Optimization Methods for Parameter Identifications in Settings with Only Partial Knowledge

Aleksei Sholokhov

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

Reading Committee
Aleksandr Aravkin (co-chair),
J. Nathan Kutz (co-chair),
James Burke,
Jonathan Wakefield (GSR)

Program Authorized to Offer Degree:

Applied Mathematics

University of Washington

University of Washington

Abstract

Optimization Methods for Parameter Identifications
in Settings with Only Partial Knowledge

Aleksei Sholokhov

Chairs of the Supervisory Committee:
Aleksandr Aravkin and J. Nathan Kutz
Department of Applied Mathematics

This work summarizes two projects focused on incorporating prior knowledge into machine learning models. In the first project we developed universal feature selection methods for linear mixed-effect models. Namely, we extended Sparse Relaxed Regularized Regression (SR3) to Linear Mixed-Effect (LME) likelihood and showed how one can minimize such likelihood with a proximal gradient descent. We also develop theoretical underpinnings of the proposed extension, including consistency results, variational properties, implementability of optimization methods, and convergence results. In particular we provide convergence analyses for a basic implementation of SR3 for LME (called MSR3) and an accelerated hybrid algorithm (called MSR3-fast). Numerical results show the utility and speed of these algorithms on realistic simulated datasets. Finally, we provide an open-source implementation of both algorithms in an open source python package `pysr3`. This package offers complete compatibility with `scikit-learn`, so all `pysr3` models can be used in pipeline with classic modelling blocks such as data pre-processors, randomized grid search, cross-validation, and quality metrics.

The second line of work develops a framework for training Reduced-Order Models (ROMs) with Physics-Informed Neural Ordinary Differential Equations (PINODE). Our innovation builds on ideas from classical collocation methods of numerical analysis and illustrates how one can use collocation points to transfer knowledge from a known equation to a model that approximates solutions of that equation. We show that the addition of our physics-informed loss allows for exceptional data supply strategies that improves the performance of ROMs in data-scarce settings, where training high-quality data-driven models is impossible. The resulting ROMs are able to extrapolate forward in time considerably more effectively, perform better for unseen initial conditions, and exhibit less sensitivity to noise. Finally, we show how such ROMs can be used as strong regularizers in single-pixel imaging (SPI) allowing to reduce data-intake by an order of magnitude relatively to current state-of-the-art algorithms.

Contents

Todo list

Chapter 1

Introduction

Edit: Add citations all over the intro

Machine learning has emerged as an important tool in applied mathematics problems as data-driven approaches have begun to outperform first-principle modeling in the tasks of classification, recognition, autonomous control, protein folding, and even video games. However, due to the absence of prior knowledge, purely data-driven approaches typically require a large amount of data to approximate the behavior of first-principle models even in the simplest scenarios. Thus, the most promising modeling approaches of today are those that integrate both data-driven and first-principle components. These latter components frequently stem from domain expertise and pre-existing beliefs. Incorporating prior knowledge into a data-driven method is a formidable task that demands a considerable amount of ingenuity. This thesis serves to illustrate two distinct examples of such endeavors. Together, these examples represent a diverse range of challenges in incorporating prior knowledge into a model and describe the strategies employed to surmount them.

An instance of such a problem is feature selection for mixed-effect models. Feature selection problems typically arise when one believes that a small subset of features within a dataset captures the majority of the variance of the quantity of interest; this subset of features is commonly known as sparse support. The canonical feature selection methods typically provide control over the sparsity of the solution and output a candidate support for it. In the absence of additional information about the support apart from its probable sparsity, these methods furnish a satisfactory toolkit for practitioners.

Real-world practitioners, however, pose expert knowledge about the covariates in their data and wish to incorporate that knowledge as a set of priors and constraints. For instance, a practitioner may desire a model that employs at most 6 out of the 40 available features, with features A and B being mandatory components. Furthermore, if feature C is integrated, it must have a negative coefficient, and feature D must be independent of any grouping parameter. Attempts to directly address such constraints often result in the formulation of mathematically intractable problems. Additionally, practitioners require the ability to swap these priors as new hypotheses and knowledge come to light. This necessitates the implementation of an optimization routine that does not rely upon these priors; otherwise it would force the practitioner to re-implement the algorithm after every such swap, rendering the method impractical. Chapter 2 elucidates my efforts in developing a universal feature selection approach for linear mixed-effects models that simultaneously offers flexibility for incorporating a variety of priors and confronts the numerical challenges that accompany such flexibility.

The second example serves to demonstrate how one can incorporate a degree of partial knowledge of physics into a deep neural network. Methods for data-driven modeling of physical phenomena have been increasingly popular in large part due to the advancements in GPU computations and automatic differentiation tools. These methods have proven particularly useful in reduced-order modeling tasks. Using data, a neural network can find a representation of a system on a reduced-order manifold via a non-linear transformation from an observable space to that

manifold. Classic techniques, such as Galerkin Proper Orthogonal Decomposition (POD) or Dynamic Mode Decomposition (DMD), are fundamentally incapable of finding such spaces because these methods are confined to linear, albeit optimal, transformations. However, the exceptional approximation capabilities of deep networks come at the cost of increased instability. In particular, the networks tend to identify manifolds that overfit the data and, consequently, fail at extrapolation tasks. Such deficiencies compromise the efficacy of the entire system in which the network is utilized, including applications in control, model identification, and compressive sensing.

Recent works have shown that one can improve a neural-network based model by incorporating partial knowledge of physics. For example, a model can find a better reduced-order manifold if it knows that it predicts the behavior of a shock wave that obeys Burger's equation. Chapter 3 is devoted to the development of a framework for training Reduced-Order Models (ROMs) with Physics-Informed Neural Ordinary Differential Equations (PINODE). It illustrates how one can transfer knowledge from a known equation to a model that predicts solutions of said equation using collocation points. The resulting ROMs are able to extrapolate forward in time considerably more effectively, perform better for unseen initial conditions, and exhibit less sensitivity to noise. Finally, in Chapter 3.5 I demonstrate the utility of PINODE models in compressive sensing applications, where they effectively "stitch" together separate timeframes and go below the Nyquist sampling limit.

Attribution Throughout my PhD I was fortunate to work with incredibly talented advisors and collaborators who significantly contributed to the content of this work. Chapter 2 captures a sequence of joint works that were completed between 2019 and 2023 at the University of Washington's Institute for Health Metrics and Evaluations (IHME) together with Dr. Aleksandr Aravkin, Dr. James Burke, and Dr. Peng Zheng. The MSR3 framework algorithm was introduced in Sholokhov et al. (2022b), with the theoretical foundations developed in Aravkin et al. (2022a), and the software implementation published in Sholokhov et al. (2023). Chapter 3 represents my work at Mitsubishi Electric Research Labs (MERL) under the guidance of Dr. Hassan Mansour and Dr. Saleh Nabi. The PINODE framework was developed in **?** and is based on our earlier joint work with Dr. Yuying Liu on Physics-Informed Koopman Networks (Liu et al. (2022)). In particular, Figure 3.3.4 and the related example come from that work. Chapter 3.5 was developed in **?** together with Dr. Hassan Mansour, Dr. Saleh Nabi, Dr. J. Nathan Kutz, and Dr. Steven Brunton at the University of Washington. Many parts of the works listed above were taken verbatim to this document per the explicit permission of my co-authors.

Chapter 2

MSR3: Sparse Relaxed Regularized Regression for Mixed-Effect Models

Abstract for Chapter 2 Linear Mixed-Effects (LME) models are a fundamental tool for modeling correlated data, including cohort studies, longitudinal data analysis, and meta-analysis. Design and analysis of variable selection methods for LMEs is more difficult than for linear regression because LME models are nonlinear. In this work we propose a relaxation strategy and optimization methods that enable a wide range of variable selection methods for LMEs using both convex and nonconvex regularizers, including ℓ_1, Adaptive-ℓ_1, SCAD, and ℓ_0. The computational framework only requires the proximal operator for each regularizer to be available, and the implementation is available in an open source `python` package `pysr3`, consistent with the `sklearn` standard. The numerical results on simulated data sets indicate that the proposed strategy improves on the state of the art for both accuracy and compute time. The variable selection techniques are also validated on a real example using a data set on bullying victimization.

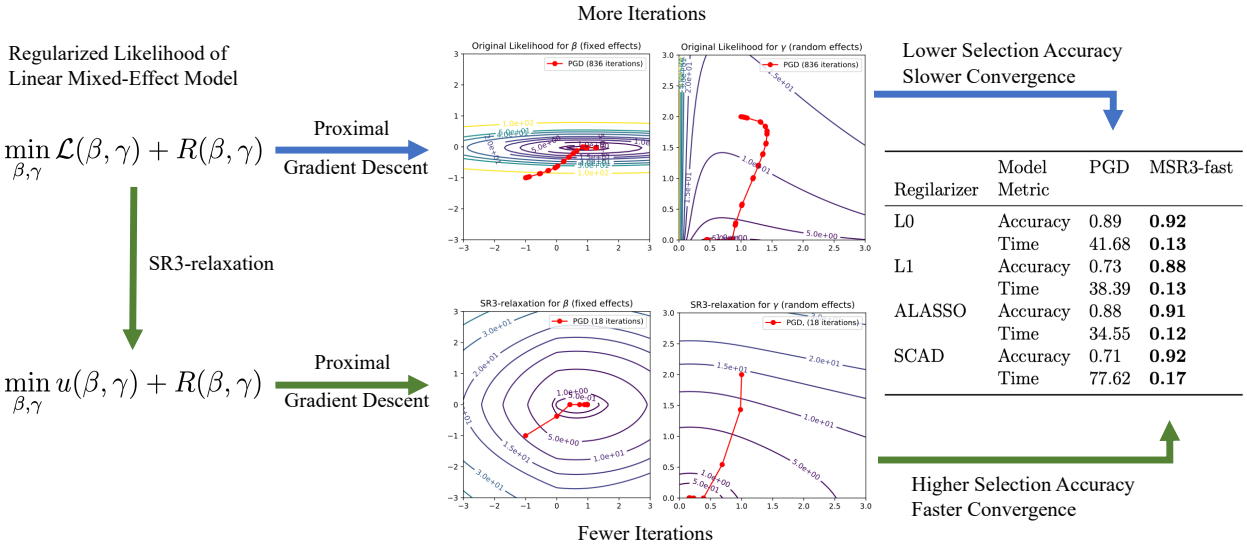
Keywords: Mixed effects models, feature selection, nonconvex optimization

2.1 Introduction

Linear mixed-effects (LME) models use covariates to explain the variability of target variables in a grouped data setting. For each group, the relationship between covariates and observations is modeled using group-specific coefficients that are linked by a common prior distribution across all groups, allowing LMEs to borrow strength across groups in order to estimate statistics for the common prior. LMEs are used in settings with insufficient data to resolve each group independently, making them fundamental tools for regression analysis in population health sciences (Reiner et al. (2020); Murray et al. (2020)), meta-analysis (DerSimonian and Laird (1986); Zheng et al. (2021)), life sciences, and as well as in many others domains (Zuur et al. (2009)).

Variable selection is a fundamental problem in all regression settings. In linear regression, the LASSO method (Tibshirani, 1996a) and related extensions have been widely used. However, variable selection for LMEs is complicated by the nonlinear structure and relative sparsity of the within-group data. While standard methods and software are available for linear regression (see e.g. `glmnet` Friedman et al. (2010)), there are few open source libraries for variable selection for LMEs. Many covariates selection algorithms for LMEs have been proposed over the last 20 years (see the survey Buscemi and Plaia (2019)), but comparison of these strategies and practical application remains difficult. Approaches vary by choice of likelihood (e.g. marginal, restricted, or h- likelihood), regularizer (e.g. ℓ_1 (Bondell et al., 2010) or SCAD Ibrahim et al. (2011a)), and information criteria (Vaida and Blanchard, 2005; Ibrahim et al., 2011b). Implementations vary as well, typically using regularizer-specific local quadratic approximations to apply solution methods for smooth problems (Newton-Raphson, EM, sequential least squares) to fit the original nonsmooth model. All of these decisions make it difficult to compare and evaluate performance of available variable selection strategies and to determine which method is best suited for a

given task. This challenge is exacerbated by the absence of standardized datasets and open source libraries for each method. Our main practical goal to fill this gap by developing a unified methodological framework that accommodates a wide variety of variable selection strategies based on a set of easily implementable regularizers, and made available in an open source library, `pysr3`[1] that is easy to use and to compare different methods. All experiments in the paper can be reproduced using `pysr3` and code in the reproducibility guide[2].



Regilarizer	Model Metric	PGD	MSR3-fast
L0	Accuracy	0.89	**0.92**
	Time	41.68	**0.13**
L1	Accuracy	0.73	**0.88**
	Time	38.39	**0.13**
ALASSO	Accuracy	0.88	**0.91**
	Time	34.55	**0.12**
SCAD	Accuracy	0.71	**0.92**
	Time	77.62	**0.17**

Figure 2.1.1: Selection of fixed and random effects for LME likelihoods $\mathcal{L}$ using 'regularization-agnostic' framework and its SR3 extension using four regularizers. SR3 relaxation accelerates algorithmic converge (middle panel), and gives better robustness and improved performance on synthetic problems across regularizers (right panel)

.

In this work we introduce a regularization-agnostic covariate selection strategy that (1) is fast and simple to implement, (2) provides robust models, and (3) is flexible enough to support most regularizers currently used in variable selection across different domains. The baseline approach uses the proximal gradient descent (PGD) method, which has been studied by the optimization community for over 40 years, but has not been widely used in LME covariate selection. In our initial numerical experiments, using a naive PGD approach indicated that, at best, the method yields only a marginal improvement over the equally unsatisfactory alternative methods in accurately determining the correct variables in our variable selection test problems. We conjecture that the weakness of the naive PGD is a result of the first-order likelihood approximation used in PGD. To overcome this problem, we propose an alternative likelihood approximation with the goal of incorporating global variational properties of the likelihood. For this purpose, we extend the sparse relaxed regularized regression (SR3) framework (Zheng et al. (2019)) to the LME setting. This idea is supported by the success of SR3 in the context of linear regression where it accelerates and improves the performance of regularization strategies.

[1] https://github.com/aksholokhov/pysr3
[2] https://github.com/aksholokhov/msr3-paper

This extension and its mathematical foundations in Aravkin et al. (2022b) constitute the major innovations of this work. Here we introduce the modeling framework and its relaxation, discuss the resulting algorithms and their implementation details, and validate the method on both simulated and real data sets, while the mathematical foundations are presented in Aravkin et al. (2022b). The SR3 framework introduces auxiliary variables x and w to decouple the likelihood $\mathcal{L}$ and sparsity regularizer R which are tied together by adding a multiple of the norm squared difference $\frac{\eta}{2}\|x-w\|^2$ to the objective. Then, fixing the variables w dedicated to the nonsmooth regularizer R, the smooth function $\mathcal{L}(x)+\frac{\eta}{2}\|x-w\|^2$ is globally optimized over the variables x to obtain an optimal value function $u_\eta(w)$. We then show that $u_\eta(w)$ is smooth. This opens the door to the application of the PGD algorithm to minimizing $u_\eta + R$ where now u_η contains global variational information on the likelihood function $\mathcal{L}$. The main obstacle in the application of this approach is the evaluation of u_η and its gradient. In Section 2.3.2 we present a method for overcoming this difficulty using variable metric techniques and interior point technology.

All new methods are implemented in an open-source library called `pysr3`, which fills a gap for python mixed-models selection tools in `Python` (Buscemi and Plaia (2019), Table 3). Our algorithms are 1-2 orders of magnitude faster than available LASSO-based libraries for mixed effects selection in `R`, see Table 2.4.2. `pysr3` enables a standardized comparison of different methods in the LME setting, and makes both the PGD framework and its SR3 extension available to practitioners working with LME models.

We begin in Section 2.2 by giving a precise description of the LME model and set the notation for the remainder of the paper. This is followed by a brief discussion of prior work on LMEs. In Section 2.3 we present our algorithms for the LME model starting with the naive PGD algorithm. This is followed by a description of the variable splitting technique used in Zheng et al. (2019) to incorporate global variation information on the likelihood function into the direction finding subproblem for the PGD algorithm. Next we tackle the problem of how to approximate the resulting optimal value function u_η and its gradient where $\eta > 0$ is the decoupling parameter. As noted, this is done using variable metric techniques and and interior point technology. We conclude Section 2.3 with a discussion of the MSR3 and MSR3-fast algorithms. In Section 2.4 we discuss how the underlying algorithmic parameters are set and test the algorithm on both simulated problems and a problem with real data. In Section 2.5 we develop theoretical underpinnings of the proposed extension, including consistency results, variational properties, implementability of optimization methods, and convergence results for both MSR3 and MSR3-fast. Section 2.6 outlines the software implementation of MSR3 and MSR3-fast in `pysr3` package. The chapter is concluded in Section 2.7 with a brief discussion of the contributions.

2.2 Linear Mixed-Effects Models: Notation and Fundamentals

Mixed-effect models describe the relationship between an outcome variable and its predictors when the observations are grouped, for example in studies or clusters. To set the notation, consider m groups of observations indexed by i, with sizes n_i, and the total number of observations equal to $n = n_1 + n_2 + \cdots + n_m$. For each group, we have design matrices for fixed features $X_i \in \mathbb{R}^{n_i \times p}$, and matrices of random features $Z_i \in \mathbb{R}^{n_i \times q}$, along with vectors of outcomes $Y_i \in \mathbb{R}^{n_i}$. Let $X = [X_1^T, X_2^T, \dots, X_m^T]^T$ and $Z = [Z_1^T, Z_2^T, \dots, Z_m^T]^T$. Following Patterson and Thompson (1971); Pinheiro and Bates (2000), we define a Linear Mixed-Effects (LME) model as

$$\begin{aligned} Y_i &= X_i\beta + Z_i u_i + \varepsilon_i, \quad i = 1 \dots m \\ u_i &\sim \mathcal{N}(0, \Gamma), \quad \Gamma \in \mathbb{S}_+^q \\ \varepsilon_i &\sim \mathcal{N}(0, \Lambda_i), \quad \Lambda_i \in \mathbb{S}_{++}^{n_i} \end{aligned} \tag{2.2.1}$$

where $\beta \in \mathbb{R}^p$ is a vector of fixed (mean) covariates, $u_i \in \mathbb{R}^q$ are unobservable random effects assumed to be distributed normally with zero mean and the unknown covariance matrix Γ, and $\mathbb{S}_+^\nu$ and $\mathbb{S}_{++}^\nu$ are the sets of real symmetric $\nu \times \nu$ positive semi-definite and positive definite matrices, respectively. Matrices Z_i encode a wide variety of models, including random intercepts (Z_i are columns of 1's that add u_i to all datapoints from the ith study) and random slopes (Z_i also scale u_i according to the magnitude of a covariate), see e.g. Pinheiro and Bates (2006). In our study, we assume that the observation error covariance matrices Λ_i are given and that the random effects covariance matrix is an unknown diagonal matrix, i.e., $\Gamma = \text{Diag}(\gamma)$, $\gamma \in \mathbb{R}_+^s$. This assumption corresponds to the meta-analysis and meta-regression branch of mixed effects problems, which is the primary focus of our applied collaborations (see e.g. Zheng et al. (2022); Lescinsky et al. (2022); Razo et al. (2022); Stanaway et al. (2022); Dai et al. (2022).) The theoretical developments in this work allow extensions to other types of repeated measure models, but practical implementation requires significant additional effort, and we leave these extensions to future work.

