Transfer Learning in ℓ_1 Regularized Regression: Hyperparameter Selection Strategy based on Sharp Asymptotic Analysis

Anonymous authors
Paper under double-blind review

Abstract

Transfer learning techniques aim to leverage information from multiple related datasets to enhance prediction quality against a target dataset. Such methods have been adopted in the context of high-dimensional sparse regression, and some Lasso-based algorithms have been invented: Trans-Lasso and Pretraining Lasso are such examples. These algorithms require the statistician to select hyperparameters that control the extent and type of information transfer from related datasets. However, selection strategies for these hyperparameters, as well as the impact of these choices on the algorithm's performance, have been largely unexplored. To address this, we conduct a thorough, precise study of the algorithm in a high-dimensional setting via an asymptotic analysis using the replica method. Our approach reveals a surprisingly simple behavior of the algorithm: Ignoring one of the two types of information transferred to the fine-tuning stage has little effect on generalization performance, implying that efforts for hyperparameter selection can be significantly reduced. Our theoretical findings are also empirically supported by real-world applications on the IMDb dataset.

1 Introduction

The increasing availability of large-scale datasets has led to an expansion of methods such as pretraining and transfer learning (Torrey & Shavlik, 2010; Weiss et al., 2016; Zhuang et al., 2021), which exploit similarities between datasets obtained from different but related sources. By capturing general patterns and features, which are assumed to be shared across data collected from multiple sources, one can improve the generalization performance of models trained for a specific task, even when the data for the task is too limited to make accurate predictions. Transfer learning has demonstrated its efficacy across a wide range of applications, such as image recognition (Zhu et al., 2011), text categorization (Zhuang et al., 2011), bioinformatics (Petegrosso et al., 2016), and wireless communications (Pan et al., 2011; Wang et al., 2021). It is reported to be effective in high-dimensional statistics (Banerjee et al., 2020; Takada & Fujisawa, 2020; Bastani, 2021; Li et al., 2021), with clinical analysis (Turki et al., 2017; Hajiramezanali et al., 2018; McGough et al., 2023; Craig et al., 2024) being a major application area due to the scarcity of patient data and the need to comply with privacy regulations. In such applications, one must handle high-dimensional data, where the number of features greatly exceeds the number of samples. This has sparked interest in utilizing transfer learning to sparse high-dimensional regression, which is the standard method for analyzing such data.

A classical method for statistical learning under high-dimensional settings is Lasso (Tibshirani, 1996), which aims at simultaneously identifying the sparse set of features and their regression coefficients that are most relevant for predicting the target variable. Its simplicity and interpretability of the results, as well as its convex property as an optimization problem, has made Lasso a popular choice for sparse regression tasks, leading to a flurry of research on its theoretical properties (Candes & Tao, 2006; Zhao & Yu, 2006; Meinshausen & Bühlmann, 2006; Wainwright, 2009; Bellec et al., 2018).

Modifications to Lasso have been proposed to incorporate and take advantage of auxiliary samples. The data-shared Lasso (Gross & Tibshirani, 2016) and stratified Lasso (Ollier & Viallon, 2017) are both methods based on multi-target learning, where one solves multiple regression problems with related data in a single fitting procedure. Two more recent approaches are Trans-Lasso (Bastani, 2021; Li et al., 2021) and Pretraining Lasso (Craig et al., 2024), both of which require two-stage regression procedures. Here, the first stage is focused on identifying common features shared across multiple related datasets, while the second stage is dedicated to fine-tuning the model on a specific target dataset. To simplify the explanation, we introduce a generalized algorithm that encompasses both methods, which is referred to as *Generalized Trans-Lasso* hereafter. Given a set of datasets of size K, $\mathcal{D} = \{\mathcal{D}^{(k)}\}_{k=1}^{K}$, each consisting of an observation vector of dimension M_k, $\boldsymbol{y}^{(k)} \in \mathbb{R}^{M_k}$, and its corresponding covariate matrix $\boldsymbol{A}^{(k)} \in \mathbb{R}^{M_k \times N}$, generalized Trans-Lasso performs the following two steps:

1. Pretraining procedure: In the first stage, all datasets are used to identify a common feature vector $\hat{\boldsymbol{x}}_1$ which captures shared patterns and relevant features across the datasets:

$$\hat{\boldsymbol{x}}^{(\mathsf{1st})} = \underset{\boldsymbol{x}\in\mathbb{R}^N}{\arg\min}\, \mathcal{L}^{(\mathsf{1st})}(\boldsymbol{x};\mathcal{D}), \qquad \mathcal{L}^{(\mathsf{1st})}(\boldsymbol{x};\mathcal{D}) = \frac{1}{2}\sum_{k=1}^{K} \|\boldsymbol{y}^{(k)} - \boldsymbol{A}^{(k)}\boldsymbol{x}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1. \tag{1}$$

2. Fine-tuning: Here, the interest is only on a specific target dataset, which is designated by $k = 1$ without loss of generality. The second stage incorporates the learned feature vector's configuration and its support set into a modified sparse regression model. This is done by first offsetting the observation vector $\boldsymbol{y}^{(1)}$ by a common prediction vector deduced from the first stage, i.e. $\boldsymbol{A}^{(1)}\hat{\boldsymbol{x}}^{(\mathsf{1st})}$. Further, the penalty factor is also modified such that variables not selected in the first stage is penalized more heavily than those selected. The above procedure is summarized by the following optimization problem:

$$\begin{aligned} \hat{\boldsymbol{x}}^{(\mathsf{2nd})} &= \underset{\boldsymbol{x}\in\mathbb{R}^N}{\arg\min}\, \mathcal{L}^{(\mathsf{2nd})}(\boldsymbol{x};\hat{\boldsymbol{x}}^{(\mathsf{1st})},\mathcal{D}^{(1)}), \\ \mathcal{L}^{(\mathsf{2nd})}(\boldsymbol{x};\hat{\boldsymbol{x}}^{(\mathsf{1st})},\mathcal{D}^{(1)}) &= \frac{1}{2}\|\boldsymbol{y}^{(1)} - \kappa\boldsymbol{A}^{(1)}\hat{\boldsymbol{x}}^{(\mathsf{1st})} - \boldsymbol{A}^{(1)}\boldsymbol{x}\|_2^2 + \sum_{i=1}^{N}\left(\lambda_2 + \Delta\lambda\mathbb{I}[\hat{x}_i^{(\mathsf{1st})} = 0]\right)|x_i|, \end{aligned} \tag{2}$$

   where $\mathbb{I}$ is the indicator function. Here, $\kappa, \Delta\lambda \geq 0$ are hyperparameters that determine the extent of information transfer from the first stage.

It is crucial to note that the performance of this algorithm is highly dependent on the choice of hyperparameters, which control the strength of knowledge transfer. Suboptimal hyperparameter selection can lead to insufficient knowledge transfer, or negative transfer (Pan et al., 2011), where the model's performance is degraded by the transfer of adversarial information. Thus, the choice of appropriate hyperparameters should be addressed with equal importance as the algorithm itself. In fact, a distinctive difference between Trans-Lasso and Pretraining Lasso is whether one should inherit support information from the first stage or not. While this adds an additional layer of complexity to hyperparameter selection, it also provides an opportunity to inherit significant information from the first stage, if the support of the feature vectors between the sources are similar. Empirical investigation of the qualitative effects of hyperparameters on the algorithm's performance can be computationally intensive, especially in high-dimensional settings.

To address the issue of hyperparameter selection without resorting to extensive empirical research, one can turn to the field of statistical physics, which provides a set of tools to analyze the performance of high-dimensional statistical models in a theoretical manner. While these theoretical studies are restricted to inevitably simplified assumptions on the statistical properties of the dataset, they often provide valuable qualitative insights generalizable to more realistic settings. The replica method (Mezard et al., 1986; Charbonneau et al., 2023), in particular, has been widely used to precisely analyze high-dimensional statistical models including Lasso (Kabashima et al., 2009; Rangan et al., 2009; Obuchi & Kabashima, 2016; Takahashi & Kabashima, 2018; Obuchi & Kabashima, 2019; Okajima et al., 2023), and has been shown to provide accurate predictions in many cases.

In this work, we conduct a sharp asymptotic analysis of the generalized Trans–Lasso's performance under a high-dimensional setting using the replica method. Our study reveals a simple heuristic strategy to select hyperparameters. More specifically, it suffices to use either the support information or the actual value of the feature vector $\hat{\boldsymbol{x}}^{(\mathsf{1st})}$ obtained from the pretraining stage to achieve near-optimal performance after fine-tuning. The theoretical findings are complemented by an application of the generalized Trans-Lasso to a real-world dataset called the IMDb dataset, which highlights the effectiveness of this heuristic.

1.1 Related Works

Trans-Lasso, first proposed by Bastani (2021) and further studied by Li et al. (2021), is a special case of the two-stage regression algorithm given by equation 1 and equation 2. In this variant, the second stage omits information transfer regarding the support of $\hat{\boldsymbol{x}}^{(\mathsf{1st})}$ (i.e., $\Delta\lambda = 0$) and sets κ to unity. This method has demonstrated efficacy in enhancing the generalization performance of Lasso regression, and has since been extended to various regression models, such as generalized linear models (Tian & Feng, 2023; Li et al., 2024), Gaussian graphical models (Li et al., 2023), and robust regression (Sun & Zhang, 2023). Moreover, the method also embodies a framework to determine which datasets are adversarial in transfer learning, allowing one to avoid negative transfer. While the problem of dataset selection is a practically important issue in transfer learning, the present study will not consider this aspect, but rather focus on the algorithm's performance given a fixed set of datasets.

On the other hand, Pretraining Lasso controls both κ and $\Delta\lambda$ using a tunable interpolation parameter $s \in [0, 1]$ as

$$\kappa = 1 - s, \qquad \Delta\lambda = \frac{1-s}{s}\lambda_2. \tag{3}$$

This formulation allows for a continuous transition between two extremes: Setting $s = 1$ reduces the problem to a standard single-dataset regression, while $s = 0$ reduces to regression constrained on the support of $\hat{\boldsymbol{x}}^{(\mathsf{1st})}$. Note that there is no choice of s which reduces Pretraining Lasso to Trans-Lasso.

Previous research has not yet established a clear understanding of the optimal hyperparameter selection for these methods. Moreover, the introduction of additional hyperparameters, such as those proposed in this study, has been largely unexplored in existing literature, possibly due to the computational demands involved. A more thorough investigation into the performance of these methods is necessary to reevaluate current, potentially suboptimal hyperparameter choices and to provide a more comprehensive understanding of the algorithm's behavior.

The analysis of our work is based on the replica method, which is a non-rigorous mathematical tool used in a wide range of theoretical studies, such as those on spin-glass systems and high-dimensional statistical models. The results from such analyses have been shown to be consistent with empirical results, and in some cases later proved by mathematically rigorous methods, such as approximate message passing theory (Bayati & Montanari, 2011; Javanmard & Montanari, 2013), adaptive interpolation (Barbier et al., 2019; Barbier & Macris, 2019), Gordon comparison inequalities (Stojnic, 2013; Thrampoulidis et al., 2018; Miolane & Montanari, 2021), and second-order Stein formulae (Bellec & Zhang, 2021; Bellec & Shen, 2022). The application of the replica method to multi-stage procedures in machine learning has also been done in the context of knowledge distillation (Saglietti & Zdeborová, 2022), self-training (Takahashi, 2024), and iterative algorithms (Okajima & Takahashi, 2024). Our analysis can be seen as an extension of these works to this generalized Trans-Lasso algorithm.

2 Problem Setup

As given in the Introduction, we consider a set of K datasets, each consisting of an observation vector $\boldsymbol{y}^{(k)} \in \mathbb{R}^{M_k}$ and covariate matrices $\boldsymbol{A}^{(k)} \in \mathbb{R}^{M_k \times N}$. The observation vector $\boldsymbol{y}^{(k)}$ is assumed to be generated from a linear model with Gaussian noise, i.e.

$$\boldsymbol{y}^{(k)} = \boldsymbol{A}^{(k)}\boldsymbol{r}^{(k)} + \boldsymbol{e}^{(k)}, \qquad \boldsymbol{e}^{(k)} \sim \mathcal{N}(0, (\sigma^{(k)})^2 \boldsymbol{I}_{M_k}). \tag{4}$$

Here, $\boldsymbol{e}^{(k)} \in \mathbb{R}^{M_k}$ is a Gaussian-distributed noise term, while $\boldsymbol{r}^{(k)} \in \mathbb{R}^N$ is the true feature vector of the underlying linear model. We consider the common and individual support model (Craig et al., 2024), where

$\{\boldsymbol{r}^{(k)}\}_{k=1}^{K}$ is assumed to have overlapping support. More concretely, let $\{\mathcal{I}(k)\}_{k=0}^{K}$ be a set of disjoint indices, where $\mathcal{I}(k) \subset \{1,2,\ldots,N\}$, $|\mathcal{I}(k)| = N_k$, for all $k = 0, \cdots, K$, and $\sum_{k=0}^{K} N_k \le N$. Then, $\boldsymbol{r}^{(k)}$ is given by

$$\boldsymbol{r}^{(k)}_{\mathcal{I}(0)} = \boldsymbol{x}_\star^{(0)} \in \mathbb{R}^{N_0}, \qquad \boldsymbol{r}^{(k)}_{\mathcal{I}(k)} = \boldsymbol{x}_\star^{(k)} \in \mathbb{R}^{N_k}, \qquad \boldsymbol{r}^{(k)}_{\backslash\{\mathcal{I}(k)\cup\mathcal{I}(0)\}} = \boldsymbol{0} \in \mathbb{R}^{N-N_k-N_0}, \tag{5}$$

and

$$\boldsymbol{y}^{(k)} = \boldsymbol{A}^{(k)}_{\mathcal{I}(0)}\boldsymbol{x}_\star^{(0)} + \boldsymbol{A}^{(k)}_{\mathcal{I}(k)}\boldsymbol{x}_\star^{(k)} + \boldsymbol{e}^{(k)}. \tag{6}$$

Here, $\boldsymbol{A}^{(k)}_{\mathcal{I}}$ denotes the submatrix of $\boldsymbol{A}^{(k)}$ with columns indexed by $\mathcal{I}$, $\boldsymbol{r}^{(k)}_{\mathcal{I}}$ denotes the subvector of $\boldsymbol{r}^{(k)}$ with indices in $\mathcal{I}$, and $\backslash\mathcal{I}$ is the shorthand of $\{1,2,\ldots,N\}\backslash\mathcal{I}$. Therefore, each linear model is always affected by the variables in the common support set $\mathcal{I}(0)$, with additional variables in the unique support set $\mathcal{I}(k)$ for each dataset $k = 1, \cdots, K$. We refer to $\boldsymbol{x}_\star^{(0)}$ as the common feature vector, while $\boldsymbol{x}_\star^{(k)}$ as the unique feature vector for class k. The implicit aim of Trans-Lasso can be seen as estimating the common feature vector $\boldsymbol{x}^{(0)}$ in the first stage, treating the unique feature vectors in each class as noise. On the other hand, the second stage is dedicated to estimating the unique feature vector $\boldsymbol{x}_\star^{(1)}$ given a noisy, biased estimator of the common feature vector $\boldsymbol{x}_\star^{(0)}$, i.e. $\hat{\boldsymbol{x}}^{(\mathsf{1st})}$.

