}(X^{(c)\top}X^{(c)})^{-1}X^{(c)\top}y^{(c)}|\tilde{x}^{(k)}] \\
&= \tilde{x}^{(k)\top}E[(X^{(c)\top}X^{(c)})^{-1}(X^{(1)\top}X^{(1)}\beta^{(1)} + X^{(2)\top}X^{(2)}\beta^{(2)})|\tilde{x}^{(k)}] \\
&= \tilde{x}^{(k)\top}\beta^{(k)} + \tilde{x}^{(k)\top}E[(X^{(c)\top}X^{(c)})^{-1}X^{(3-k)\top}X^{(3-k)}(\beta^{(3-k)} - \beta^{(k)})|\tilde{x}^{(k)}] \\
&= \tilde{x}^{(k)\top}\beta^{(k)} + \tilde{x}^{(k)\top}E[Z_{3-k}(\beta^{(3-k)} - \beta^{(k)})|\tilde{x}^{(k)}],
\end{aligned}$$

hence

$$E[\tilde{x}^{(k)\top}\beta^{(k)} - E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}])^2] = (\beta^{(1)} - \beta^{(2)})^\top E[Z_{3-k}]^\top E[\tilde{x}^{(k)}\tilde{x}^{(k)\top}]E[Z_{3-k}](\beta^{(1)} - \beta^{(2)}). \tag{5}$$

Also, we have

$$\begin{aligned}
&E\left[(\tilde{x}^{(k)\top}\beta^{(k)} - E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}])(E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}] - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})\right] \\
&= E\left[E\left[(\tilde{x}^{(k)\top}\beta^{(k)} - E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}])(E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}] - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})|\tilde{x}^{(k)}\right]\right] \\
&= 0.
\end{aligned}$$

By Lemma 3 and (5), it follows that

$$OSE(\hat{\beta}^{(c)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) = \sigma^2 + \sigma^2 tr(W_k E[(X^{(c)\top}X^{(c)})^{-1}]) + (\beta^{(1)} - \beta^{(2)})^\top E[Z_{3-k}^\top W_k Z_{3-k}](\beta^{(1)} - \beta^{(2)}).$$

□

Lemma 3. *Under Assumption 1,*

$$\begin{aligned}
E[(E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}] - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})^2] &= \sigma^2 tr\left(W_k E[(X^{(c)\top}X^{(c)})^{-1}]\right) \\
&\quad + (\beta^{(k)} - \beta^{(3-k)})^\top E[Z_{3-k}^\top W_k Z_{3-k}](\beta^{(k)} - \beta^{(3-k)}) \\
&\quad - (\beta^{(k)} - \beta^{(3-k)})^\top E[Z_{3-k}]^\top W_k E[Z_{3-k}](\beta^{(k)} - \beta^{(3-k)}).
\end{aligned}$$

Proof. We first derive

$$\begin{aligned}
E[(E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}] - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})^2] &= E[E[(E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}] - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})^2|\tilde{x}^{(k)}]] \\
&= E[Var(\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)})],
\end{aligned}$$

and

$$\begin{aligned}
Var(\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}) &= E[Var(\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|X^{(c)}, \tilde{x}^{(k)})|\tilde{x}^{(k)}] + Var(E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|X^{(c)}, \tilde{x}^{(k)\top}]|\tilde{x}^{(k)}) \\
&= \sigma^2\tilde{x}^{(k)\top}E[(X^{(c)\top}X^{(c)})^{-1}]\tilde{x}^{(k)\top} \\
&\quad + Var\left(\tilde{x}^{(k)\top}\left(\beta^{(k)} + (X^{(c)\top}X^{(c)})^{-1}X^{(3-k)\top}X^{(3-k)}(\beta^{(3-k)} - \beta^{(k)})\right)|\tilde{x}^{(k)}\right) \\
&= \sigma^2\tilde{x}^{(k)\top}E[(X^{(c)\top}X^{(c)})^{-1}]\tilde{x}^{(k)\top} + \tilde{x}^{(k)\top}Var\left(Z_{3-k}(\beta^{(k)} - \beta^{(3-k)})\right)\tilde{x}^{(k)},
\end{aligned}$$

hence

$$E[(E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}] - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})^2] = \sigma^2 tr\left(W_k E[(X^{(c)\top}X^{(c)})^{-1}]\right) + tr\left(W_k Var\left(Z_{3-k}(\beta^{(k)} - \beta^{(3-k)})\right)\right).$$

To further simplify $Var\left(Z_{3-k}(\beta^{(k)}-\beta^{(3-k)})\right)$,

$$\begin{aligned} Var\left(Z_{3-k}(\beta^{(k)}-\beta^{(3-k)})\right) &= E[Z_{3-k}(\beta^{(k)}-\beta^{(3-k)})(\beta^{(k)}-\beta^{(3-k)})^\top Z_{3-k}^\top] \\ &\quad - E[Z_{3-k}](\beta^{(k)}-\beta^{(3-k)})(\beta^{(k)}-\beta^{(3-k)})^\top E[Z_{3-k}]^\top. \end{aligned}$$

Since

$$\begin{aligned} E[Z_{3-k}(\beta^{(k)}-\beta^{(3-k)})(\beta^{(k)}-\beta^{(3-k)})^\top Z_{3-k}^\top]_{i,j} &= E[(Z_{3-k})_{i,:}(\beta^{(k)}-\beta^{(3-k)})(\beta^{(k)}-\beta^{(3-k)})^\top (Z_{3-k})_{j,:}^\top] \\ &= (\beta^{(k)}-\beta^{(3-k)})^\top E[(Z_{3-k})_{j,:}^\top (Z_{3-k})_{i,:}](\beta^{(k)}-\beta^{(3-k)}), \end{aligned}$$

where $(Z_{3-k})_{i,:}$ refers to the ith row of Z_{3-k}, it follows that

$$\begin{aligned} tr\left(W_k Var\left(Z_{3-k}(\beta^{(k)}-\beta^{(3-k)})\right)\right) &= tr\left(W_k \left[(\beta^{(k)}-\beta^{(3-k)})^\top E[(Z_{3-k})_{j,:}^\top (Z_{3-k})_{i,:}](\beta^{(k)}-\beta^{(3-k)})\right]_{i,j=1,\cdots,p}\right) \\ &\quad - (\beta^{(k)}-\beta^{(3-k)})^\top E[Z_{3-k}]^\top W_k E[Z_{3-k}](\beta^{(k)}-\beta^{(3-k)}). \end{aligned}$$

Now, because we have the following relations

$$\begin{aligned} & tr\left(W_k \left[(\beta^{(k)}-\beta^{(3-k)})^\top E[(Z_{3-k})_{j,:}^\top (Z_{3-k})_{i,:}](\beta^{(k)}-\beta^{(3-k)})\right]_{i,j=1,\cdots,p}\right) \\ &= \sum_{j=1}^{p}(W_k)_{1,j}(\beta^{(k)}-\beta^{(3-k)})^\top E[(Z_{3-k})_{j,:}^\top (Z_{3-k})_{1,:}](\beta^{(k)}-\beta^{(3-k)}) \\ &\quad + \cdots + \sum_{j=1}^{p}(W_k)_{p,j}(\beta^{(k)}-\beta^{(3-k)})^\top E[(Z_{3-k})_{j,:}^\top (Z_{3-k})_{p,:}](\beta^{(k)}-\beta^{(3-k)}) \\ &= \sum_{i=1}^{p}\sum_{j=1}^{p}(W_k)_{i,j}(\beta^{(k)}-\beta^{(3-k)})^\top E[(Z_{3-k})_{j,:}^\top (Z_{3-k})_{i,:}](\beta^{(k)}-\beta^{(3-k)}) \\ &= (\beta^{(k)}-\beta^{(3-k)})^\top \sum_{i=1}^{p}\sum_{j=1}^{p}(W_k)_{i,j} E[(Z_{3-k})_{j,:}^\top (Z_{3-k})_{i,:}](\beta^{(k)}-\beta^{(3-k)}) \\ &= (\beta^{(k)}-\beta^{(3-k)})^\top E[Z_{3-k}^\top W_k Z_{3-k}(\beta^{(k)}-\beta^{(3-k)}), \end{aligned}$$

we get

$$\begin{aligned} E[(E[\tilde{x}^{(k)\top}\hat{\beta}^{(c)}|\tilde{x}^{(k)}] - \tilde{x}^{(k)\top}\hat{\beta}^{(c)})^2] &= \sigma^2 tr\left(W_k E[(X^{(c)\top}X^{(c)})^{-1}]\right) \\ &\quad + (\beta^{(k)}-\beta^{(3-k)})^\top E[Z_{3-k}^\top W_k Z_{3-k}](\beta^{(k)}-\beta^{(3-k)}) \\ &\quad - (\beta^{(k)}-\beta^{(3-k)})^\top E[Z_{3-k}]^\top W_k E[Z_{3-k}](\beta^{(k)}-\beta^{(3-k)}). \end{aligned}$$

□

Now that the exact error is identified, the equivalent condition in Theorem 1 for error reduction can be proved.

Theorem 1. *Under Assumption 1, we have*

$$\sum_{k=1}^{2} OSE(\hat{\beta}^{(k)}, P_{\mathcal{X}\times\mathcal{Y}}^{(k)}) > \sum_{k=1}^{2} OSE(\hat{\beta}^{(c)}, P_{\mathcal{X}\times\mathcal{Y}}^{(k)})$$

if and only if

$$h(\sigma^2) > g(\beta^{(1)}, \beta^{(2)}),$$

where

$$\begin{aligned} h(x) &= A_0 x, \\ g(y,z) &= \|y-z\|_{B_0}^2, \\ A_0 &= tr\left(W_1 E\left[(X^{(1)\top}X^{(1)})^{-1} - (X^{(c)\top}X^{(c)})^{-1}\right] + W_2 E\left[(X^{(2)\top}X^{(2)})^{-1} - (X^{(c)\top}X^{(c)})^{-1}\right]\right), \textit{ and} \\ B_0 &= E[Z_1^\top W_2 Z_1] + E[Z_2^\top W_1 Z_2]. \end{aligned}$$

Proof. By Lemma 2,

$$\begin{aligned}
&\sum_{k=1}^{2}\text{OSE}(\hat\beta^{(k)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) > \sum_{k=1}^{2}\text{OSE}(\hat\beta^{(c)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) \\
&\quad\iff 2\sigma^2 + \sigma^2 tr\left(W_1 E\left[(X^{(1)\top}X^{(1)})^{-1}\right] + W_2 E\left[(X^{(2)\top}X^{(2)})^{-1}\right]\right) > \sigma^2 tr\left((W_1+W_2)E[(X^{(c)\top}X^{(c)})^{-1}]\right) \\
&\quad\quad + (\beta^{(1)}-\beta^{(2)})^\top\left(E[Z_1^\top W_2 Z_1] + E[Z_2^\top W_1 Z_2]\right)(\beta^{(1)}-\beta^{(2)}) + 2\sigma^2 \\
&\quad\iff \sigma^2 tr\left(W_1 E\left[(X^{(1)\top}X^{(1)})^{-1} - (X^{(c)\top}X^{(c)})^{-1}\right] + W_2 E\left[(X^{(2)\top}X^{(2)})^{-1} - (X^{(c)\top}X^{(c)})^{-1}\right]\right) \\
&\quad\quad > (\beta^{(1)}-\beta^{(2)})^\top\left(E[Z_1^\top W_2 Z_1] + E[Z_2^\top W_1 Z_2]\right)(\beta^{(1)}-\beta^{(2)}).
\end{aligned}$$

□

B.1.2 Replacing Parameters with Estimators

Now, we prove Lemma 1, which replaces $\beta^{(k)}$ and σ^2 with estimators by imposing high probability bounds. We first state concentration and anti-concentration inequalities from existing literature, and then develop high probability bounds on both sides of the inequality in Theorem 1.

Theorem 2. *[Hsu et al., 2011] If $K \in \mathbb{R}^{m\times n},\ L = K^\intercal K,\ \mu \in \mathbb{R}^n,\ \sigma \geq 0$, and $E[e^{\alpha^\intercal(x-\mu)}] \leq e^{\frac{1}{2}\sigma^2\|\alpha\|^2},\ \forall \alpha \in \mathbb{R}^n$, then for $\forall t > 0$*

$$\mathbb{P}\left(\|Kx\|^2 > \sigma^2\left\{tr(L) + 2\sqrt{tr(L^2)t} + 2\|L\|t\right\} + \|K\mu\|^2\sqrt{1 + 4\sqrt{\frac{\|L^2\|}{tr(L^2)}t} + \frac{4\|L\|^2}{tr(L^2)}t}\right) \leq e^{-t}.$$

Theorem 3. *[Lovett, 2010] If $V \sim N_d(0, I)$, then*

$$\mathbb{P}(\|V\|^2 \leq \epsilon) \leq \sqrt{2e\frac{\epsilon}{d}}.$$

Next, we present lemmas needed for the application of our setting.

Lemma 4. *If $Z \sim N_d(\mu, I_d)$, then*

$$\mathbb{P}(\|Z\| \leq t) \leq \sqrt{\frac{2e}{d}}(\|\mu\| + t) + 2e^{-\frac{(\|\mu\|-t)^2}{2d}} - 1$$

for $t < \|\mu\|$.

Proof. From Theorem 3, we can show that

$$\begin{aligned}
\mathbb{P}(\|Z\| \leq t) &\leq \mathbb{P}(\|Z-\mu\| \leq \|\mu\| + t) - \mathbb{P}(\|Z-\mu\| \leq \|\mu\| - t) \\
&\leq \sqrt{\frac{2e}{d}}(\|\mu\| + t) + 2e^{-\frac{(\|\mu\|-t)^2}{2d}} - 1
\end{aligned}$$

□

Based on these inequalities, a lower bound for $f(\sigma^2)$ can be obtained as follows.