Defining group-specific error terms $\omega_i = Z_i u_i + \varepsilon_i$, we get a compact formulation that recasts (2.2.1) as a correlated noise model:

$$Y_i = X_i\beta + \omega_i, \quad \omega_i \sim \mathcal{N}(0, \Omega_i(\Gamma)), \quad \Omega_i(\Gamma) = Z_i \Gamma Z_i^T + \Lambda_i. \tag{2.2.2}$$

For brevity, we refer to $\Omega_i(\Gamma)$ as just Ω_i. The reformulation (2.2.2) yields the following marginalized negative log-likelihood function of a linear mixed-effects model (Patterson and Thompson, 1971):

$$\mathcal{L}_{ML}(\beta, \Gamma) := \sum_{i=1}^{m} \frac{1}{2}(y_i - X_i\beta)^T \Omega_i^{-1}(y_i - X_i\beta) + \frac{1}{2}\ln\det\Omega_i. \tag{2.2.3}$$

Maximum likelihood estimates for β and Γ are obtained by solving the optimization problem

$$\min_{\beta,\Gamma} \ \mathcal{L}_{ML}(\beta, \Gamma) \quad \text{s.t.} \quad \Gamma \in \mathbb{S}_+^q \,. \tag{2.2.4}$$

Theorem 1 (Existence of a Minimizer, Theorem 1 from Aravkin et al. (2022a)). *Let the assumptions in the statement of problem* (2.2.4) *hold. Then optimal solutions to* (2.2.4) *exist.*

At this point, we bring in three basic definitions from variational analysis Rockafellar and Wets (2009).

Definition 1 (Epigraph and level sets). *The epigraph of a function* $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ *is defined as*

$$\operatorname{epi} f = \{(x, \alpha) : f(x) \leq \alpha\}.$$

For a given α*, the* α*-level set of* f *is defined as*

$$lev_\alpha f = \{x : f(x) \leq \alpha\}.$$

Definition 2 (Lower semicontinuity and level-boundedness). *A function* $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ *is lower semicontinous (lsc) when* $\operatorname{epi} f$ *is closed, and level-bounded when all level sets* $lev_\alpha f$ *are bounded.*

Definition 3 (Convexity). *A function* $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ *is convex when* $\operatorname{epi} f$ *is a convex set. Equivalently,*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \quad \forall x, y \in \operatorname{dom} f,\ \lambda \in (0,1),$$

where $\operatorname{dom} f := \{x \in \mathbb{R}^n \mid f(x) < +\infty\}$.

Definition 4 (Weak convexity). *A function* $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ *is* η*-weakly convex* $f(\cdot) + \frac{\eta}{2}\|\cdot\|^2$ *is convex.*

The negative log likelihood (2.2.4) is nonlinear and nonconvex, and requires an iterative numerical solver. However, it is convex with respect to β, and weakly convex with respect to γ, with a weak convexity constant $\overline{\eta}$ computed in (Aravkin et al., 2022b, Section 5.1) . The expected value of the posterior mode β given Γ has the closed form representation

$$\beta(\Gamma) = \underset{\beta}{\operatorname{argmin}}\, \mathcal{L}(\beta, \Gamma) = \left(\sum_{i=1}^{m} X_i^T \Omega_i^{-1} X_i \right)^{-1} \sum_{i=1}^{m} X_i^T \Omega_i^{-1} y_i. \tag{2.2.5}$$

The individual random effects u_i can be found through the minimization of conditional likelihood of random effects given β and Γ:

$$\min_{u_i} \mathcal{L}(u_i | \beta, \Gamma) = \min_{u_i} \left(\frac{1}{2} u_i^T \Gamma^{-1} u_i + \frac{1}{2} (Z_i u_i - Y_i + F_i(\beta))^T \Lambda_i^{-1} (Z_i u_i - Y_i + F_i(\beta)) \right), \tag{2.2.6}$$

This problem has a closed form solution commonly known as Best Linear Unbiased Predictor, or BLUP. It was first derived by Harville (1976) as an extension to Gauss-Markov theorem.

$$u_i = (\Gamma^{-1} + Z_i^T \Lambda_i^{-1} Z_i)^{-1} Z_i^T \Lambda_i^{-1} (Y_i - X_i \beta). \tag{2.2.7}$$

By using the simplification $\Gamma = \text{Diag}(\gamma)$, we obtain the problem

$$\min_{\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}^q_+} \mathcal{L}(\beta, \gamma) := \mathcal{L}_{ML}(\beta, \text{Diag}(\gamma)) . \tag{2.2.8}$$

In this setting, when an entry γ_j takes the value 0 the corresponding coordinates of all random effects u_{ij} are identically 0 for all i.

Verification of the existence to solutions to (2.2.8) and, more generally, (2.2.4) follows from the work of Zheng et al. (2021). Standalone proofs for the existence of minimizers are developed in (Aravkin et al., 2022b, Theorem 1), and extended to the presence of regularizers in (Aravkin et al., 2022b, Theorem 2).

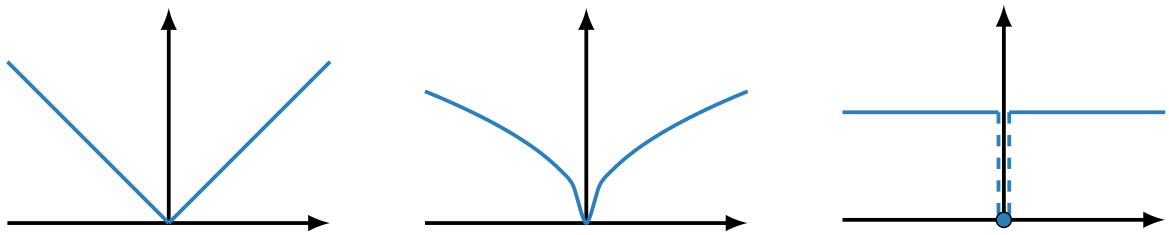
This paper focuses the case where Γ is diagonal, (often referred to as *the diagonal setup*) and all Λ_i are known (see (2.2.8)), following the meta-analysis use-case (Zheng et al., 2021) that is widely employed in epidemiological studies Murray et al. (2020). While the proposed approach can be extended to the non-diagonal case, we leave it for future work, save for a brief discussion in Section 2.4.

2.2.1 Prior Work on Feature Selection for Mixed-Effects Models

Variable (feature) selection models seeks to select or rank the most important predictors in a dataset in order to get a parsimonious model at a minimal cost to prediction quality. Feature selection may be performed both on β, to find the sparse set of covariates that best explains the mean, and on γ, to find the sparse set of covariates that best accounts for variation between groups. Both types of selection have been studied in the literature, and both are accessible using the methods developed here. If the desired number of coefficients k is given, then the feature selection problem can be formulated as the minimization of a loss function $f(\theta)$ (e.g. the negative log-likelihood) subject to a zero-norm constraint:

$$\min_{\theta} f(\theta) \quad \text{s.t.} \quad \|\theta\|_0 \leq k \tag{2.2.9}$$

where $\|\theta\|_0$ denotes the number of nonzero entries in θ, see panel (c) of Figure 2.2.1.



(a) ℓ_1 (b) $\ell_{1/2}$ (c) ℓ_0

Figure 2.2.1: Common convex and non-convex regularizers used for feature selection.

The constraint in (2.2.9) is combinatorial, and a common workaround is to relax it to a one-norm constraint, with $\|\theta\|_1$ equal to the sum of absolute values of the entries of θ. The best-known

example of this approach is the least absolute square shrinkage operator (LASSO) studied by Tibshirani (1996b) for linear regression, see panel (a) of Figure 2.2.1.

Feature selection for LMEs is more difficult than for linear regression models. In linear regression the observations are independent, whereas in mixed-effects setup they are generally correlated. In addition, LMEs have both mean effect variables β as well as random variance variables Γ. The shrinkage operator approach for linear regression (Tibshirani, 1996b) was first adapted to the problem of feature selection for the fixed effects in mixed-effect models by Lan (2006). The removal of a random effect from the model requires the elimination of an entire row and column from Γ. To make the problem more tractable, Chen and Dunson (2003) reparamtrized Γ through a modified Cholesky decomposition $\Gamma(D, L) := DLL^T D$, where D is a diagonal matrix and L is a lower-triangular matrix with ones on the main diagonal, and focused on selecting elements of D. Based on this idea, Bondell et al. (2010) extended the Adaptive LASSO regularizer (Lan (2006); Xu et al. (2015)) to mixed-effects setting using the objective $\mathcal{L}(\beta, \Gamma(D, L)) + \lambda \left(\sum_{i=1}^{p} \left| \frac{\beta_i}{\hat{\beta}_i} \right| + \sum_{j=1}^{q} \frac{D_{ii}}{\hat{D}_{ii}} \right)$, where $\hat{\beta}$ and $\hat{D}$ are the solution of a non-penalized maximum likelihood problem and λ is a tuning parameter for the weighted regularizer and is called the regularization parameter. Ibrahim et al. (2011b) use a similar approach, penalizing non-zero elements Γ_{ij} directly. Other methods that use Adaptive LASSO for simultaneous selection of fixed and random effects are Lin et al. (2013a); Fan et al. (2014); Pan and Shang (2018). Adaptive LASSO is available to practitioners via `R` packages `glmmLasso`[3] (Groll and Tutz (2014)) and `lmmLasso`[4](Schelldorfer et al. (2011)).

A popular nonconvex regularizer used for feature selection is smoothed clipped absolute deviation (SCAD) Fan and Li (2001). The adaptation of the SCAD penalty to select both fixed and random features in linear mixed models was developed by Fan and Li (2012). SCAD was also used by Chen et al. (2015) for selecting fixed effects and establishing the existence of random effects in ANOVA-type models. Finally, Ghosh and Thoresen (2018) studied SCAD regularization for selecting mean effects in high-dimensional genomics problems. One can refer to Müller et al. (2013); Buscemi and Plaia (2019) for a detailed overview of different feature selection approaches.

To better compare methods, we need to consider the tuning of the regularization parameter λ. The output of a shrinkage model critically depends on the tuning parameter λ. The entire range of λ values is captured by the notion of a "λ-path in the model space", with the best parameter and the final model chosen using information criteria. According to Müller et al. (2013), the most widely used information criterion is the marginal AIC criterion (Vaida and Blanchard (2005)), $AIC := 2\,\mathcal{L}(\hat{\theta}) + 2\alpha_n(p+q)$, where $\hat{\theta}$ includes all the estimated parameters (β, Γ), and $\alpha_n := n(n-p-q-1)$ for the finite sample case (Sugiura (1978)). Alternatively, LASSO-type methods (Bondell et al. (2010); Ibrahim et al. (2011b)) use a BIC-type information criterion, $BIC := 2\,\mathcal{L}(\hat{\theta}) + \log(n)(p+q)$. BIC performs well in practice, but does not have theoretical guarantees (Schelldorfer et al. (2011)). Finally, Hui et al. (2017) suggests penalizing the number

[3] https://rdrr.io/cran/glmmLasso/man/glmmLasso.html
[4] https://rdrr.io/cran/lmmlasso/

of non-zero random effects in the resulting model instead of penalizing the number of random features with non-zero variance, especially when random effects are optimized directly instead of using BLUP (2.2.7).

2.3 Algorithms for Feature Selection

We approach feature selection by adding a regularizer to model (2.2.8):

$$\min_{x} \mathcal{L}(x) + R(x) + \delta_{\mathcal{C}}(x), \tag{2.3.1}$$

where $x = (\beta, \gamma)$, $\mathcal{C} := \mathbb{R}^p \times \mathbb{R}_+^q$, $R : \mathbb{R}^P \times \mathbb{R}_+^q \to \overline{\mathbb{R}}_+ := \mathbb{R}_+ \cup \{+\infty\}$ is a lower semi-continuous (lsc) regularization term, and $\delta_{\mathcal{C}}$ is the convex indicator function, where $\delta_{\mathcal{C}}(x) := 0$ for $x \in \mathcal{C}$ and $+\infty$ otherwise. By (Aravkin et al., 2022b, Theorem 2), solutions to (2.3.1) always exist when R has compact lower level sets. The most common regularizers are separable taking the form

$$R(x) = \sum_{i=1}^{p} r_i(x_i), \tag{2.3.2}$$

with typical choices for the component functions r_i given in Table 2.3.1.

2.3.1 Variable Selection via Proximal Gradient Descent

Since $\mathcal{L}$ is differentiable on its domain and proximal operator for $\alpha R + \delta_{\mathcal{C}}$ is computationally tractable, the Proximal Gradient Descent (PGD) Algorithm (e.g. see Beck (2017)) offers a simple numerical strategy for estimating first-order stationary points for (2.3.1). The proximal operator for $\alpha R + \delta_{\mathcal{C}}$ is defined as the mapping $\text{prox}_{\alpha R + \delta_{\mathcal{C}}}(z) := \text{argmin}_{y \in \mathcal{C}} \; R(y) + \frac{1}{2\alpha}\|y - z\|_2^2$, and the PGD iteration is given by $x^+ = \text{prox}_{\alpha R + \delta_{\mathcal{C}}}(x - \alpha \nabla \mathcal{L}(x))$, where α is a stepsize. When $R(x)$ has the form given in (2.3.2), we have $\text{prox}_R(z) = (\text{prox}_r(z_1), \ldots, \text{prox}_r(z_q))$. Table 2.3.1 provides closed form expressions for the proximal operators of commonly used regularizers. For all of these cases, the following theorem gives closed form expressions for $\text{prox}_{\alpha R + \delta_{\mathcal{C}}}(z)$.

Regularizer	$r(x)$, $x \in \mathbb{R}$	$\text{prox}_{\alpha r}(z)$
LASSO (ℓ_1)	$\lvert x\rvert$	$\text{sign}(z)(\lvert z\rvert - \alpha)_+$
A-LASSO	$\bar{w}\lvert x\rvert$, $\bar{w} \geq 0$	$\text{sign}(z)(\lvert z\rvert - \alpha\bar{w})_+$
SCAD	$\begin{cases} \sigma\lvert x\rvert, & \lvert x\rvert \leq \sigma \\ \frac{-x^2+2\rho\sigma x-\sigma^2}{2(\rho-1)}, & \sigma < \lvert x\rvert < \rho\sigma \\ \frac{\sigma^2(\rho+1)}{2}, & \lvert x\rvert > \rho\sigma \end{cases}$	$\begin{cases} \text{sign}(z)(\lvert z\rvert - \sigma\alpha)_+, & \lvert z\rvert \leq \sigma(1+\alpha) \\ \frac{(\rho-1)z-\text{sign}(z)\rho\sigma\alpha}{\rho-1-\alpha}, & \sigma(1+\alpha) < \lvert z\rvert \leq \rho\sigma \\ z, & \lvert z\rvert > \max(\rho, 1+\alpha)\sigma \end{cases}$
$\delta_{\lVert x\rVert_0 \leq k}$ (ℓ_0 ball)	$\begin{cases} 0, & \#\{\lvert x_i\rvert \neq 0\} \leq k \\ \infty, & \text{otherwise} \end{cases}$	keep k largest $\lvert x_i\rvert$, set the rest to 0

Table 2.3.1: Proximal operators for commonly used sparsity-promoting regularizers.

Under the assumption that R is level-compact Aravkin et al. (2022a) extended Theorem 1 to the regularized case as follows:

Theorem 2 (Existence of a Minimizer for 2.3.1, Aravkin et al. (2022a), Theorem 2)**.** *Let the assumptions in the statement of problem* (2.2.8) *hold. Suppose* $\widehat{R} : \mathbb{R}^p \times \mathbb{R}^q_+ \to \mathbb{R} \cup \{+\infty\}$ *is lsc and level compact (i.e.,* $\text{epi}\, R := \{((\beta,\gamma),\nu) \mid R(\beta,\gamma) \leq \nu\}$ *is closed and* $\{(\beta,\gamma) \mid R(\beta,\gamma) \leq \nu\}$ *is bounded for all* $\nu \in \mathbb{R}$*). Then* $\mathcal{L} + \widehat{R}$ *is level compact and solutions to the following optimization problem exist:*

$$\min_{\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}^q_+} \mathcal{L}(\beta,\gamma) + \widehat{R}(\beta,\gamma). \tag{2.3.3}$$

In practice, it is often advisable to include a constraint of the form $\gamma \leq \gamma_{\max}$ for $\gamma \in \mathbb{R}^q_{++}$ chosen sufficiently large since an excessively large variance usually indicates that the model is poorly posed and needs review. Such a constraint is also numerically expedient since it prevents γ from diverging. Table 2.3.1 provides closed form expressions for the proximal operators of commonly used regularizers.

Theorem 3 (Proximal operator for bounded γ)**.** *We consider modified regularizers* $r(\gamma)$ *from the Table 2.3.1 that include an additional constraint on* γ *of the form* $0 \leq \gamma \leq \bar{\gamma}$*, for* $\bar{\gamma} \in [0, +\infty]$*. We have the following results.*

1. *For SCAD, we have for all* i *that* $\text{prox}_{(\alpha r + \delta_{[0,\bar{\gamma}]})}(\gamma_i) = \begin{cases} \text{prox}_{\alpha r}(\gamma_i), & 0 \leq \gamma_i < \bar{\gamma} \\ \bar{\gamma}, & \gamma_i \geq \bar{\gamma} \\ 0, & \textit{otherwise} \end{cases}$.

2. *For LASSO, A-LASSO we have for all* i *that* $\text{prox}_{(\alpha r + \delta_{[0,\bar{\gamma}]})}(\gamma_i) = \begin{cases} \text{prox}_{\alpha r}(\gamma_i), & 0 \leq \gamma_i < \bar{\gamma} + \alpha \\ \bar{\gamma}, & \gamma_i \geq \bar{\gamma} + \alpha \\ 0, & \textit{otherwise} \end{cases}$.

3. *For* $R(\cdot) = \delta_{\text{lev}_{\|\cdot\|_0}(k)}$ *the* $\text{prox}_{\alpha R + \delta_{\mathcal{C}}}(\gamma)$ *can be evaluated by taking* k *largest coordinates of* γ *such that* $0 \leq \gamma_i \leq \bar{\gamma}$*, and setting the remainder to* 0.

The proof of the Theorem 3 is provided in Appendix A.3. The PGD algorithm is detailed in Algorithm 1. The algorithm's step-size α depends on the Lipschitz constant; an upper-bound is given in Appendix A.5. In practice, α is computed using a line-search, since the available estimate for L is very conservative.

$x = x_0$, $\alpha < \frac{1}{L}$, where $\mathcal{L}$ is L-Lipschitz
while not converged **do**
  $x^+ = \text{prox}_{\alpha R + \delta_{\mathcal{C}}}(x - \alpha \nabla \mathcal{L}(x))$;
end

Algorithm 1: Proximal Gradient Descent for Linear Mixed-Effect Models

The main advantages of Algorithm 1 are its simplicity and flexibility. The main loop needs only the gradient and prox operator, and the structure of the algorithm is independent of the choice of R. Algorithm 1 locates first-order stationary points under weak assumptions, in particular neither the objective nor the regularizer need be convex (Beck, 2017; Attouch et al., 2013).

2.3.2 Variable Selection via MSR3

To develop an approach that is both more efficient and accurate, we extend the SR3 regularization of Zheng et al. (2019) to LMEs. We call the extension MSR3, since we are focusing on mixed effects models. Starting with the regularized likelihood (2.3.1) we introduce auxiliary parameters designed to discover the fixed and random features:

$$\min_{x,w} \mathcal{L}(x) + R(w) + \delta_{\mathcal{C}}(x) + \kappa_\eta(x - w), \tag{2.3.4}$$

where κ_η penalizes deviations between $x = (\beta, \gamma)$ and $w = (\hat{\beta}, \hat{\gamma})$, and also guarantees that the objective is convex with respect to the γ components of x for sufficiently large η:

$$\kappa_\eta(x - w) = \frac{\eta}{2}\|x - w\|^2 = \frac{\eta}{2}\|\beta - \hat{\beta}\|^2 + \frac{\eta}{2}\|\gamma - \hat{\gamma}\|^2 \tag{2.3.5}$$

with $\eta \geq \bar{\eta}$ where $\bar{\eta}$ is the weak convexity constant computed in (Aravkin et al., 2022b, Section 5.1). As $\eta \uparrow \infty$, the extended objective (2.3.4) converges in an epigraphical sense to the original objective (2.3.1). However, feature selection accuracy does not require this continuation, indeed, we show that a fixed modest value such as $\eta = 1$ can be used (Zheng et al., 2019).