We evaluate the performance of the generalized Trans-Lasso using the generalization error in each stage, which is defined as the expected mean square error of the model prediction made against a set of new observations $\tilde{\mathcal{D}} = \{(\tilde{\boldsymbol{y}}^{(k)}, \tilde{\boldsymbol{A}}^{(k)})\}_{k=1}^{K}$:

$$\epsilon^{(\mathsf{1st})} = \frac{1}{N}\sum_{k=1}^{K} \mathbb{E}_{\mathcal{D},\tilde{\mathcal{D}}}\left[\|\tilde{\boldsymbol{y}}^{(k)} - \tilde{\boldsymbol{A}}^{(k)}\hat{\boldsymbol{x}}^{(\mathsf{1st})}\|_2^2\right], \tag{7}$$

for the first stage, while the generalization error for the second stage, with specific interest on class $k = 1$ without loss of generality, is given by

$$\epsilon^{(\mathsf{2nd})} = \frac{1}{N}\mathbb{E}_{\mathcal{D},\tilde{\mathcal{D}}}\left[\|\tilde{\boldsymbol{y}}^{(1)} - \tilde{\boldsymbol{A}}^{(1)}(\kappa\hat{\boldsymbol{x}}^{(\mathsf{1st})} + \hat{\boldsymbol{x}}^{(\mathsf{2nd})})\|_2^2\right]. \tag{8}$$

The objective of our analysis is to determine the behavior of the generalization error given the choice of hyperparameters $(\lambda_1, \lambda_2, \kappa, \Delta\lambda)$.

High-dimensional setup We focus on the asymptotic behavior in a high-dimensional limit, where the length of the common and unique feature vectors, as well as the size of the datasets for each class, tend to infinity at the same rate, i.e.

$$\frac{N_0}{N} \to \pi^{(0)} = O(1), \qquad \frac{N_k}{N} \to \pi^{(k)} = O(1), \qquad \frac{M_k}{N} \to \alpha^{(k)} = O(1), \quad k = 1, \cdots, K, \qquad \text{as } N \to \infty. \tag{9}$$

Also, define the proportion of the number of variables not included in any of the support sets as $\pi^{(\mathsf{neg.})} = 1 - \sum_{k=0}^{K} \pi^{(k)}$. The covariate matrices are all assumed to have i.i.d. entries with zero mean and variance $1/N$, while the true feature vectors are also assumed to have i.i.d. standard Gaussian entries. While such a setup is a drastic simplification of real-world scenarios, it should be noted that these conditions can be further generalized to more complex settings. For instance, the analysis can be further extended to the case where the random matrix $\boldsymbol{A}^{(k)}$ inherits rotationally invariant structure (Vehkaperä et al., 2016; Takahashi & Kabashima, 2018), or to the case where the observations are corrupted by heavy-tailed noise (Adomaityte et al., 2023).

3 Results of Sharp Asymptotic Analysis via the replica method

Here, we state the result of our replica analysis. The series of calculations to obtain our results is similar to the well-established procedure used in the analyses for multi-staged procedures (Saglietti & Zdeborová, 2022; Takahashi, 2024; Okajima & Takahashi, 2024). While a brief overview of the derivation is given in the next subsection, we refer the reader to Appendix A for a detailed calculation. Although our theoretical analysis lacks a standing proof from the non-rigorousness of the replica method, we conjecture that the results are exact in the limit of large N. This is supported numerically via experiments on finite-size systems, which will be shown to be consistent with the theoretical predictions.

In fact, since the first stage of the algorithm is a standard Lasso optimization problem under Gaussian design, it is already known that at least up to the first stage of the algorithm, the replica method yields the same formula derivable from Approximate Message Passing (Bayati & Montanari, 2011) or Convex Gaussian Minmax Theorem (CGMT) (Thrampoulidis et al., 2018; Miolane & Montanari, 2021). Moreover, CGMT can be further applied to multi-stage optimization problems with independent data in each stage (Chandrasekher et al., 2023). The extension of this proof to the second stage of our algorithm, which contains data dependence among the two stages, is an unsolved problem that we leave for future work.

3.1 Overview of the calculation

Here, we briefly outline the derivation of the generalization errors $\epsilon^{(\mathsf{1st})}$ and $\epsilon^{(\mathsf{2nd})}$ using the replica method. Readers not interested in the derivation may skip this subsection and proceed to the next subsection for the main results.

Given the training dataset $\mathcal{D} = \{\mathcal{D}^{(k)}\}_{k=1}^{K}$, with $\mathcal{D}^{(k)} = (\boldsymbol{A}^{(k)}, \boldsymbol{y}^{(k)}),\ k = 1, \cdots, K$, define the following joint probability measure for $\boldsymbol{x}_1 \in \mathbb{R}^N$ and $\boldsymbol{x}_2 \in \mathbb{R}^N$ as

$$P_{\beta_1,\beta_2}(\boldsymbol{x}_1, \boldsymbol{x}_2; \mathcal{D}) \propto w_{\beta_1,\beta_2}(\boldsymbol{x}_1, \boldsymbol{x}_2; \mathcal{D}) = \exp\Big[-\beta_1 \mathcal{L}^{(\mathsf{1st})}(\boldsymbol{x}_1; \mathcal{D}) - \beta_2 \mathcal{L}^{(\mathsf{2nd})}(\boldsymbol{x}_2; \boldsymbol{x}_1, \mathcal{D}^{(1)})\Big], \tag{10}$$

Under this measure, the generalization error $\epsilon^{(\mathsf{1st})}$ and $\epsilon^{(\mathsf{2nd})}$ can be expressed as

$$\epsilon^{(\mathsf{1st})} = \lim_{\beta_2\to\infty} \lim_{\beta_1\to\infty} \lim_{t_1\to 0} \frac{1}{N} \mathbb{E}_{\mathcal{D},\tilde{\mathcal{D}}}\left[\frac{\partial}{\partial t_1} \log \int d\boldsymbol{x}_1 d\boldsymbol{x}_2 w_{\beta_1,\beta_2}(\boldsymbol{x}_1, \boldsymbol{x}_2; \mathcal{D})\ e^{t_1 \sum_{k=1}^{K} \|\tilde{\boldsymbol{y}}^{(k)} - \tilde{\boldsymbol{A}}^{(k)} \boldsymbol{x}_1\|_2^2}\right], \tag{11}$$

$$\epsilon^{(\mathsf{2nd})} = \lim_{\beta_2\to\infty} \lim_{\beta_1\to\infty} \lim_{t_2\to 0} \frac{1}{N} \mathbb{E}_{\mathcal{D},\tilde{\mathcal{D}}}\left[\frac{\partial}{\partial t_2} \log \int d\boldsymbol{x}_1 d\boldsymbol{x}_2 w_{\beta_1,\beta_2}(\boldsymbol{x}_1, \boldsymbol{x}_2; \mathcal{D})\ e^{t_2 \|\tilde{\boldsymbol{y}}^{(1)} - \tilde{\boldsymbol{A}}^{(1)}(\kappa \boldsymbol{x}_1 + \boldsymbol{x}_2)\|_2^2}\right]. \tag{12}$$

Such observation stems from the fact that, in the successive limit of $\beta_1 \to \infty$ followed by $\beta_2 \to \infty$, the joint probability measure $P_{\beta_1,\beta_2}(\boldsymbol{x}_1, \boldsymbol{x}_2; \mathcal{D})$ concentrates around $(\hat{\boldsymbol{x}}^{(\mathsf{1st})}, \hat{\boldsymbol{x}}^{(\mathsf{2nd})})$, the solution to the two-stage Trans-Lasso procedure. The above expression requires one to evaluate the average of the logarithm of the integral of $w_{\beta_1,\beta_2}(\boldsymbol{x}_1, \boldsymbol{x}_2; \mathcal{D})$, which can be alternatively expressed as

$$\mathbb{E}_{\mathcal{D}} \log \int d\boldsymbol{x}_1 d\boldsymbol{x}_2 w_{\beta_1,\beta_2}(\boldsymbol{x}_1, \boldsymbol{x}_2; \mathcal{D}) = \lim_{n\to+0} \frac{\partial}{\partial n} \log\left(\mathbb{E}_{\mathcal{D}}\Big[\int d\boldsymbol{x}_1 d\boldsymbol{x}_2 w_{\beta_1,\beta_2}(\boldsymbol{x}_1, \boldsymbol{x}_2; \mathcal{D})\Big]^n\right). \tag{13}$$

Although the right-hand side of the above expression is not directly computable for arbitrary $n \in \mathbb{R}_{\geq 0}$, it turns out that under plausible assumptions, this can be evaluated for $n \in \mathbb{N}$ in the limit $N \to \infty$. The replica method is based on the assumption that one can analytically continue this formula, defined only for $n \in \mathbb{N}$, to $n \in \mathbb{R}_{\geq 0}$. On this continuation, the limit $n \to +0$ can be taken formally.

In the process of calculating equation 13, a crucial observation is that from Gaussianity of the design matrices $\boldsymbol{A}^{(k)}$, the products between the design matrices and the feature vectors $\boldsymbol{x}_1, \boldsymbol{x}_2$ are Gaussian random variables. This not only simplifies the high-dimensional nature of the expectation over $\mathcal{D}$, but it also reduces the complexity of the analysis to consider only up to the second moments of these Gaussian random variables. These moments are characterized by the inner product of the feature vectors, which happen to concentrate to deterministic values in the limit $N \to \infty$. One is then left with an integral over this finite number of concentrating scalar variables, which can then be evaluated using the standard saddle-point method, again in the limit $N \to \infty$.

3.2 Equations of State and Generalization Error

The precise asymptotic behavior of the generalization error is given by the solution of a set of non-linear equations, which we refer to as the equations of state. They are provided as follows.

Definition 1 (Equations of state for the First Stage)**.** Let $\mathsf{z}_1, \{\mathsf{x}_\star^{(k)}\}_{k=0}^K$ be i.i.d. standard Gaussian random variables, and denote the expectation with respect to these random variables as $\mathbb{E}_1$. For finite variables $\Theta_1 = \{\{m_1^{(k)}, \hat{m}_1^{(k)}\}_{k=0}^K, q_1, \hat{q}_1, \chi_1, \hat{\chi}_1\}$, define $\{\mathsf{x}_1^{(k)}\}_{k=0}^K, \mathsf{x}_1^{(\mathsf{neg.})}$ as the solution to random optimization problems as follows:

$$
\begin{aligned}
\mathsf{x}_1^{(k)} &= \arg\min_x \mathsf{E}_1^{(k)}(x), & \mathsf{E}_1^{(k)}(x) &= \frac{\hat{q}_1}{2}x^2 - \left(\sqrt{\hat{\chi}_1}\mathsf{z}_1 + \hat{m}_1^{(k)}\mathsf{x}_\star^{(k)}\right)x + \lambda_1|x| && (14)\\
\mathsf{x}_1^{(\mathsf{neg.})} &= \arg\min_x \mathsf{E}_1^{(\mathsf{neg.})}(x), & \mathsf{E}_1^{(\mathsf{neg.})}(x) &= \frac{\hat{q}_1}{2}x^2 - \sqrt{\hat{\chi}_1}\mathsf{z}_1 x + \lambda_1|x|. && (15)
\end{aligned}
$$

Then, Θ_1 is given by the solution of the following set of equations:

$$
\begin{aligned}
q_1 &= \sum_{k=1}^K \pi^{(k)}\mathbb{E}_1\left[\left(\mathsf{x}_1^{(k)}\right)^2\right] + \pi^{(\mathsf{neg.})}\mathbb{E}_1\left[\left(\mathsf{x}_1^{(\mathsf{neg.})}\right)^2\right], & \hat{q}_1 &= \hat{m}_1^{(0)} = \frac{\alpha^{(\mathsf{tot})}}{1+\chi_1},\\
m_1^{(k)} &= \pi^{(k)}\mathbb{E}_1\left[\mathsf{x}_1^{(k)}\mathsf{x}_\star^{(k)}\right], & \hat{m}_1^{(k)} &= \frac{\alpha^{(k)}}{1+\chi_1},\\
\hat{\chi}_1 &= \sum_{k=1}^K \alpha^{(k)}\frac{q_1 - 2(m_1^{(0)} + m_1^{(k)}) + \rho^{(k)}}{(1+\chi_1)^2}, \\
\chi_1 &= \sum_{k=1}^K \pi^{(k)}\mathbb{E}_1\left[\frac{\partial}{\partial \mathsf{z}_1}\mathsf{x}_1^{(k)}\right] + \pi^{(\mathsf{neg.})}\mathbb{E}_1\left[\frac{\partial}{\partial \mathsf{z}_1}\mathsf{x}_1^{(\mathsf{neg.})}\right],
\end{aligned}
\tag{16}
$$

where we have defined $\rho^{(k)} = \pi^{(k)} + \pi^{(0)} + (\sigma^{(k)})^2$, and $\alpha^{(\mathsf{tot})} = \sum_{k=1}^K \alpha^{(k)}$.

Definition 2 (Equations of state of the Second Stage)**.** Let $\mathsf{z}_1, \{\mathsf{x}_\star^{(k)}\}_{k=0}^K$ and $\{\mathsf{x}_1^{(k)}\}_{k=0}^K$ as defined in Definition 1. For finite variables $\Theta_2 = \{\{m_2^{(k)}, \hat{m}_2^{(k)}\}_{k=0}^K, q_2, \hat{q}_2, q_r, \hat{q}_r, \chi_2, \hat{\chi}_2, \chi_r, \hat{\chi}_r\}$, and Θ_1 predefined in Definition 1, let z_2 be a Gaussian random variable with conditional distribution:

$$
\mathsf{z}_2|\mathsf{z}_1 \sim \mathcal{N}\left(\frac{\hat{\chi}_r}{\sqrt{\hat{\chi}_1\hat{\chi}_2}}\mathsf{z}_1, 1 - \frac{\hat{\chi}_r^2}{\hat{\chi}_1\hat{\chi}_2}\right), \tag{17}
$$

and furthermore, denote the joint expectation with respect to $\mathsf{z}_1, \mathsf{z}_2, \{\mathsf{x}_\star^{(k)}\}_{k=0}^K$ as $\mathbb{E}_2$. Let $\{\mathsf{x}_2^{(k)}\}_{k=0}^K, \mathsf{x}_2^{(\mathsf{neg.})}$ be the solution to random optimization problems as follows:

$$
\begin{aligned}
\mathsf{x}_2^{(k)} &= \arg\min_x \mathsf{E}_2^{(k)}(x|\mathsf{x}_1^{(k)}), & \mathsf{E}_2^{(k)}(x|\mathsf{x}_1^{(k)}) &= \frac{\hat{q}_2}{2}x^2 - \left(\sqrt{\hat{\chi}_2}\mathsf{z}_2 + \hat{m}_2^{(k)}\mathsf{x}_\star^{(k)} + \hat{q}_r\mathsf{x}_1^{(k)}\right)x + r(x|\mathsf{x}_1^{(k)}), && (18)\\
\mathsf{x}_2^{(\mathsf{neg.})} &= \arg\min_x \mathsf{E}_2^{(\mathsf{neg.})}(x|\mathsf{x}_1^{(\mathsf{neg.})}), & \mathsf{E}_2^{(\mathsf{neg.})}(x|\mathsf{x}_1^{(\mathsf{neg.})}) &= \frac{\hat{q}_2}{2}x^2 - \left(\sqrt{\hat{\chi}_2}\mathsf{z}_2 + \hat{q}_r\mathsf{x}_1^{(\mathsf{neg.})}\right)x + r(x|\mathsf{x}_1^{(\mathsf{neg.})}), && (19)
\end{aligned}
$$

where $r(x|y) = (\lambda_2 + \Delta\lambda \mathbb{I}[y=0])|x|$. Then, Θ_2 is given by the solution of the following set of equations:

$$\begin{aligned}
q_2 &= \sum_{k=1}^{K} \pi^{(k)} \mathbb{E}_2\left[\left(\mathsf{x}_2^{(k)}\right)^2\right] + \pi^{(\mathsf{neg.})} \mathbb{E}_2\left[\left(\mathsf{x}_2^{(\mathsf{neg.})}\right)^2\right], & \hat{q}_2 &= \frac{\alpha^{(1)}}{1+\chi_2}, \\
q_r &= \sum_{k=1}^{K} \pi^{(k)} \mathbb{E}_2\left[\mathsf{x}_2^{(k)} \mathsf{x}_1^{(k)}\right] + \pi^{(\mathsf{neg.})} \mathbb{E}_2\left[\mathsf{x}_2^{(\mathsf{neg.})} \mathsf{x}_1^{(\mathsf{neg.})}\right], & \hat{q}_r &= -\frac{\alpha^{(1)} A}{1+\chi_2}, \\
m_2^{(k)} &= \pi^{(k)} \mathbb{E}_2\left[\mathsf{x}_2^{(k)} \mathsf{x}_\star^{(k)}\right], & \hat{m}_2^{(k)} &= \frac{\alpha^{(1)} B}{1+\chi_2}, \\
\chi_2 &= \sum_{k=1}^{K} \pi^{(k)} \mathbb{E}_2\left[\frac{\partial}{\partial \mathsf{z}_2} \mathsf{x}_2^{(k)}\right] + \pi^{(\mathsf{neg.})} \mathbb{E}_2\left[\frac{\partial}{\partial \mathsf{z}_2} \mathsf{x}_2^{(\mathsf{neg.})}\right], \\
\hat{\chi}_2 &= \alpha^{(1)} \frac{q_2 + A^2 q_1 + 2A q_r + B^2 \rho^{(1)} - 2B(m_2^{(0)} + m_2^{(1)}) - 2AB(m_1^{(0)} + m_1^{(1)})}{(1+\chi_2)^2}, \\
\chi_r &= \sum_{k=1}^{K} \pi^{(k)} \mathbb{E}_2\left[\frac{\partial}{\partial \mathsf{z}_1} \mathsf{x}_2^{(k)}\right] + \pi^{(\mathsf{neg.})} \mathbb{E}_2\left[\frac{\partial}{\partial \mathsf{z}_1} \mathsf{x}_2^{(\mathsf{neg.})}\right], \\
\hat{\chi}_r &= -\alpha^{(1)} \frac{B\rho^{(1)} - m_2^{(0)} - m_2^{(1)} - A(m_1^{(0)} + m_1^{(1)})}{(1+\chi_1)(1+\chi_2)},
\end{aligned} \tag{20}$$

where $A = \frac{\kappa - \chi_1}{1+\chi_1}, B = 1 - \frac{\chi_r + \kappa\chi_1}{1+\chi_1}$.

Given these equations of state, the generalization error is given from the following claims:

Claim 1 (Generalization error of the first stage)**.** *Given Θ_1, the expected generalization error $\epsilon^{(\mathsf{1st})}$ converges in the large N limit as*

$$\lim_{N\to\infty} \epsilon^{(\mathsf{1st})} = \sum_{k=1}^{K} \alpha^{(k)} \left[q_1 - 2(m_1^{(0)} + m_1^{(k)}) + \rho^{(k)}\right]. \tag{21}$$

Claim 2 (Generalization error of the second stage)**.** *Given Θ_1 and Θ_2, the expected generalization error $\epsilon^{(\mathsf{2nd})}$ converges in the large N limit as*

$$\lim_{N\to\infty} \epsilon^{(\mathsf{2nd})} = \alpha^{(1)} \left[q_2 + \kappa^2 q_1 + 2\kappa q_r + \rho^{(1)} - 2(m_2^{(0)} + m_2^{(1)}) - 2\kappa(m_1^{(0)} + m_1^{(0)})\right]. \tag{22}$$

Claims 1 and 2 indicate that for a given set of hyperparameters, the asymptotic generalization error can be computed by solving a finite set of non-linear equations in Definitions 1 and 2. Therefore, one can obtain the optimal choice of hyperparameters minimizing the generalization error via a numerical root-finding and optimization procedure in finite dimension.

4 Comparison with synthetic finite-size data simulations

To verify our theoretical results, we conduct a series of synthetic numerical experiments to compare the results obtained from Claim 2 and the empirical results for the generalization error obtained from datasets of finite-size systems.

The synthetic data is specified as follows. We consider the case of $K = 2$ classes, with $\alpha^{(1)} = 0.2, 0.4$, or 0.6, and $\alpha^{(2)} = 0.8$. Two types of settings for the common and unique feature vectors are considered; one with $\pi^{(0)} = 0.10, \pi^{(1)} = \pi^{(2)} = 0.09$, and $\pi^{(0)} = 0.15, \pi^{(1)} = \pi^{(2)} = 0.04$. These two settings are chosen to represent the cases where the common feature vector is relatively small compared to the unique feature vectors, and the case where the common feature vector is relatively large, but with the total number of non-zero features underlying each class being the same ($0.19N$). Finally, the noise level of each dataset is set to $\sigma^{(1)} = \sigma^{(2)} = 0.1$. The numerical simulations are performed on 64 random realizations of data with size $N = 16,000$. The second stage generalization error is evaluated by further generating 256 random

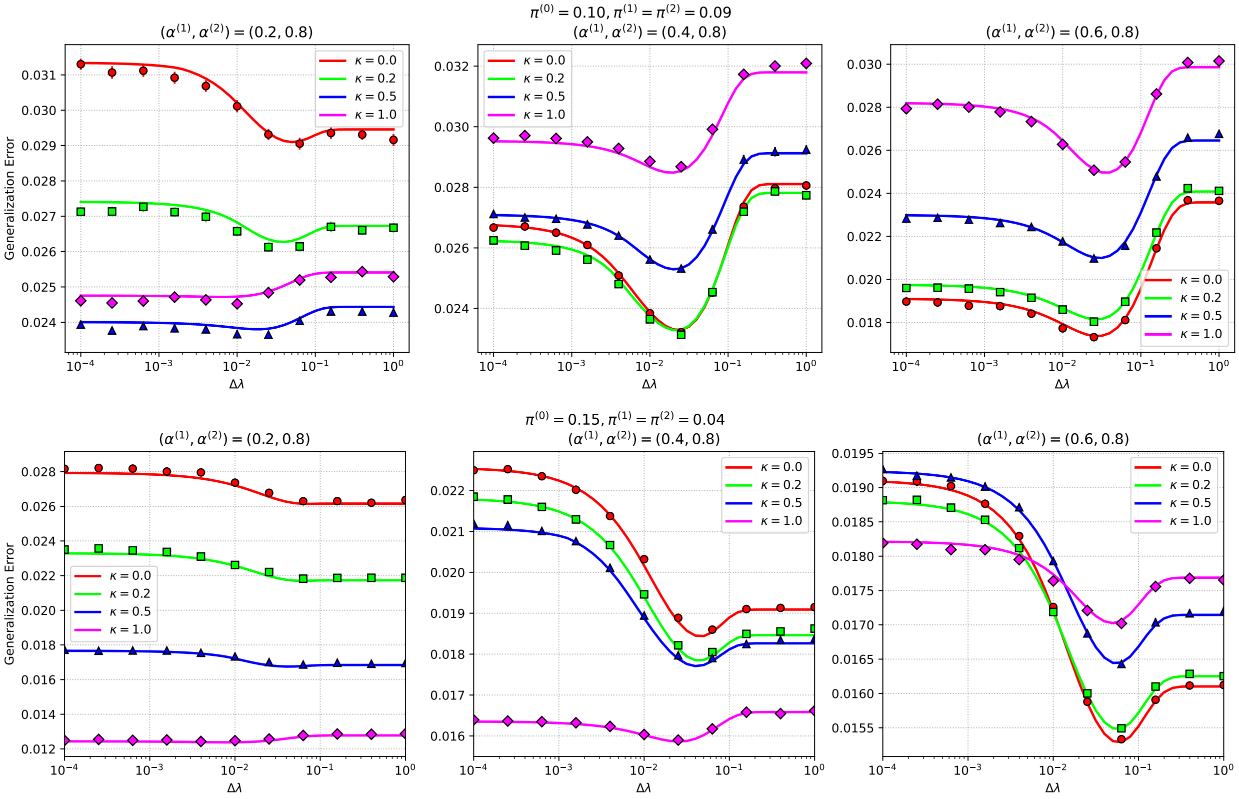

Figure 1: Comparison of generalization error obtained from Claim 2 (solid line) with finite size simulations (markers). Error bars represent standard error obtained from 64 realizations of data.

realizations of $(\tilde{\boldsymbol{A}}^{(1)}, \tilde{\boldsymbol{y}}^{(1)})$ for each dataset and calculating the average empirical generalization error over these realizations. The hyperparameters λ_1 and λ_2 are chosen such that they minimize the generalization error given in Claim 1 and 2 for each stage respectively. The finite size simulations are also performed using the same values of λ_1 and λ_2 used in the theory. In figure 1, we plot the generalization error of the second stage as a function of $\Delta\lambda$ for $\kappa = 0.0, 0.2, 0.5$, and 1.0. As evident from the plot, the generalization error is in fairly good agreement with the theoretical predictions.

Case with scarce target data Let us look closely at the left column of figure 1 where the target sample size is relatively scarce ($\alpha^{(1)} = 0.2$). For both settings of $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)})$, the generalization error is insensitive to the choice of $\Delta\lambda$ when κ is tuned properly. Another important observation is that Trans-Lasso's hyperparameter choice ($\kappa = 1, \Delta\lambda = 0$) seems comparable with the optimal choice regarding the generalization error. The same trend is also seen in the case where $\alpha^{(1)} = 0.4$ and $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.15, 0.04, 0.04)$ (bottom center subfigure of figure 1).

Case with abundant target data Next, we turn our attention to the right column of figure 1 where the target sample size is relatively abundant ($\alpha^{(1)} = 0.6$). In this case, the generalization error is sensitive to the choice of $\Delta\lambda$ but is insensitive to the choice of κ around its optimal value ($\kappa = 0.0$), for both settings of $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)})$. This indicates that under abundant target data, transferring the knowledge with respect to the support of the pretrained feature vector $\hat{\boldsymbol{x}}^{(\mathrm{1st})}$ is more crucial than the actual value of $\hat{\boldsymbol{x}}^{(\mathrm{1st})}$ itself. The same trend can be seen for the case when $\alpha^{(1)} = 0.4$ and $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.15, 0.04, 0.04)$ (top center subfigure of figure 1).

5 Hyperparameter Selection Strategy

The experiments from the last section indicate not only the accuracy of our theoretical predictions but also the practical implications for hyperparameter selection. The observations made from the two cases suggest that there are roughly two behaviors for the optimal choice of hyperparameters: One where the choice of κ is more crucial, and the other where the choice of $\Delta\lambda$ is more important. This observation suggests a simple strategy for hyperparameter selection, where one fixes $\Delta\lambda = 0$ for the former case and $\kappa = 0$ for the latter case. To further investigate the validity of this strategy, we conduct a comparative analysis of the generalization error among the following four strategies:

- $\kappa = 0$ **strategy** Here, we fix κ to zero, while letting $\Delta\lambda$ to be a tunable parameter. This reflects on the insight obtained in subsection 4 for the abundant target data case. The regularization parameters λ_1 and λ_2 are both chosen such that they minimize the generalization error in the first and second stages, respectively.
- $\Delta\lambda = 0$ **strategy** Here, we fix $\Delta\lambda$ to zero, while letting κ to be a tunable parameter. This reflects the insight obtained in subsection 4 for the scarce target data case. The regularization parameters λ_1 and λ_2 are both chosen in the same way as the $\kappa = 0$ strategy.
- **Locally Optimal (LO) strategy** Here, both κ and $\Delta\lambda$ are tunable parameters. The regularization parameters λ_1 and λ_2 are both chosen in the same way as the $\kappa = 0$ strategy.
- **Globally Optimal (GO) strategy** Here, all four parameters $(\lambda_1, \lambda_2, \kappa, \Delta\lambda)$ are tuned such that the second stage generalization error is minimized.

The last strategy requires retraining the first stage estimator $\hat{\boldsymbol{x}}^{(\mathbf{1st})}$ for each choice of $(\lambda_1, \lambda_2, \kappa, \Delta\lambda)$, which is computationally demanding. Even the LO strategy can be problematic when the dataset size is large. Basically, these two strategies are regarded as a benchmark for the other two strategies: $\kappa = 0$ and $\Delta\lambda = 0$ strategies. Below, we compare these four strategies in terms of the second-stage generalization error computed by our theoretical analysis. For simplicity, the case $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.10, 0.09, 0.09)$ is investigated. In figure 2, we plot the second stage generalization error, as well as the optimal choice of hyperparameters for each strategy as a function of σ, for $(\alpha_1, \alpha_2) = (0.2, 0.8), (0.3, 0.8)$, and $(0.4, 0.8)$. Note that the shaded green region indicates the range of σ where the $\kappa = 0$ strategy outperforms the $\Delta\lambda = 0$ strategy, while the shaded red region indicates the opposite.

Let us first examine the behavior of optimal $\Delta\lambda$ across different strategies. With the exception of the case $(\alpha_1, \alpha_2) = (0.2, 0.8)$, the optimal $\Delta\lambda$ values for each strategy (excluding the $\Delta\lambda = 0$ strategy) are approximately equal in the green region. Notably, for $(\alpha^{(1)}, \alpha^{(2)}) = (0.4, 0.8)$, the $\kappa = 0$ strategy demonstrates generalization performance comparable to the LO strategy in this region, while the $\Delta\lambda = 0$ strategy exhibits significantly lower performance. Above a certain noise threshold, the optimal value of $\Delta\lambda$ for both the LO and GO strategies becomes strictly zero. At this point, the LO strategy and the $\Delta\lambda = 0$ strategy coincide, indicating that the $\Delta\lambda = 0$ strategy can potentially be equivalent to the LO strategy.

Next, let us see the behavior of the optimal κ in each strategy. In the green region, where the $\kappa = 0$ strategy outperforms the $\Delta\lambda = 0$ strategy, both the LO and GO strategies actually exhibit finite values of κ. This means that the $\kappa = 0$ strategy is suboptimal. However, it still exhibits a comparable performance with the LO strategy and gives a significant improvement from the $\Delta\lambda = 0$ strategy. This underscores the importance of solely transferring support information; transferring the first stage feature vector itself can be less beneficial. This may be due to the crosstalk noise induced by different datasets: Using the regression results on other datasets than the target can induce noises on the feature vector components specific to the target. Consequently, it may be preferable to transfer more robust information, in this case the support configuration, rather than the noisy vector itself obtained in the first stage.

The above observations indicate that the performance closely comparable to the LO strategy can be achieved across a wide range of noise levels by simply considering either the $\kappa = 0$ or $\Delta\lambda = 0$ strategy; the $\kappa = 0$ strategy is preferred in low noise scenarios while the $\Delta\lambda = 0$ strategy is better in high noise scenarios. For practical purposes, these two strategies should be examined and the best hyperparameters should be chosen based on comparison. This is the main message for practitioners in this paper.