Lemma 5. *For $D^\top D = \{(X^{(1)\top}X^{(1)})^{-1} + (X^{(2)\top}X^{(2)})^{-1}\}^{-1}$,*

$$h(\sigma^2) \geq \phi_\delta(\hat\beta^{(1)}, \hat\beta^{(2)}, \hat\sigma^{(c)2})$$

with probability at least $1-2\delta$*, where*

$$\phi_\delta(u,v,w) = \tilde{A}_1(\delta)w + \tilde{A}_2(\delta)\|D(u-v)\|^2,$$
$$\tilde{A}_1(\delta) = \frac{n_0 A_0}{\left(n_0 + 2\sqrt{n_0\log\frac{1}{\delta}} + 2\log\frac{1}{\delta}\right)\left(3+4\sqrt{\frac{1}{n_0}\log\frac{1}{\delta}}\right)},$$
$$\tilde{A}_2(\delta) = \frac{-2A_0\left(1+2\sqrt{\frac{1}{n_0}\log\frac{1}{\delta}}\right)}{\left(n_0 + 2\sqrt{n_0\log\frac{1}{\delta}} + 2\log\frac{1}{\delta}\right)\left(3+4\sqrt{\frac{1}{n_0}\log\frac{1}{\delta}}\right)},$$

and $n_0 = n_1+n_2-p$*.*

Proof. Let us introduce additional notations for more efficient computation. Let $X^{(0)} = \begin{bmatrix} X^{(1)} & 0 \\ 0 & X^{(2)} \end{bmatrix}$, $y^{(0)} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \end{bmatrix}$, $\epsilon^{(0)} = \begin{bmatrix} \epsilon^{(1)} \\ \epsilon^{(2)} \end{bmatrix}$, and $\beta^{(0)} = \begin{bmatrix} \beta^{(1)} \\ \beta^{(2)} \end{bmatrix}$. Then the estimator for σ^2 on the combined dataset $D_1 \cup D_2$ is $\hat{\sigma}^{(c)2} = \frac{1}{n_1+n_2-p}(\|y^{(1)} - X^{(1)}\hat{\beta}^{(c)}\|^2 + \|y^{(2)} - X^{(2)}\hat{\beta}^{(c)}\|^2)$.

Let us further denote $C = \begin{bmatrix} I_p & -I_p \end{bmatrix}$ and $A = X^{(0)}(X^{(0)\top}X^{(0)})^{-1}C^\top$, and let $M = A(A^\top A)^{-1}A^\top$ and $H_{X^{(0)}} = X^{(0)}(X^{(0)\top}X^{(0)})^{-1}X^{(0)}$. Then $I - H_{X^{(0)}} + M$ is symmetric and idempotent, and $\hat{\sigma}^{(c)2} = \frac{1}{n_1+n_2-p}y^{(0)\top}(I - H_{X^{(0)}} + M)y^{(0)}$. Since $tr(I - H_{X^{(0)}} + M) = n_1+n_2-p$, $\|I - H_{X^{(0)}} + M\| = \lambda_{max}(I - H_{X^{(0)}} + M) = 1$, and

$$\begin{aligned}\|(I - H_{X^{(0)}} + M)X^{(0)}\beta^{(0)}\|^2 &= \beta^{(0)\top}X^{(0)\top}(I - H_{X^{(0)}} + M)X^{(0)}\beta^{(0)} \\ &= \beta^{(0)\top}C^\top\{C(X^{(0)\top}X^{(0)})^{-1}C^\top\}^{-1}C\beta^{(0)},\end{aligned}$$

Theorem 2 gives

$$\begin{aligned}(n_1+n_2-p)\hat{\sigma}^{(c)2} \le \sigma^2 &\left(n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}} + 2\log\frac{1}{\delta}\right) \\ &+ \left\{1 + 2\sqrt{\frac{\log\frac{1}{\delta}}{n_1+n_2-p}}\right\}\|D(\beta^{(1)} - \beta^{(2)})\|^2 \end{aligned} \tag{6}$$

with probability at least $1-\delta$, where $\lambda_{max}(\cdot)$ denotes the maximum eigenvalue of a matrix.

As $D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)}) = DC\hat{\beta}^{(0)} = DA^\intercal y^{(0)}$, Theorem 2 leads to

$$\mathbb{P}(\|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)}) - D(\beta^{(1)} - \beta^{(2)})\|^2 > \sigma^2(tr(M) + 2\sqrt{tr(M)t} + 2\|M\|t)) \le e^{-t},$$

or equivalently,

$$\|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)}) - D(\beta^{(1)} - \beta^{(2)})\| \le \sqrt{\sigma^2\left(n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}} + 2\log\frac{1}{\delta}\right)} \tag{7}$$

with probability at least $1-\delta$. Hence

$$\begin{aligned}\|D(\beta^{(1)} - \beta^{(2)})\|^2 &\le 2\|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)}) - D(\beta^{(1)} - \beta^{(2)})\|^2 + 2\|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)})\|^2 \\ &\le 2\sigma^2\left(n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}} + 2\log\frac{1}{\delta}\right) + 2\|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)})\|^2 \end{aligned} \tag{8}$$

with probability at least $1-\delta$. Combining (6) and (8) gives

$$\begin{aligned}(n_1+n_2-p)\hat{\sigma}^{(c)2} \le \sigma^2 &\left(n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}} + 2\log\frac{1}{\delta}\right)\left(3 + 4\sqrt{\frac{\log\frac{1}{\delta}}{n_1+n_2-p}}\right) \\ &+ 2\left(1 + 2\sqrt{\frac{\log\frac{1}{\delta}}{n_1+n_2-p}}\right)\|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)})\|^2,\end{aligned}$$

or equivalently,

$$h(\sigma^2) \geq \tilde{A}_1(\delta)\hat{\sigma}^{(c)2} + \tilde{A}_2(\delta)\|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)})\|^2.$$

with probability at least $1 - 2\delta$. □

Next, $g(\beta^{(1)}, \beta^{(2)})$ can also be bounded in a similar way, and we show it in the following lemma.

Lemma 6. *For* $D^\top D = \{(X^{(1)\top}X^{(1)})^{-1} + (X^{(2)\top}X^{(2)})^{-2}\}^{-1}$,

$$g(\beta^{(1)}, \beta^{(2)}) \leq \psi_\delta(\hat{\beta}^{(1)}, \hat{\beta}^{(2)}, \hat{\sigma}^{(c)2})$$

with probability at least $1 - 2\delta$*, where*

$$\psi_\delta(u, v, w) = \left\{\sqrt{g(u, v)} + T_\delta(w, \|D(u - v)\|)\right\}^2$$

and $T_\delta(x, y) = O(\max\{x, y\})$.

Proof. Let $I - H_{X^{(0)}} + M = P\Lambda P^\intercal$. Since $I - H_{X^{(0)}} + M$ is symmetric and idempotent, its eigenvalues are 0 or 1. As $tr(I - H_{X^{(0)}} + M) = n_1 + n_2 - p$, assume the first $n_1 - n_2 + p$ diagonal entries of Λ be 1 and the rest be 0 without loss of generality. Then

$$\begin{aligned} P^\top(I - H_{X^{(0)}} + M)y^{(0)} &\sim N(P^\top MX^{(0)}\beta^{(0)}, \sigma^2\Lambda) \\ &= N_{n_1+n_2}\left(P^\intercal MX^{(0)}\beta^{(0)}, \sigma^2 \begin{bmatrix} I_{n_1+n_2-p} & 0 \\ 0 & 0 \end{bmatrix}\right). \end{aligned}$$

where I_n denotes the $n \times n$ identity matrix.

Denote $F = \begin{bmatrix} I_{n_1+n_2-p} & 0 \end{bmatrix}$ and $G = \begin{bmatrix} 0 & I_p \end{bmatrix}$. For a constant c,

$$\begin{aligned} \mathbb{P}((n_1 + n_2 - p)\hat{\sigma}^{(c)2} \leq c) &= \mathbb{P}(\|P^\top(I - H_{X^{(0)}} + M)y^{(0)}\|^2 \leq c) \\ &= \mathbb{P}(\|FP^\top(I - H_{X^{(0)}} + M)y^{(0)}\|^2 + \|GP^\top(I - H_{X^{(0)}} + M)y^{(0)}\|^2 \leq c) \end{aligned}$$

Let $Z = FP^\intercal(I - H_{X^{(0)}} + M)y^{(0)}$. Then $Z \sim N(FP^\intercal MX^{(0)}\beta^{(0)}, \sigma^2 I)$, and

$$\begin{aligned} \|FP^\top MX^{(0)}\beta^{(0)}\|^2 &= \beta^{(0)\top}X^{(0)\top}M(I - H_{X^{(0)}} + M)MX^{(0)}\beta^{(0)} \\ &= \beta^{(0)\top}C^\top(C(X^{(0)\top}X^{(0)})^{-1}C^\top)^{-1}C\beta^{(0)} \\ &= \|D(\beta^{(1)} - \beta^{(2)})\|^2. \end{aligned}$$

Also, $GP^\top(I - H_{X^{(0)}} + M)y^{(0)} \sim N(GP^\top MX^{(0)}\beta^{(0)}, 0)$, and

$$\begin{aligned} \|GP^\top MX^{(0)}\beta^{(0)}\|^2 &= \beta^{(0)\top}X^{(0)\top}MP(I - \Lambda)P^\top MX^{(0)}\beta^{(0)} \\ &= \beta^{(0)\top}X^{(0)\top}M(I - I + H_{X^{(0)}} - M)MX^{(0)}\beta^{(0)} \\ &= 0. \end{aligned}$$

Combining with Lemma 4, it follows that

$$\begin{aligned} \mathbb{P}((n_1 + n_2 - p)\hat{\sigma}^{(c)2} \leq c) &\leq \frac{1}{\sigma}\sqrt{\frac{2e}{n_1 + n_2 - p}}(\|D(\beta^{(1)} - \beta^{(2)})\| + \sqrt{c}) + 2e^{-\frac{(\|D(\beta^{(1)} - \beta^{(2)})\| - \sqrt{c})^2}{2(n_1+n_2-p)}} - 1 \\ &\leq \frac{2}{\sigma}\sqrt{\frac{2e}{n_1 + n_2 - p}}\|D(\beta^{(1)} - \beta^{(2)})\| + 2e^{-\frac{(\|D(\beta^{(1)} - \beta^{(2)})\| - \sqrt{c})^2}{2(n_1+n_2-p)}} - 1 \end{aligned}$$

for $c \leq \|D(\beta^{(1)} - \beta^{(2)})\|^2$. Hence

$$
\begin{aligned}
(n_1 + n_2 - p)\hat{\sigma}^{(c)2} &\geq \left\{\|D(\beta^{(1)} - \beta^{(2)})\| + \sigma\sqrt{2(n_1 + n_2 - p)\log\frac{2}{1 + \delta - 2\|D(\beta^{(1)} - \beta^{(2)})\|\sqrt{\frac{2e}{n_1+n_2-p}}\frac{1}{\sigma}}}\right\}^2 \\
&\geq \left\{\|D(\beta^{(1)} - \beta^{(2)})\| + \sigma\sqrt{2(n_1 + n_2 - p)}\left(\sqrt{\log\frac{4}{1+\delta}} + \frac{1}{2\sqrt{\log\frac{4}{1+\delta}}} - \frac{1}{8\|D(\beta^{(1)} - \beta^{(2)})\|}\right.\right. \\
&\quad \left.\left.\times\sqrt{\frac{n_1 + n_2 - p}{2e}}\frac{1+\delta}{\sqrt{\log\frac{4}{1+\delta}}}\sigma\right)\right\}^2
\end{aligned}
$$

with probability at least $1 - \delta$ as the function $\xi : s \mapsto \sqrt{\log\frac{1}{a - \frac{b}{s}}}$ satisfies

$$
\xi(s) \geq \xi\left(\frac{2b}{a}\right) + \xi'\left(\frac{2b}{a}\right)\left(s - \frac{2b}{a}\right) = -\frac{a}{4b\sqrt{\log\frac{4}{a}}}s + \sqrt{\log\frac{4}{a}} + \frac{1}{2\sqrt{\log\frac{4}{a}}}.
$$

Rearranging the terms gives

$$
\begin{aligned}
\|D(\beta^{(1)} - \beta^{(2)})\|\sqrt{(n_1 + n_2 - p)\hat{\sigma}^{(c)2}} &\geq -\frac{n_1 + n_2 - p}{8}\frac{1+\delta}{\sqrt{e\log\frac{4}{1+\delta}}}\sigma^2 + +\|D(\beta^{(1)} - \beta^{(2)})\|^2 \\
&\quad + \|D(\beta^{(1)} - \beta^{(2)})\|\sqrt{2(n_1 + n_2 - p)}\left(\sqrt{\log\frac{4}{1+\delta}} + \frac{1}{2\sqrt{\log\frac{4}{1+\delta}}}\right)\sigma \qquad (9)
\end{aligned}
$$

with probability at least $1 - \delta$.

Meanwhile, from (7),

$$
\begin{aligned}
\|D(\beta^{(1)} - \beta^{(2)})\| &\leq \|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)}) - D(\beta^{(1)} - \beta^{(2)})\| + \|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)})\| \\
&\leq \sqrt{\sigma^2(n_1 + n_2 - p + 2\sqrt{n_1 + n_2 - p}\sqrt{\log\frac{1}{\delta}} + 2\log\frac{1}{\delta})} + \|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)})\| \qquad (10)
\end{aligned}
$$

and

$$
\begin{aligned}
\|D(\beta^{(1)} - \beta^{(2)})\| &\geq \|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)})\| - \|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)}) - D(\beta^{(1)} - \beta^{(2)})\| \\
&\geq \|D(\hat{\beta}^{(1)} - \hat{\beta}^{(2)})\| - \sqrt{\sigma^2(n_1 + n_2 - p + 2\sqrt{n_1 + n_2 - p}\sqrt{\log\frac{1}{\delta}} + 2\log\frac{1}{\delta})} \qquad (11)
\end{aligned}
$$

with probability at least $1 - \delta$.