To understand the algorithm and logic behind the objective (2.3.4), we define an optimal value function $u_\eta(w)$ and the solution set $S_\eta(w)$:

$$\begin{aligned} u_\eta(w) &= \min_x \ \mathcal{L}(x) + \delta_{\mathcal{C}}(x) + \kappa_\eta(x - w) \\ S_\eta(w) &= \operatorname*{argmin}_x \mathcal{L}(x) + \delta_{\mathcal{C}}(x) + \kappa_\eta(x - w). \end{aligned} \tag{2.3.6}$$

Substituting (2.3.6) into (2.3.4) transforms (2.3.4) into

$$\min_{w} u_\eta(w) + R(w) \tag{2.3.7}$$

Here we have transformed the original regularized likelihood (2.3.1) through relaxation and partial minimization to obtain an equivalent problem (2.3.7) for w with the same regularizer. The value function u_η encapsulates global variational information on the function $\mathcal{L}(x) + \delta_{\mathcal{C}}(x)$ relative to w.

In the case of linear regression, the function u_η has a closed form solution Zheng et al. (2019). However, in both the linear regression context of Zheng et al. (2019) and in the LME context studied here, we need only compute $S_\eta(w)$ in order to optimize (2.3.7). Indeed, in (Aravkin et al., 2022b, Section 5) it is shown that there exists a computable $\bar{\eta} > 0$, which we have called the weak convexity constant, such that $\mathcal{L} + \delta_{\mathcal{C}} + \kappa_\eta(\cdot - w)$ is strongly convex for all $\eta > \bar{\eta}$ regardless of the choice of w. This allows us to show that u_η is well-defined, differentiable, and Lipschitz continuous, with

$$\nabla u_\eta(w) = \nabla_w k_\eta(x - w)|_{x=S_\eta(w)} = \eta(w - S_\eta(w)). \tag{2.3.8}$$

Our empirical studies indicate that (2.3.7) has advantages over (2.3.1) from an optimization perspective since u_η typically has nearly spherical level-sets while keeping the position of minima close to those of $\mathcal{L}(x)$. This effect is extensively studied and validated for a quadratic loss function in the original work of Zheng et al. (2019). In the center panel of Figure 2.1.1, we plot the level-sets of $\mathcal{L}(x) + \|x\|_1$ (left column) and $u_\eta + \|\cdot\|_1$ (right column) for the same mixed-effect problem. The more spherical geometry of the latter allows the Algorithm 2 (described below) to converge in 21 iterations, whereas Algorithm 1 takes 1284 iterations. The difference is most pronounced when the minimum sits on the boundary of the feasible set, which is always the case for the variable selection problems with sparse support.

We apply PGD to optimize the regularized value function u_η which yields the iteration

$$w^+ = \text{prox}_{\alpha^{-1}R}(w - \alpha \nabla u_\eta(w)) \tag{2.3.9}$$

The results in Aravkin et al. (2022b) show that all components of the iteration (2.3.9) are well-defined. The equivalence of Algorithm 2 and (2.3.9) is established in the following lemma, which extends the relationship studied by Zheng et al. (2019) to the case of $x = (\beta, \gamma)$.

Lemma 4 (Equivalence of Algorithms)**.** *Algorithm 2 is equivalent to* (2.3.9)*.*

Proof. Substituting (2.3.8) into (2.3.9), we see that the iteration (2.3.9) is equivalent to the alternating minimization scheme outlined in the Algorithm 2. □

1 $w = w_0$
2 **while** not converged **do**
3 $\quad x^+ = \arg\min_x \mathcal{L}(x) + \delta_{\mathcal{C}}(x) + \kappa_\eta(x - w)$
4 $\quad w^+ = \text{prox}_{\alpha^{-1}R}(x^+)$
5 **end**

Algorithm 2: Proximal Gradient Descent for Value Function

In (Aravkin et al., 2022b, Theorem 6), it is shown that for any sequence $\eta_k \uparrow \infty$ the associated optimal solutions (x^k, w^k) to (11) satisfy $\mathcal{L}(x^k)+R(w^k) \uparrow \inf_{x\in\mathbb{R}^p\times\mathbb{R}^q_+} \mathcal{L}(x)+R(x)$ with $\|x^k - w^k\| \to 0$. In particular, every cluster point of the sequences $\{x^k\}$ and $\{w^k\}$ are solutions to (2.2.8), where such cluster points exist whenever the function R is coercive, i.e. $\lim_{\|x\|\uparrow\infty} R(x) = +\infty$. Just how close w^k is to a solution to (2.2.8) remains an open question, however, our numerical studies in Section 2.4 show that η can be chosen surprisingly small. Indeed, we typically take $\eta = 1$.

In the linear regression setting of Zheng et al. (2019), Algorithm 2 can be implemented exactly. In the nonlinear case, evaluating x^+ requires an iterative algorithm. For this we use an interior point method which replaces the indicator function $\delta_{\mathcal{C}}$ by a smooth log-barrier term. This allows us to approximate both u_η and its gradient where the degree of the approximation is controlled by the convergence criteria of the interior point algorithm.

An Interior Point Method for Approximating u_η. In order to solve for the x^+ update in line 2 of Algorithm 2, we must optimize a convex loss with linear inequality constraints, that is, for a fixed $w = (\hat{\beta}, \hat{\gamma})$, we need to solve

$$\min_{\beta,\gamma} \mathcal{L}(\beta, \gamma) + \kappa_\eta(\beta - \hat{\beta}, \gamma - \hat{\gamma}) \quad \text{s.t.} \quad 0 \leq \gamma. \tag{2.3.10}$$

This problem is well suited for an interior point approach (Kojima et al., 1991; Nesterov and Nemirovskii, 1994; Wright, 1997a; Vanderbei and Shanno, 1999). First, the inequality constraint $0 \leq \gamma$ is relaxed using the perspective of the negative log, i.e $\varphi : \mathbb{R}^q \times \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ given by

$$\varphi(\gamma, \mu) := \begin{cases} -\mu \sum_{i=1}^q \ln(\gamma_i/\mu) & , \ \mu > 0, \\ \delta_{\mathbb{R}^q_+}(\gamma) & , \ \mu = 0, \\ +\infty & , \ \mu < 0. \end{cases}$$

The mapping φ is known to be a closed proper convex function and, for $\mu > 0$, it is essentially equivalent to the well-known log-barrier function. For more information on the perspective mapping, its calculus, and perspective duality, we refer the reader to Aravkin et al. (2018, 2013). We call η the coupling parameter and μ the log-barrier parameter and write $\phi_\mu(\cdot) := \varphi(\cdot, \mu)$. The relaxed problem employs auxiliary variables $(\tilde{\beta}, \tilde{\gamma})$ and relaxation parameters $0 \leq \eta$ and $0 \leq \mu$ to obtain the problem

$$\begin{aligned} \min_{(\beta,\gamma),(\tilde{\beta},\tilde{\gamma})} \ & \mathcal{L}(\beta, \gamma) + \phi_\mu(\gamma) + \kappa_\eta(\beta - \tilde{\beta}, \gamma - \tilde{\gamma}) + R(\tilde{\beta}, \tilde{\gamma}) \\ \text{s.t.} \ & \tilde{\gamma} \geq 0 \ . \end{aligned} \tag{2.3.11}$$

We rewrite (2.3.11) so as to separate the smooth and nonsmooth components to obtain

$$\min_{(\beta,\gamma),(\tilde{\beta},\tilde{\gamma})} \mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma}) \ , \tag{2.3.12}$$

where

$$\mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) := \mathcal{L}(\beta,\gamma) + \phi_\mu(\gamma) + \kappa_\eta(\beta-\tilde{\beta},\gamma-\tilde{\gamma}). \tag{2.3.13}$$

Observe that, for all $\mu,\eta \in \mathbb{R}_+$, $\mathcal{L}_{\eta,\mu}$, $\nabla \mathcal{L}_{\eta,\mu}$ and $\nabla^2 \mathcal{L}_{\eta,\mu}$ are continuous on $(\mathbb{R}^p \times \mathrm{dom}(\phi_\mu)) \times (\mathbb{R}^p \times \mathbb{R}^q)$ (see Appendix A.2) so that $\mathcal{L}_{\eta,\mu}$ is smooth on its domain. As in Zheng et al. (2019), we use the decoupling to write (2.3.12) as an iterated optimization problem over the smooth components of the objective. This yields a representation of the form

$$\min_{(\tilde{\beta},\tilde{\gamma})} u_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma}), \tag{2.3.14}$$

where

$$u_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) := \min_{(\beta,\gamma)} \mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) \ . \tag{2.3.15}$$

This is the formulation of the mixed-effects variable selection problem we study. Notice that this value function differs from a simplified definition in (2.3.6): we replaced the linearity constraint with a perspective function and got another parameter μ due to it. The log-barrier penalty approximates the indicator function to the positive orthant as μ decreases; indeed, the function $\gamma \mapsto \mu \ln(\gamma)$ epi-converges to the indicator function $\delta_{\mathbb{R}^n_+}(\gamma)$ as $\mu \downarrow 0$ (Rockafellar and Wets (2009)). The penalty (homotopy) parameter μ is progressively decreased to 0 as the algorithm proceeds as described below. We refer to (2.3.15) when we say "value function" from now on.

Our focus is on the optimal value function $u_{\eta,\mu}$ which captures global variational information about the function $\mathcal{L}$ over its domain. In Section 2.5 we show that $u_{\eta,\mu}$ has a locally Lipschitz continuous gradient and that the evaluation of $u_{\eta,\mu}$ and $\nabla u_{\eta,\mu}$ is accomplished by optimizing a well conditioned strongly convex function. This allows us to apply the PGD algorithm to the function $u_{\eta,\mu}$ rather than the function $\mathcal{L}$. Our numerical studies in Section 2.4 show that the global information captured by $u_{\eta,\mu}$ significantly improves both the accuracy of the solution obtained and the overall numerical efficiency of the algorithm.

For $\gamma > 0$, the necessary optimality conditions for $\mathcal{L}_{\mu,\eta}$ in γ give us the relation

$$\nabla_\gamma \mathcal{L}_{\mu,\eta}(\beta,\gamma) = \nabla_\gamma \mathcal{L}(\beta,\gamma) + \eta(\gamma-\hat{\gamma}) - \mu \mathrm{Diag}\,(\gamma)^{-1}\mathbf{1} = 0, \tag{2.3.16}$$

where $\mathbf{1}$ is the vector of all ones of the appropriate dimension. By setting $v = \nabla_\gamma \mathcal{L}_{\mu,\eta}(\beta,\gamma) + \eta(\gamma-\hat{\gamma})$, we can rewrite this equation as

$$v \odot \gamma - \mu\mathbf{1} = 0, \tag{2.3.17}$$

where $\mathbf{1}$ is the vector of all ones of the appropriate dimension and "$\odot$" denotes the Hadamard (or simply element-wise) product. The complete set of optimality conditions for (2.3.15) can

now be written as

$$G_{\mu,\eta}(v,\beta,\gamma) := \begin{bmatrix} v \odot \gamma - \mu \mathbf{1} \\ \nabla_\beta \mathcal{L}(\beta,\gamma) + \eta(\beta - \hat{\beta}) \\ \nabla_\gamma \mathcal{L}(\beta,\gamma) + \eta(\gamma - \hat{\gamma}) - v \end{bmatrix} = 0. \tag{2.3.18}$$

We then apply Newton's method to (2.3.18), that is, in each iteration the search direction $[\Delta v, \Delta\beta, \Delta\gamma]$ solves the linear system

$$\nabla G_{\mu,\eta}(v,\beta,\gamma) \begin{bmatrix} \Delta v \\ \Delta\beta \\ \Delta\gamma \end{bmatrix} = -G_{\mu,\eta}(v,\beta,\gamma), \quad \nabla G_{\mu,\eta}(v,\beta,\gamma) = \begin{bmatrix} \mathrm{Diag}\,(\gamma) & 0 & \mathrm{Diag}\,(v) \\ 0 & \nabla^2_{\beta\beta}\mathcal{L} + \eta I & \nabla^2_{\beta\gamma}\mathcal{L} \\ -I & \nabla^2_{\gamma\beta}\mathcal{L} & \nabla^2_{\gamma\gamma}\mathcal{L} + (\eta + \overline{\lambda})I \end{bmatrix}$$

and we have used the fact that $v \odot \gamma = \mathrm{Diag}\,(v)\,\gamma = \mathrm{Diag}\,(\gamma)\,v$. The exact formulae for the derivatives of $\mathcal{L}$ are provided in the Appendix A.2.

The general structure of the algorithm is as follows. Given a search direction $[\Delta v^{(k)}, \Delta\beta^{(k)}, \Delta\gamma^{(k)}]$, choose a step of size $\alpha_k > 0$ so that the update

$$\begin{pmatrix} v^{(k+1)} & \beta^{(k+1)} & \gamma^{(k+1)} \end{pmatrix} = \begin{pmatrix} v^{(k)} & \beta^{(k)} & \gamma^{(k)} \end{pmatrix} + \alpha_k \begin{pmatrix} \Delta v^{(k)} & \Delta\beta^{(k)} & \Delta\gamma^{(k)} \end{pmatrix}$$

satisfies the conditions

$$\begin{aligned} \textit{Positivity:} \quad & \gamma^{(k+1)} > 0, \ v^{(k+1)} > 0 \\ \textit{Sufficient Descent:} \quad & \|G_{\eta,\mu}(v^{(k+1)}, \beta^{(k+1)}, \gamma^{(k+1)})\| \leq 0.99 \|G_{\eta,\mu}(v^{(k)}, \beta^{(k)}, \gamma^{(k)})\|, \end{aligned}$$

where the parameter 0.99 is used to bias toward the acceptance of a full Newton step. At each iteration the relaxation parameter μ is updated by the formula $\mu^{(k+1)} = {v^{(k)}}^T \gamma^{(k)}/q$, where ${v^{(k)}}^T \gamma^{(k)}$ is the duality gap at iteration k. The algorithm terminates when the criteria

$$\begin{aligned} \|G_{\mu,\eta}(v^{(k+1)}, \beta^{(k+1)}, \gamma^{(k+1)})\| &\leq \texttt{tol} \\ \mu &\leq \texttt{tol} \end{aligned}$$

are both satisfied, so the interior point problem is nearly stationary, and closely approximates the original problem (2.3.10). MSR3 is summarized in Algorithm 3, which approximates Algorithm 2 as the tolerance goes to 0. In the numerical experiments, we use $\texttt{tol} = 10^{-5}$, and accuracy does not change as the tolerance parameter decreases.

```
1 w = w_0
2 while not converged do
3 |   x^+ satisfies ||G_{μ,η}(v^+, x^+)|| ≤ tol, μ ≤ tol
4 |   w^+ = prox_{α^{-1}R}(x^+)
5 end
```

Algorithm 3: MSR3

Positive Approximation of the Hessian For many datasets the weak convexity constant $\overline{\eta}$ can be extremely large and difficult to compute. However, if η is too small $\nabla^2_{\gamma\gamma}\,\mathcal{L}_{\mu,\eta}(\beta,\gamma)$ is negative-(semi)definite. Negative definite Hessians can hamper the convergence of second-order methods (e.g., see Nocedal and Wright (2006)). Therefore, one must take care in selecting η. For this, we recall from (Aravkin et al., 2022b, Lemma 3) that

$$\nabla^2\,\mathcal{L}\,(\beta,\gamma) = \sum_{i=1}^{m} S_i^T \begin{bmatrix} X_i^T \\ -Z_i^T \end{bmatrix} \Omega_i(\gamma)^{-1} \begin{bmatrix} X_i & -Z_i \end{bmatrix} S_i - \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2}(Z_i^T\Omega_i(\gamma)^{-1}Z_i)^{\circ 2} \end{bmatrix}.$$

This implies that negative eigenvalues for the Hessian must arise from the Hessian with respect to γ, $\nabla^2_{\gamma\gamma}\,\mathcal{L}(\beta,\gamma)$, and more specifically, the term $(Z_i^T\Omega_i(\gamma)^{-1}Z_i)^{\circ 2}$. A positive semidefinite approximation to the Hessian is obtained by simply dropping this term.

2.3.3 Relaxation and Efficient Algorithms: MSR3 and MSR3-Fast

While algorithm (2) is modular, it requires solving a nonlinear optimization problem in $x = (\beta,\gamma)$ for each single update of $w = (\hat{\beta},\hat{\gamma})$. To make the implementation as efficient as possible, we designed a more balanced updating scheme, that alternates Newton iterations as described in the interior point algorithm with w updates. We update w whenever we are sufficiently close to the 'central path' in the interior point method, a condition that can be checked rigorously using optimality conditions. This scheme is detailed in Algorithm 4.

In designing Algorithm 4, we chose a particular central path parameter, $\tau = 0.5$ in line 8, that controls how far the interior point method needs to proceed before we take a proximal gradient step. We explored the effect of this parameter on performance and timing in Section 2.4.2 and found that it did not have any effect on either for values between 0.1 and 0.9. MSR3-fast was competitive with respect to time compared to PGD and PGD with line search (also as reported in Chapter 2.4.2) for problems up to 1000 features.

2.4 Verifications

2.4.1 MSR3 for Covariate Selection

In this section we compare the feature selection accuracy and the numerical efficiency of Algorithms 1 and 4 when using the LASSO, A-LASSO, SCAD, and L0 sparsity regularizers. We begin by describing how the data is generated for our numerical simulations followed by a description of how the regularization parameter λ and the coupling parameter η were chosen. Our experiments on real data are presented in Section 2.4.3.

Experimental Setup. The number of fixed effects p and random effects q are set at 20 with $\beta = \gamma = [\frac{1}{2}, \frac{2}{2}, \frac{3}{2}, \ldots, \frac{10}{2}, 0, 0, 0, \ldots, 0]$, i.e. the first 10 covariates are increasingly important and

progress $\leftarrow$ **True**; iter = 0;
$\beta^+, \tilde{\beta}^+ \leftarrow \beta_0$; $\gamma^+, \tilde{\gamma}^+ \leftarrow \gamma_0$; $v^+ \leftarrow 1 \in \mathbb{R}^q$; $\mu \leftarrow \frac{{v^+}^T \gamma^+}{10q}$
while iter $<$ max_iter and $\|G_{\eta,\mu}(\beta^+, \gamma^+, v^+)\| >$ tol and progress **do**
  $\beta \leftarrow \beta^+$; $\gamma \leftarrow \gamma^+$; $\tilde{\beta} \leftarrow \tilde{\beta}^+$; $\tilde{\gamma} \leftarrow \tilde{\gamma}^+$
  $[dv, d\beta, d\gamma] \leftarrow \nabla G_{\eta,\mu}((\beta, \gamma, v), (\tilde{\beta}, \tilde{\gamma}))^{-1} G_{\eta,\mu}((\beta, \gamma, v), (\tilde{\beta}, \tilde{\gamma}))$ // Newton Iteration
  $\alpha \leftarrow 0.99 \times \min\left(1, -\frac{\gamma_i}{d\gamma_i}, \forall i : \ d\gamma_i < 0\right)$
  $\beta^+ \leftarrow \beta + \alpha d\beta$; $\gamma^+ = \gamma + \alpha d\gamma$; $v^+ \leftarrow v + \alpha dv$
  if $\|\gamma^+ \odot v^+ - q^{-1}{\gamma^+}^T v^+ \mathbf{1}\| > 0.5 q^{-1} {v^+}^T \gamma^+$ **then**
    continue // Keep doing Newton iterations
  end
  else
    $\tilde{\beta}^+ = \text{prox}_{\alpha R}(\beta^+)$; $\tilde{\gamma}^+ = \text{prox}_{\alpha R + \delta_{\mathbb{R}_+}}(\gamma^+)$; $\mu = \frac{1}{10}\frac{{v^+}^T \gamma^+}{q}$ // Near central path
  end
  progress = ($\|\beta^+ - \beta\| \geq$ tol or $\|\gamma^+ - \gamma\| \geq$ tol or $\|\tilde{\beta}^+ - \tilde{\beta}\| \geq$ tol or $\|\tilde{\gamma}^+ - \tilde{\gamma}\| \geq$ tol)
  iter += 1
end
return $\tilde{\beta}^+$, $\tilde{\gamma}^+$

Algorithm 4: MSR3-fast (Optimized Proximal Gradient Descent for the Value function)

the last 10 covariates are not. The data is generated as

$$
\begin{aligned}
y_i &= X_i \beta + Z_i u_i + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, 0.3^2 I) \\
X_i &\sim \mathcal{N}(0, I)^p, \quad Z_i = X_i \\
u_i &\sim \mathcal{N}(0, \text{Diag}(\gamma)),
\end{aligned}
$$

with 9 groups of sizes $[10, 15, 4, 8, 3, 5, 18, 9, 6]$. The data generation is repeated 100 times in order to estimate the uncertainty bounds. The smallest non-zero components in the generated signals are just above the level of observation noise.