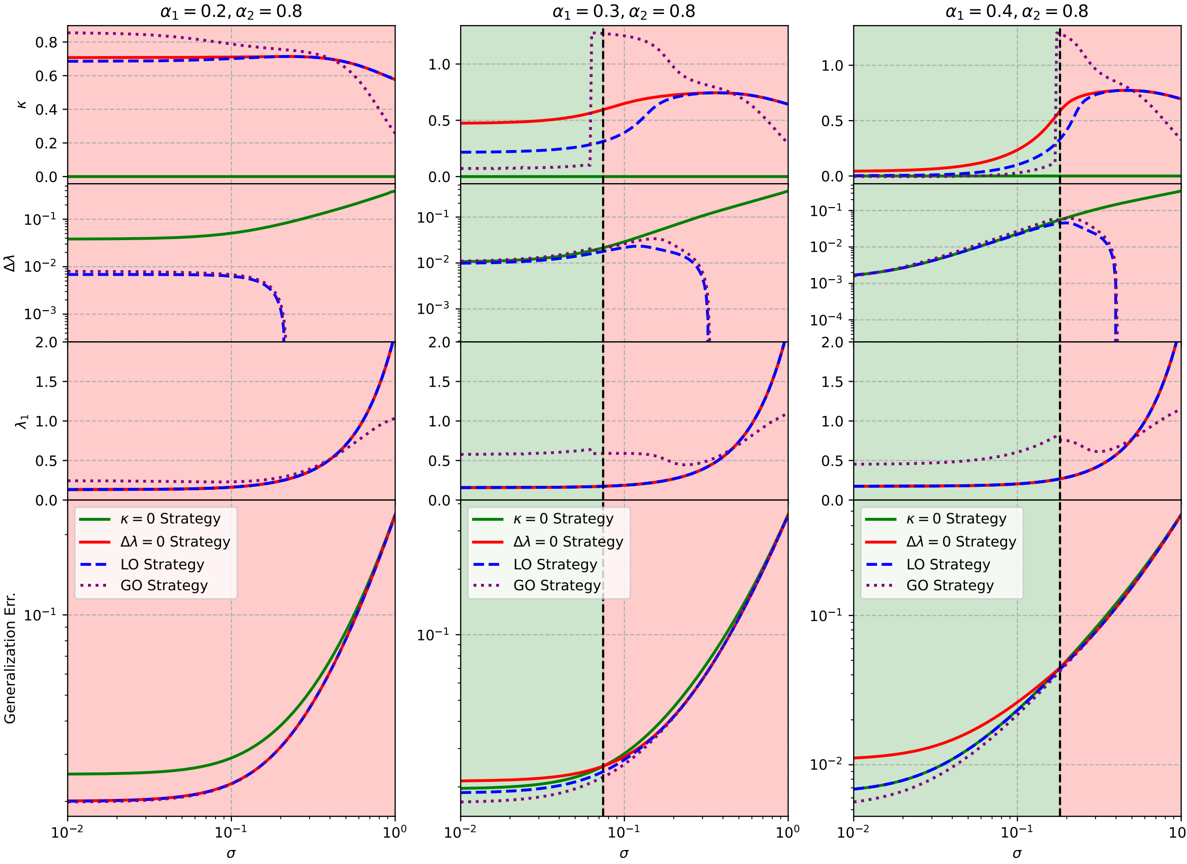

Figure 2: Generalization error and optimal hyperparameters for each strategy as a function of σ for $(\alpha_1, \alpha_2) = (0.2, 0.8), (0.3, 0.8), (0.4, 0.8)$. The region shaded in red indicates the range of σ where the $\Delta\lambda = 0$ strategy outperforms the $\kappa = 0$ strategy, while the region shaded in green indicates the opposite.

Comparison with the original Trans-Lasso The $\Delta\lambda = 0$ strategy provides a lower bound for the generalization error of the Trans-Lasso, as the latter constrains κ to unity instead of treating it as a tunable parameter. Our results demonstrate that the Trans-Lasso is suboptimal compared to strategies that incorporate support information in the second stage, particularly in the green region of figure 2. Notably, for $(\alpha_1, \alpha_2) = (0.4, 0.8)$ and $\sigma = 0.01$, the $\Delta\lambda = 0$ strategy reduces the generalization error by over 30% compared to the Trans-Lasso.

Ubiquity of the simplified hyperparameter selection To further confirm the efficacy of the two simple strategies, we investigate the difference in generalization error between the LO strategy and $\kappa = 0$ or $\Delta\lambda = 0$ strategy. More specifically, let $\epsilon_{\rm LO}$, $\epsilon_{\kappa=0}$ and $\epsilon_{\Delta\lambda=0}$ be the generalization error of the corresponding strategies. Figure 3 shows the minimum ratio $\min\{\epsilon_{\Delta\lambda=0}, \epsilon_{\kappa=0}\}/\epsilon_{\rm LO}$ for various values of (α_1, α_2). For comparison, the same values are calculated for the Pretraining Lasso and Trans-Lasso, i.e. $\epsilon_{\rm Pretrain}/\epsilon_{\rm LO}$ and $\epsilon_{\rm Trans}/\epsilon_{\rm LO}$, with $\epsilon_{\rm Pretrain}$ and $\epsilon_{\rm Trans}$ being the generalization error with the parameters chosen optimally (note that λ_1 and λ_2 are chosen in the same way as the LO strategy). The results indicate that, across a broad range of parameters, at least one of the simplified strategies (either $\Delta\lambda = 0$ or $\kappa = 0$) achieves a generalization error within 10% of that obtained by the more complex LO strategy. Notably, substantial relative differences are observed only within a narrow region of the parameter space $(\alpha^{(1)}, \alpha^{(2)})$, highlighting the effectiveness of the simplified approach across a wide range of problem settings. We also report that for the case $\sigma = 0.5$, the $\Delta\lambda = 0$ strategy always coincided with the LO strategy in the region $(\alpha^{(1)}, \alpha^{(2)}) \in [0.05, 0.45] \times [0.1, 0.9]$. Results consistent with the above observations were also seen for other settings of $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)})$; see Appendix B.

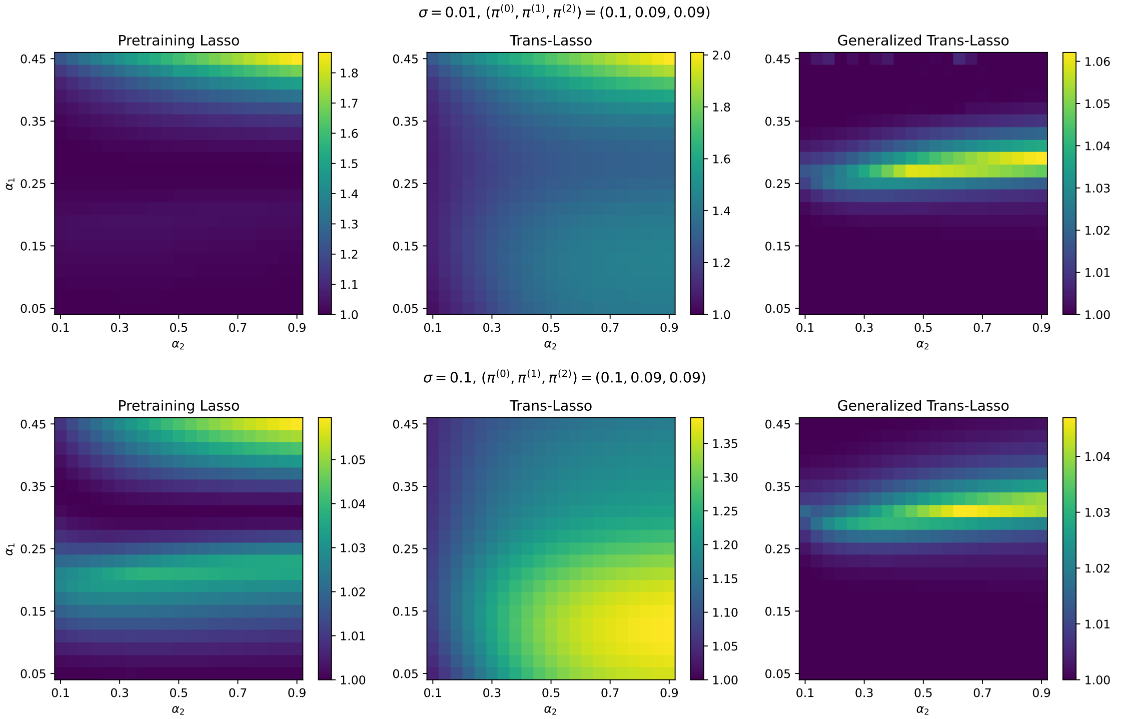

Figure 3: Ratios $\min\{\epsilon_{\Delta\lambda=0}, \epsilon_{\kappa=0}\}/\epsilon_{\mathrm{LO}}, \epsilon_{\mathrm{Pretrain}}/\epsilon_{\mathrm{LO}}$ and $\epsilon_{\mathrm{Trans}}/\epsilon_{\mathrm{LO}}$ under setting $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.1, 0.09, 0.09)$ and noise level $\sigma = 0.01$ (top figure) and 0.1 (bottom figure) for various values of $(\alpha^{(1)}, \alpha^{(2)})$. Note that each heatmap has its own color scale bar; see Appendix B for a direct comparison with shared color scale bars between the $\kappa = 0$ or $\Delta\lambda = 0$ strategy with Pretraining-Lasso and Trans-Lasso.

The role of λ_1 For low noise scenarios, the GO strategy tends to prefer larger λ_1 in the first stage compared to the other strategies. This observation may be beneficial when the noise level of each dataset is known a priori: We may tune the λ_1 value to a larger value than the one obtained from the LO or its proxy strategies ($\kappa = 0$ or $\Delta\lambda = 0$ strategies). This offers an opportunity to enhance our prediction beyond the LO strategy while avoiding the complicated retraining process necessary for the GO strategy.

6 Real Data Application: IMDb Dataset

To verify whether the above insights can be generalized to real-world scenarios, we here conduct experiments on the IMDb dataset. The IMDb dataset comprises a pair of 25,000 user-generated movie reviews from the Internet Movie Database, each being devoted to training and test datasets. The dataset is evenly split into highly positive or negative reviews, with negative labels assigned to ratings given 4 or lower out of 10, and positive labels to ratings of 7 or higher. Movies labeled with only one of the three most common genres in the dataset, drama, comedy, and horror, are considered. By representing the reviews as binary feature vectors using bag-of-words while ignoring words with less than 5 occurrences, we are left with 27743 features in total, with 8286 drama, 5027 comedy, and 3073 horror movies in the training dataset, and 9937 drama, 4774 comedy, and 3398 horror movies in the test dataset. The datasets can be acquired from Supplementary Materials of Gross & Tibshirani (2016).

For sake of completeness, we calculate the test error for $(\kappa, \Delta\lambda/\lambda_2) \in \{0.0, 0.1, 0.2, \cdots, 1.5\} \times \{0.0, 0.1, 0.2, \cdots, 1.0\}$ to see if our insights are qualitatively consistent with experiments on real data. For each pair of $(\kappa, \Delta\lambda/\lambda_2)$, λ_1 and λ_2 are determined by a 10-fold cross-validation procedure, where the training dataset is further split into a learning dataset and a validation dataset. Since λ_1 (and consequently $\hat{\boldsymbol{x}}^{(1\mathrm{st})}$) must be selected oblivious to the validation dataset, we perform Approximate Leave-One-Out Cross-Validation (Obuchi & Kabashima, 2016; Rad & Maleki, 2020; Stephenson & Broderick, 2020) on the learning

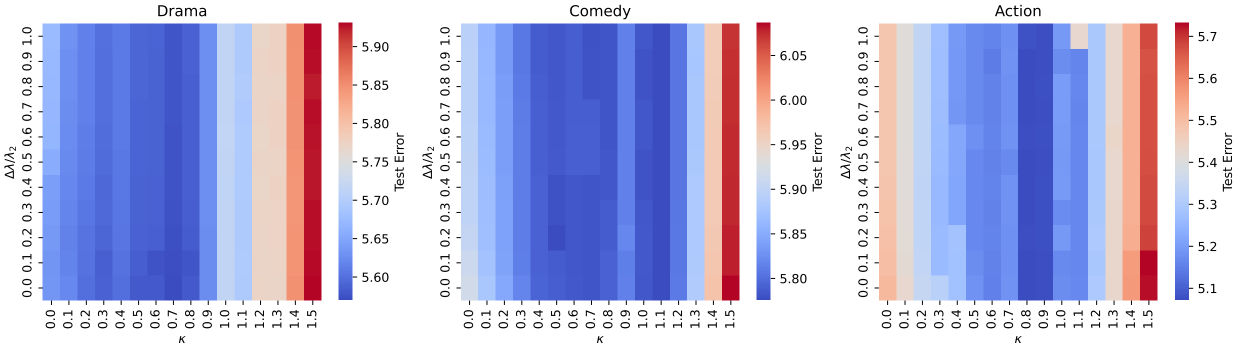

Figure 4: Second stage test error for each IMDB genre plotted against varying values of hyperparameters κ and $\Delta\lambda/\lambda_2$. The plots demonstrate that the effect of $\Delta\lambda$ on generalization performance is tiny across genres.

Method	Drama	Comedy	Action
Gen. Trans-Lasso, $\Delta\lambda = 0$	**5.58**	**5.78**	**5.07**
Gen. Trans-Lasso, $\kappa = 0$	5.67	5.90	5.48
Gen. Trans-Lasso, LO	**5.58**	**5.78**	5.19
Pretraining Lasso	5.63	**5.78**	5.18
Trans-Lasso	5.71	**5.78**	5.17

Table 1: Comparison of test errors for various Lasso-based methods and hyperparameter strategies applied to the IMDb dataset.

dataset, which allows one to estimate the leave-one-out cross-validation error deterministically without actually performing the computationally expensive procedure. For the sake of comparison, we also compare with the performance of Pretraining Lasso and Trans-Lasso, with λ_1 and λ_2 chosen using the same method as above.

In Figure 4, we present the second stage test error for each genre across various values of $(\kappa, \Delta\lambda/\lambda_2)$. The plot reveals that $\Delta\lambda$ is essentially a redundant hyperparameter having almost no influence on generalization performance. This observation strongly supports the efficacy of the $\Delta\lambda = 0$ strategy in the present dataset. We compare the test errors obtained from three approaches within our generalized Trans-Lasso framework: The $\Delta\lambda = 0$ strategy, the $\kappa = 0$ strategy, and LO strategy. Additionally, we include test errors from Pretraining Lasso and the original Trans-Lasso for comparison, whose results are summarized in Table 1. Recall that Trans-Lasso fixes κ to unity, while Pretraining Lasso employs $(\kappa, \Delta\lambda/\lambda_2) = (1-s, (1-s)/s)$ for hyperparameter $s \in [0,1]$. In the experiments, we choose s from $\{0.0, 0.1, 0.2, \cdots, 1.0\}$ via cross-validation. Our analysis demonstrates that the hyperparameter choices made by both the original Trans-Lasso and Pretraining Lasso are suboptimal compared to the $\Delta\lambda = 0$ strategy of our generalized Trans-Lasso algorithm. Note that the LO strategy can potentially select hyperparameters exhibiting higher test error compared to the $\Delta\lambda = 0$ strategy (as seen in the Action genre), as they are chosen via cross-validation and the true generalization error is not directly minimized.

7 Conclusion

In this work, we have conducted a sharp asymptotic analysis of the generalized Trans-Lasso algorithm, precisely characterizing the effect of hyperparameters on its generalization performance. Our theoretical calculations reveal that near-optimal generalization in this transfer learning algorithm can be achieved by focusing on just one of two modes of knowledge transfer from the source dataset. The first mode transfers only the support information of the features obtained from all source data, while the second transfers the actual configuration of the feature vector obtained from all sources. Experiments on the IMDb dataset confirm that this simple hyperparameter strategy is effective, outperforming conventional Lasso-type transfer algorithms.

References

Urte Adomaityte, Leonardo Defilippis, Bruno Loureiro, and Gabriele Sicuro. High-dimensional robust regression under heavy-tailed data: Asymptotics and universality. *arXiv preprint arXiv:2309.16476*, 2023.