Combining (9), (10), and (11) gives

$$
\begin{aligned}
&\sqrt{(n_1+n_2-p)\hat{\sigma}^{(c)2}}\left(\sqrt{\sigma^2(n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}}+2\log\frac{1}{\delta})}+\|D(\hat{\beta}^{(1)}-\hat{\beta}^{(2)})\|\right)\\
&\geq \|D(\beta^{(1)}-\beta^{(2)})\|\sqrt{(n_1+n_2-p)\hat{\sigma}^{(c)2}}\\
&\geq -\frac{n_1+n_2-p}{8}\frac{1+\delta}{\sqrt{e\log\frac{4}{1+\delta}}}\sigma^2+\|D(\beta^{(1)}-\beta^{(2)})\|^2\\
&\quad+\|D(\beta^{(1)}-\beta^{(2)})\|\sqrt{2(n_1+n_2-p)}\left(\sqrt{\log\frac{4}{1+\delta}}+\frac{1}{2\sqrt{\log\frac{4}{1+\delta}}}\right)\sigma\\
&\geq -\frac{n_1+n_2-p}{8}\frac{1+\delta}{\sqrt{e\log\frac{4}{1+\delta}}}\sigma^2+\left(\sqrt{\log\frac{4}{1+\delta}}+\frac{1}{2\sqrt{\log\frac{4}{1+\delta}}}\right)\sqrt{2(n_1+n_2-p)}\sigma\\
&\quad\times\left(\|D(\hat{\beta}^{(1)}-\hat{\beta}^{(2)})\|-\sigma\sqrt{(n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}}+2\log\frac{1}{\delta})}\right)\\
&\quad+\left(\|D(\hat{\beta}^{(1)}-\hat{\beta}^{(2)})\|-\sqrt{\sigma^2(n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}}+2\log\frac{1}{\delta})}\right)^2
\end{aligned}
$$

with probability at least $1-2\delta$. Solving σ leads to

$$
\sigma\leq c_0:=\frac{c_1-c_4+\sqrt{(c_1-c_4)^2-4c_3(c_5-c_2)}}{2c_3}
$$

with probability at least $1-2\delta$, and the parameters being

$$
\begin{aligned}
c_1&=\sqrt{(n_1+n_2-p)\hat{\sigma}^{(c)2}}\sqrt{n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}}+2\log\frac{1}{\delta}},\\
c_2&=\sqrt{(n_1+n_2-p)\hat{\sigma}^{(c)2}}\|D(\hat{\beta}^{(1)}-\hat{\beta}^{(2)})\|,\\
c_3&=-\frac{n_1+n_2-p}{8}\frac{1+\delta}{\sqrt{e\log\frac{4}{1+\delta}}}-\sqrt{2(n_1+n_2-p)}\left(\sqrt{\log\frac{4}{1+\delta}}+\frac{1}{2\sqrt{\log\frac{4}{1+\delta}}}\right)\\
&\quad\times\sqrt{n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}}+2\log\frac{1}{\delta}}+n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}}+2\log\frac{1}{\delta},\\
c_4&=\|D(\hat{\beta}^{(1)}-\hat{\beta}^{(2)})\|\left\{\sqrt{2(n_1+n_2-p)}\left(\sqrt{\log\frac{4}{1+\delta}}+\frac{1}{2\sqrt{\log\frac{4}{1+\delta}}}\right)\right.\\
&\quad\left.-2\sqrt{n_1+n_2-p+2\sqrt{n_1+n_2-p}\sqrt{\log\frac{1}{\delta}}+2\log\frac{1}{\delta}}\right\},\ \text{and}\\
c_5&=\|D(\hat{\beta}^{(1)}-\hat{\beta}^{(2)})\|^2.
\end{aligned}
$$

Now, for $B_0=B^\top B$, applying Theorem 2 leads to

$$
\begin{aligned}
\|B(\beta^{(1)}-\beta^{(2)})-B(\hat{\beta}^{(1)}-\hat{\beta}^{(2)})\|^2&=\|BC(X^{(0)\top}X^{(0)})^{-1}X^{(0)\top}(X^{(0)}\beta^{(0)}-y^{(0)})\|^2\\
&\leq\sigma^2\left\{tr(\Sigma)+2\sqrt{tr(\Sigma^2)\log\frac{1}{\delta}}+2\|\Sigma\|\log\frac{1}{\delta}\right\}
\end{aligned}
$$

with probability at least $1-\delta$, and $\Sigma = X^{(0)}(X^{(0)\top}X^{(0)})^{-1}C^\top B^\top BC(X^{(0)\top}X^{(0)})^{-1}X^{(0)\top}$. Therefore,

$$\begin{aligned} g(\beta^{(1)},\beta^{(2)}) &\le \left\{\|B(\beta^{(1)}-\beta^{(2)}) - B(\hat\beta^{(1)}-\hat\beta^{(2)})\| + \|B(\hat\beta^{(1)}-\hat\beta^{(2)})\|\right\}^2 \\ &\le \left\{\sigma\sqrt{tr(\Sigma)+2\sqrt{tr(\Sigma^2)\log\frac{1}{\delta}}+2\|\Sigma\|\log\frac{1}{\delta}} + \sqrt{g(\hat\beta^{(1)},\hat\beta^{(2)})}\right\}^2 \end{aligned} \tag{12}$$

with probability at least $1-\delta$. With (12), the bound on $g(\beta^{(1)},\beta^{(2)})$ finally becomes

$$g(\beta^{(1)},\beta^{(2)}) \le \left\{c_0\sqrt{tr(\Sigma)+2\sqrt{tr(\Sigma^2)\log\frac{1}{\delta}}+2\|\Sigma\|\log\frac{1}{\delta}} + \sqrt{g(\hat\beta^{(1)},\hat\beta^{(2)})}\right\}^2$$

with probability at least $1-3\delta$. □

Finally, the proof of Lemma 1 is a natural implication of Lemma 5 and 6.

Lemma 1. *(restated) With probability at least $1-5\delta$, we have $f(\sigma^2) \ge g(\beta^{(1)},\beta^{(2)})$ if*

$$\phi_\delta(\hat\beta^{(1)},\hat\beta^{(2)},\hat\sigma^{(c)2}) \ge \psi_\delta(\hat\beta^{(1)},\hat\beta^{(2)},\hat\sigma^{(c)2}).$$

Proof. From Lemma 5,

$$h(\sigma^2) \ge \phi_\delta(\hat\beta^{(1)},\hat\beta^{(2)},\hat\sigma^{(c)2})$$

with probability at least $1-2\delta$. From Lemma 6,

$$g(\beta^{(1)},\beta^{(2)}) \le \psi_\delta(\hat\beta^{(1)},\hat\beta^{(2)},\hat\sigma^{(c)2})$$

with probability at least $1-3\delta$. Therefore, if

$$\phi_\delta(\hat\beta^{(1)},\hat\beta^{(2)},\hat\sigma^{(c)2}) \ge \psi_\delta(\hat\beta^{(1)},\hat\beta^{(2)},\hat\sigma^{(c)2}),$$

then $f(\sigma^2) \ge g(\beta^{(1)},\beta^{(2)})$ with probability at least $1-5\delta$. □

B.1.3 Theoretical Guarantee for Consistency of Algorithm 2

In this section, we justify the consistency of our estimation on $\widehat{\text{OSE}}$ for Algorithm 2. Proposition 1 guarantees that if we combine the two datasets D_1 and D_2 whenever $\phi_\delta > \psi_\delta$ and do not merge otherwise, then we have $\sum_{k=1}^2 \text{OSE}(\hat\beta^{(k)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) > \sum_{k=1}^2 \text{OSE}(\hat\beta^{(c)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}})$ with probability $1-5\delta$. Next we will show that the estimator $\widehat{\text{OSE}}$ used in Algorithm 2 is consistent.

We aim to develop algorithm with high SR, which is computationally infeasible if we only have access to empirical distribution. From (4), a natural idea is to estimate the out-of-sample errors to approximate SR. With the lack of generating distribution, one can only rely on the existing sampled dataset $\{D_{all}\}$, and the problem becomes how to provide a valid estimation for $\text{OSE}(\hat f, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) = E_{P^{(k)}_{\mathcal{X}\times\mathcal{Y}}}[(\tilde y - \hat f(\tilde x))^2]$ for some function $\hat f$ where $(\tilde x,\tilde y)\sim P^{(k)}_{\mathcal{X}\times\mathcal{Y}}$.

In practice, our estimator $\widehat{\text{OSE}}\left(\hat\beta, D_k^{out}\right)$ is computed by the method of bootstrap. As we do not assume knowledge of the underlying distribution, this nonparametric method is coherent with our assumption. Specifically, the m-th bootstrap samples $\{(x^*_{i,m}, y^*_{i,m})\}_{i=1}^{\tilde n_k}$ are generated from D_k^{out} for $m=1,\cdots,M$. We estimate the error by $\widehat{\text{OSE}}_M(\hat\beta, D_k^{out}) = \frac{1}{M}\sum_{m=1}^M \frac{1}{\tilde n_k}\sum_{i=1}^{\tilde n_k}(y^*_{i,m} - x^{*\top}_{i,m}\hat\beta)^2$. By boostrapping, we get a consistent estimator of the out-of-sample error.

As described below, we can have the following theoretical guarantee under some technical assumption. For a reminder, samples $(X^{(k)}, y^{(k)})$ are generated from the distribution $P^{(k)}_{\mathcal{X}\times\mathcal{Y}}$ and $P^{(k)}_X$ is the marginal distribution. Let $X^{(c)}$ denote the combined data $[X^{(1)\top}, X^{(2)\top}]^\top$. Moreover, let μ_k and Σ_k be the mean and the variance of $P^{(k)}_X$.

Assumption 2. *Denote $b_k = E[\hat\beta^{(c)}|X^{(c)}] - \beta^{(k)}$, and its second moment $E[b_k b_k^\top]$ to be Λ_k. We define $\Omega_k = \mathbb{E}[(X^{(k)\top}X^{(k)})^{-1}]$ and $\Omega_c = \mathbb{E}[(X^{(c)\top}X^{(c)})^{-1}]$. Finally, we assume $\Omega_k\mu_k \ne 0$ and $\sigma^2\Omega_c\mu_k + \Lambda_k\mu_k \ne 0$.*

Theorem 4. *With Assumption 1 and 2, we have $\widehat{OSE}_M(\hat\beta^{(k)}, D_k^{out}) - OSE(\hat\beta^{(k)}, P^k_{X\times Y}) \xrightarrow{p} 0$ as $\tilde n_k \to \infty$.*

B.1.4 Consistency of Errors

Next, we prove Theorem 4, the consistency of the estimator of the out-of-sample error. We first make the following assumption.

Assumption 2. *(restated) Denote $b_k = E[\hat{\beta}^{(c)}|X^{(c)}] - \beta^{(k)}$. Let $\Lambda_k = E[b_k b_k^\top]$, $\Omega_k = \mathbb{E}[(X^{(k)\top}X^{(k)})^{-1}]$ and $\Omega_c = \mathbb{E}[(X^{(c)\top}X^{(c)})^{-1}]$ be given. We assume $\Omega_k\mu_k \neq 0$ and $\sigma^2\Omega_c\mu_k + \Lambda_k\mu_k \neq 0$.*

Consistency result is built upon the following theorem on central limit theorem of bootstrapping.

Theorem 5. *[Gill, 1989] Let $\mathcal{F}$ denote the space of all cumulative distribution functions. Assume $X_1, \cdots, X_n$ are iid samples from a distribution $F \in \mathcal{F}$ and $X_1^*, \cdots, X_n^*$ are bootstrapped samples from their empirical distribution $\hat{F}_n$. Let us denote the empirical distribution of the bootstrapped samples as $\hat{F}_n^*$. For $\mu = E[X_1]$, if a functional T on $\mathcal{F}$ satisfies $T(F) = g(\mu)$ for some continuously differentiable function g and $g'(\mu) \neq 0$, then*

$$\sup_u \left| \mathbb{P}_{\hat{F}_n}\left(\sqrt{n}(T(\hat{F}_n^*) - T(\hat{F}_n) \leq u\right) - \mathbb{P}_F\left(\sqrt{n}(T(\hat{F}_n) - T(F) \leq u\right) \right| \to 0$$

almost surely as $n \to \infty$.

Theorem 4 is proved by comparing the limit of estimated out-of-sample error of $\hat{\beta}^{(k)}$ on original samples and bootstrapped samples.

Theorem 4. *(restated) Under Assumption 1 and 2,*

$$\widehat{OSE}_M(\hat{\beta}^{(k)}, D_k^{out}) - OSE(\hat{\beta}^{(k)}, P_{\mathcal{X}\times\mathcal{Y}}^{(k)}) \xrightarrow{p} 0$$

as $\tilde{n}_k \to \infty$.

Proof. The out-of-sample error of $\hat{\beta}^{(k)}$ on $P_{X\times Y}^{(k)}$ can be rewritten as

$$\begin{aligned}\text{OSE}(\hat{\beta}^{(k)}, P_{X\times Y}^k) &= \sigma^2 + \sigma^2 tr(E[\tilde{x}^{(k)}\tilde{x}^{(k)\top}]E[(X^{(k)\top}X^{(k)})^{-1}]) \\ &= \sigma^2 + \sigma^2 tr(Var(\tilde{x}^{(k)})\Omega_k) + \sigma^2 tr(E[\tilde{x}^{(k)}]E[\tilde{x}^{(k)}]^\top\Omega_k),\end{aligned}$$

where $\tilde{x}^{(k)}$ denotes an out-of-sample from the marginal distribution $P_{\mathcal{X}}^{(k)}$. Note that $\text{OSE}(\hat{\beta}^{(k)}, P_{X\times Y}^k)$ is differentiable with respect to $E[\tilde{x}^{(k)}]$ and $\frac{d\text{OSE}(\hat{\beta}^{(k)}, P_{X\times Y}^k)}{dE[\tilde{x}^{(k)}]} = 2\sigma^2\Omega_k E[\tilde{x}^{(k)}] \neq 0$.

The estimated OSE built upon the out-of-samples $\{(\tilde{x}_i^{(k)}, \tilde{y}_i^{(k)})\}_{i=1}^{\tilde{n}_k}$ from D_k^{out} is $\widehat{\text{OSE}}(\hat{\beta}^{(k)}, D_k^{out}) = \frac{1}{\tilde{n}_k}\sum_{i=1}^{\tilde{n}_k}(\tilde{y}_i^{(k)} - \tilde{x}_i^{(k)}\hat{\beta}^{(k)})^2$. Let us choose $\tilde{n}_k = n_k$. By the central limit theorem, $\sqrt{n_k}(\widehat{\text{OSE}}(\hat{\beta}^{(k)}, D_k^{out}) - \text{OSE}(\widehat{\beta^{(k)}}, P_{\mathcal{X}\times\mathcal{Y}}^{(k)})) \xrightarrow{d} N(0,1)$ as $n_k \to \infty$, hence $\widehat{\text{OSE}}(\hat{\beta}^{(k)}, D_k^{out}) - \text{OSE}(\widehat{\beta^{(k)}}, P_{\mathcal{X}\times\mathcal{Y}}^{(k)}) \xrightarrow{p} 0$.

The estimated OSE on the bootstraps $\{\{(x_{i,m}^*, y_{i,m}^*)\}_{i=1}^{\tilde{n}_k}\}_{m=1}^M$ from D_k^{out} is $\widehat{\text{OSE}}_M(\hat{\beta}^{(k)}, D_k^{out}) = \frac{1}{M}\sum_{m=1}^M \frac{1}{\tilde{n}_k}\sum_{i=1}^{\tilde{n}_k}(y_{i,m}^* - x_{i,m}^{*\top}\beta^{(k)})^2$. By Theorem 5, $\sqrt{\tilde{n}_k}\left(\widehat{\text{OSE}}_M(\hat{\beta}^{(k)}, D_k^{out}) - \widehat{\text{OSE}}(\hat{\beta}^{(k)}, D_k^{out})\right)$ and $\sqrt{n_k}(\widehat{\text{OSE}}(\hat{\beta}^{(k)}, D_k^{out}) - \text{OSE}(\widehat{\beta^{(k)}}, P_{\mathcal{X}\times\mathcal{Y}}^{(k)}))$ asymptotically have the same distribution almost surely as $\tilde{n}_k \to \infty$. It follows that $\widehat{\text{OSE}}_M(\hat{\beta}^{(k)}, D_k^{out}) - \text{OSE}(\hat{\beta}^{(k)}, P_{\mathcal{X}\times\mathcal{Y}}^{(k)}) \xrightarrow{p} 0$. □

Additionally, Theorem 6 establishes the consistency of the estimated out-of-sample error of $\hat{\beta}^{(c)}$.