Parameter Selection. The regularization parameter λ multiplying R and the coupling parameter η restricting the difference between (β, γ) and $(\tilde{\beta}, \tilde{\gamma})$ are chosen to maximize a classic BIC criterion from Jones (2011). We set a log-uniform grid of 20 candidate values for the parameter $\eta \in [10^{-3}, 10^2]$. For each value of η, the BIC is optimized using a golden search in $\lambda \in [0, 10^5]$. The final values of η and λ are chosen to maximize the BIC criterion.

Figure 2.4.2 shows the dependence of accuracy on the values of η for the first data set generated in our test set. There are three distinct regions, corresponding to loose, moderate, and tight

Regularizer	Model Metric	PGD	MSR3	MSR3-fast
L0	Accuracy	0.89	**0.92**	**0.92**
	Time	47.47	109.86	**0.36**
L1	Accuracy	0.73	**0.89**	0.88
	Time	43.02	13.74	**0.35**
ALASSO	Accuracy	0.88	**0.91**	**0.91**
	Time	38.68	81.52	**0.45**
SCAD	Accuracy	0.71	**0.92**	**0.92**
	Time	87.24	104.20	**0.45**

Table 2.4.1: Comparison of performance of algorithms measured as accuracy of selecting the correct covariates and run-time. The L0 strategy stands out over other standard regularizers. MSR3 improves performance significantly for all regularizers, while MSR3-fast improves convergence speed while preserving the accuracy of MSR3. More detailed results are in the Table A.6.1 of Appendix A.6.

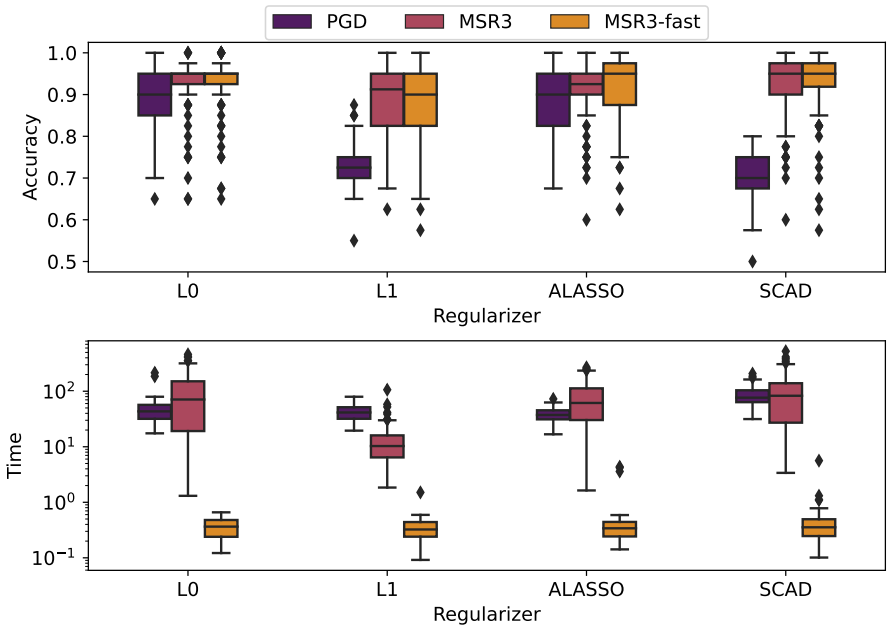


Figure 2.4.1: Feature selection accuracy and execution time in seconds for PGD (Algorithm 1), MSR3 (Algorithm 2), and MSR3-fast (Algorithm 4) with various regularizers. MSR3-Fast has the same accuracy as MSR3 and significantly decreases computation time.

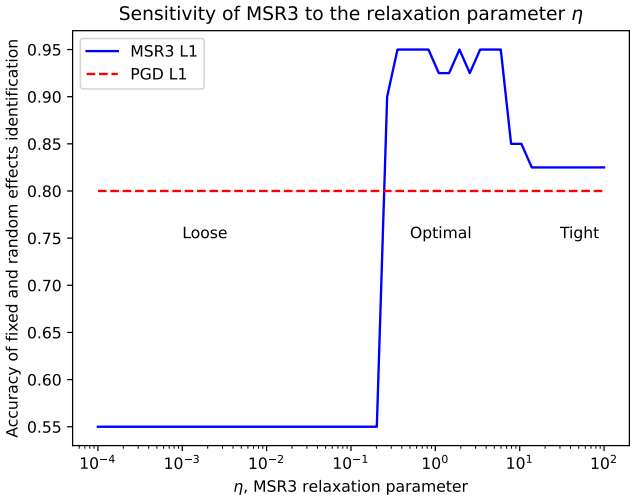


Figure 2.4.2: Dependence of model performance on the relaxation η for a sample problem.

levels of coupling. When η is small the coupling term does not have sufficient strength and the training does not progress far from the initial point (a fully dense vector $\mathbf{1}$ in this case). When the coupling is tight, the level-sets and minimizers are closer to those of the the original problem. For the values in between, the coupling significantly improves the model's accuracy. These results are consistent with experiments in the sparse linear regression setting Zheng et al. (2019).

Results. The experimental results are presented in the Table 2.4.1 and Figure 2.4.1. MSR3 improves the selection accuracy of most regularization techniques described in Table 2.3.1, showing a near-perfect performance, while converging two orders of magnitude faster in wall-clock time.

Comparison to `glmmLasso` and `lmmLasso`. We used (Buscemi and Plaia, 2019, Table 3) as a reference for feature selection libraries. Of the 17 entries mentioned, the four libraries that successfully ran on our synthetic data described above were packages `glmmLasso`[5] (Groll and Tutz (2014)), `lmmLasso`[6](Schelldorfer et al. (2011)), `fence`[7] (Jiang et al. (2008)) and `PCO` (Lin et al. (2013b)) libraries. `fence` caused a memory overflow on the experimental system during the performance evaluation on the datasets described above. We could not evaluate `PCO` because it did not support datasets where the total number of random effects mq exceeded the total number of observations n. We compare performance of MSR3 (available through the open source `pysr3` library) to the performance of the `R` packages `glmmLasso`[8] (Groll and Tutz

[5] https://rdrr.io/cran/glmmLasso/man/glmmLasso.html
[6] https://rdrr.io/cran/lmmlasso/
[7] https://rdrr.io/cran/fence/
[8] https://rdrr.io/cran/glmmLasso/man/glmmLasso.html

(2014)) and `lmmLasso`[9] (Schelldorfer et al. (2011)) which are the functionally closest libraries available online. As of this writing, `glmmlasso` does not allow the user to specify Γ as a diagonal matrix. Since the diagonal specification simplifies the problem, this puts `glmmlasso` package at a disadvantage in our numerical comparison. We evaluate all algorithms' performance on the same set of problems as described above. We tuned the hyperparameters of `glmmLasso` and `lmmLasso` by minimizing the BIC scores provided by the libraries over $\lambda \in [0, 10^5]$. The results are presented in Table 2.4.2. Overall, MSR3 executes, on average, 5 times faster in wall-clock time than `glmmLasso` and 60 times faster than `lmmLasso` and shows much higher accuracy in selecting correct fixed and random effects simultaneously. The accuracy of `glmmLasso` is lower relative to the other libraries' scores likely due to its BIC selection criterion choosing dense models. The package `lmmLasso` supports the diagonal specification of Γ, thus allowing a direct comparison with the scores from `pysr3`. `lmmLasso` yields a competitive accuracy of selecting random effects but `lmmLasso` provides dense solutions for fixed effects β for chosen values of λ.

Algorithm	Units (perc. / 100 runs)	MSR3-Fast (ℓ_1)	glmmLasso	lmmLasso
Accuracy	% (5%-95%)	**88 (72-98)**	48 (42-55)	66 (55-73)
FE Accuracy	% (5%-95%)	**86 (64-100)**	52 (40-66)	47 (45-55)
RE Accuracy	% (5%-95%)	**91 (74-100)**	45 (45-45)	84 (55-100)
F1	% (5%-95%)	**89 (73-97)**	63 (60-66)	65 (0-77)
FE F1	% (5%-95%)	**88 (69-100)**	64 (57-70)	57 (0-64)
RE F1	% (5%-95%)	**90 (73-100)**	62 (62-62)	78 (0-100)
Time	sec. (5%-95%)	**0.19 (0.14-0.24)**	1.37 (0.78-1.89)	11.51 (5.35-23.66)
Iterations	num. (5%-95%)	**34 (28-45)**	50 (33-77)	-

Table 2.4.2: Comparison of performance of MSR3-Fast for ℓ_1 regularizer vs `glmmLasso`. MSR3-Fast executes 5 times faster in wall time and has higher accuracy of selecting correct covariates.

2.4.2 Scalability and Sensitivity Analysis

Scalability

We tested the scalability of the new approach (MSR3-fast) compared to proximal gradient descent and proximal gradient descent with line search. To do this, chose an initially small problem and we scaled the number of features in the data from 100 to 1000, while scaling the number of observations proportionally, and tested the time to completion of these three methods, averaged over 100 replicates. Namely, each problem had 8 groups of $10A$ observations each, β and γ had $20A$ features equally split between 0 and 1. To get the problems of different sizes we assigned A to be 1, 2, 5, 10, 20, 50, and 100, and for each choice of A we generated 100 random problems. Since MSR3-fast has a relaxation parameter η, we evaluated MSR3-fast across different η values to also test the effect of η on timing. For each experiment, we also

[9] https://rdrr.io/cran/lmmlasso/

computed the accuracy of the feature selection, to make sure that there was no degradation in performance. The results are presented in Tables 2.4.3 and 2.4.4. In terms of timing, we indeed see a superlinear increase in computational complexity with respect to the number of features. Nonetheless, MSR3-fast is competitive with the alternatives across the experiments, and the results are far more accurate. Larger problems could likely significantly benefit from iterated solvers within the interior point framework.

Algorithm	MSR3-Fast							PGD	PGD-LineSearch
η	0.01	0.05	0.10	0.50	1.00	5.00	10.00		
# Features									
100	0m 7s	0m 7s	0m 7s	0m 6s	0m 7s	0m 8s	0m 10s	2m 44s	4m 44s
200	0m 36s	0m 39s	0m 36s	0m 39s	0m 39s	0m 49s	1m 8s	7m 43s	11m 28s
400	5m 2s	4m 51s	4m 34s	4m 26s	5m 16s	7m 33s	10m 38s	47m 46s	12m 36s
1000	59m 10s	57m 12s	60m 30s	69m 57s	68m 55s	111m 31s	139m 47s	469m 16s	55m 8s

Table 2.4.3: Execution time for feature selection problems of varying sizes. Each cell shows total time, including grid-search with respect to the sparsity parameter λ. Each cell shows averaged value over 100 randomly-generated problems.

Algorithm	MSR3-Fast							PGD	PGD-LineSearch
η	0.01	0.05	0.10	0.50	1.00	5.00	10.00		
# Features									
100	0.94	0.94	0.95	0.94	0.91	0.86	0.84	0.77	0.77
200	0.99	0.99	0.99	0.98	0.98	0.97	0.95	0.78	0.82
400	0.99	0.99	0.99	0.99	0.99	1.00	1.00	0.80	0.84
1000	0.99	0.98	0.98	0.98	0.98	1.00	1.00	0.83	0.87

Table 2.4.4: Accuracy of feature selection problems of varying sizes. Each cell shows averaged value over 100 randomly-generated problems.

Closeness to the Central Path for IP

The τ parameter of MSR3-fast controls how close the interior point method gets to the central path before taking a prox-gradient step. This is a heuristic parameter in the algorithm, and to understand its impact we tested the sensitivity of the execution time and accuracy for a problem with 200 features for four selections of relaxation parameter η. The problems were identical to those from the second row of Table 2.4.3. The results are reported in Tables 2.4.5 and 2.4.6. Neither time nor accuracy were affected by τ across the levels of η.

Algorithm	MSR3-Fast			
η	0.01	0.10	1.00	10.00
τ				
0.1	0m 41s	0m 40s	0m 41s	1m 12s
0.3	0m 35s	0m 36s	0m 38s	1m 1s
0.5	0m 34s	0m 35s	0m 36s	0m 57s
0.7	0m 33s	0m 33s	0m 35s	0m 59s
0.9	0m 33s	0m 33s	0m 35s	0m 52s

Table 2.4.5: Execution time of MSR3-fast for different values of τ - a parameter that controls how close the IP needs to be to the central path before doing a projection step. Each cell shows total time, including grid-search with respect to the sparsity parameter λ. Each cell shows averaged value over 100 randomly-generated problems.

Algorithm	MSR3-Fast			
η	0.01	0.10	1.00	10.00
τ				
0.1	0.99	0.99	0.99	0.95
0.3	0.99	0.99	0.98	0.95
0.5	0.99	0.99	0.98	0.95
0.7	0.99	0.99	0.98	0.95
0.9	0.99	0.99	0.98	0.95

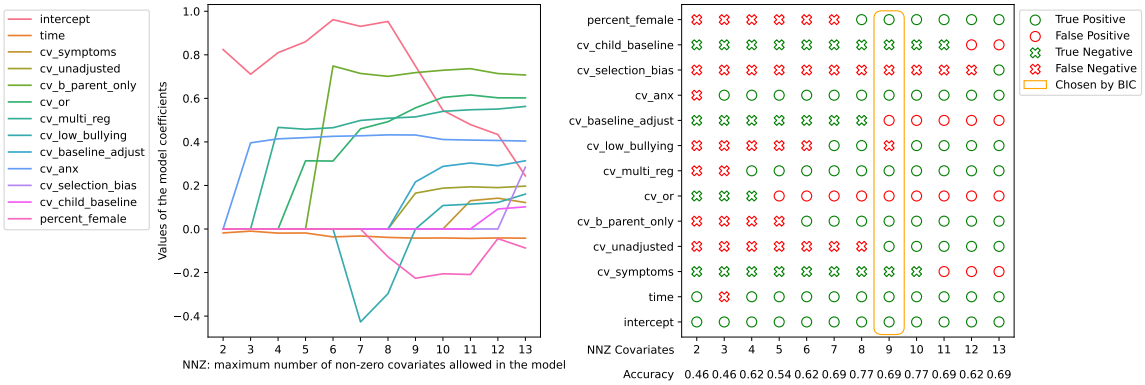
Table 2.4.6: Accuracy of MSR3-fast for different values of τ - a parameter that controls how close the IP needs to be to the central path before doing a projection step. Each cell shows averaged value over 100 randomly-generated problems.

2.4.3 Application to Real Data: Anxiety and Depression as a Result of Bullying Victimization

In this section we validate the MSR3-empowered ℓ_0-regularized mixed-effect model ($R(x) = \delta_{\|x\|_0 \leq k}$ from Table 2.3.1) by using it to identify the most important covariates in real data on relative risk of anxiety and depressive disorders depending on the exposure to bullying in young age[10]. This research has been a part of Global Burden of Diseases study for the last several years. The end goal is to estimate the burden through disability adjusted life years (DALYs) (Murray and Acharya, 1997) of major depressive disorder (MDD) and anxiety disorders that are caused by bullying. For this risk factor, the exposure is primarily concentrated in childhood and adolescents, but the risk for MDD and anxiety disorders is anticipated to continue

[10]Institute for Health Metrics and Evaluation (IHME). Bullying Victimization Relative Risk Bundle GBD 2020. Seattle, United States of America (USA), 2021.

Figure 2.4.3: Validation of Random Feature Selection for Bullying Data from GBD 2020. Left panel shows coefficient paths across numbers of nonzero covariates allowed in the model using the ℓ_0 regularizer. Right panel evaluates each choice against expert knowledge. The algorithm picks seven historically significant covariates and two historically insignificant, for the model selected using the BIC criteria. See the Appendix A.7.1 for covariates description and assessment of significance.

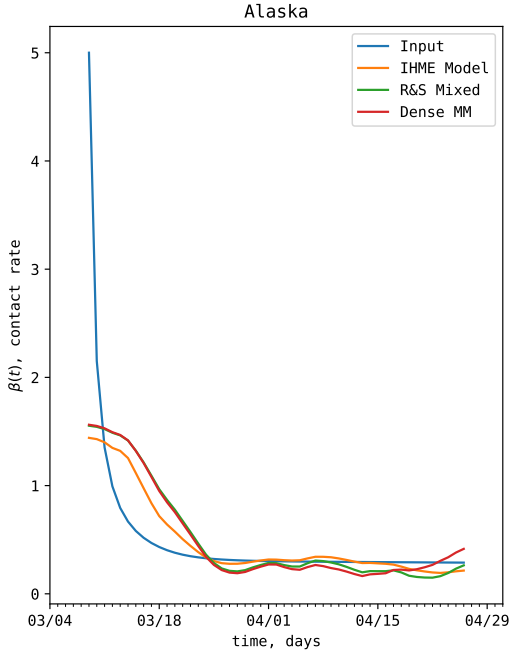


well into adulthood. This elevated risk is, however, expected to decrease with time as other risk factors come into play in adulthood (unemployment, relationship issues, etc.). To accommodate this, the research team uses the models which estimate the relative risk (RR) of MDD and anxiety disorders among persons exposed to bullying depending on how many years it has been since the first exposure. Studies informing the model were sourced from a systematic review and consist of longitudinal cohort studies. They measure exposure to bullying at baseline, and then follow up years later and assess them for MDD or anxiety disorders. The detailed description of the covariates can be found in Appendix A.7.1.

The feature selection process is illustrated on Figure 2.4.3. Here, the BIC criterion from Jones (2011) was used to select k, which suggests $k = 4$ or 5. For the $k = 4$ case, the selected covariates (`intercept`, `time`, `cv_threshold_bullying`, `cv_b_parent_only`) are known as important and were used in the analysis in previous years of GBD. For the $k = 5$ case, the algorithm also selects `cv_child_baseline` and `cv_or`, which were not used before. The `cv_child_baseline` covariate describes whether the midpoint in the sample is above or below 13. The `cv_or` variable describes whether the estimate is a relative risk or odds ratio. The selection of these variables suggests a closer look at the data reporting mechanisms across studies. For example, there is an active literature on converting estimates between relative risks and odds ratios Grant (2014); Wang (2013).