Trambak Banerjee, Gourab Mukherjee, and Wenguang Sun. Adaptive sparse estimation with side information. *Journal of the American Statistical Association*, 115(532):2053–2067, 2020.

Jean Barbier and Nicolas Macris. The adaptive interpolation method: a simple scheme to prove replica formulas in bayesian inference. *Probability theory and related fields*, 174:1133–1185, 2019.

Jean Barbier, Florent Krzakala, Nicolas Macris, Léo Miolane, and Lenka Zdeborová. Optimal errors and phase transitions in high-dimensional generalized linear models. *Proceedings of the National Academy of Sciences*, 116(12):5451–5460, 2019.

Hamsa Bastani. Predicting with proxies: Transfer learning in high dimension. *Management Science*, 67(5): 2964–2984, 2021.

Mohsen Bayati and Andrea Montanari. The Lasso risk for gaussian matrices. *IEEE Transactions on Information Theory*, 58(4):1997–2017, 2011.

Pierre C Bellec and Yiwei Shen. Derivatives and residual distribution of regularized m-estimators with application to adaptive tuning. In *Conference on Learning Theory*, pp. 1912–1947. PMLR, 2022.

Pierre C. Bellec and Cun-Hui Zhang. Second-order Stein: SURE for SURE and other applications in high-dimensional inference. *The Annals of Statistics*, 49(4):1864 – 1903, 2021.

Pierre C. Bellec, Guillaume Lecué, and Alexandre B. Tsybakov. Slope meets Lasso: Improved oracle bounds and optimality. *The Annals of Statistics*, 46(6B):3603 – 3642, 2018.

Emmanuel J. Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

Kabir Aladin Chandrasekher, Ashwin Pananjady, and Christos Thrampoulidis. Sharp global convergence guarantees for iterative nonconvex optimization with random data. *The Annals of Statistics*, 51(1):179 – 210, 2023.

Patrick Charbonneau, Enzo Marinari, Marc Mézard, Giorgio Parisi, Federico Ricci-Tersenghi, Gabriele Sicuro, and Francesco Zamponi. *Spin Glass Theory and Far Beyond*. WORLD SCIENTIFIC, 2023.

Erin Craig, Mert Pilanci, Thomas Le Menestrel, Balasubramanian Narasimhan, Manuel Rivas, Roozbeh Dehghannasiri, Julia Salzman, Jonathan Taylor, and Robert Tibshirani. Pretraining and the Lasso. *arXiv preprint arXiv:2401.1291*, 2024.

Cedric Gerbelot, Alia Abbara, and Florent Krzakala. Asymptotic errors for teacher-student convex generalized linear models (or: How to prove Kabashima's replica formula). *IEEE Transactions on Information Theory*, 69(3):1824–1852, 2023.

Samuel M. Gross and Robert Tibshirani. Data shared Lasso: A novel tool to discover uplift. *Computational Statistics and Data Analysis*, 101:226–235, 2016.

Ehsan Hajiramezanali, Siamak Zamani Dadaneh, Alireza Karbalayghareh, Mingyuan Zhou, and Xiaoning Qian. Bayesian multi-domain learning for cancer subtype discovery from next-generation sequencing count data. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Adel Javanmard and Andrea Montanari. State evolution for general approximate message passing algorithms, with applications to spatial coupling. *Information and Inference: A Journal of the IMA*, 2(2):115–144, 2013.

Yoshiyuki Kabashima, Tadashi Wadayama, and Toshiyuki Tanaka. A typical reconstruction limit for compressed sensing based on ℓ_p-norm minimization. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(09):L09003, 2009.

Sai Li, T. Tony Cai, and Hongzhe Li. Transfer Learning for High-Dimensional Linear Regression: Prediction, Estimation and Minimax Optimality. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):149–173, 11 2021.

Sai Li, T Tony Cai, and Hongzhe Li. Transfer learning in large-scale gaussian graphical models with false discovery rate control. *Journal of the American Statistical Association*, 118(543):2171–2183, 2023.

Sai Li, Linjun Zhang, T Tony Cai, and Hongzhe Li. Estimation and inference for high-dimensional generalized linear models with knowledge transfer. *Journal of the American Statistical Association*, 119(546):1274–1285, 2024.

Sarah F. McGough, Svetlana Lyalina, Devin Incerti, Yunru Huang, Stefka Tyanova, Kieran Mace, Chris Harbron, Ryan Copping, Balasubramanian Narasimhan, and Robert Tibshirani. Prognostic pan-cancer and single-cancer models: A large-scale analysis using a real-world clinico-genomic database. *medRxiv preprint 2023.12.18.23300166*, 2023.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436 – 1462, 2006.

M Mezard, G Parisi, and M Virasoro. *Spin Glass Theory and Beyond*. WORLD SCIENTIFIC, 1986.

Léo Miolane and Andrea Montanari. The distribution of the Lasso: Uniform control over sparse balls and adaptive parameter tuning. *The Annals of Statistics*, 49(4), 2021.

Tomoyuki Obuchi and Yoshiyuki Kabashima. Cross validation in LASSO and its acceleration. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(5):053304, 2016.

Tomoyuki Obuchi and Yoshiyuki Kabashima. Semi-analytic resampling in Lasso. *Journal of Machine Learning Research*, 20(70):1–33, 2019.

Koki Okajima and Takashi Takahashi. Asymptotic dynamics of alternating minimization for bilinear regression. *arXiv preprint arXiv:2402.04751*, 2024.

Koki Okajima, Xiangming Meng, Takashi Takahashi, and Yoshiyuki Kabashima. Average case analysis of Lasso under ultra sparse conditions. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 11317–11330. PMLR, 25–27 Apr 2023.

E. Ollier and V. Viallon. Regression modelling on stratified data with the Lasso. *Biometrika*, 104(1):83–96, 2017.

Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

Raphael Petegrosso, Sunho Park, Tae Hyun Hwang, and Rui Kuang. Transfer learning across ontologies for phenome–genome association prediction. *Bioinformatics*, 33(4):529–536, 11 2016.

Kamiar Rahnama Rad and Arian Maleki. A scalable estimate of the out-of-sample prediction error via approximate leave-one-out cross-validation. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4):965–996, 2020.

Sundeep Rangan, Vivek Goyal, and Alyson K Fletcher. Asymptotic analysis of map estimation via the replica method and compressed sensing. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

Luca Saglietti and Lenka Zdeborová. Solvable model for inheriting the regularization through knowledge distillation. In *Mathematical and Scientific Machine Learning*, pp. 809–846. PMLR, 2022.

William Stephenson and Tamara Broderick. Approximate cross-validation in high dimensions with guarantees. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2424–2434. PMLR, 26–28 Aug 2020.

Mihailo Stojnic. A framework to characterize performance of Lasso algorithms. *arXiv preprint arXiv:1303.7291*, 2013.

Fei Sun and Qi Zhang. Robust transfer learning of high-dimensional generalized linear model. *Physica A: Statistical Mechanics and its Applications*, 618:128674, 2023.

Masaaki Takada and Hironori Fujisawa. Transfer learning via ℓ_1 regularization. *Advances in Neural Information Processing Systems*, 33:14266–14277, 2020.

Takashi Takahashi. The role of pseudo-labels in self-training linear classifiers on high-dimensional gaussian mixture data. *arXiv preprint arXiv:2205.07739*, 2024.

Takashi Takahashi and Yoshiyuki Kabashima. A statistical mechanics approach to de-biasing and uncertainty estimation in Lasso for random measurements. *Journal of Statistical Mechanics: Theory and Experiment*, 2018(7):073405, 2018.

Christos Thrampoulidis, Ehsan Abbasi, and Babak Hassibi. Precise error analysis of regularized m-estimators in high dimensions. *IEEE Transactions on Information Theory*, 64(8):5592–5628, 2018.

Ye Tian and Yang Feng. Transfer learning under high-dimensional generalized linear models. *Journal of the American Statistical Association*, 118(544):2684–2697, 2023.

Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

Lisa Torrey and Jude Shavlik. *Transfer Learning. In: Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques.* IGI Global, 2010.

Turki Turki, Zhi Wei, and Jason T. L. Wang. Transfer learning approaches to improve drug sensitivity prediction in multiple myeloma patients. *IEEE Access*, 5:7381–7393, 2017.

Mikko Vehkaperä, Yoshiyuki Kabashima, and Saikat Chatterjee. Analysis of regularized LS reconstruction and random matrix ensembles in compressed sensing. *IEEE Transactions on Information Theory*, 62(4): 2100–2124, 2016.

Martin J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1 -constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009.

Meiyu Wang, Yun Lin, Qiao Tian, and Guangzhen Si. Transfer learning promotes 6G wireless communications: Recent advances and future challenges. *IEEE Transactions on Reliability*, 70(2):790–807, 2021.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3:1–40, 2016.

Peng Zhao and Bin Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7 (90):2541–2563, 2006.

Yin Zhu, Yuqiang Chen, Zhongqi Lu, Sinno Jialin Pan, Gui-Rong Xue, Yong Yu, and Qiang Yang. Heterogeneous transfer learning for image classification. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pp. 1304–1309. AAAI Press, 2011.

Fuzhen Zhuang, Ping Luo, Hui Xiong, Qing He, Yuhong Xiong, and Zhongzhi Shi. Exploiting associations between word clusters and document classes for cross-domain text categorization. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(1):100–114, 2011.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2021.

A Detailed derivation of Asymptotic analysis

Here, we provide a detailed derivation for the asymptotic formulae for the generalization errors $\epsilon^{(\mathsf{1st})}$ and $\epsilon^{(\mathsf{2nd})}$. Recall that the objective of our analysis is to calculate the data average of the $n-$th power of

$$\Phi(\mathcal{D},\tilde{\mathcal{D}}) := \int d\boldsymbol{x}_1 d\boldsymbol{x}_2 e^{-\beta_1\mathcal{L}^{(\mathsf{1st})}(\boldsymbol{x}_1;\mathcal{D})-\beta_2\mathcal{L}^{(\mathsf{2nd})}(\boldsymbol{x}_2;\boldsymbol{x}_1,\mathcal{D}^{(1)})+t_1\sum_{k=1}^K\|\tilde{\boldsymbol{y}}^{(k)}-\tilde{\boldsymbol{A}}^{(k)}\boldsymbol{x}_1\|_2^2+t_2\|\tilde{\boldsymbol{y}}^{(1)}-\tilde{\boldsymbol{A}}^{(1)}(\kappa\boldsymbol{x}_1+\boldsymbol{x}_2)\|_2^2}, \tag{23}$$

in the successive limit $\lim_{\beta_2\to\infty}\lim_{\beta_1\to\infty}$. For $n\in\mathbb{N}$, one can express this as

$$\begin{aligned}
\mathbb{E}_{\mathcal{D},\tilde{\mathcal{D}}}\Big[\Phi^n(\mathcal{D},\tilde{\mathcal{D}})\Big] &= \int\left(\prod_{a=1}^n d\boldsymbol{x}_{1,a}d\boldsymbol{x}_{2,a}\right)\exp\Big[-\sum_{a=1}^n\Big(\beta_1\lambda_1\|\boldsymbol{x}_{1,a}\|_1+\beta_2 r(\boldsymbol{x}_{2,a}|\boldsymbol{x}_{1,a})\Big)\Big]\ \mathcal{L},\\
\mathcal{L} =& \mathbb{E}_{\mathcal{D},\tilde{\mathcal{D}}}\prod_{\mu=1}^M\prod_{a=1}^n\exp\Bigg[-\frac{\beta_1}{2}\sum_{k=1}^K\big(y_\mu^{(k)}-\boldsymbol{A}_{\mu,:}^{(k)}\cdot\boldsymbol{x}_{1,a}\big)^2-t_1\sum_{k=1}^K\big(\tilde{y}_\mu^{(k)}-\tilde{\boldsymbol{A}}_{\mu,:}^{(k)}\cdot\boldsymbol{x}_{1,a}\big)^2\\
&-\frac{\beta_2}{2}\big(y_\mu^{(1)}-\boldsymbol{A}_{\mu,:}^{(1)}\cdot(\kappa\boldsymbol{x}_{1,a}+\boldsymbol{x}_{2,a})\big)^2-t_2\big(\tilde{y}_\mu^{(1)}-\tilde{\boldsymbol{A}}_{\mu,:}^{(1)}\cdot(\kappa\boldsymbol{x}_{1,a}+\boldsymbol{x}_{2,a})\big)^2\Bigg].
\end{aligned}\tag{24}$$

Conditioned on $\{\boldsymbol{x}_{1,a},\boldsymbol{x}_{2,a}\}_{a=1}^n$, the distribution over $\{\boldsymbol{A}_{\mu,:}^{(k)}\cdot\boldsymbol{x}_{1,a},\tilde{\boldsymbol{A}}_{\mu,:}^{(k)}\cdot\boldsymbol{x}_{1,a},\boldsymbol{A}_{\mu,:}^{(1)}\cdot\boldsymbol{x}_{2,a},\tilde{\boldsymbol{A}}_{\mu,:}^{(1)}\cdot\boldsymbol{x}_{2,a},y_\mu,\tilde{y}_\mu\}_{\mu=1}^M$ is i.i.d. with respect to μ, whose profile is identical to that of the random variables

$$h_{1,a}^{(k)} := \boldsymbol{a}^{(k)}\cdot\boldsymbol{x}_{1,a},\qquad h_{2,a} := \boldsymbol{a}^{(1)}\cdot\boldsymbol{x}_{2,a},\qquad \tilde{h}_{1,a}^{(k)} := \tilde{\boldsymbol{a}}^{(k)}\cdot\boldsymbol{x}_{1,a},\qquad \tilde{h}_{2,a} := \tilde{\boldsymbol{a}}^{(1)}\cdot\boldsymbol{x}_{2,a}, \tag{25}$$

$$H^{(k)} := \boldsymbol{a}_{\mathcal{I}(k)}^{(k)}\cdot\boldsymbol{x}_\star^{(k)}+\boldsymbol{a}_{\mathcal{I}(0)}^{(k)}\cdot\boldsymbol{x}_\star^{(0)}+\xi^{(k)},\qquad \tilde{H}^{(k)} := \tilde{\boldsymbol{a}}_{\mathcal{I}(k)}^{(k)}\cdot\boldsymbol{x}_\star^{(k)}+\tilde{\boldsymbol{a}}_{\mathcal{I}(0)}^{(k)}\cdot\boldsymbol{x}_\star^{(0)}+\tilde{\xi}^{(k)}, \tag{26}$$

with $\{\boldsymbol{a}^{(k)},\tilde{\boldsymbol{a}}^{(k)}\}_{k=1}^K$ being a set of N-dimensional centered Gaussian vectors with i.i.d. elements of variance $1/N$, and $\xi^{(k)},\tilde{\xi}^{(k)}$ being independent centered Gaussian random variables with variance $(\sigma^{(k)})^2$ for $k=1,\cdots,K$. Note that the random variables with different index k are independent of each other. Following this observation, $\mathcal{L}$ can be expressed alternatively as $\mathcal{L} := \prod_{k=1}^K(\mathbb{E}L^{(k)})^{M^{(k)}}(\mathbb{E}\tilde{L}^{(k)})^{M^{(k)}}$, where

$$L^{(k)} = \prod_{a=1}^n\exp\Big[-\frac{\beta_1}{2}(H^{(k)}-h_{1,a}^{(k)})^2-\frac{\delta_{1k}\beta_2}{2}(H^{(1)}-\kappa h_{1,a}^{(1)}-h_{2,a})^2\Big], \tag{27}$$

$$\tilde{L}^{(k)} = \prod_{a=1}^n\exp\Big[-t_1(\tilde{H}^{(k)}-\tilde{h}_{1,a}^{(k)})^2-\delta_{1k}t_2(\tilde{H}^{(1)}-\kappa\tilde{h}_{1,a}^{(1)}-\tilde{h}_{2,a})^2\Big]. \tag{28}$$