Theorem 6. *Under Assumption 1 and 2,*

$$\widehat{OSE}_M(\hat{\beta}^{(c)}, D_k^{out}) - OSE(\hat{\beta}^{(c)}, P_{\mathcal{X}\times\mathcal{Y}}^{(k)}) \xrightarrow{p} 0$$

as $\tilde{n}_k \to \infty$.

Proof. The out-of-sample error of $\hat{\beta}^{(c)}$ on $P^{(k)}_{\mathcal{X}\times\mathcal{Y}}$ can be rewritten as

$$\begin{aligned}
\text{OSE}(\hat{\beta}^{(k)}, P^{k}_{X\times Y}) &= \sigma^2 + \sigma^2 tr(E[\tilde{x}^{(k)}\tilde{x}^{(k)\top}]E[(X^{(c)\top}X^{(c)})^{-1}]) + (\beta^{(1)} - \beta^{(2)})^\top E[X^{(3-k)\top}X^{(3-k)}(X^{(c)\top}X^{(c)})^{-1} \\
&\quad \times E[\tilde{x}^{(k)}\tilde{x}^{(k)\top}](X^{(c)\top}X^{(c)})^{-1}X^{(3-k)\top}X^{(3-k)}](\beta^{(1)} - \beta^{(2)}) \\
&= \sigma^2 + \sigma^2 tr(Var(\tilde{x}^{(k)})\Omega_k) + \sigma^2 E[\tilde{x}^{(k)}]^\top \Omega_k E[\tilde{x}^{(k)}] + tr(E[\tilde{x}^{(k)}\tilde{x}^{(k)\top}]E[(X^{(c)\top}X^{(c)})^{-1} \\
&\quad \times X^{(3-k)\top}X^{(3-k)}(\beta^{(1)} - \beta^{(2)})(\beta^{(1)} - \beta^{(2)})^\top X^{(3-k)\top}X^{(3-k)}(X^{(c)\top}X^{(c)})^{-1}]).
\end{aligned}$$

As

$$\begin{aligned}
(X^{(c)\top}X^{(c)})^{-1}X^{(3-k)\top}X^{(3-k)}(\beta^{(3-k)} - \beta^{(k)}) &= \sum_{j=1}^{2}(X^{(c)\top}X^{(c)})^{-1}X^{(j)\top}X^{(j)}\beta^{(j)} - \beta^{(k)} \\
&= E[\hat{\beta}^{(c)}|X^{(c)}] - \beta^{(k)} \\
&= b_k,
\end{aligned}$$

the above expression becomes

$$\text{OSE}(\hat{\beta}^{(k)}, P^{k}_{X\times Y}) = \sigma^2 + \sigma^2 tr(Var(\tilde{x}^{(k)})\Omega_k) + \sigma^2 E[\tilde{x}^{(k)}]^\top \Omega_k E[\tilde{x}^{(k)}] + tr(Var(\tilde{x}^{(k)})\Lambda_k) + E[\tilde{x}^{(k)}]^\top \Lambda_k E[\tilde{x}^{(k)}].$$

Lastly, $\text{OSE}(\hat{\beta}^{(k)}, P^{k}_{X\times Y})$ is differentiable with respect to $E[\tilde{x}^{(k)}]$ and $\frac{d\text{OSE}(\hat{\beta}^{(k)}, P^{k}_{X\times Y})}{dE[\tilde{x}^{(k)}]} = 2\sigma^2\Omega_k E[\tilde{x}^{(k)}] + 2\Lambda_k E[\tilde{x}^{(k)}] \neq 0$. By the same reasoning as in Theorem 4, it can be shown that $\widehat{\text{OSE}}_M(\hat{\beta}^{(c)}, D^{out}_k) - \text{OSE}(\hat{\beta}^{(c)}, P^{(k)}_{\mathcal{X}\times\mathcal{Y}}) \xrightarrow{p} 0$. □

B.2 CLASSIFICATION

In contrast to the regression where the out-of-sample error could be precisely computed, estimators for classification usually do not have an explicit form, let alone the loss. Instead of attempting to find the exact error, we bound the out-of-sample error for classification with high probability. Based on the bounds, we propose an equivalent condition under which the error bound decreases when datasets are merged.

In this section, we consider the problem of combining two datasets on a classification task with a focus on neural networks. A classification dataset is composed of a feature of a data x and the corresponding class label y. For a new data with an unknown label, the goal of classification is to correctly predict the label. It is common in practice that multiple datasets are available to train a model, but is not immediately clear how to leverage the datasets to their full potential. The natural question that arises is that when the datasets can be safely combined.

A criterion for deciding when to merge datasets with theoretical support would be highly appreciated as there is a tradeoff between generalization and approximation. For a brief illustration, suppose two datasets are given. On one hand, regarding all the data of the two datasets as observations from the same distribution would bring one large dataset, which is a union of the two datasets. A single model will be trained on the combined dataset, and this same model will make predictions on both of the datasets. The increased number of samples will reduce generalization error and help the model approach the best model among all the candidates. However, approximation error could be large as the same model is fitted to both of the datasets, hence restricting the model flexibility or expressivity. This tradeoff might be detrimental especially if the two datasets were sampled from significantly heterogeneous sources. On the other hand, training two models separately on each dataset would result in models more tailored to individual datasets. Nevertheless, the models would suffer from comparatively larger generalization error due to smaller number of samples, rendering the models to be far from optimal.

To capture and better understand what happens in common across multiple neural networks for classification tasks, we employ linear models in our theoretical analysis. This choice of model is motivated from the commonalities among numerous neural networks. Modern neural networks have assorted architectures, ranging from convolutional neural networks to Transformers. Their variation can be as simple as just having different size of dimensions or the number of layers, or can be as complicated as using different type of layers and connections. While there are countless number of possible designs, one thing in common among most networks for classification is that they have a linear layer at the end. Output of the linear layer followed by a softmax function represents a predicted probability distribution over classes. In order to analyze various neural networks in a more abstract setting, we focus on classifcation with linear models.

Our main result is as follows. We propose and prove error bounds of the models on individual datasets and combined dataset. Then we present a theorem that provides an equivalence condition under which the error bound is minimized if the datasets are combined. The implication of the theorem is that combining the datasets culminates in reduced error bound if the underlying distributions of the two datasets are close to each other. This interpretation is consistent with what is intuitively expected about the problem. If the datasets have similar characteristics or they are generated from similar distributions, merging them would definitely be beneficial as the model could enjoy the enlarged size of the dataset.

B.2.1 Error Bound on a Single Dataset

Let us formulate the problem more formally. Let $\mathcal{X}$ be a feature space and $\mathcal{Y} = \{1, \cdots, C\}$ be a set of all possible categories. Suppose a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ is given. A family of linear models $\beta = \{\beta_c\}_{c=1}^C$ takes as input the feature x_i and generates as output a probability distribution $softmax(\beta_1^\top x_i, \cdots, \beta_C^\top x_i)$. The model is trained with cross entropy loss, which is a common practice in classification tasks. The cross entropy loss of a model $\hat{\beta}$ augmented by regularization is defined as

$$l_\lambda(\hat{\beta}; x, y) = \sum_{c=1}^{C} -y_c \log \hat{y}_c + \frac{\lambda}{2} \sum_{c=1}^{C} \|\hat{\beta}_c\|^2,$$

where $\hat{y}_c$ denotes the predicted probability of the class c. We slightly abuse the notation and view y as either an integer or an one hot vector indicating a class when there is no confusion. Specifically, if the class of a data is c, then y could be either c or one hot vector with only the cth entry being 1, depending on the context. For the purpose of theoretical analysis, we make separability assumption of the data.

Assumption 3. *There exists a margin $\gamma_0 > 0$ such that for all data (x_i, y_i), if the label of y_i is c_i, then $\beta_{c_i}^\top x_i < 0$ and $\beta_c^\top x_i > \gamma_0$ for $\forall c \neq c_i$. In other words,*

$$y_i = \frac{1}{2} \left(\mathbf{1}_C - [sign(\beta_1^\top x_i), \cdots, sign(\beta_C^\top x)]^\top \right)$$

where $\mathbf{1}_C \in \mathbb{R}^C$ denotes a column vector with all entries being 1.

Let us concatenate all the true parameters as $\beta = [\beta_1^\top, \cdots, \beta_C^\top]^\top$. A family of linear models $\hat{\beta} = [\hat{\beta}_1^\top, \cdots, \hat{\beta}_C^\top]^\top$ is trained to estimate the true value of β and forecast label of a new data. Total C linear models, followed by a softmax function, predicts the probability of the data belonging to each category.

We also introduce a surrogate loss to approximate the cross entropy loss. We define the surrogate loss as

$$l_{\lambda,\gamma}(\hat{\beta}, \beta; x) = \sum_{c=1}^{C} \|\beta_c - \hat{\beta}_c\| \|x\| + \sum_{c=1}^{C} -\beta_c^\top x g_\gamma(\beta_c^\top x) + \log \sum_{c=1}^{C} e^{\hat{\beta}_c^\top x} + \frac{\lambda}{2} \sum_{c=1}^{C} \|\hat{\beta}_c\|^2,$$

where $g_\gamma(t) = \mathbb{1}_{(-\infty,0)} + \mathbb{1}_{[0,\gamma]}(1 - \frac{t}{\gamma})$ is the well known ramp loss. The surrogate loss $l_{\lambda,\gamma}$ serves as a continuous bound of the cross entropy with appropriate choice of γ. Namely,

$$l_\lambda(\hat{\beta}; x, y) \leq l_{\lambda,\gamma}(\hat{\beta}, \beta; x)$$

if $(x, y) \sim \mathcal{M}(\beta)$ and $\gamma \leq \gamma_0$. The corresponding empirical loss and population loss are defined as

$$\hat{L}_{\lambda,\gamma}(\hat{\beta}, \beta; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} l_{\lambda,\gamma}(\hat{\beta}, \beta; x_i),$$
$$L_{\lambda,\gamma}(\hat{\beta}, \beta) = E_{\tilde{x}}[l_{\lambda,\gamma}(\hat{\beta}, \beta; \tilde{x})],$$

respectively. Population loss of l_λ is defined as

$$L_\lambda(\hat{\beta}, \beta) = E_{\tilde{x},\tilde{y}}[l_\lambda(\hat{\beta}; \tilde{x}, \tilde{y})].$$

It suffices to bound $L_{\lambda,\gamma}$ in order to find an upper bound of the performance L_λ of a model.

Given two datasets $\mathcal{D}_1 = \{(x_i^{(1)}, y_i^{(1)})\}_{i=1}^{n_1}$ and $\mathcal{D}_2 = \{(x_i^{(2)}, y_i^{(2)})\}_{i=1}^{n_2}$, our goal is to determine whether training a single family of linear models on the joint dataset would be beneficial, compared to fitting two families separately on each dataset.

Namely, we would like to compare the performance of a model $\hat{\beta}^{(m)}$ trained on a single dataset $\mathcal{D}_m$ for $m = 1, 2$ and a model $\hat{\beta}^{(1,2)}$ trained on the joint dataset $\mathcal{D}_1 \cup \mathcal{D}_2$.

It is desirable to determine whether to combine the datasets by comparing the performance of the best model based on the data. Let us denote the minimizers of the empirical loss and the population loss as

$$\hat{\beta}^* = \underset{\hat{\beta}}{\text{argmin}}\ \hat{L}_{\lambda,\gamma}(\hat{\beta}, \beta; \mathcal{D}),$$
$$\beta^* = \underset{\hat{\beta}}{\text{argmin}}\ L_{\lambda,\gamma}(\hat{\beta}, \beta),$$

respectively, where β is the true parameter generating $\mathcal{D}$. Note that such minimizers exist as the loss is nonnegative and convex, and are unique due to the regularization. Unfortunately, finding the best models precisely is intractable as the minimization problems do not have closed form solutions, opposed to the regression problem. In practice, gradient descent is one of the popular choice to approximate the optimizer. As the surrogate loss is subdifferentiable, we employ subgradient method to estimate the optimal solution. Starting with an initial estimate $\hat{\beta}^{[1]}$, the estimator is iteratively updated by

$$\hat{\beta}^{[k+1]} = \hat{\beta}^{[k]} - \eta^{[k]} g^{[k]},$$

where $\eta^{[k]}$ is the kth step size and $g^{[k]} \in \partial L_{\lambda,\gamma}(\hat{\beta}^{[k-1]}, \beta; \mathcal{D})$ is a subgradient. Instead of analyzing the performance of the empirical risk minimizer $L_\lambda(\hat{\beta}^*, \beta)$, we focus on that of the subgradient method estimator $L_\lambda(\hat{\beta}^{[k]}, \beta)$.

Now that the problem is well formulated, let us examine error bounds of each model. We first scrutinize excess risk of an estimator with high probability, which involves Rademacher complexity. We further bound the Rademacher complexity of the hypothesis class of interest. The subgradient estimator will then be compared with empirical risk minimizer. Joining all the results leads to complete analysis of the model performance. We start by making an additional assumption of boundedness of the feature.

Assumption 4. *For all data (x_i, y_i), the features are bounded by $\|x_i\| \leq B$.*

While there is no explicit restriction to the scope of the hypothesis class, candidate parameters naturally satisfy

$$\|\hat{\beta}_c\| \leq \sqrt{\frac{2 \log C}{\lambda}}$$

for all $c = 1, \cdots, C$ due to the regularization. This boundedness can be easily confirmed. Suppose $\|\hat{\beta}_{c_0}\| > \sqrt{\frac{2 \log C}{\lambda}}$ for some c_0. Then $l_\lambda(\hat{\beta}) \geq \frac{\lambda}{2}\|\hat{\beta}_{c_0}\|^2 \geq \log C = l_\lambda(0)$, hence $\hat{\beta}$ cannot minimize the loss. Henceforth, we implicitly assume that $\hat{\beta}_c$ is bounded.

We first scrutinize the error on a single dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where $(x_i, y_i) \sim \mathcal{M}(\beta)$. Under the assumptions, we can bound the difference between the empirical loss and the population loss of an estimator. Constructing a bound requires McDiarmid's inequality, which is stated below.

Lemma 7. *[McDiarmid, 1989] Let $f : \mathbb{R}^m \to \mathbb{R}$ be a function of bounded difference, that is, $|f(z_1, \cdots, z_m) - f(z_1, \cdots, z_i', \cdots, z_m)| \leq c_i$ for $\forall i, z_1, \cdots, z_m, z_i'$. Let $Z_1, \cdots, Z_m$ be independent random variables. Then*

$$\mathbb{P}(f(Z_1, \cdots, Z_m) - E[f(Z_1, \cdots, Z_m)] \geq \epsilon) \leq e^{\frac{-2\epsilon^2}{\sum_{i=1}^n c_i^2}}$$

for all $\epsilon > 0$.