2.4.4 Application to Real Data: COVID-19 Transmission Factor

In this section we apply our method to a COVID-19 Contact Rate Forecasting problem. The global pandemic created an unprecedented need for robust and accurate disease transmission



RMSE:
  IHME   : 3.85e+00
  Dense MM : 3.98e+00   +3%
  R&S Mixed : 4.00e+00   +4%

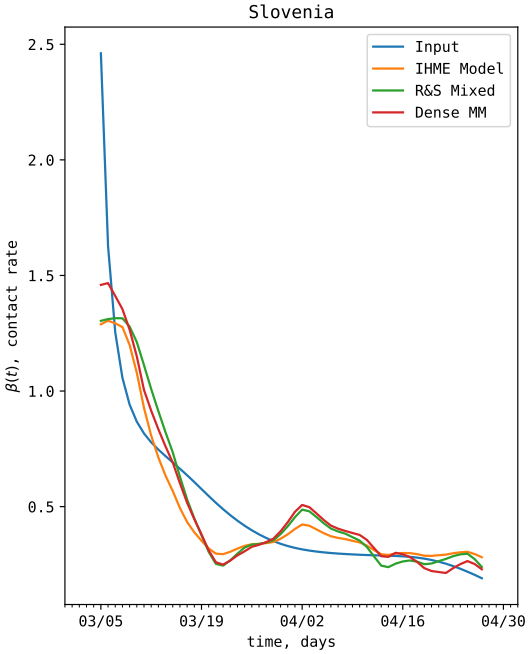
Full MM Coefficients:

name	local	mean	RE	Var
intercept	1.23e+01	1.34e+01	-1.08e+00	5.66e-01
temperature	-6.75e+02	-6.75e+02	0.00e+00	2.19e-20
mobility_lift	6.44e+01	6.13e+01	3.09e+00	1.83e+03
proportion_over_1k	-7.14e+00	-7.14e+00	0.00e+00	9.61e-21
testing_reference	6.11e+00	-2.21e+00	8.32e+00	5.09e+02

R&S Mixed Coefficients:

name	local	mean	RE	Var
intercept	1.22e+01	1.43e+01	-2.05e+00	7.95e+01
temperature	-6.75e+02	-6.75e+02	0.00e+00	0.00e+00
mobility_lift	5.78e+01	6.00e+01	-2.20e+00	1.83e+03
proportion_over_1k	0.00e+00	0.00e+00	0.00e+00	0.00e+00
testing_reference	0.00e+00	0.00e+00	0.00e+00	0.00e+00

Legend:
  Both Fixed and Random
  Fixed Only
  Excluded



RMSE:
  IHME   : 1.46e+00
  Dense MM : 1.40e+00   -4%
  R&S Mixed : 1.56e+00   +7%

Full MM Coefficients:

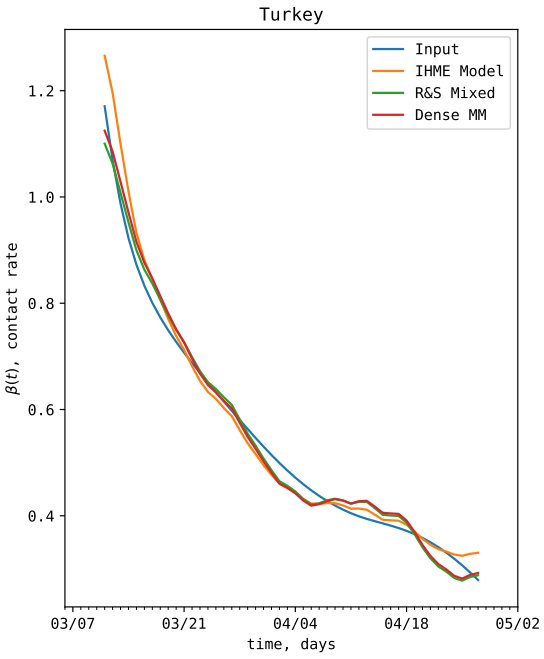
name	local	mean	RE	Var
intercept	1.31e+01	1.34e+01	-3.62e-01	5.66e-01
temperature	-6.75e+02	-6.75e+02	0.00e+00	2.19e-20
mobility_lift	3.28e+01	6.13e+01	-2.85e+01	1.83e+03
proportion_over_1k	-7.14e+00	-7.14e+00	0.00e+00	9.61e-21
testing_reference	-3.16e+01	-2.21e+00	-2.93e+01	5.09e+02

R&S Mixed Coefficients:

name	local	mean	RE	Var
intercept	1.27e+01	1.43e+01	-1.55e+00	7.95e+01
temperature	-6.75e+02	-6.75e+02	0.00e+00	0.00e+00
mobility_lift	4.27e+01	6.00e+01	-1.73e+01	1.83e+03
proportion_over_1k	0.00e+00	0.00e+00	0.00e+00	0.00e+00
testing_reference	0.00e+00	0.00e+00	0.00e+00	0.00e+00

Legend:
  Both Fixed and Random
  Fixed Only
  Excluded

Figure 2.4.4: Comparison of fits of two different models (fully dense linear mixed model(MM) and MSR3-ℓ_0 (R&S) to the original IHME Projections for Alaska and Slovenia. The quality (RMSE) achieved by a sparse fit is within 10% from a quality of both dense models which is evidences that the model picked up informative features.

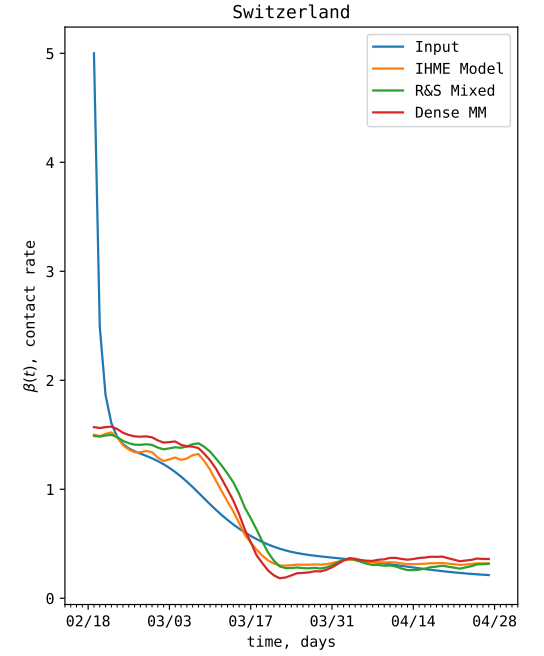



```
RMSE:
  IHME      : 2.67e-01
  Dense MM  : 1.85e-01   -31%
  R&S Mixed : 1.71e-01   -36%

Full MM Coefficients:
name                      local       mean        RE       Var
  intercept            1.28e+01  1.34e+01 -6.56e-01  5.66e-01
  temperature         -6.75e+02 -6.75e+02  0.00e+00  2.19e-20
  mobility_lift        3.05e+01  6.13e+01 -3.08e+01  1.83e+03
  proportion_over_1k  -7.14e+00 -7.14e+00  0.00e+00  9.61e-21
  testing_reference    1.49e+00 -2.21e+00  3.70e+00  5.09e+02

R&S Mixed Coefficients:
name                      local       mean        RE       Var
  intercept            1.26e+01  1.43e+01 -1.69e+00  7.95e+01
  temperature         -6.75e+02 -6.75e+02  0.00e+00  0.00e+00
  mobility_lift        2.88e+01  6.00e+01 -3.12e+01  1.83e+03
  proportion_over_1k   0.00e+00  0.00e+00  0.00e+00  0.00e+00
  testing_reference    0.00e+00  0.00e+00  0.00e+00  0.00e+00

Legend:
  Both Fixed and Random
  Fixed Only
  Excluded
```




```
RMSE:
  IHME      : 3.79e+00
  Dense MM  : 3.88e+00   +2%
  R&S Mixed : 3.98e+00   +5%

Full MM Coefficients:
name                      local       mean        RE       Var
  intercept            1.29e+01  1.34e+01 -5.34e-01  5.66e-01
  temperature         -6.75e+02 -6.75e+02  0.00e+00  2.19e-20
  mobility_lift        4.82e+01  6.13e+01 -1.31e+01  1.83e+03
  proportion_over_1k  -7.14e+00 -7.14e+00  0.00e+00  9.61e-21
  testing_reference   -2.31e+01 -2.21e+00 -2.09e+01  5.09e+02

R&S Mixed Coefficients:
name                      local       mean        RE       Var
  intercept            1.27e+01  1.43e+01 -1.61e+00  7.95e+01
  temperature         -6.75e+02 -6.75e+02  0.00e+00  0.00e+00
  mobility_lift        6.65e+01  6.00e+01  6.49e+00  1.83e+03
  proportion_over_1k   0.00e+00  0.00e+00  0.00e+00  0.00e+00
  testing_reference    0.00e+00  0.00e+00  0.00e+00  0.00e+00

Legend:
  Both Fixed and Random
  Fixed Only
  Excluded
```

Figure 2.4.5: Comparison of fits of two different models (fully dense linear mixed model (MM) and MSR3-ℓ_0) (R&S) to the original IHME Projections for Turkey and Switzerland. The quality (RMSE) achieved by a sparse fit is within 10% from a quality of both dense models which is evidences that the model picked up informative features.

forecasting. Since the beginning of the pandemic Institute for Health Metrics and Evaluation has been providing guidance to local authorities across the world with their pioneering COVID-19 Projections tool IHME (2020). The key methodology which was essential for success in their forecasting of the disease's dynamics was transforming the death data into the contact rate time series data, and then relating this contact rate to available covariates such as temperature, social mobility, population, and others IHME COVID-19 Forecasting Team (2020). All these covariates were collected in real time using limited human resources, and identifying the most important covariates was crucial for distributing these resources effectively. In the example below we show how MSR3 can be used for making covariate selection on IHME data.

The dataset consists of $m = 60$ groups (countries or states), the detailed description of groups sizes (n_i) and time spans is provided in the Appendix 2.4.4. The target variable Y_i was the contact rate – the coefficient $\beta(t)$ from an SEIR model (not to be confused with β – vector of fixed effects). The covariates – columns of X_i and Z_i – were: `intercept` –a column of ones, `temperature` – average air temperature in degrees Fahrenheit, `mobility_lift` – social mobility, `proportion_over_1k` – population size threshold, and `testing_reference` – testing. The observation error's variance σ_i was set to be 0.1.

Feature selection results for four particular locations (Alaska, Slovenia, Turkey, and Switherland) are presented on Figures 2.4.4 and 2.4.5, with coefficients for the rest of the locations attached in supplementary materials. MSR3 was charged with a task to produce a fit using only three covariates, one of which had to be mean-only (no random effects). On the left we see the original data (`Data`, in blue), and predictions of three models: IHME Projections (`IHME`, in orange), a linear mixed model fit with no selection (`Dense MM`, in red), and a sparse fit of MSR3 (MSR3 in green). The MSR3 model has chosen to use `intercept`, `mobility_lift`, and `temperature` covariates, with the later only as a fixed covariate; `testing_reference` and `proportion_over_1k` were left out. On the plots to the left we see that the exclusion of two covariates did not significantly affect the quality of predictions. The residual errors (RMSE) to the right support this statement: the difference in RSME is within 10% of RSME of a "dense" model which uses all the covariates, as well as from IHME's original model which fit all the locations via sequence of independent regressions. This choice also seems reasonable given the timespan: the proliferation of testing during the spring was not yet significant, so it did not inform predictions in a major way. The exclusion of `proportion_over_1k` could have been due to the scale of grouping: locations were grouped on the level of states and countries, not on the level of individual counties where the influence of population-based covariates could have been more significant.

2.5 Theoretical Analysis

In this section we provide a theoretical justification for the algorithmic approach to the solution of the marginalized maximum likelihood estimation problems presented in Aravkin et al. (2022a)

as well as in Section 2.3.2. The approach is motivated by the sparse relaxed regularization regression (SR3) strategy developed in Zheng et al. (2019). While the analysis of Zheng et al. (2019) relies on the least squares data-fitting term, here we develop the algorithmic design and analysis required for the nonlinear and nonconvex LME extension.

This section proceeds as follows. We study the relaxation strategy defined in (2.3.11) where the the relaxation depends on a decoupling parameter η and log-barrier smoothing parameter μ. In particular, we introduce the optimal value function $u_{\eta,\mu}$ (2.3.15) used to exploit the decoupling of the likelihood function from the sparsity regularizer. Here $u_{\eta,\mu}$ is obtained by partially minimizing the decoupled variables in the likelihood while keeping those in the regularizer fixed. Section 2.5.1 introduces the MSR3 algorithm as the PGD algorithm applied to the sum of $u_{\eta,\mu}$ and the sparsity regularizer. A brief discussion the basic assumptions typically required for establishing the viability of the PGD algorithm for this formulation is given. Section 2.5.2 is the theoretical core of the work. In this section we show that the optimal value function function $u_{\eta,\mu}$ satisfies the properties necessary for the application of the PGD algorithm. In particular, we establish the Lipschitz continuity of $\nabla u_{\eta,\mu}$ (Lemma 16). The convergence results for MSR3 are presented in Section 2.5.3 for fixed values of η and μ. In Section 2.5.4 we address the key issues surrounding the initialization of the coupling and smoothing parameters η and μ when only approximate values for $u_{\eta,\mu}$ and $\nabla u_{\eta,\mu}$ are known. Here we appeal to both variable metric ideas as well as properties of the interior point algorithm.

2.5.1 Proximal Gradient Descent for the Regularized Value Function

We follow the analysis of the PGD algorithm given in (Beck, 2017, Chapter 10) as it applies to the objective

$$\Phi_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) := u_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) + \widetilde{R}(\tilde{\beta},\tilde{\gamma}), \quad \text{where } \widetilde{R}(\tilde{\beta},\tilde{\gamma}) := R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma}). \tag{2.5.1}$$

Since $u_{\eta,\mu}$ is nonconvex, one typically applies a line search method to select stepsize. However, this is often not required in practice. For this reason we state the algorithm with and without a line search.

In Algorithm 3, the parameter L is assumed to be a global Lipschitz constant for $\nabla u_{\eta,\mu}$. In Section 2.5.3, we show that the existence of L is not needed. In both algorithms we introduce the requirement that $\gamma^k \leq \gamma_{\max}$. While it is possible to include an explicit constraint of this form in the optimal variable selection problem (2.3.1), we do not do so since we assume that $\gamma_{\max}$ is chosen so large that, from a practical perspective, the violation of this constraint indicates that the model is poorly posed and the algorithm needs to be terminated. We base our analysis of the convergence properties of Algorithms 1 and 2 on (Beck, 2017, Theorem 10.15) which makes use of the following three basic assumptions:

Basic Assumptions for the PGD Algorithm

Initialize: $\theta \in (0,1),\ \tau \in (0,1),\ \eta > 0,\ \mu > 0,\ \epsilon_{\text{Tol}} \geq 0\ k = 0,\ t_0 > 0,$
$\bar{w}^0 = (\tilde{\beta}^0, \tilde{\gamma}^0) \in \mathbb{R}^p \times \mathbb{R}^q_+$ with $\inf \Phi_{\eta,\mu} < \Phi_{\eta,\mu}(\bar{w}^0)$, $\gamma_{\max} > \tilde{\gamma}^0$,
$w^0 = \text{prox}_{t_0 \widetilde{R}}(\bar{w}^0 - t_0 \nabla u_{\eta,\mu}(\bar{w}^0))$.

while $\underline{\left\|w^k - \bar{w}^k\right\| > \epsilon_{\text{Tol}} \text{ and } \gamma^k \leq \gamma_{\max}}$ **do**

(i) $t_{k+1} = \max \left\{ t \;\middle|\; \begin{array}{l} s \in \mathbb{W},\ t = t_0 \theta^s,\ w = \text{prox}_{t\widetilde{R}}(w^k - t \nabla u_{\eta,\mu}(w^k)) \\ \phi(w) \leq \phi(w^k) - \tau t \left\|w^k - w\right\|^2 \end{array} \right\}.$

(ii) $\bar{w}^{k+1} = w^k$

$w^{k+1} = \text{prox}_{t_{k+1}\widetilde{R}}(w^k - t_{k+1} \nabla u_{\eta,\mu}(w^k))$

$k = k + 1$

end

Algorithm 5: Proximal Gradient Descent fo $\Phi_{\eta,\mu}$ with Backtracking

(A) $\widetilde{R} : \mathbb{R}^p \times \mathbb{R}^q \to \overline{\mathbb{R}}$ is a closed proper convex function.

(B) $u_{\eta,\mu} : \mathbb{R}^p \times \mathbb{R}^q \to \overline{\mathbb{R}}$ is closed and proper, $\text{dom}\, u_{\eta,\mu}$ is convex, $\text{dom}\, \widetilde{R} \subset \text{int}\,(\text{dom}\, u_{\eta,\mu})$, and $u_{\eta,\mu}$ is $L_{\eta,\mu}$-smooth over $\text{int}\,(\text{dom}\, u_{\eta,\mu})$.

(C) Problem (2.3.14) has an optimal solution with optimal value Φ_{OPT}.

We assume that (A) holds. This is not an overly restrictive assumption since it is satisfied by most of the standard variable selection regularizers. We show that (C) holds when R satisfies an additional coercivity hypothesis (Theorem 7). On the other hand, establishing that (B) holds in a concrete setting such as ours can be quite difficult. In particular, just as with $\mathcal{L}$, $u_{\eta,\mu}$ may fail to be globally Lipschitz. Validating Assumption (B) as well as developing a technique for circumventing the need for a global Lipschitz constant for $\nabla u_{\eta,\mu}$ consumes the majority of the theoretical development.

2.5.2 The Smoothness of the Value Function

We investigate the relationship between the problems (2.3.1) and (2.3.12), the existence of solutions to (2.3.12), and the properties of the function $u_{\eta,\mu}$ and its derivative.

Underlying convexity

Lemma 5 ($\mathcal{L} + \phi_\mu$ is Weakly Convex)**.** *Let $\mathcal{L}$ be as given in* (2.2.8)*. Then*

$$\nabla^2 \mathcal{L}\,(\beta, \gamma) = \sum_{i=1}^{m} S_i^T \begin{bmatrix} X_i^T \\ -Z_i^T \end{bmatrix} \Omega_i(\gamma)^{-1} \begin{bmatrix} X_i & -Z_i \end{bmatrix} S_i - \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2}(Z_i^T \Omega_i(\gamma)^{-1} Z_i)^{\circ 2} \end{bmatrix}, \tag{2.5.2}$$

for all $(\beta, \gamma) \in \mathbb{R}^p \times \mathbb{R}^q_+$*, where*

$$S_i := \begin{bmatrix} I_q & 0 \\ 0 & \operatorname{Diag}\left(Z_i^T \Omega_i^{-1}(X_i\beta - Y_i)\right) \end{bmatrix}$$

and, for any $A \in \mathbb{R}^{t\times t}$*,* $A^{\circ 2} := A \circ A$*. In particular, this implies that the matrix*

$$\begin{bmatrix} \nabla_{\beta\beta}\mathcal{L}(\beta,\gamma) & \nabla_{\gamma\beta}\mathcal{L}(\beta,\gamma) \\ \nabla_{\beta\gamma}\mathcal{L}(\beta,\gamma) & \nabla_{\gamma\gamma}\mathcal{L}(\beta,\gamma) + \bar{\eta} I \end{bmatrix} \tag{2.5.3}$$

is positive semidefinite for $\bar{\eta} = \nu m$*, where*

$$\nu := max\left\{(1/2)\mu_{min}(\Lambda_i)^{-2}\sigma^4_{max}(Z_i) \mid i = 1, \dots, m\right\},$$

$\mu_{min}(\Lambda_i)$ *is the smallest eigenvalue of* Λ_i*, and* $\sigma_{max}(Z_i)$ *is the largest singular value of* Z_i*,* $i = 1, \dots, m$*. Consequently, for any* $(\tilde{\beta}, \tilde{\gamma}) \in \operatorname{dom} \widetilde{R}$ *and* $\mu \geq 0$*, the mapping* $(\beta, \gamma) \mapsto \mathcal{L}_{\eta,\mu}((\beta, \gamma), (\tilde{\beta}, \tilde{\gamma}))$ *is convex for all* $\eta \geq \bar{\eta} := \nu m$*. In particular, this implies that* $\mathcal{L} + \phi_\mu$ *is weakly convex for any* $\mu \geq 0$*, and the mapping* $(\beta, \gamma) \mapsto \mathcal{L}_{\eta,\mu}((\beta, \gamma), (\tilde{\beta}, \tilde{\gamma}))$ *is strongly convex for* $\eta > \bar{\eta}$ *with modulus of strong convexity* $(\eta - \bar{\eta})$ *regardless of the choice of* $(\tilde{\beta}, \tilde{\gamma}) \in \mathbb{R}^p \times \mathbb{R}^q$*.*

Proof. The formula for $\nabla^2 \mathcal{L}$ is given in Appendix A.2 (see (2.5.2)). By (Aravkin et al., 2021, Theorem 3.1), $\mu_{\max}\left(\left(Z_i^T \Omega_i(\gamma)^{-1} Z\right) \leq \lambda^{-1}_{\min}\sigma^2_{\max}(Z_i)$, and since $\mu_{\max}(H^{\circ 2}) \leq \mu^2_{\max}(H)$ for all $H \in \mathbb{S}^q_+$ Horn and Johnson (1985), we have

$$\mu_{\max}\left(\frac{1}{2}(Z_i^T \Omega_i(\gamma)^{-1} Z_i)^{\circ 2}\right) \leq (1/2)\lambda^{-2}_{\min}\sigma^4_{\max}(Z_i) =: \nu_i \qquad i = 1, \dots, m.$$

This establishes that the matrix in (2.5.3) is positive semidefinite. Since ϕ_μ is convex, the mapping

$$(\beta, \gamma) \mapsto \mathcal{L}_{\eta,\mu}((\beta, \gamma), (\tilde{\beta}, \tilde{\gamma}))$$

is strongly convex for any choice of $\bar{\eta} > \nu m$, where

$$\nu := \max_{i=1,\dots,m} \nu_i. \tag{2.5.4}$$

□

For the remainder of the paper, we assume that

$$\eta > \nu m =: \bar{\eta} \tag{2.5.5}$$

so that the mapping $(\beta, \gamma) \mapsto \mathcal{L}_{\eta,\mu}((\beta, \gamma), (\tilde{\beta}, \tilde{\gamma}))$ is strongly convex with positive definite Hessian regardless of the choice of $\tilde{\gamma} \in \mathbb{R}^q$. With this in mind, the function $u_{\eta,\mu}$ defined by (2.3.15) resembles a Moreau envelope. However, this is misleading since, in particular, we are not even assured of the existence of solutions to the optimization problem defining $u_{\eta,\mu}$.