Here, $\delta_{ij}=\mathbb{I}[i=j]$ denotes the Kronecker delta, which is unity if $i=j$ and zero otherwise. From the Gaussian nature of $\{\boldsymbol{a}^{(k)},\tilde{\boldsymbol{a}}^{(k)}\}_{k=1}^K$ and $\{\xi^{(k)},\tilde{\xi}^{(k)}\}_{k=1}^K$, the random variables $\{h_{1,a}^{(k)}\}_{a,k=1}^{n,K}$, $\{h_{2,a}\}_{a=1}^n$, and $\{H^{(k)}\}_{k=1}^K$ are all centered Gaussians with the following covariance structure:

$$
\begin{aligned}
\mathbb{E}[h_{1,a}^{(k)}h_{1,b}^{(l)}] &= \delta_{kl}Q_1^{ab}, \qquad Q_1^{ab} := \frac{1}{N}\boldsymbol{x}_{1,a}\cdot\boldsymbol{x}_{1,b} = \frac{1}{N}\left(\sum_{k'=0}^{K}\boldsymbol{x}_{1,a}^{(k')}\cdot\boldsymbol{x}_{1,b}^{(k')} + \boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{1,b}^{(\mathsf{neg.})}\right),\\
\mathbb{E}[h_{2,a}h_{2,b}] &= Q_2^{ab}, \qquad Q_2^{ab} := \frac{1}{N}\boldsymbol{x}_{2,a}\cdot\boldsymbol{x}_{2,b} = \frac{1}{N}\left(\sum_{k'=0}^{K}\boldsymbol{x}_{2,a}^{(k')}\cdot\boldsymbol{x}_{2,b}^{(k')} + \boldsymbol{x}_{2,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{2,b}^{(\mathsf{neg.})}\right),\\
\mathbb{E}[h_{1,a}^{(k)}h_{2,a}] &= \delta_{1k}Q_r^{ab}, \qquad Q_r^{ab} := \frac{1}{N}\boldsymbol{x}_{1,a}\cdot\boldsymbol{x}_{2,b} = \frac{1}{N}\left(\sum_{k'=0}^{K}\boldsymbol{x}_{1,a}^{(k')}\cdot\boldsymbol{x}_{2,b}^{(k')} + \boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{2,b}^{(\mathsf{neg.})}\right),\\
\mathbb{E}[h_{1,a}^{(k)}H^{(l)}] &= \delta_{kl}\big(m_1^{(k)} + m_1^{(0)}\big), \qquad m_{1,a}^{(k')} := \frac{1}{N}\boldsymbol{x}_{1,a}^{(k')}\cdot\boldsymbol{x}_\star^{(k')} \quad (k' = 0, 1, \cdots, K),\\
\mathbb{E}[h_{2,a}H^{(k)}] &= \delta_{k1}\big(m_2^{(k)} + m_2^{(0)}\big), \qquad m_{2,a}^{(k')} := \frac{1}{N}\boldsymbol{x}_{2,a}^{(k')}\cdot\boldsymbol{x}_\star^{(k')} \quad (k' = 0, 1, \cdots, K),\\
\mathbb{E}[H^{(k)}H^{(l)}] &= \delta_{kl}\left[\frac{1}{N}\|\boldsymbol{x}_\star^{(k)}\|_2^2 + \frac{1}{N}\|\boldsymbol{x}_\star^{(0)}\|_2^2 + (\sigma^{(k)})^2\right] = \delta_{kl}\rho^{(k)},
\end{aligned}
\tag{29}
$$

for $1 \le k, l \le K$ and $1 \le a, b \le n$. Here, $\boldsymbol{x}_{1,a}^{(k)}$ and $\boldsymbol{x}_{2,a}^{(k)}$ are defined as the subvectors of $\boldsymbol{x}_{1,a}$ and $\boldsymbol{x}_{2,a}$ indexed by $\mathcal{I}(k)$ respectively, while $\boldsymbol{x}_{1,a}^{(\mathsf{neg.})}$ and $\boldsymbol{x}_{2,a}^{(\mathsf{neg.})}$ are both subvectors of $\boldsymbol{x}_{1,a}$ and $\boldsymbol{x}_{2,a}$ whose indices are not included in $\bigcup_{k=0}^{K}\mathcal{I}(k)$. The same covariance structure also follows for the random variables $\{\tilde{h}_{1,a}^{(k)}\}_{a,k=1}^{n,K}$, $\{\tilde{h}_{2,a}\}_{a=1}^{n}$, and $\{\tilde{H}^{(k)}\}_{k=1}^{K}$. Define the average with respect to the random variables given in equation 29, conditioned on $\Omega = \{Q_1^{ab}, Q_2^{ab}, Q_r^{ab}, m_{1,a}^{(k)}, m_{2,a}^{(k)}\}$, as $\mathbb{E}_{|\Omega}$. Inserting the trivial identities based on the delta function corresponding to the definitions of Ω given in equation 29, we can rewrite equation 24 up to a trivial multiplicative constant as

$$
\begin{aligned}
&\int d\Omega \int \left(\prod_{a=1}^{n}\prod_{k=0}^{K} d\boldsymbol{x}_{1,a}^{(k)}d\boldsymbol{x}_{2,a}^{(k)}\right)\exp\left[-\sum_{a=1}^{n}\Big(\beta_1\lambda_1\|\boldsymbol{x}_{1,a}\|_1 + \beta_2 r(\boldsymbol{x}_{1,a}|\boldsymbol{x}_{2,a})\Big) + \sum_{k=1}^{K} M^{(k)}\log(\mathbb{E}_{|\Omega}L^{(k)})(\mathbb{E}_{|\Omega}\tilde{L}^{(k)})\right]\\
&\times\prod_{a\le b=1}^{n}\delta\left(NQ_1^{ab} - \sum_{k'=0}^{K}\boldsymbol{x}_{1,a}^{(k')}\cdot\boldsymbol{x}_{1,b}^{(k')} - \boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{1,b}^{(\mathsf{neg.})}\right)\delta\left(NQ_2^{ab} - \sum_{k'=0}^{K}\boldsymbol{x}_{2,a}^{(k')}\cdot\boldsymbol{x}_{2,b}^{(k')} - \boldsymbol{x}_{2,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{2,b}^{(\mathsf{neg.})}\right)\\
&\times\prod_{a,b=1}^{n}\delta\left(NQ_r^{ab} - \sum_{k'=0}^{K}\boldsymbol{x}_{1,a}^{(k')}\cdot\boldsymbol{x}_{2,b}^{(k')} - \boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{2,b}^{(\mathsf{neg.})}\right)\mathbb{E}_\star\prod_{a=1}^{n}\prod_{k=0}^{K}\delta\big(Nm_{1,a}^{(k)} - \boldsymbol{x}_{1,a}^{(k)}\cdot\boldsymbol{x}_\star^{(k)}\big)\delta\big(Nm_{2,a}^{(k)} - \boldsymbol{x}_{2,a}^{(k)}\cdot\boldsymbol{x}_\star^{(k)}\big),
\end{aligned}
\tag{30}
$$

where $\mathbb{E}_\star$ is the average with respect to the set of ground truth vectors $\{\boldsymbol{x}_\star^{(k)}\}_{k=0}^{K}$, whose elements are i.i.d. according to a standard normal distribution. Note that the delta functions can be expressed alternatively using its Fourier representation as

$$
\begin{aligned}
\delta\left(NQ_1^{ab} - \sum_{k'=0}^{K}\boldsymbol{x}_{1,a}^{(k')}\cdot\boldsymbol{x}_{1,b}^{(k')} - \boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{1,b}^{(\mathsf{neg.})}\right) &= \int_{\mathbb{C}} d\hat{Q}_1^{ab} e^{\hat{Q}_1^{ab}\left(NQ_1^{ab} - \sum_{k'=0}^{K}\boldsymbol{x}_{1,a}^{(k')}\cdot\boldsymbol{x}_{1,b}^{(k')} - \boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{1,b}^{(\mathsf{neg.})}\right)},\\
\delta\left(NQ_2^{ab} - \sum_{k'=0}^{K}\boldsymbol{x}_{2,a}^{(k')}\cdot\boldsymbol{x}_{2,b}^{(k')} - \boldsymbol{x}_{2,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{2,b}^{(\mathsf{neg.})}\right) &= \int_{\mathbb{C}} d\hat{Q}_2^{ab} e^{\hat{Q}_2^{ab}\left(NQ_2^{ab} - \sum_{k'=0}^{K}\boldsymbol{x}_{2,a}^{(k')}\cdot\boldsymbol{x}_{2,b}^{(k')} - \boldsymbol{x}_{2,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{2,b}^{(\mathsf{neg.})}\right)},\\
\delta\left(NQ_r^{ab} - \sum_{k'=0}^{K}\boldsymbol{x}_{1,a}^{(k')}\cdot\boldsymbol{x}_{2,b}^{(k')} - \boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{2,b}^{(\mathsf{neg.})}\right) &= \int_{\mathbb{C}} d\hat{Q}_r^{ab} e^{\hat{Q}_r^{ab}\left(NQ_r^{ab} - \sum_{k'=0}^{K}\boldsymbol{x}_{1,a}^{(k')}\cdot\boldsymbol{x}_{2,b}^{(k')} - \boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{2,b}^{(\mathsf{neg.})}\right)},\\
\delta\big(Nm_{1,a}^{(k)} - \boldsymbol{x}_{1,a}^{(k)}\cdot\boldsymbol{x}_\star^{(k)}\big) &= \int_{\mathbb{C}} d\hat{m}_{1,a}^{(k)} e^{\hat{m}_{1,a}^{(k)}\left(Nm_{1,a}^{(k)} - \boldsymbol{x}_{1,a}^{(k)}\cdot\boldsymbol{x}_\star^{(k)}\right)},\\
\delta\big(Nm_{2,a}^{(k)} - \boldsymbol{x}_{2,a}^{(k)}\cdot\boldsymbol{x}_\star^{(k)}\big) &= \int_{\mathbb{C}} d\hat{m}_{2,a}^{(k)} e^{\hat{m}_{2,a}^{(k)}\left(Nm_{2,a}^{(k)} - \boldsymbol{x}_{2,a}^{(k)}\cdot\boldsymbol{x}_\star^{(k)}\right)},
\end{aligned}
\tag{31}
$$

up to a trivial multiplicative constant.

To obtain an expression that is analytically continuable to $n \to 0$, we introduce the replica symmetric ansatz (Charbonneau et al., 2023), which assumes that the integral over Ω is dominated by the contribution of the subspace where Ω satisfy the following constraints:

$$\begin{gathered} Q_1^{ab} = q_1 + (1-\delta_{ab})\frac{\chi_1}{\beta_1}, \qquad Q_2^{ab} = q_2 + (1-\delta_{ab})\frac{\chi_2}{\beta_2}, \qquad Q_r^{ab} = q_r + (1-\delta_{ab})\frac{\chi_r}{\beta_1}, \\ m_{1,a}^{(k)} = m_1^{(k)}, \qquad m_{2,a}^{(k)} = m_2^{(k)}, \end{gathered} \tag{32}$$

for $1 \le a, b \le n$ and $0 \le k \le K$. Although a general proof of this ansatz itself is still lacking, calculations based on this assumption are known to be asymptotically exact in the limit $N \to \infty$ for convex generalized linear models (Gerbelot et al., 2023) and Bayes-optimal estimation (Barbier et al., 2019). Consequently, the conjugate variables $\{\hat{Q}_1^{ab}, \hat{Q}_2^{ab}, \hat{Q}_r^{ab}, \hat{m}_{1,a}^{(k)}, \hat{m}_{2,a}^{(k)}\}$ are also assumed to be replica symmetric:

$$\begin{gathered} \hat{Q}_1^{ab} = \beta_1\hat{q}_1 - (1-\delta_{ab})\beta_1^2\hat{\chi}_1, \qquad \hat{Q}_2^{ab} = \beta_2\hat{q}_2 - (1-\delta_{ab})\beta_2^2\hat{\chi}_2, \qquad \hat{Q}_r^{ab} = -\beta_2\hat{q}_r + (1-\delta_{ab})\beta_1\beta_2\hat{\chi}_r, \\ \hat{m}_{1,a}^{(k)} = -\beta_1\hat{m}_1^{(k)}, \qquad \hat{m}_{2,a}^{(k)} = -\beta_2\hat{m}_2^{(k)}. \end{gathered} \tag{33}$$

Inserting equation 32 and equation 33 to equation 30, and rewriting $\mathbb{E}_{|\Omega}$ as $\mathbb{E}_{|\Theta}$ given this simplified profile of Ω offers

$$\begin{aligned} &\mathbb{E}_\star \int_{\mathbb{C}} d\Theta_1 d\Theta_2 \int \Bigg(\prod_{a=1}^n \prod_{k=0}^K d\boldsymbol{x}_{1,a}^{(k)} d\boldsymbol{x}_{2,a}^{(k)}\Bigg) \exp\Bigg[\sum_{k=1}^K M^{(k)} \log(\mathbb{E}_{|\Theta} L^{(k)}) + \sum_{k=1}^K M^{(k)} \log(\mathbb{E}_{|\Theta} \tilde{L}^{(k)})\Bigg] \\ &\times \exp nN\Bigg[\beta_1\Big(\frac{q_1\hat{q}_1 - \chi_1\hat{\chi}_1}{2} - \sum_{k=0}^K m_1^{(k)}\hat{m}_1^{(k)}\Big) + \beta_2\Big(\frac{q_2\hat{q}_2 - \chi_2\hat{\chi}_2}{2} - q_r\hat{q}_r - \chi_r\hat{\chi}_r - \sum_{k=0}^K m_2^{(k)}\hat{m}_2^{(k)}\Big)\Bigg] \\ &\times \exp\Bigg[-\beta_1\sum_{a=1}^n\sum_{k=0}^K\Big(\frac{\hat{q}_1}{2}\|\boldsymbol{x}_{1,a}^{(k)}\|_2^2 - \hat{m}_1\boldsymbol{x}_\star^{(k)}\cdot\boldsymbol{x}_{1,a}^{(k)} + \lambda_1\|\boldsymbol{x}_{1,a}^{(k)}\|_1\Big) - \beta_1\sum_{a=1}^n\Big(\frac{\hat{q}_1}{2}\|\boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\|_2^2 + \lambda_1\|\boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\|_1\Big) \\ &\qquad - \beta_2\sum_{a=1}^n\sum_{k=0}^K\Big(\frac{\hat{q}_2}{2}\|\boldsymbol{x}_{2,a}^{(k)}\|_2^2 - (\hat{m}_2\boldsymbol{x}_\star^{(k)} + \hat{q}_r\boldsymbol{x}_{1,a}^{(k)})\cdot\boldsymbol{x}_{2,a}^{(k)} + r(\boldsymbol{x}_{2,a}^{(k)}|\boldsymbol{x}_{1,a}^{(k)})\Big) \\ &\qquad - \beta_2\sum_{a=1}^n\Big(\frac{\hat{q}_2}{2}\|\boldsymbol{x}_{2,a}^{(\mathsf{neg.})}\|_2^2 - \hat{q}_r\boldsymbol{x}_{1,a}^{(\mathsf{neg.})}\cdot\boldsymbol{x}_{2,a}^{(\mathsf{neg.})} + r(\boldsymbol{x}_{2,a}^{(\mathsf{neg.})}|\boldsymbol{x}_{1,a}^{(\mathsf{neg.})})\Big) \\ &\qquad + \frac{\beta_1^2\hat{\chi}_1}{2}\Big\|\sum_{a=1}^n\boldsymbol{x}_{1,a}\Big\|_2^2 + \frac{\beta_2^2\hat{\chi}_2}{2}\Big\|\sum_{a=1}^n\boldsymbol{x}_{2,a}\Big\|_2^2 + \beta_1\beta_2\hat{\chi}_r\Big(\sum_{a=1}^n\boldsymbol{x}_{1,a}\Big)\cdot\Big(\sum_{a=1}^n\boldsymbol{x}_{2,a}\Big)\Bigg]. \end{aligned} \tag{34}$$