We can now analyze the excess risk of an estimator. The following proposition proposes an error bound, the proof of which uses ordinary arguments in generalization bounds. See Bartlett and Mendelson [2002] for example.

Proposition 2. *If the dataset $\mathcal{D}$ is generated by $(x_i, y_i) \sim \mathcal{M}(\beta)$, then the difference between the empirical loss and the population loss is bounded by*

$$\hat{L}_{\lambda,\gamma}(\hat{\beta}, \beta; \mathcal{D}) - L_{\lambda,\gamma}(\hat{\beta}, \beta) \leq \sqrt{\frac{1}{2n} \log \frac{1}{\delta}} \omega(\beta) + 2R_n(l_{\lambda,\gamma})$$

for all $\hat{\beta}$ with probability at least $1-\delta$, where $\omega(\beta) = O(\|\beta\|)$,

$$R_n(l_{\lambda,\gamma}) = E\left[\sup_{\hat{\beta}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i l_{\lambda,\gamma}(\hat{\beta},\beta;x_i)\right]$$

is Rademacher complexity, and σ_i is Rademacher random variable.

Proof. Let $\omega(\beta) = 2CB\sqrt{\frac{\log C}{\lambda}} + 2B\sum_{c=1}^{C}\|\beta_c\| + (C+1)\log C$. It is straightforward that $l_{\lambda,\gamma}(\hat{\beta},\beta;x_i) \le \omega(\beta)$. Let $f(\mathcal{D}) = \sup_{\hat{\beta}}\left\{\hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}) - L_{\lambda,\gamma}(\hat{\beta},\beta)\right\}$. Then

$$\begin{aligned}|f(\mathcal{D}) - f_m(x_1,y_1,\cdots,x_i',y_i',\cdots,x_n,y_n)| \le \Bigg| &\sup_{\hat{\beta}}\left\{\hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}) - L_{\lambda,\gamma}(\hat{\beta},\beta)\right\} \\ &- \sup_{\hat{\beta}}\left\{\hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}) - L_{\lambda,\gamma}(\hat{\beta},\beta) + \frac{1}{n}l_{\lambda,\gamma}(\hat{\beta},\beta;x_i') - \frac{1}{n}l_{\lambda,\gamma}(\hat{\beta},\beta;x_i')\right\}\Bigg| \\ \le\ & \frac{\omega(\beta)}{n}.\end{aligned}$$

Hence

$$\mathbb{P}(f(\mathcal{D}) - E[f(\mathcal{D})] \ge \epsilon) \le e^{-\frac{2n}{\omega(\beta)^2}\epsilon^2}$$

for all $\epsilon > 0$. Moreover, the expectation can be bounded by the Rademacher complexity. Let $\mathcal{D}' = \{(x_i', y_i')\}_{i=1}^{n}$ be a set of independent copies of each (x_i, y_i). Then

$$\begin{aligned}E[f(\mathcal{D})] &= E\left[\sup_{\hat{\beta}}\left\{\hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}) - L_{\lambda,\gamma}(\hat{\beta},\beta)\right\}\right] \\ &= E\left[\sup_{\hat{\beta}}\left\{\hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}) - E[\hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}')]\right\}\right] \\ &= E\left[\sup_{\hat{\beta}} E\left[\hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}) - \hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}')|\mathcal{D}\right]\right] \\ &\le E\left[\sup_{\hat{\beta}}\left\{\hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}) - \hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}')\right\}\right] \\ &= E\left[\sup_{\hat{\beta}}\frac{1}{n}\sum_{i=1}^{n}\left\{l_{\lambda,\gamma}(\hat{\beta},\beta;x_i) - l_{\lambda,\gamma}(\hat{\beta},\beta;x_i')\right\}\right] \\ &= E\left[\sup_{\hat{\beta}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i\left\{l_{\lambda,\gamma}(\hat{\beta},\beta;x_i) - l_{\lambda,\gamma}(\hat{\beta},\beta;x_i')\right\}\right] \\ &\le E\left[\sup_{\hat{\beta}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i l_{\lambda,\gamma}(\hat{\beta},\beta;x_i) + \sup_{\hat{\beta}}\frac{1}{n}\sum_{i=1}^{n} -\sigma_i l_{\lambda,\gamma}(\hat{\beta},\beta;x_i')\right] \\ &= 2R_n(l_{\lambda,\gamma}).\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{P}\left(f(\mathcal{D}) \ge \epsilon\right) &\le e^{-\frac{2n}{\omega(\beta)^2}(\epsilon - E[f(\mathcal{D})])^2} \\ &\le e^{-\frac{2n}{\omega(\beta)^2}(\epsilon - 2R_n(l_{\lambda,\gamma}))^2}.\end{aligned}$$

Setting $\delta = e^{-\frac{2n}{\omega(\beta)^2}(\epsilon - 2R_n(l_{\lambda,\gamma}))^2}$ gives the result. □

Similar result on the difference between the population loss and the empirical loss can be obtained by the same argument. We state the result and omit the proof.

Proposition 3. *Under the same assumption of Proposition 2, the difference between the population loss and the empirical loss is bounded by*

$$L_{\lambda,\gamma}(\hat{\beta},\beta) - \hat{L}_{\lambda,\gamma}(\hat{\beta},\beta;\mathcal{D}) \leq \sqrt{\frac{1}{2n}\log\frac{1}{\delta}\omega(\beta)} + 2R_n(l_{\lambda,\gamma})$$

for all $\hat{\beta}$ with probability at least $1-\delta$.

Previous lemmas bound the difference between $L_{\lambda,\gamma}$ and $\hat{L}_{\lambda,\gamma}$ with high probability using the Rademacher complexity. While the bounds provide some insight into the excess risk, it would give rise to better comprehension if the Rademacher complexity is replaced by more interpretable terms. To this end, we further bound the Rademacher complexity. The following two lemmas will be useful when we analyze the Rademacher complexity.

Lemma 8. *For $n \in \mathbb{N}$,*

$$\frac{1}{2^n}\sum_{k=0}^{[\frac{n}{2}]}(n-2k)\binom{n}{k} \leq \sqrt{\frac{3n}{4\pi}}.$$

Proof. If n is even,

$$\begin{aligned}
\frac{1}{2^n}\sum_{k=0}^{[\frac{n}{2}]}(n-2k)\binom{n}{k} &= \frac{n}{2^n}\sum_{k=0}^{\frac{n}{2}}\binom{n}{k} - \frac{n}{2^{n-1}}\sum_{k=1}^{\frac{n}{2}}\binom{n-1}{k-1} \\
&= \frac{n}{2^n}\left(\frac{2^n-\binom{n}{\frac{n}{2}}}{2} + \binom{n}{\frac{n}{2}}\right) - \frac{n}{2^{n-1}}2^{n-2} \\
&= \frac{n}{2^{n+1}}\binom{n}{\frac{n}{2}} \\
&\leq \frac{n}{2^{n+1}}\frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}}}{2\pi\frac{n}{2}\left(\frac{n}{2e}\right)^n e^{\frac{2}{6n+1}}} \\
&\leq \sqrt{\frac{n}{2\pi}}.
\end{aligned}$$

If n is odd,

$$\begin{aligned}
\frac{1}{2^n}\sum_{k=0}^{[\frac{n}{2}]}(n-2k)\binom{n}{k} &= \frac{n}{2^n}\sum_{k=0}^{\frac{n-1}{2}}\binom{n}{k} - \frac{n}{2^{n-1}}\sum_{k=1}^{\frac{n-1}{2}}\binom{n-1}{k-1} \\
&= \frac{n}{2^n}\frac{2^n}{2} - \frac{n}{2^{n-1}}\left(\frac{2^{n-1}-\binom{n-1}{\frac{n-1}{2}}}{2}\right) \\
&= \frac{n}{2^n}\binom{n-1}{\frac{n-1}{2}} \\
&\leq \frac{n}{2^n}\frac{\sqrt{2\pi(n-1)}\left(\frac{n-1}{e}\right)^{n-1} e^{\frac{1}{12n-12}}}{2\pi\frac{n-1}{2}\left(\frac{n-1}{2e}\right)^{n-1} e^{\frac{2}{6n-5}}} \\
&\leq \sqrt{\frac{3n}{4\pi}}.
\end{aligned}$$

□

The proof of the following lemma is motivated by Ledoux and Talagrand [2011].

Lemma 9. *Let $f : \mathbb{R}^m \to \mathbb{R}$ be an L-Lipschitz function, $g_i : \mathbb{R} \to \mathbb{R}$ for $i = 1, \cdots, n$ where $n \geq 0$, and $h : \mathbb{R}^m \to \mathbb{R}$. Then*

$$E\left[\sup_{\theta_1,\cdots,\theta_m} h(\theta_1,\cdots,\theta_m) + \sum_{i=1}^n \sigma_i f(g_i(\theta_1),\cdots,g_i(\theta_m))\right] \leq E\left[\sup_{\theta_1,\cdots,\theta_m} h(\theta_1,\cdots,\theta_m) + L\sum_{i=1}^n \sigma_i \sum_{j=1}^m g_i(\theta_j)\right]$$

Proof. For $n = 0$, the inequality holds. Assume that the inequality holds for $n-1$. Then

$$\begin{aligned}
&E\left[\sup_{\theta_1,\cdots,\theta_m} \sum_{i=1}^n h(\theta_1,\cdots,\theta_m) + \sigma_i f(g_i(\theta_1),\cdots,g_i(\theta_m))\right] \\
&= \frac{1}{2}E\left[\sup_{\theta_1,\cdots,\theta_m} h(\theta_1,\cdots,\theta_m) + \sum_{i=1}^{n-1} \sigma_i f(g_i(\theta_1),\cdots,g_i(\theta_m)) + f(g_n(\theta_1),\cdots,g_n(\theta_m))\right] \\
&+ \frac{1}{2}E\left[\sup_{\theta_1,\cdots,\theta_m} h(\theta_1,\cdots,\theta_m) + \sum_{i=1}^{n-1} \sigma_i f(g_i(\theta_1),\cdots,g_i(\theta_m)) - f(g_n(\theta_1),\cdots,g_n(\theta_m))\right] \\
&= \frac{1}{2}E\Bigg[\sup_{\theta_1,\cdots,\theta_m,\theta'_1,\cdots,\theta'_m} h(\theta_1,\cdots,\theta_m) + h(\theta'_1,\cdots,\theta'_m) + \sum_{i=1}^{n-1} \sigma_i \left\{f(g_i(\theta_1),\cdots,g_i(\theta_m)) + f(g_i(\theta'_1),\cdots,g_i(\theta'_m))\right\} \\
&\quad + |f(g_n(\theta_1),\cdots,g_n(\theta_m)) - f(g_n(\theta'_1),\cdots,g_n(\theta'_m))|\Bigg] \\
&\leq \frac{1}{2}E\Bigg[\sup_{\theta_1,\cdots,\theta_m,\theta'_1,\cdots,\theta'_m} h(\theta_1,\cdots,\theta_m) + h(\theta'_1,\cdots,\theta'_m) + \sum_{i=1}^{n-1} \sigma_i \left\{f(g_i(\theta_1),\cdots,g_i(\theta_m)) + f(g_i(\theta'_1),\cdots,g_i(\theta'_m))\right\} \\
&\quad + L\left\|\begin{bmatrix} g_n(\theta_1) \\ \vdots \\ g_n(\theta_m)\end{bmatrix} - \begin{bmatrix} g_n(\theta'_1) \\ \vdots \\ g_n(\theta'_m)\end{bmatrix}\right\|\Bigg] \\
&\leq \frac{1}{2}E\Bigg[\sup_{\theta_1,\cdots,\theta_m,\theta'_1,\cdots,\theta'_m} h(\theta_1,\cdots,\theta_m) + h(\theta'_1,\cdots,\theta'_m) + \sum_{i=1}^{n-1} \sigma_i \left\{f(g_i(\theta_1),\cdots,g_i(\theta_m)) + f(g_i(\theta'_1),\cdots,g_i(\theta'_m))\right\} \\
&\quad + L\sum_{j=1}^m |g_n(\theta_j) - g_n(\theta'_j)|\Bigg] \\
&= E\left[\sup_{\theta_1,\cdots,\theta_m} h(\theta_1,\cdots,\theta_m) + \sum_{i=1}^{n-1} \sigma_i f(g_i(\theta_1),\cdots,g_i(\theta_m)) + L\sigma_n \sum_{j=1}^m g_n(\theta_j)\right] \\
&\leq E\left[\sup_{\theta_1,\cdots,\theta_m} h(\theta_1,\cdots,\theta_m) + L\sum_{i=1}^n \sigma_i \sum_{j=1}^m g_i(\theta_j)\right] \text{ by induction hypothesis.}
\end{aligned}$$

Hence the inequality holds for all $n \geq 0$. □

Now we can bound the Rademacher complexity of the hypothesis class of interest with the help of the preceding lemmas. Let us define $|||\beta||| = \sum_{c=1}^C \|\beta_c\|$ to further simplify the notation.

Proposition 4. *The Rademacher complexity is bounded by*

$$R_n(l_{\lambda,\gamma}) \leq \frac{1}{2}\sqrt{\frac{3}{\pi n}}B\left(\sqrt{\frac{2\log C}{\lambda}}C + |||\beta|||\right) + \frac{BC}{\sqrt{n}}\sqrt{\frac{2\log C}{\lambda}} + \frac{1}{2}\sqrt{\frac{3}{\pi n}}\log C$$

Proof. By definition,

$$
\begin{aligned}
R_n(l_{\lambda,\gamma}) &= E[\sup_{\hat\beta} \frac{1}{n}\sum_{i=1}^{n} \sigma_i l_{\lambda,\gamma}(\hat\beta, \beta; x_i)] \\
&= E\left[\sup_{\hat\beta} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \left\{ \sum_{c=1}^{C} \|\beta_c - \hat\beta_c\|\|x_i\| + \sum_{c=1}^{C} -\beta_c^\top x_i g_\gamma(\beta_c^\top x_i) + \log\sum_{c=1}^{C} e^{\hat\beta_c^\top x_i} + \frac{\lambda}{2}\sum_{c=1}^{C}\|\hat\beta_c\|^2 \right\}\right] \\
&\leq E\left[\sup_{\hat\beta} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \sum_{c=1}^{C} \|\beta_c - \hat\beta_c\|\|x_i\|\right] + E\left[\sup_{\hat\beta} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \sum_{c=1}^{C} -\beta_c^\top x_i g_\gamma(\beta_c^\top x_i)\right] + E\left[\sup_{\hat\beta} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \log\sum_{c=1}^{C} e^{\hat\beta_c^\top x_i}\right] \\
&\quad + E\left[\sup_{\hat\beta} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \frac{\lambda}{2}\sum_{c=1}^{C}\|\hat\beta_c\|^2\right].
\end{aligned}
$$

The first term can be bounded by

$$
\begin{aligned}
E\left[\sup_{\hat\beta} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \sum_{c=1}^{C} \|\beta_c - \hat\beta_c\|\|x_i\|\right] &\leq E\left[\sup_{\hat\beta} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \sum_{c=1}^{C} B\|\beta_c - \hat\beta_c\|\right] \\
&= E\left[\frac{B}{n}\sum_{i=1}^{n} \sigma_i \sum_{c=1}^{C} \left(\|\beta_c\| + \sqrt{\frac{2\log C}{\lambda}}\right) \mathbb{1}_{\sum_{i=1}^{n}\sigma_i > 0}\right] \\
&= \frac{B}{n}\left(C\sqrt{\frac{2\log C}{\lambda}} + \sum_{c=1}^{C}\|\beta_c\|\right)\frac{1}{2^n}\sum_{k=0}^{\left[\frac{n}{2}\right]}(n-2k)\binom{n}{k} \\
&\leq \sqrt{\frac{3}{4\pi n}} B\left(C\sqrt{\frac{2\log C}{\lambda}} + \sum_{c=1}^{C}\|\beta_c\|\right)
\end{aligned}
$$

The second term is 0 as it does not contain any $\hat\beta$ and all the x_i's and σ_i's are independent. Especially, σ_i is symmetric around 0.