Existence and consistency

To establish the existence of solutions to the relaxed optimization problems (2.3.12) and the problems defining the parametrized family $u_{\eta,\mu}$ in (2.3.15), we assume that R is 1-coercive.

Lemma 6. *Given $\mu > 0$ let ϕ_μ be as defined above, and assume that $R : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R} \cup \{+\infty\}$ is 1-coercive, i.e., $\liminf_{\|(\tilde{\beta},\tilde{\gamma})\|\to\infty} \|(\tilde{\beta},\tilde{\gamma})\|^{-1} R(\tilde{\beta},\tilde{\gamma}) > 0$. Then $\phi_\mu + R$ is level compact.*

Proof. If $\mu = 0$, then the result is trivially true, so we assume that $\mu > 0$. Let $\{(\beta^k, \gamma^k)\} \subset \mathbb{R}^p \times \mathbb{R}^q_+$ be such that $\left\|(\beta^k, \gamma^k)\right\| \uparrow \infty$. We need to show that $\phi_\mu(\gamma^k) + R(\beta^k, \gamma^k) \to \infty$. If $\{\gamma^k\}$ is bounded, then $\phi_\mu(\gamma^k) + R(\beta^k, \gamma^k) \to \infty$ since in this case $\phi_\mu(\gamma^k)$ is bounded below. So assume that $\{\gamma^k\}$ is unbounded which implies that $\phi_\mu(\gamma^k) \to -\infty$. Since R is 1-coercive, we know that there is an $\hat{\alpha} > 0$ such that, for k sufficiently large, $R(\beta^k, \gamma^k) \geq \hat{\alpha} \sum_{i=1}^q \gamma_i^k$. But then $\phi_\mu(\gamma^k) + R(\beta^k, \gamma^k) \geq \sum_{i=1}^q (\hat{\alpha}\gamma_i^k - \mu \ln(\gamma_i^k))$ where the right-hand side diverges to $+\infty$ as $k \uparrow \infty$. Hence, $\phi_\mu(\gamma^k) + R(\beta^k, \gamma^k) \to \infty$. □

Theorem 7. *Let $\mathcal{L}$ be as in Theorem 2 and let $\eta > 0$ satisfy* (2.5.5)*. Let $\mu \geq 0$. If $\mu = 0$, assume that $R : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}_+ \cup \{+\infty\}$ is level compact; otherwise, assume R is 1-coercive. Then solutions to* (2.3.12) *always exist.*

Proof. Let v^* be the optimal value in (2.3.12) and let $\{((\beta^k, \gamma^k), (\tilde{\beta}^k, \tilde{\gamma}^k))\} \subset (\mathbb{R}^p \times \mathbb{R}^q_+)^2$ be such that $\mathcal{L}_{\eta,\mu}((\beta^k, \gamma^k), (\tilde{\beta}^k, \tilde{\gamma}^k)) + R(\tilde{\beta}^k, \tilde{\gamma}^k) \downarrow v^*$. By (A.4.3) and (A.4.4), it must be the case that

$$\begin{aligned} &\mathcal{L}_{\eta,\mu}((\beta^k, \gamma^k), (\tilde{\beta}^k, \tilde{\gamma}^k)) + R(\tilde{\beta}^k, \tilde{\gamma}^k) \\ &\quad \geq \frac{n+1}{2} \ln(\tilde{\alpha}) + \phi_\mu(\gamma^k) + \kappa_\eta(\beta^k - \tilde{\beta}^k, \gamma^k - \tilde{\gamma}^k) + R(\tilde{\beta}^k, \tilde{\gamma}^k) \\ &\quad \geq \frac{n+1}{2} \ln(\tilde{\alpha}) + \phi_\mu(\gamma^k) + \frac{\bar{\eta}}{2} \left\|\gamma^k - \tilde{\gamma}^k\right\|^2 + R(\tilde{\beta}^k, \tilde{\gamma}^k). \end{aligned} \tag{2.5.6}$$

If $v^* = -\infty$, then (2.5.6) tells us that

$$\phi_\mu(\gamma^k) + \frac{\bar{\eta}}{2} \left\|\gamma^k - \tilde{\gamma}^k\right\|^2 + R(\tilde{\beta}^k, \tilde{\gamma}^k) \to -\infty. \tag{2.5.7}$$

This in turn implies that $\mu > 0$, $\phi_\mu(\gamma^k) \to -\infty$ and $\left\|\gamma^k\right\| \to \infty$. Since R is 1-coercive and $\left\|\gamma^k\right\| \to \infty$, we can assume with no loss in generality that there is an $\bar{\alpha} > 0$ such that

$R(\tilde{\beta}^k, \tilde{\gamma}^k) \geq \bar{\alpha} \sum_{i=1}^q \tilde{\gamma}_i^k$ for all $k \in \mathbb{N}$. Consequently,

$$\begin{aligned}
\phi_\mu(\gamma^k) + \frac{\bar{\eta}}{2} \left\|\gamma^k - \tilde{\gamma}^k\right\|^2 + R(\tilde{\beta}^k, \tilde{\gamma}^k) \\
&\geq \sum_{i=1}^q \left(-\mu \ln(\gamma_i^k/\mu) + \frac{\bar{\eta}}{2}(\gamma_i^k - \tilde{\gamma}_i^k)^2 + \bar{\alpha}\tilde{\gamma}_i^k\right) \\
&= \sum_{i=1}^q \left((-\mu \ln(\gamma_i^k/\mu) + \bar{\alpha}\gamma_i^k) + \frac{\bar{\eta}}{2}(\gamma_i^k - \tilde{\gamma}_i^k)^2 - \bar{\alpha}(\gamma_i^k - \tilde{\gamma}_i^k)\right) \\
&= \sum_{i=1}^q \left((-\mu \ln(\gamma_i^k/\mu) + \bar{\alpha}\gamma_i^k) + \frac{\bar{\eta}}{2}\left[(\gamma_i^k - \tilde{\gamma}_i^k - \frac{\bar{\alpha}}{\bar{\eta}})^2 - (\frac{\bar{\alpha}}{\bar{\eta}})^2\right]\right) \\
&\geq -q\frac{\bar{\alpha}^2}{2\bar{\eta}} + \sum_{i=1}^q (-\mu \ln(\gamma_i^k/\mu) + \bar{\alpha}\gamma_i^k) \\
&= -q\frac{\bar{\alpha}^2}{2\bar{\eta}} + \phi_\mu(\gamma^k) + \bar{\alpha}\left\|\gamma^k\right\|_1 \ \to +\infty,
\end{aligned} \tag{2.5.8}$$

which is a contradiction. Hence $v^* > -\infty$.

Let $\rho > v^* > -\infty$. If $\{\gamma^k\} \subset \mathbb{R}_+^q$ is unbounded, we may assume with no loss in generality that $\left\|\gamma^k\right\| \to +\infty$. If $\mu = 0$, then, by (2.5.6), $\rho > \frac{n+1}{2}\ln(\tilde{\alpha}) + R(\tilde{\beta}^k, \tilde{\gamma}^k) \uparrow +\infty$, a contradiction, and so we can assume that $\mu > 0$ and R is 1-coercive. Using (2.5.6) we may proceed as in (2.5.8) to find that

$$\rho > \frac{n+1}{2}\ln(\tilde{\alpha}) - q\frac{\bar{\alpha}^2}{2\bar{\eta}} + \sum_{i=1}^q (-\mu \ln \gamma_i^k + \bar{\alpha}\gamma_i^k) \ \to +\infty, \tag{2.5.9}$$

again a contradiction, so the sequence $\{\gamma^k\}$ is bounded. Therefore, the first inequality in (2.5.6) tells us that the entire sequence $\{((\beta^k, \gamma^k), (\tilde{\beta}^k, \tilde{\gamma}^k))\}$ is necessarily bounded. Consequently, a limit point of the sequence $\{((\beta^k, \gamma^k), (\tilde{\beta}^k, \tilde{\gamma}^k))\}$ exists and, since R is lsc, any such limit point is a solution to (2.3.12). □

Next we fix $\mu \geq 0$ and show that as $\eta \uparrow \infty$ the solutions to (2.3.12) converge to solutions of

$$\min_{(\beta,\gamma)\in\mathcal{C}} \mathcal{L}(\beta, \gamma) + \phi_\mu(\gamma) + R(\beta, \gamma). \tag{2.5.10}$$

In particular, for $\mu = 0$, they converge to solutions of (2.3.1).

Theorem 8 (Consistency as $\eta \to \infty$)**.** *Let $\mathcal{L}$ and R be as in Theorem 7 and fix $\mu \geq 0$. Let $\{\eta_k\} \subset \mathbb{R}_{++}$ be such that $\eta_k < \eta_{k+1}$ with $\eta_k \uparrow \infty$, and let $((\beta^k, \gamma^k), (\tilde{\beta}^k, \tilde{\gamma}^k))$ be an optimal solution to* (2.3.12) *for $(\eta, \mu) = (\eta_k, \mu)$, $k \in \mathbb{N}$. Then any limit point (equivalently, cluster point) $((\bar{\beta}, \bar{\gamma}), (\hat{\beta}, \hat{\gamma}))$ of $\{((\beta^k, \gamma^k), (\tilde{\beta}^k, \tilde{\gamma}^k))\}$ satisfies $(\bar{\beta}, \bar{\gamma}) = (\hat{\beta}, \hat{\gamma})$ with $(\bar{\beta}, \bar{\gamma})$ being an optimal solution to* (2.5.10)*.*

Proof. With no loss in generality $\eta_k > \bar{\eta}$ for all k. Set

$$\left.\begin{aligned}
a_k(x, w) &:= \mathcal{L}_{\eta_k,\mu}(x, w) + R(w) \\
b_k(x, w) &:= \mathcal{L}(x) + \phi_\mu(\gamma) + R(w) \\
c_k(x, w) &:= \kappa_{\eta_k}(\beta - \tilde{\beta}, \gamma - \tilde{\gamma})
\end{aligned}\right\} \quad \forall k \in \mathbb{N},$$

where $x = (\beta, \gamma)$ and $w = (\tilde{\beta}, \tilde{\gamma})$ with κ_η defined in (2.3.5). Set $x^k = (\beta^k, \gamma^k)$, $\bar{x} = (\bar{\beta}, \bar{\gamma})$, $w^k = (\tilde{\beta}^k, \tilde{\gamma}^k)$ and $\hat{w} = (\hat{\beta}, \hat{\gamma})$. By Lemma 6 and Theorem 2 with $\widehat{R} = \phi_\mu + R$, there is an optimal solution x_μ to (2.5.10) yielding an optimal value of v_μ for which $a_k(x^k, w^k) \leq a_k(x_\mu, x_\mu) = v_\mu$ for all $k \in \mathbb{N}$. Hence, the sequence $\{a_k(x^k, w^k)\}$ is upper bounded by v_μ. Since

$$a_k(x^k, w^k) \leq a_k(x^{k+1}, w^{k+1}) \leq a_{k+1}(x^{k+1}, w^{k+1}),$$

there exists $\tilde{v}$ such that $a_k(x^k, w^k) \uparrow \tilde{v} \leq v_\mu$. Next, observe that

$$a_k(x^k, w^k) \leq a_k(x^{k+1}, w^{k+1}) \quad \text{and} \quad a_{k+1}(x^{k+1}, w^{k+1}) \leq a_{k+1}(x^k, w^k).$$

By adding these inequalities together we find that $\left\|x^{k+1} - w^{k+1}\right\| \leq \left\|x^k - w^k\right\|$ so that $\left\|x^k - w^k\right\| \downarrow \tilde{\kappa}$ for some $\tilde{\kappa} \geq 0$. We also have

$$\begin{aligned} b_k(x^k, w^k) + (\eta_k/2)\left\|x^k - w^k\right\| &= a_k(x^k, w^k) \\ &\leq a_k(x^{k+1}, w^{k+1}) \\ &= b_{k+1}(x^{k+1}, w^{k+1}) + (\eta_k/2)\left\|x^{k+1} - w^{k+1}\right\| \\ &\leq b_{k+1}(x^{k+1}, w^{k+1}) + (\eta_k/2)\left\|x^k - w^k\right\|, \end{aligned}$$

which gives $b_k(x^k, w^k) \leq b_{k+1}(x^{k+1}, w^{k+1}) \leq \tilde{v}$. Therefore, $b_k(x^k, w^k) \uparrow \hat{v}$ for some $\hat{v} \leq \tilde{v}$. Consequently,

$$\tilde{\kappa} = \lim_k \left\|x^k - w^k\right\| = \lim_k \eta_k^{-1}[a_k(x^k, w^k) - b_k(x^k, w^k)] = 0.$$

Therefore, if $(\bar{x}, \bar{w})$ is any limit point of the sequence $\{(x^k, w^k)\}$, then $\bar{x} = \bar{w}$ and $\mathcal{L}(\bar{x}) + \phi_\mu(\bar{\gamma}) + R(\bar{x}) = v_\mu$ since $\mathcal{L}(x^k) + \phi_\mu(\gamma^k) + R(w^k) \leq a_k(x^k, w^k) \leq v_\mu$ for all $k \in \mathbb{N}$. □

We now pair Theorem 8 with a consistency result for the barrier parameter μ.

Theorem 9 (Consistency as $\mu \to 0$)**.** *Let $\mathcal{L}$ and R be as in Theorem 7. For every $\mu \geq 0$, problem* (2.5.10) *has a solution (β_μ, γ_μ). Moreover, if $\{\mu_k\} \subset \mathbb{R}_{++}$ is such that $\mu_k \downarrow 0$, then the sequence $\{(\beta_{\mu_k}, \gamma_{\mu_k})\}$ is bounded and every limit point of the sequence is a solution to* (2.3.1).

Proof. The existence of (β_μ, γ_μ) for all $\mu \geq 0$ follows immediately from Lemma 6 and Theorem 2 with $\widehat{R} = R + \phi_\mu$. Let $\mu_k \downarrow 0$ and set $(\beta^k, \gamma^k) := (\beta_{\mu_k}, \gamma_{\mu_k})$. Set $\widetilde{\mathcal{L}} := \mathcal{L} + R + \delta_{\mathbb{R}^p \times \mathbb{R}^q_+}$ so that the objective in (2.5.10) is $\widetilde{\mathcal{L}} + \phi_\mu$ and the objective in (2.3.1) is $\widetilde{\mathcal{L}}$ with (β_0, γ_0) a solution to (2.3.1) by definition. Observe that

$$\begin{aligned} \widetilde{\mathcal{L}}(\beta^k, \gamma^k) + \phi_{\mu_k}(\gamma^k) \leq \widetilde{\mathcal{L}}(\beta^{k+1}, \gamma^{k+1}) + \phi_{\mu_k}(\gamma^{k+1}) \quad \text{and} \\ \widetilde{\mathcal{L}}(\beta^{k+1}, \gamma^{k+1}) + \phi_{\mu_{k+1}}(\gamma^{k+1}) \leq \widetilde{\mathcal{L}}(\beta^k, \gamma^k) + \phi_{\mu_{k+1}}(\gamma^k) \end{aligned}$$

Summing these inequalities yields the inquality

$$(\mu_k - \mu_{k+1}) \sum_{i=1}^q \ln(\gamma_i^k) \geq (\mu_k - \mu_{k+1}) \sum_{i=1}^q \ln(\gamma_i^{k+1}),$$

so $\{\sum_{i=1}^{q} \ln(\gamma_i^k)\}$ is a non-increasing sequence. Therefore,

$$\begin{aligned}\widetilde{\mathcal{L}}(\beta^{k+1},\gamma^{k+1}) + \phi_{\mu_{k+1}}(\gamma^{k+1}) &\leq \widetilde{\mathcal{L}}(\beta^{k},\gamma^{k}) + \phi_{\mu_{k+1}}(\gamma^{k}) \\ &\leq \widetilde{\mathcal{L}}(\beta^{k},\gamma^{k}) + \phi_{\mu_{k+1}}(\gamma^{k+1})\end{aligned}$$

which implies that $\{\widetilde{\mathcal{L}}(\beta^k,\gamma^k)\}$ is also a non-increasing sequence and bounded below by $\widetilde{\mathcal{L}}(\beta_0,\gamma_0)$. Since Theorem 2 tells us that $\widetilde{\mathcal{L}}$ is level compact, the sequence $\{(\beta^k,\gamma^k)\}$ is bounded. Let $(\bar{\beta},\bar{\gamma}) \in \mathbb{R}^p \times \mathbb{R}^q_+$ be any limit point of $\{(\beta^k,\gamma^k)\}$ and let $J \subset \mathbb{N}$ be such that $(\beta^k,\gamma^k) \overset{J}{\to} (\bar{\beta},\bar{\gamma})$. Then

$$\widetilde{\mathcal{L}}(\beta^k,\gamma^k) + \phi_{\mu_k}(\gamma^k) \leq \widetilde{\mathcal{L}}(\beta,\gamma) + \phi_{\mu_k}(\gamma) \quad \forall\, (\beta,\gamma) \in \mathbb{R}^p \times \mathbb{R}^q_{++}\,.$$

Since $\widetilde{\mathcal{L}}$ is continuous on $\mathbb{R}^p \times \mathbb{R}^q_+$ and the perspective function $\phi_\mu(\gamma) = \varphi(\mu,\gamma)$ is lsc on $\mathbb{R}^p \times \mathbb{R}^q_+$, we have

$$\widetilde{\mathcal{L}}(\bar{\beta},\bar{\gamma}) \leq \liminf_{k\in J}(\widetilde{\mathcal{L}}(\beta^k,\gamma^k) + \phi_{\mu_k}(\gamma^k)) \leq \widetilde{\mathcal{L}}(\beta,\gamma) \quad \forall\, (\beta,\gamma) \in \mathbb{R}^p \times \mathbb{R}^q_{++}\,.$$

Consequently, the continuity of $\widetilde{\mathcal{L}}$ on $\mathbb{R}^p \times \mathbb{R}^q_+$ implies that $(\bar{\beta},\bar{\gamma})$ solves (2.3.1). □

The continuity and differentiability of $u_{\eta,\mu}$

The continuity of $u_{\eta,\mu}$ is closely tied to the continuity of the associated solution mapping $\mathcal{S}_{\eta,\mu} : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^p \times \mathrm{dom}(\phi_\mu)$ given by

$$\mathcal{S}_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) := \operatorname*{argmin}_{(\beta,\gamma)} \mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma}))\,. \tag{2.5.11}$$

Theorem 10 (Continuity of $u_{\eta,\mu}$ and $\mathcal{S}_{\eta,\mu}$)**.** *Let the assumptions of Theorem 7 hold. For every $(\mu,\eta) \in \mathbb{R}_+ \times \mathbb{R}_{++}$, the function $u_{\eta,\mu}$ defined in* (2.3.15) *is well-defined and continuous on $\mathbb{R}^p \times \mathbb{R}^q$. In addition, the solution mapping $\mathcal{S}_{\eta,\mu}$ is well-defined, single-valued and continuous on $\mathbb{R}^p \times \mathbb{R}^q$.*

Proof. Since $\eta > \bar{\eta} = \nu m$, Lemma 5 tells us that the objective in (2.3.15) is strongly convex, and so (2.3.15) has a unique solution. Consequently, $\mathcal{S}_{\eta,\mu}$ is well-defined and single-valued on $\mathbb{R}^p \times \mathbb{R}^q$. This implies that $u_{\eta,\mu}$ is also well defined on $\mathbb{R}^p \times \mathbb{R}^q$ since

$$u_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) = \mathcal{L}_{\eta,\mu}(\mathcal{S}_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}),(\tilde{\beta},\tilde{\gamma})) \quad \forall\, (\tilde{\beta},\tilde{\gamma}) \in \mathbb{R}^p \times \mathbb{R}^q.$$

The result follows once it is shown that $\mathcal{S}_{\eta,\mu}$ is continuous.