The last equation can be further simplified using the equality

$$\exp\Bigg[\frac{\beta_1^2\hat{\chi}_1}{2}\Big\|\sum_{a=1}^n\boldsymbol{x}_{1,a}\Big\|_2^2 + \frac{\beta_2^2\hat{\chi}_2}{2}\Big\|\sum_{a=1}^n\boldsymbol{x}_{2,a}\Big\|_2^2 + \beta_1\beta_2\hat{\chi}_r\Big(\sum_{a=1}^n\boldsymbol{x}_{1,a}\Big)\cdot\Big(\sum_{a=1}^n\boldsymbol{x}_{2,a}\Big)\Bigg] \tag{35}$$

$$= \mathbb{E}_{\boldsymbol{z}_1,\boldsymbol{z}_2}\exp\Bigg[\beta_1\sqrt{\hat{\chi}_1}\sum_{a=1}^n\boldsymbol{z}_1\cdot\boldsymbol{x}_{1,a} + \beta_2\sqrt{\hat{\chi}_2}\sum_{a=1}^n\boldsymbol{z}_2\cdot\boldsymbol{x}_{2,a}\Bigg] \tag{36}$$

where $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathbb{R}^N$ are random Gaussian vectors with elements i.i.d. according to $z_{1,i} \sim \mathsf{z}_1, z_{2,i} \sim \mathsf{z}_2$ for all $i = 1, \cdots, N$. From this decomposition, the integrals over $\boldsymbol{x}_{1,a}$ and $\boldsymbol{x}_{2,a}$ decouple over both vector coordinates $i = 1, \cdots, N$ and replica indices $a = 1, \cdots, n$:

$$\begin{gathered} \int d\Theta_1 d\Theta_2 \int e^{\sum_{k=1}^K M^{(k)}\log\mathbb{E}_{|\Theta}L^{(k)} + \sum_{k=1}^K M^{(k)}\log\mathbb{E}_{|\Theta}\tilde{L}^{(k)}} \prod_{k=0}^K \Big\{\mathbb{E}_2\Big[\int dx_1 dx_2 e^{-\beta_1\mathsf{E}_1^{(k)}(x_1) - \beta_2\mathsf{E}_2^{(k)}(x_2|x_1)}\Big]^n\Big\}^{N^{(k)}} \\ \times\Big\{\mathbb{E}_2\Big[\int dx_1 dx_2 e^{-\beta_1\mathsf{E}_1^{(\mathsf{neg.})}(x_1) - \beta_2\mathsf{E}_2^{(\mathsf{neg.})}(x_2|x_1)}\Big]^n\Big\}^{N-\sum_{k=0}^K N^{(k)}}. \end{gathered} \tag{37}$$

In the successive limit of $\beta_1 \to \infty$ and $\beta_2 \to \infty$, the integrals over x_1 and x_2 can be evaluated using Laplace's method, yielding the asymptotic form:

$$\begin{aligned}\int d\Theta_1 d\Theta_2 \prod_{k=1}^{K} e^{M^{(k)} \log \mathbb{E}_{|\Theta} L^{(k)} + M^{(k)} \log \mathbb{E}_{|\Theta} \tilde{L}^{(k)}} \prod_{k=0}^{K} \left\{ \mathbb{E}_2 \left[e^{-\beta_1 \mathsf{E}_1^{(k)}(\mathsf{x}_1^{(k)}) - \beta_2 \mathsf{E}_2^{(k)}(\mathsf{x}_2^{(k)} | \mathsf{x}_1^{(k)})} \right]^n \right\}^{N^{(k)}} \\ \times \left\{ \mathbb{E}_2 \left[e^{-\beta_1 \mathsf{E}_1^{(\mathsf{neg.})}(\mathsf{x}_1^{(\mathsf{neg.})}) - \beta_2 \mathsf{E}_2^{(\mathsf{neg.})}(\mathsf{x}_2^{(\mathsf{neg.})} | \mathsf{x}_1^{(\mathsf{neg.})})} \right]^n \right\}^{N - \sum_{k=0}^{K} N^{(k)}}. \end{aligned} \tag{38}$$

where the definitions of $\mathbb{E}_2$ and $\{\mathsf{x}_1^{(k)}, \mathsf{x}_2^{(k)}\}_{k=0}^{K}, \mathsf{x}_1^{(\mathsf{neg.})}, \mathsf{x}_2^{(\mathsf{neg.})}$ follow from Definitions 1 and 2.

Let us proceed with the calculation of $\mathbb{E}_{|\Theta} L^{(k)}$. Define the centered Gaussian variables $\{H^{(k)}, h_1^{(k)}, h_2^{(k)}\}$ and $\{u_{1,a}, u_{2,a}\}_{a=1}^{n}$ as

$$\begin{pmatrix} H^{(k)} \\ h_1^{(k)} \\ h_2^{(k)} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}_3, \begin{pmatrix} \rho^{(k)} & m_1^{(k)} & m_2^{(k)} \\ m_1^{(k)} & q_1 & q_r \\ m_2^{(k)} & q_r & q_2 \end{pmatrix} \right), \qquad \begin{pmatrix} u_{1,a} \\ u_{2,a} \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}_2, \begin{pmatrix} \chi_1/\beta_1 & \chi_r/\beta_1 \\ \chi_r/\beta_1 & \chi_2/\beta_2 \end{pmatrix} \right), \tag{39}$$

where $\{u_{1,a}, u_{2,a}\}$ are independent for all $a = 1, \cdots, n$. Then, we can see that the random variables admit the decomposition $h_{1,a}^{(k)} = h_1^{(k)} + u_{1,a}, h_{2,a} = h_2^{(k)} + u_{2,a}$, offering the expression

$$\mathbb{E}_{|\Theta} L^{(k)} = \mathbb{E}_{h_1^{(k)}, h_2^{(k)}, H^{(k)}} \prod_{a=1}^{n} \mathbb{E}_{u_{1,a}, u_{2,a}} \exp\left[-\frac{\beta_1}{2}(H^{(k)} - h_1^{(k)} - u_{1,a})^2 - \frac{\beta_2 \delta_{1k}}{2} \left(H^{(k)} - \kappa(h_1^{(k)} + u_{1,a}) - h_2^{(k)} - u_{2,a} \right)^2 \right]. \tag{40}$$

Taking into account that $\beta_1 \gg \beta_2$, the Gaussian measure over $(u_{1,a}, u_{2,a})$ can be written as

$$C_{\beta_1, \beta_2} \exp\left[-\frac{\beta_1}{2\chi_1}(1 + c_{\beta_1}) u_{1,a}^2 - \frac{\beta_2}{2\chi_2} \left(u_{2,a} - \frac{\chi_r}{\chi_1} u_{1,a} \right)^2 \right] du_{1,a} du_{2,a}, \tag{41}$$

where C_{β_1, β_2} is a number subexponential in β_1 and β_2, and c_{β_1} is a real number which converges to zero as $\beta_1 \to \infty$. Using this expression to equation 40 yields

$$\begin{aligned} \mathbb{E}_{|\Theta} L^{(k)} = & \mathbb{E}_{h_1^{(k)}, h_2^{(k)}, H^{(k)}} \left\{ \int du_1 du_2 \exp\left[-\frac{\beta_1}{2\chi_1}(1 + c_{\beta_1}) u_1^2 - \frac{\beta_1}{2}(H^{(k)} - h_1^{(k)} - u_1)^2 \right.\right. \\ & \left.\left. - \frac{\beta_2}{2\chi_2} u_2^2 - \frac{\beta_2 \delta_{1k}}{2} \left(H^{(k)} - \kappa(h_1^{(k)} + u_1) - h_2^{(k)} - u_2 - \frac{\chi_r}{\chi_1} u_1 \right)^2 + \log C_{\beta_1, \beta_2} \right] \right\}^n. \end{aligned} \tag{42}$$

The same procedure can be applied to $\tilde{L}^{(k)}$, this time accounting for $t_1, t_2 \ll \beta_2$, offering

$$\mathbb{E}_{|\Theta} \tilde{L}^{(k)} = \mathbb{E}_{\tilde{h}_1^{(k)}, \tilde{h}_2^{(k)}, \tilde{H}^{(k)}} \exp n \left[-t_1 \left(\tilde{H}^{(k)} - \tilde{h}_1^{(k)} \right)^2 - t_2 \delta_{1k} \left(\tilde{H}^{(k)} - \kappa \tilde{h}_1^{(k)} - \tilde{h}_2^{(k)} \right)^2 + \log C_{\beta_1, \beta_2} \right]. \tag{43}$$

Note that the expressions for $\tilde{L}^{(k)}$ and $L^{(k)}$ are simple Gaussian integrals that can be computed explicitly. Now that all the terms in equation 30 have been expressed in analytical form with respect to n, the limit $n \to 0$ can be taken formally. Evaluating equation 38, equation 42 and equation 43 up to first order of n, and neglecting subleading terms with respect to β_1 and β_2, we finally obtain an expression of the form

$$\mathbb{E}_{\mathcal{D}, \tilde{\mathcal{D}}} \left[\Phi^n(\mathcal{D}, \tilde{\mathcal{D}}) \right] = \int d\Theta_1 d\Theta_2 \exp N \left[n \mathcal{G}(\Theta_1, \Theta_2) + O(n^2) \right], \tag{44}$$

where

$$\mathcal{G}(\Theta_1, \Theta_2) = \beta_1 G_1(\Theta_1) + \beta_2 G_2(\Theta_2 | \Theta_1) + t_1 \epsilon_1 + t_2 \epsilon_2, \tag{45}$$

$$
\begin{aligned}
G_1(\Theta_1) = {} & \frac{q_1\hat{q}_1 - \chi_1\hat{\chi}_1}{2} - \sum_{k=0}^{K} m_1^{(k)}\hat{m}_1^{(k)} - \sum_{k=0}^{K} \pi^{(k)}\mathbb{E}_1\Big[\mathsf{E}_1^{(k)}\big(\mathsf{x}_1^{(k)}\big)\Big] - \pi^{(\mathsf{neg.})}\mathbb{E}_1\Big[\mathsf{E}_1^{(\mathsf{neg.})}\big(\mathsf{x}_1^{(\mathsf{neg.})}\big)\Big] \\
& - \sum_{k=1}^{K} \frac{\alpha^{(k)}}{2}\frac{q_1 - 2(m_1^{(0)} + m_1^{(k)}) + \rho^{(k)}}{1+\chi_1}, \qquad (46) \\
G_2(\Theta_2|\Theta_1) = {} & \frac{q_2\hat{q}_2 - \chi_2\hat{\chi}_2}{2} - q_r\hat{q}_r - \chi_r\hat{\chi}_r - \sum_{k=0}^{K} m_2^{(k)}\hat{m}_2^{(k)} \\
& - \sum_{k=0}^{K} \pi^{(k)}\mathbb{E}_2\Big[\mathsf{E}_2^{(k)}\big(\mathsf{x}_2^{(k)}|\mathsf{x}_1^{(k)}\big)\Big] - \pi^{(\mathsf{neg.})}\mathbb{E}_2\Big[\mathsf{E}_2^{(\mathsf{neg.})}\big(\mathsf{x}_2^{(\mathsf{neg.})}|\mathsf{x}_1^{(\mathsf{neg.})}\big)\Big] \\
& - \frac{\alpha^{(1)}}{2}\frac{q_2 + A^2 q_1 + 2Aq_r + B^2\rho^{(1)} - 2B(m_2^{(0)} + m_2^{(1)}) - 2AB(m_1^{(0)} + m_1^{(1)})}{1+\chi_2}, \qquad (47)
\end{aligned}
$$

and

$$
\begin{aligned}
\epsilon_1 &= \sum_{k=1}^{K} \pi^{(k)}\Big[q_1 - 2(m_1^{(0)} + m_1^{(k)}) + \rho^{(k)}\Big], \qquad (48) \\
\epsilon_2 &= \pi^{(1)}\Big[q_2 + \kappa^2 q_1 + 2\kappa q_r + \rho^{(1)} - 2(m_2^{(0)} + m_2^{(1)}) - 2\kappa(m_1^{(0)} + m_1^{(1)})\Big]. \qquad (49)
\end{aligned}
$$

For large N and finite n, the integral over Θ_1 and Θ_2 can be evaluated using the saddle point method, where the integral is dominated by the stationary point of the exponent. This finally yields

$$
\lim_{n\to 0}\lim_{N\to\infty} \frac{1}{N}\frac{\partial}{\partial n}\log \mathbb{E}_{\mathcal{D},\tilde{\mathcal{D}}}\Big[\Phi^n(\mathcal{D},\tilde{\mathcal{D}})\Big] = \underset{\Theta_1,\Theta_2}{\mathrm{Extr}}\, \mathcal{G}(\Theta_1,\Theta_2), \qquad (50)
$$

where Extr denotes the extremum operation of the function. The stationary conditions for Θ_1 and Θ_2, in the successive limit of $\beta_1 \to \infty$ and $\beta_2 \to \infty$ are given by the equations of state in Definitions 1 and 2. Note that the term $t_1\epsilon_1 + t_2\epsilon_2$ does not contribute to the stationary condition in the limit $t_1, t_2 \to 0$, and thus can be neglected until one takes the derivative with respect to t_1 and t_2, as in equation 11 and equation 12.

B Additional numerical experiments

Here, we provide more numerical experiments highlighting the difference between the Generalized Trans-Lasso, Pretraining Lasso and Trans-Lasso. In figures 5 and 6, we show the ratios of $\min\{\epsilon_{\Delta\lambda=0}, \epsilon_{\kappa=0}\}, \epsilon_{\mathrm{Pretrain}}$ and $\epsilon_{\mathrm{Trans}}$ against ϵ_{LO} for the cases $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.15, 0.04, 0.04)$ and $(0.05, 0.14, 0.14)$, respectively. The former case represents the problem setting where the underlying common feature vector among the datasets is large, while the latter represents the problem setting where the underlying common feature vector is small. In both cases, the minimum of the $\kappa = 0$ or $\Delta\lambda = 0$ strategy can obtain generalization errors close to the one from the LO strategy for both low or moderate noise levels. This is not a property exhibited in the Trans-Lasso or Pretraining Lasso, where both algorithms have relatively low generalization performance when $\sigma = 0.01$ and $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.15, 0.04, 0.04)$. Note that when $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.05, 0.14, 0.14)$, all three algorithms yield comparable generalization performance.

To directly compare the $\kappa = 0$ or $\Delta\lambda = 0$ strategy with the Pretraining Lasso and Trans-Lasso, in figures 7, 8 and 9, we also plot the same ratios with shared color scale bars. As evident from the plot, the $\kappa = 0$ or $\Delta\lambda = 0$ strategy exhibits a clear improvement in generalization performance over Trans-Lasso. While the difference between Pretraining Lasso and $\kappa = 0$ or $\Delta\lambda = 0$ strategy is comparable for $\sigma = 0.1$, we can see a consistent advantage for lower noise levels ($\sigma = 0.01$).