As the log-sum-exp function is a contraction with respect to $\|\cdot\|_\infty$, the third term can be bounded by

$$
\begin{aligned}
E\left[\sup_{\hat\beta} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \log\sum_{c=1}^{C} e^{\hat\beta_c^\top x_i}\right] &\leq E\left[\sup_{\hat\beta} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \sum_{c=1}^{C} \hat\beta_c^\top x_i\right] \\
&= E\left[\sup_{\hat\beta} \frac{1}{n}\sum_{c=1}^{C} \hat\beta_c^\top \sum_{i=1}^{n} \sigma_i x_i\right] \\
&\leq E\left[\frac{1}{n}\sum_{c=1}^{C} \sqrt{\frac{2\log C}{\lambda}} \|\sum_{i=1}^{n}\sigma_i x_i\|\right] \\
&\leq \frac{1}{n}\sqrt{\frac{2\log C}{\lambda}} \sum_{c=1}^{C} \sqrt{E\left[\|\sum_{i=1}^{n}\sigma_i x_i\|^2\right]} \\
&= \frac{1}{n}\sqrt{\frac{2\log C}{\lambda}} \sum_{c=1}^{C} \sqrt{E\left[\sum_{i=1}^{n}\|x_i\|^2\right]} \\
&\leq \frac{1}{\sqrt{n}}\sqrt{\frac{2\log C}{\lambda}} CB \text{ using lemma 9.}
\end{aligned}
$$

The last term can be bounded by

$$\begin{aligned} E\left[\sup_{\hat{\beta}} \frac{1}{n}\sum_{i=1}^{n}\sigma_i \frac{\lambda}{2}\sum_{c=1}^{C}\|\hat{\beta}_c\|^2\right] &= E\left[\frac{\lambda}{2n}C\frac{2\log C}{\lambda}\mathbb{1}_{\sum_{i=1}^{n}\sigma_i>0}\sum_{i=1}^{n}\sigma_i\right] \\ &= \frac{C\log C}{n}\sum_{k=0}^{\left[\frac{n}{2}\right]}(n-2k)\frac{\binom{n}{k}}{2^n} \\ &\le \frac{C\log C}{n}\sqrt{\frac{3n}{4\pi}} \text{ by lemma 8.} \end{aligned}$$

Combining all the bounds gives the desired result. □

As explained before, estimators are obtained through subgradient method since direct optimization of the loss is intractable. The following lemma provides a guarantee that the lowest loss among all the updates is close to the optimal loss under suitable conditions.

Lemma 10. *[Shor, 1985] Let $f : \mathbb{R}^m \to \mathbb{R}$ be subdifferentiable with bounded subgradients such that $\|g\| \le G$ for all $g \in \partial f(x)$. Let $x^{[k+1]} = x^{[k]} - \eta^{[k]}g^{[k]}$ where $g^{[k]} \in \partial f(x^{[k]})$ and $\eta^{[k]}$ is a step size at the kth step. If $f(x^*) = \min f(x)$, then*

$$\min\{f(x^{[k]})|k = 1, \cdots, K\} - f(x^*) \le \frac{\|x^{[1]} - x^*\|^2 + G\sum_{k=1}^{K}\eta^{[k]2}}{2\sum_{k=1}^{K}\eta^{[k]}}.$$

The above lemma gives the following guarantee for subgradient method estimator for the surrogate loss, which is stated in the following proposition. For simplicity, we choose to use the same step size and the number of steps for all datasets, but it is also possible to consider different step sizes and the number of steps.

Proposition 5. *Let the subgradient method estimator on $\mathcal{D}$ be iteratively defined as $\hat{\beta}^{[k+1]} = \hat{\beta}^{[k]} - \eta^{[k]}g^{[k]}$ where $\hat{\beta}^{[1]}$ is an initial estimator, $\eta^{[k]}$ is the kth step size, and $g^{[k]} \in \partial\hat{L}_{\lambda,\gamma}(\hat{\beta}^{[k]}, \beta; \mathcal{D})$. Then*

$$\min\{\hat{L}_{\lambda,\gamma}(\hat{\beta}^{[k]}; \mathcal{D})|k = 1, \cdots, K\} - \hat{L}_{\lambda,\gamma}(\hat{\beta}^*; \mathcal{D}) \le \frac{8\log C + G\lambda\sum_{k=1}^{K}\eta^{[k]2}}{2\lambda\sum_{k=1}^{K}\eta^{[k]}}$$

where $G = 2CB + C\sqrt{2\lambda\log C}$.

Proof. Note that

$$l_{\lambda,\gamma}(\hat{\beta}, \beta; x) = \sum_{c=1}^{C}\|\beta_c - \hat{\beta}_c\|\|x\| + \sum_{c=1}^{C} -\beta_c^\top x g_\gamma(\beta_c^\top x) + \log\sum_{c=1}^{C} e^{\hat{\beta}_c^\top x} + \frac{\lambda}{2}\sum_{c=1}^{C}\|\hat{\beta}_c\|^2.$$

The term $\|\beta_c - \hat{\beta}_c\|$ is subdifferentiable with respect to $\hat{\beta}_c$ at $\hat{\beta}_c = \beta_c$ with subdifferential being a unit ball and differentiable elsewhere with $\nabla_{\hat{\beta}_c}\|\beta_c - \hat{\beta}_c\| = \frac{\beta_c - \hat{\beta}_c}{\|\hat{\beta}_c - \beta_c\|}$. The other terms are all differentiable with $\nabla_{\hat{\beta}_c}\log\sum_{c'=1}^{C} e^{\hat{\beta}_{c'}^\top x} = \frac{e^{\hat{\beta}_c^\top x}}{\sum_{c'=1}^{C} e^{\hat{\beta}_{c'}^\top x}}x$ and $\nabla_{\hat{\beta}_c}\frac{\lambda}{2}\sum_{c'=1}^{C}\|\hat{\beta}_{c'}\|^2 = \lambda\hat{\beta}_c$. As $\hat{L}_{\lambda,\gamma}(\hat{\beta}, \beta; \mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} l_{\lambda,\gamma}(\hat{\beta}, \beta; x_i)$, the subgradients of $\hat{L}_{\lambda,\gamma}(\hat{\beta}), \beta; \mathcal{D})$ are bounded by G. The result follows by applying the above lemma and recalling that both $\hat{\beta}^{[1]}$ and $\hat{\beta}^*$ are bounded. □

Integrating all the propositions leads to the main theorem, which guarantees the subgradient method estimator on a single dataset. In the following theorem, the error bound is split into several parts for better interpretation. Details will be elucidated after the proof.

Theorem 7. *Let $\hat{\beta}^{[k]}$ be the subgradient method estimator of $\hat{L}_{\lambda,\gamma}(\hat{\beta}, \beta; \mathcal{D})$. Let $s = \underset{k\le K}{argmin}\ \hat{L}_{\lambda,\gamma}(\hat{\beta}^{[k]}; \mathcal{D})$ denote the step that achieved the lowest loss during K subgradient method updates. Then*

$$L_\lambda(\hat{\beta}^{[s]}; \beta) \le \Phi(n, \beta; \delta) + L_{\lambda,\gamma}(\beta^*, \beta)$$

with probability at least $1 - 2\delta$ *where* $\Phi = O(\sqrt{\frac{1}{n}\log\frac{1}{\delta}})$ *and*

$$\beta^* = \underset{\hat{\beta}}{\textit{argmin}}\ L_{\lambda,\gamma}(\hat{\beta}, \beta)$$

denotes the minimizer for the population surrogate loss.

Proof. By the analysis on the excess risk in Proposition 2 and 3,

$$L_{\lambda,\gamma}(\hat{\beta}^{[s]}, \beta) - \hat{L}_{\lambda,\gamma}(\hat{\beta}^{[s]}, \beta; \mathcal{D}) \le \sqrt{\frac{1}{2n}\log\frac{1}{\delta}}\omega(\beta) + 2R_n(l_{\lambda,\gamma})$$

with probability at least $1 - \delta$ and

$$\hat{L}_{\lambda,\gamma}(\hat{\beta}^{*(m)}, \beta^{(m)}; \mathcal{D}^{(m)}) - L_{\lambda,\gamma}(\hat{\beta}^{*(m)}, \beta^{(m)}) \le \sqrt{\frac{1}{2n}\log\frac{1}{\delta}}\omega(\beta) + 2R_n(l_{\lambda,\gamma})$$

with probability at least $1 - \delta$. Also,

$$\hat{L}_{\lambda,\gamma}(\hat{\beta}^{[s]}; \mathcal{D}) - \hat{L}_{\lambda,\gamma}(\hat{\beta}^*; \mathcal{D}) \le \frac{8\log C + G\lambda\sum_{k=1}^{K}\eta^{[k]2}}{2\lambda\sum_{k=1}^{K}\eta^{[k]}}$$

by the subgradient method analysis in Proposition 5 and

$$\hat{L}_{\lambda,\gamma}(\hat{\beta}^*; \mathcal{D}) \le \hat{L}_{\lambda,\gamma}(\beta^*; \mathcal{D})$$

by the definition of $\hat{\beta}^*$. Adding all the inequalities leads to

$$L_{\lambda,\gamma}(\hat{\beta}^{[s]}, \beta) - L_{\lambda,\gamma}(\hat{\beta}^*, \beta) \le 2\sqrt{\frac{1}{2n}\log\frac{1}{\delta}}\omega(\beta) + 4R_n(l_{\lambda,\gamma}) + \frac{8\log C + G\lambda\sum_{k=1}^{K}\eta^{[k]2}}{2\lambda\sum_{k=1}^{K}\eta^{[k]}}$$

with probability at least $1 - 2\delta$. The proof concludes by recalling that $L_{\lambda,\gamma}$ serves as an upper bound of L_λ and applying the bound on the Rademacher complexity in Proposition 4. Specifically, Φ has the form of

$$\Phi(n, \beta; \delta) = \phi_1(n, \beta; \delta) + \phi_2(n, \beta) + \phi_3$$

where

$$\begin{aligned} \phi_1(n, \beta; \delta) &= \sqrt{\frac{2}{n}\log\frac{1}{\delta}}\omega(\beta), \\ \phi_2(n, \beta) &= \frac{a_1 + a_2|||\beta|||}{\sqrt{n}}, \\ \phi_3 &= \frac{8\log C + G\lambda\sum_{k=1}^{K}\eta^{[k]2}}{2\lambda\sum_{k=1}^{K}\eta^{[k]}} \end{aligned}$$

for constants $a_1 = \left(2\sqrt{\frac{6}{\pi}} + 4\sqrt{2}\right)\sqrt{\frac{\log C}{\lambda}}BC + 2\sqrt{\frac{3}{\pi}}C\log C$ and $a_2 = 2\sqrt{\frac{3}{\pi}}B$. □

The bound Φ on the excess risk is composed with three parts. The first part, denoted as ϕ_1 in the theorem, is the error stemming from the randomness of the samples. It is directly connected to the confidence level by being a function of δ. The second part, which is ϕ_2, is the error related to the complexity of the hypothesis class. The last part ϕ_3 is the error due to the subgradient method.

This theorem provides full analysis of the performance of the subgradient method estimator $L_\lambda(\hat{\beta}^{[s]}; \beta)$ in terms of the number of samples, distributional properties, and the step sizes of the subgradient method. Suppose the step size $\eta^{[k]}$ converges to 0 and $\sum_{k=1}^{\infty}\eta^{[k]} = \infty$. For instance, taking the step size to be $\eta^{[k]} = \frac{1}{k}$ satisfies both conditions. Then the error due to subgradient method converges to 0, implying that the subgradient method converges. Under this scenario, the error bound is related to the size of the dataset n_m. Especially, the theorem theoretically proves that the error bound decreases if there are more samples, as one might have intuitively expected.

The bound also depends on the size of the hypothesis class, which is controlled by the regularization coefficient λ. The familiy of candidate models grows larger as λ decreases. The effect of λ is hidden in the coefficients a_1 and a_3. For smaller λ, these coefficiets becomes larger, which in turn implies that the number of samples required to achieve the same error bound increases.

While the above theorem compares the performance of $\hat{\beta}^{[s](m)}$ with $\beta^{*(m)}$ on a single dataset $\mathcal{D}_m$, it could also be compared with

$$\beta^{*(1,2)} = \operatorname*{argmin}_{\hat{\beta}} \sum_{m=1}^{2} L_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)}),$$

the minimizer of the population surrogate loss on the combined datset. The proof uses the same line of reasoning. This another comparison will later be used to construct a criterion for merging the datasets.

B.2.2 Error Bound on the Combined Dataset

Performance guarantee on the combined dataset can be proved in a similar way. Most parts of the proof on the combined dataset are essentially identical to the proof for the single dataset. We therefore only demonstrate the parts where the reasoning deviates essentially from the previous proof. We begin by proposing another concentration inequality on the joint loss. Note that in the following proposition, the same model $\hat{\beta}$ is applied to both datasets, whereas there was one model per one dataset previously. We also assume that features on the both datasets are bounded by the same constant B, but our analysis could be easily extended to the case where the bounds are different on each dataset.