Let $\{(\tilde{\beta}^k,\tilde{\gamma}^k)\} \subset \mathbb{R}^p \times \mathbb{R}^q$ and $(\tilde{\beta}^*,\tilde{\gamma}^*) \in \mathbb{R}^p \times \mathbb{R}^q$ be such that $(\tilde{\beta}^k,\tilde{\gamma}^k) \to (\tilde{\beta}^*,\tilde{\gamma}^*)$. Set $(\hat{\beta}^k,\hat{\gamma}^k) := \mathcal{S}_{\eta,\mu}(\tilde{\beta}^k,\tilde{\gamma}^k),\ k \in \mathbb{N}$ and $(\bar{\beta},\bar{\gamma}) = \mathcal{S}_{\eta,\mu}(\tilde{\beta}^*,\tilde{\gamma}^*)$. We must show that $(\hat{\beta}^k,\hat{\gamma}^k) \to (\bar{\beta},\bar{\gamma})$. We begin by showing that the sequence $\{(\hat{\beta}^k,\hat{\gamma}^k)\}$ is bounded. By Lemma 5, the mapping $(\beta,\gamma) \mapsto \mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma}))$ is strongly convex with modulus of strong convexity η for all

$(\tilde{\beta}, \tilde{\gamma}) \in \mathbb{R}^p \times \mathbb{R}^q$. In particular, this implies that

$$\begin{aligned} \mathcal{L}_{\eta,\mu}((\bar{\beta},\bar{\gamma}),(\tilde{\beta}^k,\tilde{\gamma}^k)) + \Big\langle \nabla_{(\beta,\gamma)} \mathcal{L}_{\eta,\mu}((\bar{\beta},\bar{\gamma}),(\tilde{\beta}^k,\tilde{\gamma}^k)),\, (\hat{\beta}^k,\hat{\gamma}^k) - (\bar{\beta},\bar{\gamma}) \Big\rangle \\ + \frac{\eta}{2} \left\| (\hat{\beta}^k,\hat{\gamma}^k) - (\bar{\beta},\bar{\gamma}) \right\|^2 \\ \leq \mathcal{L}_{\eta,\mu}((\hat{\beta}^k,\hat{\gamma}^k),(\tilde{\beta}^k,\tilde{\gamma}^k)) \\ \leq \mathcal{L}_{\eta,\mu}((\bar{\beta},\bar{\gamma}),(\tilde{\beta}^k,\tilde{\gamma}^k)). \end{aligned} \tag{2.5.12}$$

Since $(\tilde{\beta}^k, \tilde{\gamma}^k) \to (\tilde{\beta}^*, \tilde{\gamma}^*)$ and both $\nabla_{(\beta,\gamma)} \mathcal{L}_{\eta,\mu}((\bar{\beta},\bar{\gamma}),\cdot)$ and $\mathcal{L}_{\eta,\mu}((\bar{\beta},\bar{\gamma}),\cdot)$ are continuous at $(\tilde{\beta}^*, \tilde{\gamma}^*)$, we can assume with no loss in generality that there is a constant $c > 0$ such that

$$\left\| \nabla_{(\beta,\gamma)} \mathcal{L}_{\eta,\mu}((\bar{\beta},\bar{\gamma}),(\tilde{\beta}^k,\tilde{\gamma}^k)) \right\| \leq c \text{ and } |\, \mathcal{L}_{\eta,\mu}((\bar{\beta},\bar{\gamma}),(\tilde{\beta}^k,\tilde{\gamma}^k))| \leq c \quad \forall\, k \in \mathbb{N}\,.$$

Plugging this into (2.5.12) and simplifying gives

$$\frac{\eta}{2} \left\| (\hat{\beta}^k,\hat{\gamma}^k) - (\bar{\beta},\bar{\gamma}) \right\|^2 \leq c(1 + \left\| (\hat{\beta}^k,\hat{\gamma}^k) - (\bar{\beta},\bar{\gamma}) \right\|).$$

Therefore the sequence $\{(\hat{\beta}^k, \hat{\gamma}^k)\}$ must be bounded.

Let (β_0, γ_0) be any limit point of $\{(\hat{\beta}^k, \hat{\gamma}^k)\}$ and let $J \subset \mathbb{N}$ be such that $(\hat{\beta}^k, \hat{\gamma}^k) \xrightarrow{J} (\beta_0, \gamma_0)$. Then, by the final inequality in (2.5.12), we can take the limit in $k \in J$ to find that $\mathcal{L}_{\eta,\mu}((\beta_0,\gamma_0),(\tilde{\beta}^*,\tilde{\gamma}^*)) \leq \mathcal{L}_{\eta,\mu}((\bar{\beta},\bar{\gamma}),(\tilde{\beta}^*,\tilde{\gamma}^*))$. The uniqueness of $(\bar{\beta},\bar{\gamma})$ tells us that $(\beta_0,\gamma_0) = (\bar{\beta},\bar{\gamma})$. Since (β_0,γ_0) was any limit point of the bounded sequence $\{(\hat{\beta}^k,\hat{\gamma}^k)\}$, we have $(\hat{\beta}^k,\hat{\gamma}^k) \to (\bar{\beta},\bar{\gamma})$ which implies that $\mathcal{S}_{\eta,\mu}$ is continuous on $\mathbb{R}^p \times \mathbb{R}^q$. □

We now consider the differentiability of $u_{\eta,\mu}$. For this we make use of the following lemma.

Lemma 11 (Local uniform level boundedness of $\mathcal{L}_{\eta,\mu}$)**.** *Let $\mu \geq 0$, $\eta > \bar{\eta}$ and suppose that the assumptions of Theorem 7 hold. Set $x = (\beta,\gamma)$ and $w = (\tilde{\beta},\tilde{\gamma})$. Then the function $\mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma}))$ is level bounded in (β,γ) locally uniformly in $(\tilde{\beta},\tilde{\gamma})$ for all $(\tilde{\beta},\tilde{\gamma}) \in \mathbb{R}^p \times \mathbb{R}^q$. That is, for every $(\tilde{\beta},\tilde{\gamma}) \in \mathbb{R}^p \times \mathbb{R}^q$ and $\rho \in \mathbb{R}$, there are $\nu \in \mathbb{R}$ and $\epsilon > 0$ such that $\big\{(\beta,\gamma) \;\big|\; \mathcal{L}_{\eta,\mu}((\beta,\gamma),(\bar{\beta},\bar{\gamma}) \leq \rho\big\} \subset \nu\,\mathbb{B}$ for all $(\tilde{\beta},\tilde{\gamma}) \in (\bar{\beta},\bar{\gamma}) + \epsilon\,\mathbb{B}$.*

Proof. Set $x = (\beta,\gamma)$, $w = (\tilde{\beta},\tilde{\gamma})$, and $\bar{w} = (\bar{\beta},\bar{\gamma})$. If the result is false, there exists $\bar{w} \in \mathbb{R}^p \times \mathbb{R}^q$, $\rho > 0$, and a sequence $\{(x^k, w^k)\} \subset (\mathbb{R}^p \times \operatorname{dom}(\phi_\mu)) \times (\mathbb{R}^p \times \mathbb{R}^q)$ such that $w^k \to \bar{w}$ and $\left\|x^k\right\| \uparrow \infty$ with $x^k \in \big\{x \;\big|\; \mathcal{L}_{\eta,\mu}(x, w^k) \leq \rho\big\}$ for all $k \in \mathbb{N}$. By Lemma 5, the mappings $x \mapsto \mathcal{L}_{\eta,\mu}(x,w)$ are strongly convex with modulus $\hat{\eta} := \eta - \bar{\eta} > 0$ for all $w \in \mathbb{R}^p \times \mathbb{R}^q$. Let $\hat{x} \in \mathbb{R}^p \times \operatorname{dom}\phi_\mu$. Then $(\hat{x},\bar{w})$ is a point of continuity for $\nabla_x \mathcal{L}_{\eta,\mu}$, so with no loss in generality there is a $c_0 > 0$ such that $\left\|\nabla_x \mathcal{L}_{\eta,\mu}(\hat{x}, w^k)\right\| \leq c_0$ for all $k \in \mathbb{N}$. Then strong convexity implies that

$$\begin{aligned} \mathcal{L}_{\eta,\mu}(\hat{x}, w^k) - c_0 \left\|x^k - \hat{x}\right\| + \frac{\hat{\eta}}{2} \left\|x^k - \hat{x}\right\|^2 \\ \leq \mathcal{L}_{\eta,\mu}(\hat{x}, w^k) + \left\langle \nabla_x \mathcal{L}_{\eta,\mu}(\hat{x}, w^k),\, x^k - \hat{x} \right\rangle + \frac{\hat{\eta}}{2} \left\|x^k - \hat{x}\right\|^2 \\ \leq \mathcal{L}_{\eta,\mu}(x^k, w^k) \leq \rho. \end{aligned}$$

But $\mathcal{L}_{\eta,\mu}(\hat{x}, w^k) - c_0 \left\|x^k - \hat{x}\right\| + \frac{\hat{\eta}}{2} \left\|x^k - \hat{x}\right\|^2 \uparrow \infty$ since $\left\|x^k\right\| \uparrow \infty$. This contradiction establishes the result. □

Theorem 12 (Differentiability of $u_{\eta,\mu}$)**.** *Let $\mu \geq 0$, $\eta > \bar{\eta}$ and suppose that the assumptions of Theorem 7 hold. Then the function $u_{\eta,\mu}$ defined in* (2.3.15) *is continuously differentiable on $\mathbb{R}^p \times \mathbb{R}^q$ with*

$$\nabla u_{\eta,\mu}(\tilde{\beta}, \tilde{\gamma}) = \nabla_{(\tilde{\beta},\tilde{\gamma})} \kappa_\eta(\tilde{\beta} - \hat{\beta}, \tilde{\gamma} - \hat{\gamma}) = \eta \begin{pmatrix} \tilde{\beta} - \hat{\beta} \\ \tilde{\gamma} - \hat{\gamma} \end{pmatrix}, \text{ where } (\hat{\beta}, \hat{\gamma}) = \mathcal{S}_{\eta,\mu}(\tilde{\beta}, \tilde{\gamma}). \tag{2.5.13}$$

Proof. We show that the result follows from (Rockafellar and Wets, 2009, Theorem 10.58). Set $x = (\beta, \gamma)$ and $w = (\tilde{\beta}, \tilde{\gamma})$. The objective function in the definition of $u_{\eta,\mu}$ is $\mathcal{L}_{\eta,\mu}(x, w)$, where $\mathcal{L}_{\eta,\mu}$ is proper and lsc. Moreover, Lemma 11 tells us that $\mathcal{L}_{\eta,\mu}(x, w)$ is level bounded in x locally uniformly in w for all $w \in \mathbb{R}^p \times \mathbb{R}^q$. We have already observed that, for all $\mu \in \mathbb{R}_+$ and $\eta > \bar{\eta}$, $\mathcal{L}_{\eta,\mu}$ and $\nabla \mathcal{L}_{\eta,\mu}$ are continuous on $(\mathbb{R}^p \times \operatorname{dom}(\phi_\mu)) \times (\mathbb{R}^p \times \mathbb{R}^q)$. Therefore, by (Rockafellar and Wets, 2009, Theorem 10.58) and Theorem 10, $u_{\eta,\mu}$ is locally upper-$\mathcal{C}^1$ and strictly differentiable at every point $w \in \mathbb{R}^p \times \mathbb{R}^q$ with $\nabla u_{\eta,\mu}(w) = \nabla_w \mathcal{L}_{\eta,\mu}(\mathcal{S}_{\eta,\mu}(w))$. In addition, $\mathcal{S}_{\eta,\mu}$ is continuous on $\mathbb{R}^p \times \mathbb{R}^q$. The result follows since $\nabla_w \mathcal{L}_{\eta,\mu}((\beta, \gamma), (\tilde{\beta}, \tilde{\gamma})) = \eta \begin{pmatrix} \tilde{\beta} - \beta \\ \tilde{\gamma} - \gamma \end{pmatrix}$. □

The Lipschitz Continuity of the Gradient of the Value Function

Since our goal is to employ the PGD algorithm to solve the relaxed problems (2.3.11), we require that $\nabla u_{\eta,\mu}(\tilde{\beta}, \tilde{\gamma})$ be Lipschitz continuous. Formula (2.5.13) tells us that the Lipschitz continuity of $\nabla u_{\eta,\mu}(\tilde{\beta}, \tilde{\gamma})$ is equivalent to that of the solution mapping $\mathcal{S}_{\eta,\mu}$. To study the Lipschitz continuity of $\mathcal{S}_{\eta,\mu}$ we make use of the mapping $G : \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^p \to \mathbb{R}^p \times \mathbb{R}^q$ be given by

$$G_{\eta,\mu}((\beta, \gamma, v), (\tilde{\beta}, \tilde{\gamma})) := \begin{bmatrix} \nabla_\beta \mathcal{L}(\beta, \gamma) + \eta(\beta - \tilde{\beta}) \\ \nabla_\gamma \mathcal{L}(\beta, \gamma) + \eta(\gamma - \tilde{\gamma}) - v \\ v \odot \gamma - \mu \mathbf{1} \end{bmatrix}. \tag{2.5.14}$$

Observe that, for $\mu > 0$, $(\hat{\beta}, \hat{\gamma}) = \mathcal{S}_{\eta,\mu}(\tilde{\beta}, \tilde{\gamma})$ if and only if

$$\hat{\gamma}, \hat{v} \in \mathbb{R}^q_+ \quad \text{and} \quad G_{\eta,\mu}((\hat{\beta}, \hat{\gamma}, \hat{v}), (\tilde{\beta}, \tilde{\gamma})) = 0, \tag{2.5.15}$$

since the equation $v \odot \gamma = \mu \mathbf{1}$ implies that $v = -\nabla \phi_\mu(\gamma)$. In addition, when $\mu = 0$, condition (2.5.15) is equivalent to $(\hat{\beta}, \hat{\gamma}, v)$ being a KKT point for the optimization problem in (2.3.15) which, in turn, is equivalent to $(\hat{\beta}, \hat{\gamma}) = \mathcal{S}_{\eta,\mu}(\tilde{\beta}, \tilde{\gamma})$ by Theorem 10. We record these observations in the following lemma.

Lemma 13. *Let the assumptions of Theorem 7 hold. Then, for every $(\mu, \eta) \in \mathbb{R}_+ \times \mathbb{R}_{++}$, $(\hat{\beta}, \hat{\gamma}) = \mathcal{S}_{\eta,\mu}(\tilde{\beta}, \tilde{\gamma})$ if and only if there is a vector $\hat{v} \in \mathbb{R}^q_+$ such that $G_{\eta,\mu}((\hat{\beta}, \hat{\gamma}, \hat{v}), (\tilde{\beta}, \tilde{\gamma})) = 0$. If $\mu > 0$, then $\hat{v} = -\nabla \phi_\mu(\hat{\gamma})$, and if $\mu = 0$, then $\hat{v}$ is the unique KKT multiplier associated with the constraint $0 \leq \gamma$.*

Our approach to establishing the Lipschitz continuity of $\mathcal{S}_{\eta,\mu}$ is to first show that $\mathcal{S}_{\eta,\mu}$ is differentiable and then obtain a bound on its Jacobian. As usual, diffentiability follows by applying the implicit function theorem to $G_{\eta,\mu}$.

Lemma 14 (The invertibility of $\nabla_{(\beta,\gamma,v)}G_{\eta,\mu}$)**.** *Let the assumptions of Theorem 7 hold and let $G_{\eta,\mu}$ be as given in* (2.5.14)*. Let $(\tilde{\beta},\tilde{\gamma})\in\mathbb{R}^p\times\mathbb{R}^q$ and $(\hat{\beta},\hat{\gamma},\hat{v})\in\mathbb{R}^p\times\mathbb{R}^q_+\times\mathbb{R}^q_+$ be such that $G_{\eta,\mu}((\hat{\beta},\hat{\gamma},\hat{v}),(\tilde{\beta},\tilde{\gamma}))=0$. Then $\nabla_{(\beta,\gamma,v)}G_{\eta,\mu}((\hat{\beta},\hat{\gamma},\hat{v}),(\tilde{\beta},\tilde{\gamma}))$ is invertible if and only if*

$$0<\hat{v}_i+\hat{\gamma}_i,\ i=1,\dots,q,\qquad \textit{(strict complementary slackness)} \tag{2.5.16}$$

which automatically holds if $\mu>0$. In this case, the inverse is given by

$$\begin{bmatrix} H^{-1}-\begin{bmatrix}\hat{R}\\ \hat{H}_2\end{bmatrix}(D(\hat{\gamma})+D(\hat{v})\hat{H}_2)^{-1}D(\hat{v})[\hat{R}^T\ \hat{H}_2] & \begin{bmatrix}\hat{R}\\ \hat{H}_2\end{bmatrix}(D(\hat{\gamma})+D(\hat{v})\hat{H}_2)^{-1} \\ -(D(\hat{\gamma})+D(\hat{v})\hat{H}_2)^{-1}D(\hat{v})[\hat{R}^T\ \hat{H}_2] & (D(\hat{\gamma})+D(\hat{v})\hat{H}_2)^{-1}\end{bmatrix}, \tag{2.5.17}$$

where $D(\hat{\gamma}):=\operatorname{Diag}(\hat{\gamma})$, $D(\hat{v}):=\operatorname{Diag}(\hat{v})$,

$$\begin{aligned} H=\begin{bmatrix}H_1 & R\\ R^T & H_2\end{bmatrix}&:=\begin{bmatrix}\nabla_{\beta\beta}\mathcal{L}(\hat{\beta},\hat{\gamma})+\eta I & \nabla_{\gamma\beta}\mathcal{L}(\hat{\beta},\hat{\gamma})\\ \nabla_{\beta\gamma}\mathcal{L}(\hat{\beta},\hat{\gamma}) & \nabla_{\gamma\gamma}\mathcal{L}(\hat{\beta},\hat{\gamma})+\eta I\end{bmatrix} \qquad \textit{and}\\ H^{-1}&=\begin{bmatrix}H_1^{-1}+H_1^{-1}R(H_2-R^TH_1^{-1}R)^{-1}R^TH_1^{-1} & -H_1^{-1}R(H_2-R^TH_1^{-1}R)^{-1}\\ -(H_2-R^TH_1^{-1}R)^{-1}R^TH_1^{-1} & (H_2-R^TH_1^{-1}R)^{-1}\end{bmatrix}\\ &=:\begin{bmatrix}\hat{H}_1 & \hat{R}\\ \hat{R}^T & \hat{H}_2\end{bmatrix}. \end{aligned}$$

Proof. Observe that

$$\nabla_{(\beta,\gamma,v)}G_{\eta,\mu}((\beta,\gamma,v),(\tilde{\beta},\tilde{\gamma}))=\begin{bmatrix}\nabla_{\beta\beta}\mathcal{L}(\beta,\gamma)+\eta I & \nabla_{\gamma\beta}\mathcal{L}(\beta,\gamma) & 0\\ \nabla_{\beta\gamma}\mathcal{L}(\beta,\gamma) & \nabla_{\gamma\gamma}\mathcal{L}(\beta,\gamma)+\eta I & -I\\ 0 & \operatorname{Diag}(v) & \operatorname{Diag}(\gamma)\end{bmatrix}. \tag{2.5.18}$$

Let us first assume that $\nabla_{(\beta,\gamma,v)}G_{\eta,\mu}((\hat{\beta},\hat{\gamma},\hat{v}),(\tilde{\beta},\tilde{\gamma}))$ is invertible and, for simplicity write

$$\nabla_{(\beta,\gamma,v)}G_{\eta,\mu}((\hat{\beta},\hat{\gamma},\hat{v}),(\tilde{\beta},\tilde{\gamma}))=\begin{bmatrix}H & A\\ B^T & D\end{bmatrix},$$

where $A:=[0,\ -I]^T$, $B=[0,\ \operatorname{Diag}(\hat{v})]^T$ and $D:=\operatorname{Diag}(\hat{\gamma})$. Since $H\in\mathbb{S}^{p+q}_{++}$, the matrix

$$\begin{aligned}\begin{bmatrix}I & 0\\ -B^TH^{-1} & I\end{bmatrix}\begin{bmatrix}H & A\\ B^T & D\end{bmatrix}\begin{bmatrix}I & -H^{-1}A\\ 0 & I\end{bmatrix}&=\begin{bmatrix}H & 0\\ 0 & D-B^TH^{-1}A\end{bmatrix}\\ &=\begin{bmatrix}H & 0\\ 0 & \operatorname{Diag}(\hat{\gamma})+\operatorname{Diag}(\hat{v})\,\hat{H}_2\end{bmatrix}\end{aligned} \tag{2.5.19}$$

is nonsingular. In particular, the matrix $\operatorname{Diag}(\hat{\gamma})+\operatorname{Diag}(\hat{v})\,\hat{H}_2$ is necessarily invertible. But if there is an i such that $0=\hat{\gamma}_i+\hat{v}_i$, then $\hat{\gamma}_i=\hat{v}_i=0$ so that the matrix $\operatorname{Diag}(\hat{\gamma})+\operatorname{Diag}(\hat{v})\,H_{22}$ has a zero row and so is singular. Since this cannot be that case, (2.5.16) must hold.