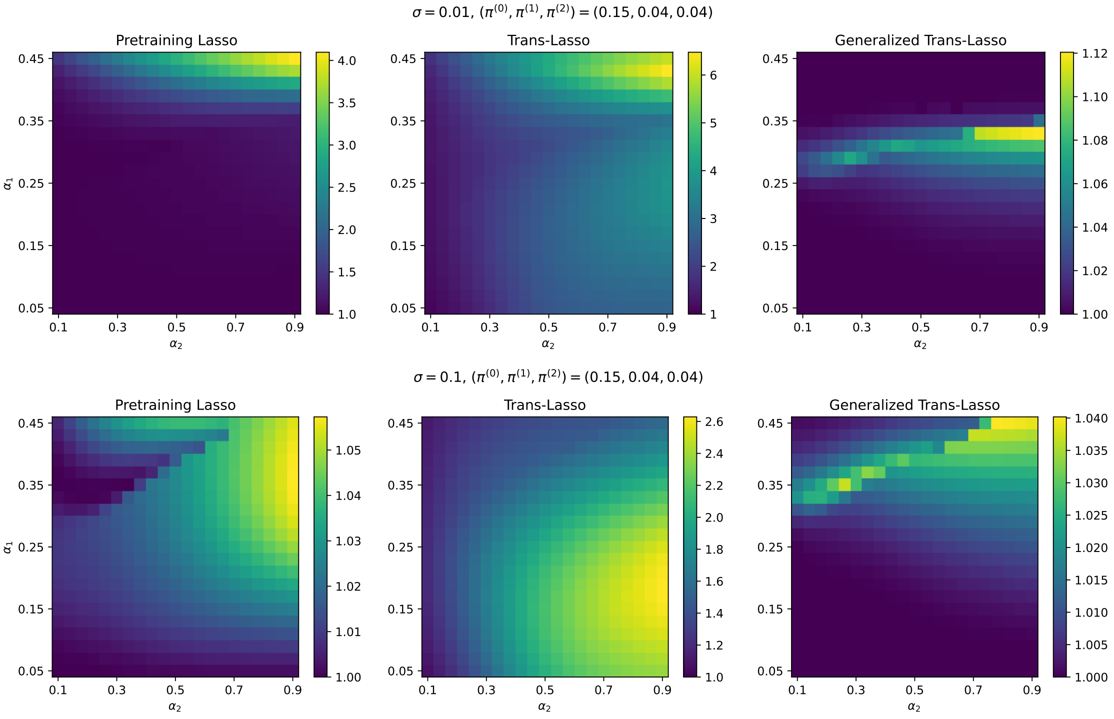

Figure 5: Ratios $\min\{\epsilon_{\Delta\lambda=0}, \epsilon_{\kappa=0}\}/\epsilon_{\text{LO}}, \epsilon_{\text{Pretrain}}/\epsilon_{\text{LO}}$ and $\epsilon_{\text{Trans}}/\epsilon_{\text{LO}}$ under setting $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.15, 0.04, 0.04)$ and noise level $\sigma = 0.01$ (top figure) and 0.1 (bottom figure) for various values of $(\alpha^{(1)}, \alpha^{(2)})$.

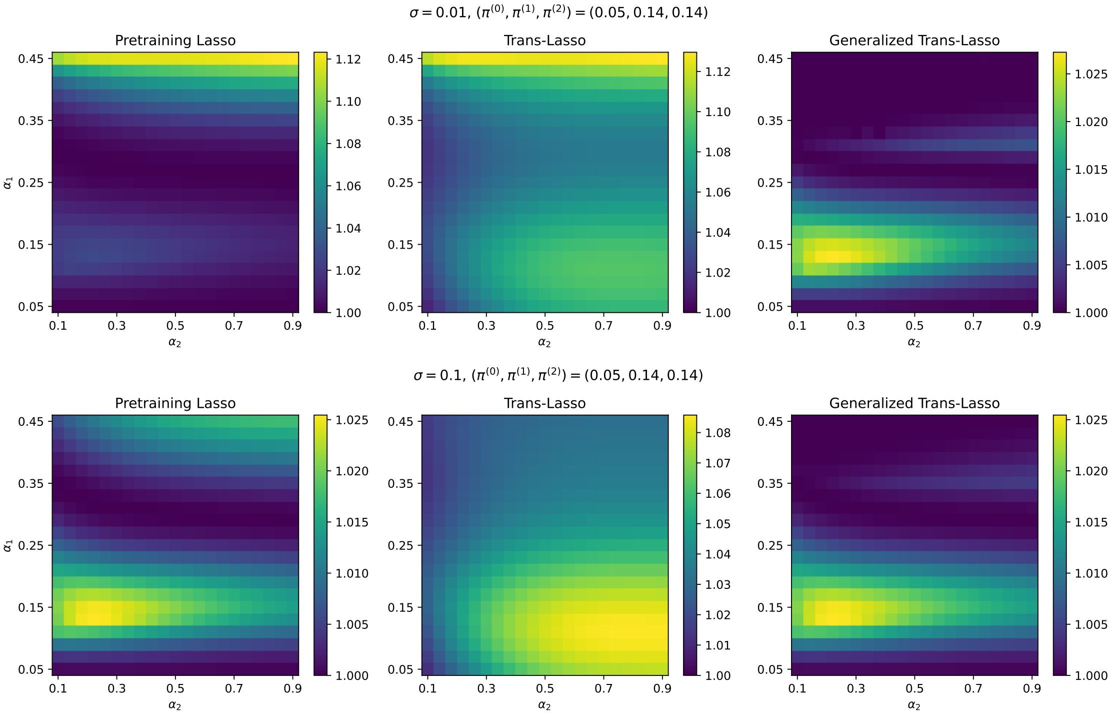

Figure 6: Ratios $\min\{\epsilon_{\Delta\lambda=0}, \epsilon_{\kappa=0}\}/\epsilon_{\text{LO}}, \epsilon_{\text{Pretrain}}/\epsilon_{\text{LO}}$ and $\epsilon_{\text{Trans}}/\epsilon_{\text{LO}}$ under setting $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.05, 0.14, 0.14)$ and noise level $\sigma = 0.01$ (top figure) and 0.1 (bottom figure) for various values of $(\alpha^{(1)}, \alpha^{(2)})$.

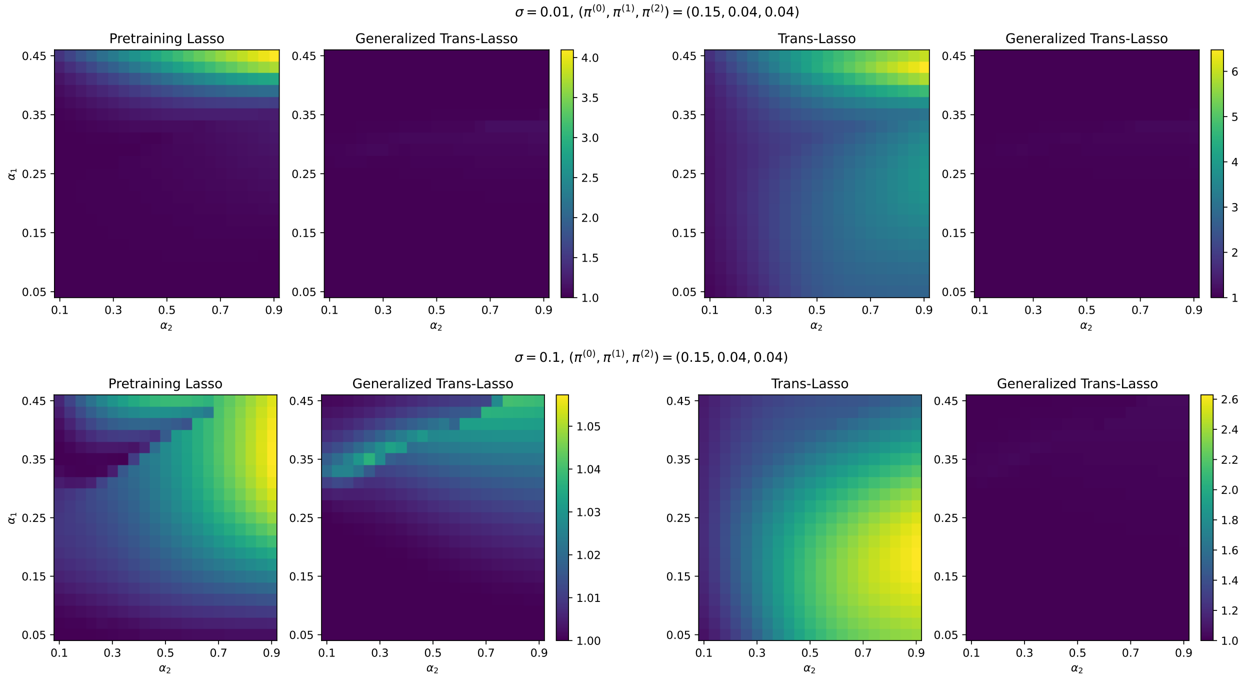

Figure 7: Comparisons between ratios $\min\{\epsilon_{\Delta\lambda=0}, \epsilon_{\kappa=0}\}/\epsilon_{\mathrm{LO}}$, $\epsilon_{\mathrm{Pretrain}}/\epsilon_{\mathrm{LO}}$ and $\epsilon_{\mathrm{Trans}}/\epsilon_{\mathrm{LO}}$ under setting $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.15, 0.04, 0.04)$.

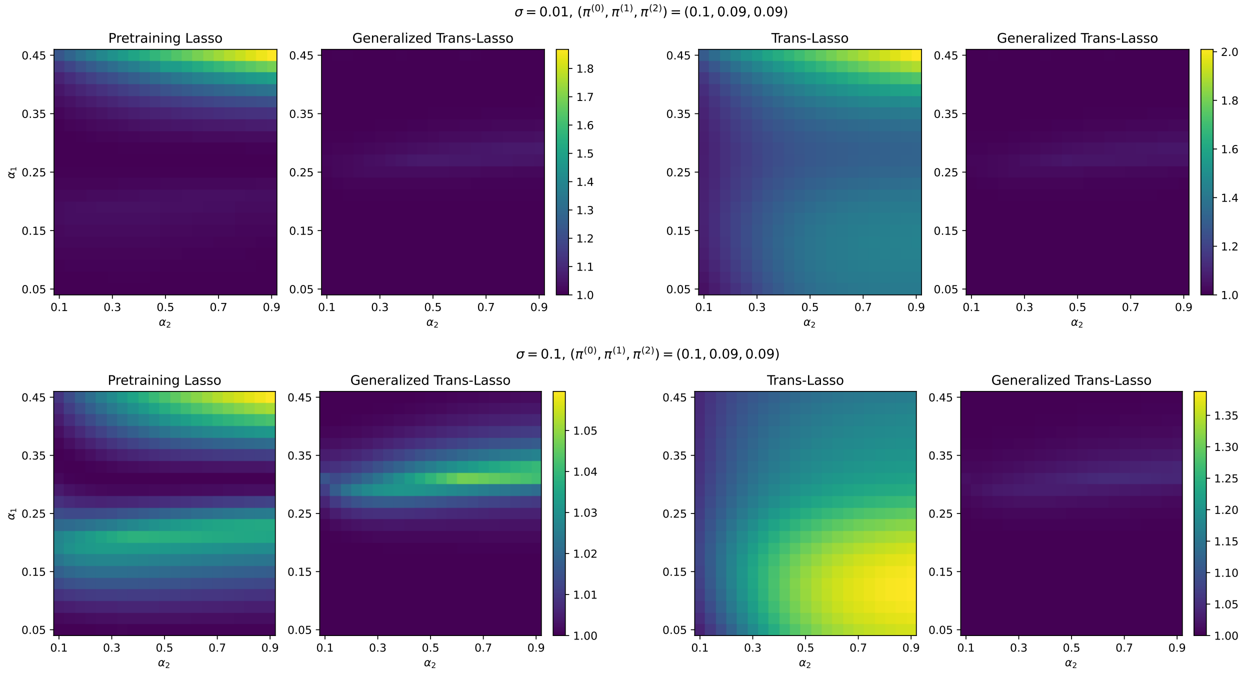

Figure 8: Comparisons between ratios $\min\{\epsilon_{\Delta\lambda=0}, \epsilon_{\kappa=0}\}/\epsilon_{\mathrm{LO}}$, $\epsilon_{\mathrm{Pretrain}}/\epsilon_{\mathrm{LO}}$ and $\epsilon_{\mathrm{Trans}}/\epsilon_{\mathrm{LO}}$ under setting $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.10, 0.09, 0.09)$.

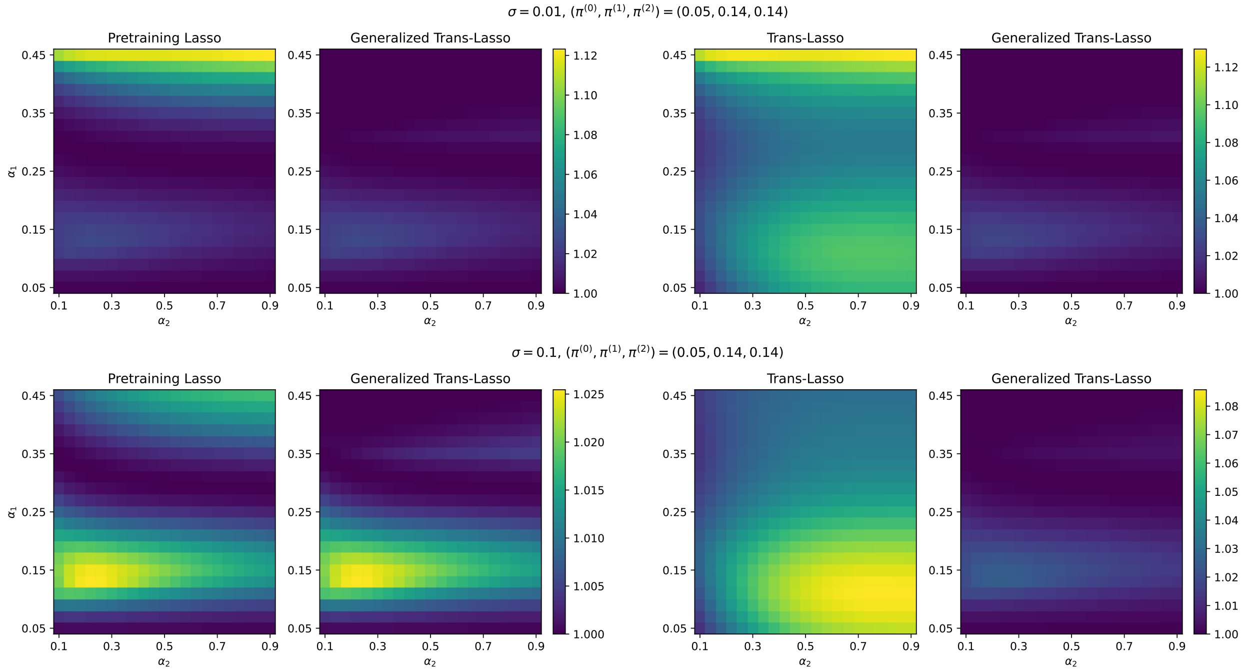

Figure 9: Comparisons between ratios $\min\{\epsilon_{\Delta\lambda=0}, \epsilon_{\kappa=0}\}/\epsilon_{\text{LO}}$, $\epsilon_{\text{Pretrain}}/\epsilon_{\text{LO}}$ and $\epsilon_{\text{Trans}}/\epsilon_{\text{LO}}$ under setting $(\pi^{(0)}, \pi^{(1)}, \pi^{(2)}) = (0.05, 0.14, 0.14)$.