Proposition 6. *The difference between the empirical loss and the population loss on $\mathcal{D}_1 \cup \mathcal{D}_2$ is bounded by*

$$\sum_{m=1}^{2} \left\{ \hat{L}_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)}; \mathcal{D}_m) - L_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)}) \right\} \leq \sqrt{\frac{1}{2}\left(\frac{\omega(\beta^{(1)})^2}{n_1} + \frac{\omega(\beta^{(2)})^2}{n_2} \right) \log\frac{1}{\delta}} + 2R_{n_1,n_2}(l_{\lambda,\gamma})$$

for all $\hat{\beta}$ with probability at least $1-\delta$ where

$$R_{n_1,n_2}(l_{\lambda,\gamma}) = E\left[\sup_{\hat{\beta}} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} l_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)}; x_i^{(m)}) \right].$$

Also,

$$\sum_{m=1}^{2} \left\{ L_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)}) - \hat{L}_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)}; \mathcal{D}_m) \right\} \leq \sqrt{\frac{1}{2}\left(\frac{\omega(\beta^{(1)})^2}{n_1} + \frac{\omega(\beta^{(2)})^2}{n_2} \right) \log\frac{1}{\delta}} + 2R_{n_1,n_2}(l_{\lambda,\gamma})$$

for all $\hat{\beta}$ with probability at least $1-\delta$.

Proof. Both of the inequalities can be proved in the same way. We show the first inequality in this proof. Let $f(\mathcal{D}_1, \mathcal{D}_2) = \sup_{\hat{\beta}} \sum_{m=1}^{2} \left\{ \hat{L}_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)}; \mathcal{D}_m) - L_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)}) \right\}$. Then

$$|f(\mathcal{D}_1, \mathcal{D}_2) - f(x_1^{(1)}, y_1^{(1)}, \cdots, x_i^{(1)\prime}, y_i^{(1)\prime}, \cdots, x_{n_1}^{(1)}, y_{n_1}^{(1)}, \mathcal{D}_2)| \leq \frac{\omega(\beta^{(1)})}{n_1} \text{ and}$$
$$|f(\mathcal{D}_1, \mathcal{D}_2) - f(\mathcal{D}_1, x_1^{(2)}, y_1^{(2)}, \cdots, x_j^{(2)\prime}, y_j^{(2)\prime}, \cdots, x_{n_2}^{(2)}, y_{n_2}^{(2)})| \leq \frac{\omega(\beta^{(2)})}{n_2}$$

as before. Hence

$$\mathbb{P}(f(\mathcal{D}_1, \mathcal{D}_2) - E[f(\mathcal{D}_1, \mathcal{D}_2)] \geq \epsilon) \leq e^{-\frac{2}{A}\epsilon^2}$$

for all $\epsilon > 0$ where $A = \frac{\omega(\beta^{(1)})}{n_1} + \frac{\omega(\beta^{(2)})}{n_2}$. The expectation of f is bounded by the Rademacher complexity. Therefore,

$$\mathbb{P}\left(f(\mathcal{D}_1, \mathcal{D}_2) \geq \epsilon\right) \leq e^{-\frac{2}{A}(\epsilon - 2R_{n_1,n_2}(l_{\lambda,\gamma}))^2}.$$

Set δ to be the right hand side of the inequality to obtain the result. □

The Rademacher complexity on the joint dataset can be further bounded in the following way.

Proposition 7. *The Rademacher complexity is bounded by*

$$R_{n_1,n_2}(l_{\lambda,\gamma}) \le \left\{ \frac{1}{8}\sqrt{\frac{3}{\pi}} B \left(\left\vert\!\left\vert\!\left\vert \beta_c^{(1)} - \beta_c^{(2)} \right\vert\!\right\vert\!\right\vert + \left\vert\!\left\vert\!\left\vert \beta_c^{(1)} + \beta_c^{(2)} \right\vert\!\right\vert\!\right\vert \right) + \left(1 + \frac{1}{2}\sqrt{\frac{3}{\pi}}\right) \sqrt{\frac{2\log C}{\lambda}} BC + \frac{1}{2}\sqrt{\frac{3}{\pi}} C \log C \right\}$$
$$\times \sum_{m=1}^{2} \frac{1}{\sqrt{n_m}} + \frac{1}{4}\sqrt{\frac{3}{\pi}} B \sum_{m=1}^{2} \frac{\left\vert\!\left\vert\!\left\vert \beta^{(m)} \right\vert\!\right\vert\!\right\vert}{\sqrt{n_m}}$$

Proof. By definition,

$$\begin{aligned}
R_{n_1,n_2}(l_{\lambda,\gamma}) &= E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} l_{\lambda,\gamma}(\hat\beta, \beta^{(m)}; x_i^{(m)}) \right] \\
&= E\left[\sup_{\hat\beta} \frac{1}{n} \sum_{i=1}^{n} \sigma_i^{(m)} \left\{ \sum_{c=1}^{C} \|\beta_c^{(m)} - \hat\beta_c\| \|x_i^{(m)}\| + \sum_{c=1}^{C} -\beta_c^{(m)\top} x_i^{(m)} g_\gamma(\beta_c^{(m)\top} x_i^{(m)}) \right.\right. \\
&\quad \left.\left. + \log \sum_{c=1}^{C} e^{\hat\beta_c^\top x_i^{(m)}} + \frac{\lambda}{2} \sum_{c=1}^{C} \|\hat\beta_c\|^2 \right\} \right] \\
&\le E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} \sum_{c=1}^{C} \|\beta_c^{(m)} - \hat\beta_c\| \|x_i^{(m)}\| \right] \\
&\quad + E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} \sum_{c=1}^{C} -\beta_c^{(m)\top} x_i^{(m)} g_\gamma(\beta_c^{(m)\top} x_i^{(m)}) \right] \\
&\quad + E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} \log \sum_{c=1}^{C} e^{\hat\beta_c^\top x_i^{(m)}} \right] \\
&\quad + E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} \frac{\lambda}{2} \sum_{c=1}^{C} \|\hat\beta_c\|^2 \right].
\end{aligned}$$

The first term can be bounded by

$$\begin{aligned}
E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} \left\vert\!\left\vert\!\left\vert \beta_c^{(m)} - \hat\beta_c \right\vert\!\right\vert\!\right\vert \|x_i^{(m)}\| \right] &\le E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} B \left\vert\!\left\vert\!\left\vert \beta_c^{(m)} - \hat\beta_c \right\vert\!\right\vert\!\right\vert \right] \\
&\le E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} B \left\vert\!\left\vert\!\left\vert \beta_c^{(m)} - \hat\beta_c \right\vert\!\right\vert\!\right\vert \mathbb{1}_{\sum_{i=1}^{n_1} \sigma_i^{(1)} > 0, \sum_{i=1}^{n_2} \sigma_i^{(2)} > 0} \right] \\
&\quad + E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} B \left\vert\!\left\vert\!\left\vert \beta_c^{(m)} - \hat\beta_c \right\vert\!\right\vert\!\right\vert \mathbb{1}_{\sum_{i=1}^{n_1} \sigma_i^{(1)} > 0, \sum_{i=1}^{n_2} \sigma_i^{(2)} < 0} \right] \\
&\quad + E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} B \left\vert\!\left\vert\!\left\vert \beta_c^{(m)} - \hat\beta_c \right\vert\!\right\vert\!\right\vert \mathbb{1}_{\sum_{i=1}^{n_1} \sigma_i^{(1)} < 0, \sum_{i=1}^{n_2} \sigma_i^{(2)} > 0} \right] \\
&\le E\left[\sup_{\hat\beta} \sum_{m=1}^{2} \frac{1}{n_m} \sum_{i=1}^{n_m} \sigma_i^{(m)} B \left\vert\!\left\vert\!\left\vert \beta_c^{(m)} - \hat\beta_c \right\vert\!\right\vert\!\right\vert \mathbb{1}_{\sum_{i=1}^{n_1} \sigma_i^{(1)} > 0, \sum_{i=1}^{n_2} \sigma_i^{(2)} > 0} \right] \\
&\quad + \frac{1}{4}\sqrt{\frac{3}{\pi}} \sum_{m=1}^{2} \frac{1}{\sqrt{n_m}} B \left(\sqrt{\frac{2\log C}{\lambda}} C + \left\vert\!\left\vert\!\left\vert \beta_c^{(m)} - \hat\beta_c \right\vert\!\right\vert\!\right\vert \right)
\end{aligned}$$

where the last inequality comes from the proof for Rademacher complexity bound on a single dataset and by the fact that

$\mathbb{P}(\sum_{i=1}^{n_m}\sigma_i^{(m)} < 0) \leq \frac{1}{2}$ for $m = 1, 2$. Moreover,

$$\begin{aligned}
&E\left[\sup_{\hat{\beta}}\sum_{m=1}^{2}\frac{1}{n_m}\sum_{i=1}^{n_m}\sigma_i^{(m)}B\sum_{c=1}^{C}\|\beta_c^{(m)}-\hat{\beta}_c\|\mathbb{1}_{\sum_{i=1}^{n_1}\sigma_i^{(1)}>0,\sum_{i=1}^{n_2}\sigma_i^{(2)}>0}\right]\\
&\leq E\left[\sup_{\hat{\beta}}\sum_{m=1}^{2}\frac{B}{n_m}\sum_{i=1}^{n_m}\sigma_i^{(m)}\sum_{c=1}^{C}\left(\left\|\frac{\beta_c^{(1)}+\beta_c^{(2)}}{2}-\hat{\beta}_c\right\|+\left\|\frac{\beta_c^{(1)}-\beta_c^{(2)}}{2}\right\|\right)\mathbb{1}_{\sum_{i=1}^{n_1}\sigma_i^{(1)}>0,\sum_{i=1}^{n_2}\sigma_i^{(2)}>0}\right]\\
&\leq E\left[\sup_{\hat{\beta}}\sum_{m=1}^{2}\frac{B}{n_m}\sum_{i=1}^{n_m}\sigma_i^{(m)}\sum_{c=1}^{C}\left\|\frac{\beta_c^{(1)}+\beta_c^{(2)}}{2}-\hat{\beta}_c\right\|\mathbb{1}_{\sum_{i=1}^{n_1}\sigma_i^{(1)}>0,\sum_{i=1}^{n_2}\sigma_i^{(2)}>0}\right]\\
&\quad+\sum_{c=1}^{C}\frac{\|\beta_c^{(1)}-\beta_c^{(2)}\|}{2}\sum_{m=1}^{2}E\left[\frac{B}{n_m}\sum_{i=1}^{n_m}\sigma_i^{(m)}\mathbb{1}_{\sum_{i=1}^{n_1}\sigma_i^{(1)}>0,\sum_{i=1}^{n_2}\sigma_i^{(2)}>0}\right]\\
&\leq\sum_{c=1}^{C}\left(\frac{\|\beta_c^{(1)}+\beta_c^{(2)}\|}{2}+\sqrt{\frac{2\log C}{\lambda}}\right)\frac{1}{2}\sum_{m=1}^{2}E\left[\frac{B}{n_m}\sum_{i=1}^{n_m}\sigma_i^{(m)}\mathbb{1}_{\sum_{i=1}^{n_m}\sigma_i^{(m)}>0}\right]\\
&\quad+\sum_{c=1}^{C}\frac{\|\beta_c^{(1)}-\beta_c^{(2)}\|}{2}\sum_{m=1}^{2}E\left[\frac{B}{n_m}\sum_{i=1}^{n_m}\sigma_i^{(m)}\mathbb{1}_{\sum_{i=1}^{n_1}\sigma_i^{(1)}>0,\sum_{i=1}^{n_2}\sigma_i^{(2)}>0}\right]
\end{aligned}$$

and

$$\begin{aligned}
\sum_{m=1}^{2}E\left[\frac{B}{n_m}\sum_{i=1}^{n_m}\sigma_i^{(m)}\mathbb{1}_{\sum_{i=1}^{n_1}\sigma_i^{(1)}>0}\mathbb{1}_{\sum_{i=1}^{n_2}\sigma_i^{(2)}>0}\right]&\leq\sum_{m=1}^{2}\frac{1}{2}\frac{B}{n_m}E\left[\sum_{i=1}^{n_m}\sigma_i^{(m)}\mathbb{1}_{\sum_{i=1}^{n_m}\sigma_i^{(m)}>0}\right]\\
&\leq\sum_{m=1}^{2}\frac{1}{2}\sqrt{\frac{3}{4\pi n_m}}B.
\end{aligned}$$

Hence we get

$$\begin{aligned}
E\left[\sup_{\hat{\beta}}\sum_{m=1}^{2}\frac{1}{n_m}\sum_{i=1}^{n_m}\sigma_i^{(m)}\sum_{c=1}^{C}\|\beta_c^{(m)}-\hat{\beta}_c\|\|x_i^{(m)}\|\right]&\leq\frac{1}{4}\sqrt{\frac{3}{\pi}}\sum_{m=1}^{2}\frac{1}{\sqrt{n_m}}B\left(\sqrt{\frac{2\log C}{\lambda}}C+\sum_{c=1}^{C}\|\beta_c^{(m)}\|\right)\\
&\quad+\sum_{m=1}^{2}\frac{1}{4}\sqrt{\frac{3}{\pi}}\frac{B}{\sqrt{n_m}}\sum_{c=1}^{C}\left(\frac{\|\beta_c^{(1)}-\beta_c^{(2)}\|}{2}+\frac{\|\beta_c^{(1)}+\beta_c^{(2)}\|}{2}\right)\\
&\quad+\frac{1}{4}\sqrt{\frac{3}{\pi}}BC\sqrt{\frac{2\log C}{\lambda}}\sum_{m=1}^{2}\frac{1}{\sqrt{n_m}}
\end{aligned}$$

The rest of the terms are bounded in the same way as in one dataset case. Adding all the bounds up concludes the proof. □

Subgradient method estimator on the combined dataset also has a similar error bound.

Proposition 8. *Let the subgradient method estimator on $\mathcal{D}_1 \cup \mathcal{D}_2$ be iteratively defined by $\hat{\beta}^{[k+1]} = \hat{\beta}^{[k]} - \eta^{[k]}g^{[k]}$ where $\hat{\beta}^{[1]}$ is an initial estimator, $\eta^{[k]}$ is the kth step size, and $g^{[k]} \in \partial\sum_{m=1}^{2}\hat{L}_{\lambda,\gamma}(\hat{\beta}^{[k](m)}, \beta^{(m)}; \mathcal{D}_m)$. Then*

$$\min\{\sum_{m=1}^{2}\hat{L}_{\lambda,\gamma}(\hat{\beta}^{[k](m)};\mathcal{D}_m)|k=1,\cdots,K\}-\sum_{m=1}^{2}\hat{L}_{\lambda,\gamma}(\hat{\beta}^{*(1,2)};\mathcal{D}_m)\leq\frac{8\log C+2G\lambda\sum_{k=1}^{K}\eta^{[k]2}}{\lambda\sum_{k=1}^{K}\eta^{[k]}}$$

where $\hat{\beta}^{*(1,2)} = \underset{\hat{\beta}}{\text{argmin}}\ \sum_{m=1}^{2}\hat{L}_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)}; \mathcal{D}_m)$ is the empirical risk minimizer on the combined dataset.

Integrating all the propositions gives the following guarantee on the combined dataset.

Theorem 8. *Let $\hat{\beta}^{[k]}$ be the subgradient method estimator of $\sum_{m=1}^{2} \hat{L}_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)}; \mathcal{D}_m)$. Let $s = \underset{k \leq K}{\textit{argmin}} \sum_{m=1}^{2} \hat{L}_{\lambda,\gamma}(\hat{\beta}^{[k]}; \mathcal{D}_m)$ denote the step that achieved the lowest loss during all the subgradient steps. Then*

$$\sum_{m=1}^{2} L_{\lambda}(\hat{\beta}^{[s]}; \beta^{(m)}) \leq \Psi(n_1, n_2, \beta^{(1)}, \beta^{(2)}; \delta) + \sum_{m=1}^{2} L_{\lambda,\gamma}(\beta^{*(1,2)}, \beta^{(m)})$$

with probability at least $1 - 2\delta$ where $\Psi = O\left(\left(\sqrt{\frac{1}{n_1}} + \sqrt{\frac{1}{n_2}}\right)\sqrt{\log \frac{1}{\delta}}\right)$ and

$$\beta^{*(1,2)} = \underset{\hat{\beta}}{\textit{argmin}} \sum_{m=1}^{2} L_{\lambda,\gamma}(\hat{\beta}, \beta^{(m)})$$

is the minimizer for the population surrogate loss on the combined dataset.

Proof. Define

$$\Psi(n_1, n_2, \beta^{(1)}, \beta^{(2)}; \delta) = \psi_1(n_1, n_2, \beta^{(1)}, \beta^{(2)}; \delta) + \psi_2(n_1, n_2, \beta^{(1)}, \beta^{(2)}) + \psi_3$$

where

$$\begin{aligned}
\psi_1(n_1, n_2, \beta^{(1)}, \beta^{(2)}; \delta) &= \sqrt{2\left(\frac{\omega(\beta^{(1)})^2}{n_1} + \frac{\omega(\beta^{(2)})^2}{n_2}\right)\log\frac{1}{\delta}}, \\
\psi_2(n_1, n_2, \beta^{(1)}, \beta^{(2)}) &= \left(\frac{1}{\sqrt{n_1}} + \frac{1}{\sqrt{n_2}}\right)\left\{b_1\left(\left|\!\left|\!\left|\beta^{(1)} - \beta^{(2)}\right|\!\right|\!\right| + \left|\!\left|\!\left|\beta^{(1)} + \beta^{(2)}\right|\!\right|\!\right|\right) + b_2\right\} \\
&\quad + b_3\left(\frac{|\!|\!|\beta^{(1)}|\!|\!|}{\sqrt{n_1}} + \frac{|\!|\!|\beta^{(2)}|\!|\!|}{\sqrt{n_2}}\right), \\
\psi_3 &= \frac{8\log C + 2G\lambda\sum_{k=1}^{K}\eta^{[k]2}}{\lambda\sum_{k=1}^{K}\eta^{[k]}},
\end{aligned}$$

for constants $b_1 = \frac{1}{2}\sqrt{\frac{3}{\pi}}B$, $b_2 = \left(4 + 2\sqrt{\frac{3}{\pi}}\right)BC\sqrt{\frac{2\log C}{\lambda}} + 2\sqrt{\frac{3}{\pi}}C\log C$, and $b_3 = \sqrt{\frac{3}{\pi}}B$. Then the theorem can be proved by using the same line of reasoning as in Theorem 7. □

As before, the bound Ψ consists of three parts. The first part ψ_1 is the error stemming from the randomness of the samples, and is connected to δ. The second part ψ_2 is the error related to the complexity of the hypothesis class. ψ_2 has some overlapping terms that appeared in Theorem 7 as well as new terms that reveal the effect of relationship between $\beta^{(1)}$ and $\beta^{(2)}$. The last part ψ_3 is the error due to the subgradient method.

Once again, the theorem displays the effect of size of the datasets and the hypothesis class on the performance of the subgradient method estimator $\sum_{m=1}^{2} L_{\lambda}(\hat{\beta}^{[s]}; \beta^{(m)})$. Suppose the step size satisfies conditions as before, which implies that the subgradient method converges. The error bound decreases if there are more samples in $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(1)}$. Similar interpretation can be made on the size of the hypothesis class.

A new implication from this theorem is that relationship between the true parameters also influences the error bound. The error bound decreases if the true parameters are similar. Indeed, if $\beta^{(1)}$ and $\beta^{(2)}$ are close to each other, then $|\!|\!|\beta^{(1)} - \beta^{(2)}|\!|\!|$ is small, reducing $\psi_{2,3}$. It aligns with the intuition that merging the datasets would undoubtedly be advantageous if they are sampled from the same distribution.

B.2.3 An Equivalent Condition for Error Bound Reduction

We propose to combine two datasets based on the error bounds. Recall from Theorem 7 that Φ serves as an upper bound of the performance of the model trained on a single dataset . Also, Ψ serves as an upper bound of the performance of the model trained on the joint dataset by Theorem 8. Merging the datasets when the sum of Φ's on each dataset is greater than Ψ would decrease the error bound, leading to a better guarantee of the model. Specifically, let $\hat{\beta}^{[s_m](m)}$ denote the subgradient method estimator that achieves the lowest loss on $\mathcal{D}_m$ alone for $m = 1, 2$. Similarly, let $\hat{\beta}^{[s_{1,2}](1,2)}$ denote the subgradient method estimator on the joint dataset $\mathcal{D}_1 \cup \mathcal{D}_2$. As in Theorem 8, let $\beta^{*(1,2)}$ denote the minimizer for the population loss on the joint dataset. Then

$$L_\lambda(\hat{\beta}^{[s_m](m)}; \beta^{(m)}) \leq \Phi(n_m, \beta^{(m)}; \delta) + L_{\lambda,\gamma}(\beta^{*(1,2)}, \beta^{(m)})$$

for $m = 1, 2$ and

$$\sum_{m=1}^{2} L_\lambda(\hat{\beta}^{[s_{1,2}](1,2)}; \beta^{(m)}) \leq \Psi(n_1, n_2, \beta^{(1)}, \beta^{(2)}; \delta) + \sum_{m=1}^{2} L_{\lambda,\gamma}(\beta^{*(1,2)}, \beta^{(m)}).$$

Hence the datasets should be combined if $\Psi(n_1, n_2, \beta^{(1)}, \beta^{(2)}; \delta) \leq \sum_{m=1}^{2} \Phi(n_m, \beta^{(m)}; \delta)$. The following theorem states an equivalent condition for the comparison.

Theorem 9. *Let Φ and Ψ be the functions defined in Theorem 7 and 8, respectively. Then the error bound of the model trained on the combined dataset is smaller, or equivalently,*

$$\Psi(n_1, n_2, \beta^{(1)}, \beta^{(2)}; \delta) \leq \sum_{m=1}^{2} \Phi(n_m, \beta^{(m)}; \delta)$$

if and only if

$$\begin{aligned} &\sqrt{2 \log \frac{1}{\delta} \left(\frac{\omega(\beta^{(1)})^2}{n_1} + \frac{\omega(\beta^{(2)})^2}{n_2} \right)} + \sqrt{\frac{3}{\pi}} B \left(\frac{1}{\sqrt{n_1}} + \frac{1}{\sqrt{n_2}} \right) \left(\frac{|||\beta^{(1)} - \beta^{(2)}|||}{2} + \frac{|||\beta^{(1)} + \beta^{(2)}|||}{2} \right) \\ &\leq \sqrt{2 \log \frac{1}{\delta}} \left(\sqrt{\frac{\omega(\beta^{(1)})^2}{n_1}} + \sqrt{\frac{\omega(\beta^{(2)})^2}{n_2}} \right) + \sqrt{\frac{3}{\pi}} B \left(\frac{|||\beta^{(1)}|||}{\sqrt{n_1}} + \frac{|||\beta^{(2)}|||}{\sqrt{n_2}} \right) + \frac{4 \log C}{\lambda \sum_{k=1}^{K} \eta^{[k]}}. \end{aligned}$$

The above condition agrees with what one would intuitively expect when combining two datasets. Suppose the true parameters $\beta^{(1)}$ and $\beta^{(2)}$ are are close to each other. Then the distance $\left|\left|\left| \hat{\beta}^{(1)} - \hat{\beta}^{(2)} \right|\right|\right|$ is small, decreasing the left-hand-side of the inequality. In turn, the above condition would possibly hold, which further indicates that the datasets should be combined.

C EXPERIMENTAL DETAILS

In this section, we provide details for the implementation of the neural network model, and compare the algorithm with other clustering method for multitask learning. The method that is most relevant to our study is BiFactor MTL proposed by Murugesan et al. [2017].

C.1 NEURAL NETWORK SETUP

We employ two simple neural network architectures: a single hidden layer multi-layer perceptron (MLP1) and a two hidden layer multi-layer perceptron (MLP2). The following details outline the configurations:

- **Network Architecture**: The input datasets consist of Boom Bikes (BB) [Mishra, 2021], Demand Forecast for Optimized Inventory Planning (DFOIP) [Aguilar, 2023], and Walmart Data Analysis (WDAF) [Sahu, 2023], with input feature dimensions of 7, 9, and 12, respectively. As datasets have different number of features, models are built based on the input dimension accordingly. In both MLP1 and MLP2, the hidden layers have the same dimension as the input feature dimension of each input.
- **Training the Representation**: For each dataset, we train the models using the entire dataset to enable the capture of more complex patterns and generate richer feature representations. It is important to note that this step is not intended to produce the most accurate predictive model but rather to focus on learning a more informative representation of the data for subsequent tasks. The models are trained with a batch size of 64 over 10 epochs, with a learning rate of 1e-4 and weight decay of 0.9. The Mean Squared Error (MSE) loss function is used, optimized by stochastic gradient descent. Training was conducted on a single Nvidia RTX 6000 (Ampere Version) GPU.

Table 7: Algorithm 3 on Representation with $\widehat{\text{OSE}}$

	BB	DFOIP	WDAF
Individual	2.822	3.097×10^3	1.495×10^2
Alg. 3	2.359	9.105×10^2	3.562×10^1
Reduction	16.41%	70.60%	76.17%
	BB	DFOIP	WDAF
Individual	3.308	2.908×10^3	1.114×10^2
BiFactor	3.239	1.167×10^3	9.697×10^1
Reduction	2.086%	59.87%	12.90%

- **Integration with Algorithm 3**: Let $f(\cdot)$ and $g(\cdot)$ represent the trained MLP1 and MLP2 models, respectively. Both models consist of hidden layers followed by a linear layer, expressed as $f(\cdot) = l \circ f_h(\cdot)$ and $g(\cdot) = l \circ g_h(\cdot)$, where $f_h(\cdot)$ and $g_h(\cdot)$ represent the hidden layer functions, and $l(\cdot)$ represents the linear layer. For datasets $\{(x_i, y_i)\}_{i=1}^n$, we transform the input into $\{(e_i, y_i)\}_{i=1}^n$, where e_i is the representation from the penultimate layer. That is, either $e_i = f_h(x_i)$ or $e_i = g_h(x_i)$, depending on the model we use. Algorithm 3 can then be applied to the transformed dataset $\{(e_i, y_i)\}_{i=1}^n$, as the linear layer structure aligns perfectly with the regression setting.

C.2 CHOICE OF HYPERPARAMETERS

We further provide more details on the choice of hyperparameters of our algorithms.

- **Grid search of α in Algorithm 2**: The grid search of α in Part1 is applied over the interval $[\alpha_{min}, \alpha_{max}] = [2, 10]$ with window size $\eta = 0.01$.
- **Threshold λ in Algorithm 2** and 3: λ is fixed as 0.9 throughout all experiments.
- **The number of iterations in Algorithm 2**: In Part1, $max_iterations$ is set as 1000.
- **Size of training dataset**: We chose n_1 and n_2 as 50 or 100, depending on the feature size p.
- **The number of out-of-samples**: Across all experiments, the number of out-of-samples is fixed as 1000.
- **Performance comparison in Section 3.1**: The columns `Algorithm 2` and `Direct Comparison` in Table 3 and Table 4 are obtained from 1000 repetitions.

C.3 COMPARISON WITH OTHER MODELS

To the best of our knowledge, no existing work specifically focuses on combining datasets with the goal of minimizing out-of-sample error. However, the algorithm developed by Murugesan et al. [2017] could serve as an alternative approach. In Murugesan et al. [2017], factor models are employed to decompose the weight matrix into feature clusters and task clusters. To facilitate a comparison between the two methods, we apply each approach to the features obtained by applying the MLP1 model to the datasets introduced earlier.

The result of Algorithm 3 and BiFactor MTL is summarized in Table 7. BB is divided into three parts based on the weather condition, DFOIP is partitioned into four parts based on store types, and WDAF is split into five parts according to store locations. For each dataset, only a subset of the entire dataset was chosen for this additional experiment to expedite the clustering. The reported values of $\widehat{\text{OSE}}$ on individual datasets may be different for the two methods as they both have some randomness when selecting out-of-samples and computing the $\widehat{\text{OSE}}$.

As shown in Table 7, Algorithm 3 significantly reduces $\widehat{\text{OSE}}$, whereas the reduction for the BiFactor MTL method is less significant. This discrepancy in $\widehat{\text{OSE}}$ reduction is attributed to the different dataset combinations selected by each algorithm. For example, Algorithm 3 opted to combine the datasets for weather conditions 1 and 2 while keeping condition 3 separate on BB, whereas BiFactor MTL merged the datasets for weather conditions 2 and 3. Our algorithm effectively identifies relevant datasets for combination, leading to lower errors on the merged datasets.

C.4 COMPLEXITY ANALYSIS

Algorithm 3 uses pairwise comparison, which effectively reduces the running time by having a quadratic growth on the number of datasets. As our method employs linear models, the actual execution time does not extend uncontrollably when the number of datasets scales. For instance, it took 10 minutes for RSS and 3 hours for CFGSF to train an MLP and apply Algorithm 3, on a standard CPU without parallelization.

To our knowledge, there is no off-the-shelf clustering method applicable to our setting to effectively reduce the computational complexity due to the following reason. Most clustering methods commonly assume that objects to be clustered have representative features. However, our algorithm only has a notion of pairwise similarity between two datasets and no universal feature space of the datasets themselves, which renders those clustering algorithms inapplicable. Other clustering methods that work without features usually require a similarity matrix which contains information about similarity between any two objects. However, complexity of computing the similarity matrix amounts to that of Algorithm 3 in the worst case. Hence we believe that applying the greedy algorithm not only simplifies the clustering part, which is not the most core part of our method, but also potentially reduces the computational cost. Given the computational resources commonly available today, we do not view the computational complexity as a significant limitation. Developing a clustering-based method within our framework would require substantial additional work, and we hope exploration on clustering strategies could be done in future research.