Conversely, suppose $(r^T, s^T, t^T)^T$ is in the nullspace of $\nabla_{(\beta,\gamma,v)} G_{\eta,\mu}((\hat{\beta},\hat{\gamma},\hat{v}),(\tilde{\beta},\tilde{\gamma}))$. Then $0 = \text{Diag}\,(\hat{v})\,s + \text{Diag}\,(\hat{\gamma})\,t$. This combined with (2.5.16) implies that $s^T t = 0$. Consequently,

$$0 = \begin{pmatrix} r \\ s \end{pmatrix}^T \begin{pmatrix} 0 \\ t \end{pmatrix} = \begin{pmatrix} r \\ s \end{pmatrix}^T H \begin{pmatrix} r \\ s \end{pmatrix},$$

which implies that $(r, s) = (0, 0)$ since H is positive definite. Therefore, $t = 0$ which shows that $\nabla_{(\beta,\gamma,v)} G_{\eta,\mu}((\hat{\beta},\hat{\gamma},\hat{v}),(\tilde{\beta},\tilde{\gamma}))$ is nonsingular.

The formula for the inverse follows from (2.5.19) which tells us that

$$\begin{bmatrix} H & A \\ B^T & D \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ -B^T H^{-1} & I \end{bmatrix} \begin{bmatrix} H^{-1} & 0 \\ 0 & (\text{Diag}\,(\hat{\gamma}) + \text{Diag}\,(\hat{v})\,\hat{H}_2)^{-1} \end{bmatrix} \begin{bmatrix} I & -H^{-1}A \\ 0 & I \end{bmatrix}. \quad (2.5.20)$$

Alternatively, one can apply the formulas in Lu and Shiou (2002). □

Using Lemma 14, we apply the implicit function theorem to the equation

$$G_{\eta,\mu}((\hat{\beta},\hat{\gamma},\hat{v}),(\tilde{\beta},\tilde{\gamma})) = 0$$

and obtain the following result.

Theorem 15 (Differentiability of $\mathcal{S}_{\eta,\mu}$)**.** *Let the hypotheses and notation of Lemma 14 hold and let $\mathcal{S}_{\eta,\mu}$ be as defined in* (2.5.11)*. Given $\eta, \mu \in \mathbb{R}_+ \times \mathbb{R}_+$, define $\widehat{\mathcal{S}}_{\eta,\mu} : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^p \times \mathbb{R}^q_+ \times \mathbb{R}^q_+$ by*

$$\widehat{\mathcal{S}}_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) = \left\{ (\hat{\beta},\hat{\gamma},\hat{v}) \;\middle|\; \hat{\gamma},\hat{v} \in \mathbb{R}^q_+ \;\text{ and } 0 = G_{\eta,\mu}((\hat{\beta},\hat{\gamma},\hat{v}),(\tilde{\beta},\tilde{\gamma})) \right\} \quad (2.5.21)$$

Suppose $(\tilde{\beta},\tilde{\gamma}) \in \mathbb{R}^p \times \mathbb{R}^q$ and $(\bar{\beta},\bar{\gamma},\bar{v}) = \widehat{\mathcal{S}}_{\eta,\mu}(\tilde{\beta},\tilde{\gamma})$ with $\bar{\gamma},\bar{v} \in \mathbb{R}^q_+$ and such that (2.5.16) *holds. Then there exist open neighborhoods $\widetilde{\mathcal{N}}$ of $(\tilde{\beta},\tilde{\gamma})$ and such that $\mathcal{S}_{\eta,\mu}$ and $\widehat{\mathcal{S}}_{\eta,\mu}$ are differentiable on $\widetilde{\mathcal{N}}$ with*

$$\nabla \widehat{\mathcal{S}}_{\eta,\mu}(\beta,\gamma) = \eta \begin{bmatrix} H^{-1} - \begin{bmatrix} \hat{R} \\ \hat{H}_2 \end{bmatrix} (D(\hat{\gamma}) + D(\hat{v})\hat{H}_2)^{-1} D(\hat{v}) [\hat{R}^T \;\; \hat{H}_2] \\ -(D(\hat{\gamma}) + D(\hat{v})\hat{H}_2)^{-1} D(\hat{v}) [\hat{R}^T \;\; \hat{H}_2] \end{bmatrix},$$

$$\nabla \mathcal{S}_{\eta,\mu}(\beta,\gamma) = \eta \left[H^{-1} - \begin{bmatrix} \hat{R} \\ \hat{H}_2 \end{bmatrix} (D(\hat{\gamma}) + D(\hat{v})\hat{H}_2)^{-1} D(\hat{v}) [\hat{R}^T \;\; \hat{H}_2] \right]$$

for all $(\beta,\gamma) \in \widetilde{\mathcal{N}}$ and $(\hat{\beta},\hat{\gamma},\hat{v}) = \widehat{\mathcal{S}}_{\eta,\mu}(\beta,\gamma)$. In particular, this implies that both $\widehat{\mathcal{S}}_{\eta,\mu}$ and $\mathcal{S}_{\eta,\mu}$ are continuously differentiable on $\mathbb{R}^p \times \mathbb{R}^q$.

Using the notation of Lemma 14 the expression for $\nabla \mathcal{S}_{\eta,\mu}(\beta,\gamma)$ in Theorem 15 can be simplified when $\mu > 0$ to

$$\nabla \mathcal{S}_{\eta,\mu}(\beta,\gamma) = \eta \left[H^{-1} - \begin{bmatrix} -H_1^{-1}R \\ I \end{bmatrix} \hat{H}_2 (\mu^{-1} \text{Diag}\,(\hat{\gamma})^2 + \hat{H}_2)^{-1} \hat{H}_2 [-R^T H_1^{-1} \;\; I] \right].$$

By combining this with the Shur complement formula (e.g., see (2.5.19) and (2.5.20))

$$H^{-1} = \begin{bmatrix} I & -H_1^{-1}R \\ 0 & I \end{bmatrix} \begin{bmatrix} H_1^{-1} & 0 \\ 0 & (H_2 - R^T H_1^{-1} R)^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -R^T H_1^{-1} & I \end{bmatrix},$$

where $\hat{H}_2 = (H_2 - R^T H_1^{-1} R)^{-1}$ is positive definite, we obtain

$$\nabla \mathcal{S}_{\eta,\mu}(\beta,\gamma) = \eta \begin{bmatrix} I & -H_1^{-1}R \\ 0 & I \end{bmatrix} \begin{bmatrix} H_1^{-1} & 0 \\ 0 & \hat{H}_2 - \hat{H}_2(\mu^{-1}\mathrm{Diag}\,(\hat{\gamma})^2 + \hat{H}_2)^{-1}\hat{H}_2 \end{bmatrix} \begin{bmatrix} I & 0 \\ -R^T H_1^{-1} & I \end{bmatrix}$$

Since the matrix

$$\hat{H}_2 - \hat{H}_2(\mu^{-1}\mathrm{Diag}\,(\hat{\gamma})^2 + \hat{H}_2)^{-1}\hat{H}_2 = \hat{H}_2^{1/2}[I - (I + \mu^{-1}\hat{H}_2^{-1/2}\mathrm{Diag}\,(\hat{\gamma})^2\,\hat{H}_2^{-1/2})^{-1}]\hat{H}_2^{1/2}$$

is positive definite, we have that

$$\|\nabla \mathcal{S}_{\eta,\mu}(\beta,\gamma)\| \leq \eta(1 + \left\|H_1^{-1}R\right\|^2) \max\{\left\|H_1^{-1}\right\|, \left\|(H_2 - R^T H_1^{-1} R)^{-1}\right\|\}. \tag{2.5.22}$$

Since $H_1 = \nabla_{\beta\beta}\,\mathcal{L}(\hat{\beta},\hat{\gamma}) + \eta I$, we have $\left\|H_1^{-1}\right\| \leq \eta^{-1} < (\eta - \bar{\eta})^{-1}$. We now show that $(\eta - \bar{\eta})^{-1}$ bounds $\left\|(H_2 - R^T H_1^{-1} R)^{-1}\right\|$. For this it is sufficient to show that $(\eta - \bar{\eta}) \leq \mu_{\min}(H_2 - R^T H_1^{-1} R)$. By Lemma 5, the matrix in (2.5.3) is positive semidefinite. Since $\nabla_{\beta\beta}\,\mathcal{L}(\beta,\gamma)$ is positive definite, the Shur complement $\nabla_{\gamma\gamma}\,\mathcal{L}(\beta,\gamma) + \bar{\eta} I - \nabla_{\beta\gamma}\,\mathcal{L}(\beta,\gamma)\nabla_{\beta\beta}\,\mathcal{L}(\beta,\gamma)^{-1}\nabla_{\gamma\beta}\,\mathcal{L}(\beta,\gamma)$ is positive semidefinite. Consequently,

$$\begin{aligned} H_2 - R^T H_1^{-1} R = (\eta - \bar{\eta})I + (\nabla_{\gamma\gamma}\,\mathcal{L}(\beta,\gamma) + \bar{\eta} I - R^T \nabla_{\beta\beta}\,\mathcal{L}(\beta,\gamma)^{-1} R) \\ + R^T(\nabla_{\beta\beta}\,\mathcal{L}(\beta,\gamma)^{-1} - (\nabla_{\beta\beta}\,\mathcal{L}(\beta,\gamma) + \eta I)^{-1})R \;\succeq (\eta - \bar{\eta})I, \end{aligned}$$

since $\nabla_{\beta\beta}\,\mathcal{L}(\beta,\gamma)^{-1} - (\nabla_{\beta\beta}\,\mathcal{L}(\beta,\gamma) + \eta I)^{-1}$ is positive definite. Therefore, $(\eta - \bar{\eta}) \leq \mu_{\min}(H_2 - R^T H_1^{-1} R)$. By combining this with (2.5.22) we obtain the bound

$$\|\nabla \mathcal{S}_{\eta,\mu}(\beta,\gamma)\| \leq \frac{\eta}{\eta - \bar{\eta}} \left(1 + \left\|H_1^{-1}R\right\|^2\right), \tag{2.5.23}$$

where

$$\begin{aligned} H_1^{-1}R &= -(X^T\Omega(\hat{\gamma})^{-1}X + \eta I)^{-1} \sum_{i=1}^{m} X_i^T \Omega_i(\hat{\gamma})^{-1} Z_i \mathrm{Diag}\left(Z_i^T \Omega_i(\hat{\gamma})^{-1}(X_i\hat{\beta} - Y_i)\right) \\ &= -(X^T\Omega(\hat{\gamma})^{-1}X + \eta I)^{-1} X^T \Omega(\hat{\gamma})^{-1} \widehat{Z}\, \mathrm{Diag}\left(\widehat{Z}^T \Omega(\hat{\gamma})^{-1} r(\hat{\beta})\right), \end{aligned}$$

with $r(\hat{\beta}) := X\beta - y$ and $\widehat{Z} = \mathrm{Diag}\,(Z_1, Z_2, \ldots, Z_m)$. Therefore, as in Lemma 5, we obtain the bound

$$\left\|H_1^{-1}R\right\| \leq \eta^{-1}\mu_{\min}^{-2}(\Lambda)\sigma_{\max}(X)\sigma_{\max}^2(Z)\left\|X\hat{\beta} - y\right\|. \tag{2.5.24}$$

This inequality can be used to show that $\nabla u_{\eta,\mu}$ is bounded on the lower level sets of $u_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma})$ if $\mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma})$ is level compact. However, we only know that this is true if we can bound the values of $\hat{\gamma}$ over these sets. In practice, the values of $\hat{\gamma}$ are bounded if the model is well posed since these values are tied to the variances of the random effects. One can accommodate this by adding a constraint of the form $\gamma \leq \gamma_{\max}$ for $\gamma_{\max} \in \mathbb{R}^q_{++}$ chosen sufficiently large.

Lemma 16 (Lipschitz Continuity of $\nabla u_{\eta,\mu}$)**.** *Let the assumptions of Theorem 7 hold and suppose $\mu > 0$ and $\gamma_{max} \in \mathbb{R}^q_{++}$. Let $\zeta \in \mathbb{R}$ and set*

$$\widehat{\mathcal{V}}(\eta,\mu,\gamma_{max},\zeta) := \left\{ ((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) \;\middle|\; \begin{array}{c} \mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma}) \leq \zeta, \\ \gamma,\ \tilde{\gamma} \leq \gamma_{max} \end{array} \right\}, \text{ and}$$

$$\mathcal{V}(\eta,\mu,\gamma_{max},\zeta) := \left\{ (\tilde{\beta},\tilde{\gamma}) \;\middle|\; u_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma}) \leq \zeta,\ \tilde{\gamma} \leq \gamma_{max} \right\}.$$

Then

1. *both $\widehat{\mathcal{V}}(\eta,\mu,\gamma_{max},\zeta)$ and $\mathcal{V}(\eta,\mu,\gamma_{max},\zeta)$ are compact with*

$$\mathcal{V}(\eta,\mu,\gamma_{max},\zeta) \subset \left\{ (\tilde{\beta},\tilde{\gamma}) \;\middle|\; \begin{array}{r} \tilde{\gamma} \leq \gamma_{max} \text{ and } \exists\, (\beta,\gamma) \in \mathbb{R}^p \times \mathbb{R}^q_{++} \text{ s.t.} \\ ((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) \in \widehat{\mathcal{V}}(\eta,\mu,\gamma_{max},\zeta) \end{array} \right\}, \tag{2.5.25}$$

2. *the set $\mathcal{V}(\eta,\mu,\gamma_{max},\zeta)$ has nonempty interior if $\zeta > u_{\eta,\mu}(\tilde{\beta},\tilde{\gamma})$ for some $(\tilde{\beta},\tilde{\gamma}) \in \mathbb{R}^p \times \mathbb{R}^q$, and*

3. *the set $\widetilde{\mathcal{V}}(\eta,\mu,\gamma_{max},\zeta,\omega) := \overline{\text{conv}}(\mathcal{V}(\eta,\mu,\gamma_{max},\zeta) + \omega\mathbb{B})$ is a compact, convex set with nonempty interior whenever $\mathcal{V}(\eta,\mu,\gamma_{max},\zeta) \neq \emptyset$.*

Moreover, $\nabla u_{\eta,\mu}$ is Lipschitz on $\overline{\text{conv}}(\mathcal{V}(\eta,\mu,\gamma_{max},\zeta) + \omega\mathbb{B})$ for every $\omega \geq 0$, where

$$\mathbb{B} := \{ (\beta,\gamma) \in \mathbb{R}^p \times \mathbb{R}^q \mid \|(\beta,\gamma)\| \leq 1 \}.$$

Proof. Since Theorem 1 tells us that $\mathcal{L}$ is bounded below, $\mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma})$ is not level compact if and only if there is an unbounded sequence in a lower level set of $\mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma})$ for which $\gamma^k \uparrow \infty$. Therefore, the compactness of $\widehat{\mathcal{V}}(\eta,\mu,\gamma_{\max},\zeta)$ follows from the lower semicontinuity of $\mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma})$. Since

$$u_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma}) \leq \mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta},\tilde{\gamma})) + R(\tilde{\beta},\tilde{\gamma}) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma}) \quad \forall\, (\beta,\gamma) \in \mathbb{R}^p \times \mathbb{R}^q,$$

the inclusion (2.5.25) holds. In addition, the set on the right hand side of (2.5.25) is the projection of $\widehat{\mathcal{V}}(\eta,\mu,\gamma_{\max},\zeta)$ onto its first components (β,γ) and so is compact. This in turns tells us that $\mathcal{V}(\eta,\mu,\gamma_{\max},\zeta)$ is compact. Hence, $\overline{\text{conv}}(\mathcal{V}(\eta,\mu,\gamma_{\max},\zeta) + \omega\mathbb{B})$ is also compact. The continuity of $u_{\eta,\mu}$ implies that $\mathcal{V}(\eta,\mu,\gamma_{\max},\zeta)$ has nonempty interior if $\zeta > u_{\eta,\mu}(\tilde{\beta},\tilde{\gamma})$ for some $(\tilde{\beta},\tilde{\gamma}) \in \mathbb{R}^p \times \mathbb{R}^q$. Theorem 10 shows that $\mathcal{S}_{\eta,\mu}$ is continuous on $\mathbb{R}^p \times \mathbb{R}^q$ so the bound (2.5.24) combined with Theorem 15 implies that $\mathcal{S}_{\eta,\mu}$ is locally Lipschitz on $\mathbb{R}^p \times \mathbb{R}^q$. Hence, by (2.5.13), $\nabla u_{\eta,\mu}$ is locally Lipschitz on $\mathbb{R}^p \times \mathbb{R}^q$. The compactness of $\overline{\text{conv}}(\mathcal{V}(\eta,\mu,\gamma_{\max},\zeta) + \omega\mathbb{B})$ tells us that $\nabla u_{\eta,\mu}$ is Lipschitz on this set for all $\omega \geq 0$. □

2.5.3 Convergence of the PGD Algorithm for Regularized MSR3 Likelihood

The convergence of the PGD algorithm for fixed valued of the relaxation parameters η and μ appeals to the standard convergence theory as presented in (Beck, 2017, Chapter 10) which requires the use of Assumptions (A)–(C) in Section 2.5.1. We assume that the variable selection regularizer R is chosen so that Assumption (A) holds. In addition, under the assumptions of Theorem 5, Theorem 12 tells us that the function $u_{\eta,\mu}$ is well defined and continuously differentiable on all of $\mathbb{R}^p \times \mathbb{R}^q$ with the solution mapping $\mathcal{S}_{\eta,\mu}(\tilde{\beta}, \tilde{\gamma})$ well defined, single valued, and differentiable on $\mathbb{R}^p \times \mathbb{R}^q$ (Theorem 15). Therefore, Assumption (C) is satisfied as is much of assumption (B). However, as is commonly the case in a specific application, the $L_{\eta,\mu}$-smoothness of $u_{\eta,\mu}$ over $\operatorname{int}(\operatorname{dom} u_{\eta,\mu}) = \mathbb{R}^p \times \mathbb{R}^q$ fails. This drawback is remedied by observing that the PGD algorithm is a descent algorithm. This allows us to focus on the behavior of the functions over the lower level sets described in Lemma 16.

Let $\bar{w}^0 = (\tilde{\beta}^0, \tilde{\gamma}^0) \in \mathbb{R}^p \times \mathbb{R}^q_+$ be the point at which Algorithm 3 is initiated and let $\zeta > \mathcal{L}_{\eta,\mu}((\beta,\gamma),(\tilde{\beta}^0,\tilde{\gamma}^0)) + R(\tilde{\beta}^0,\tilde{\gamma}^0) + \delta_{\mathbb{R}^q_+}(\tilde{\gamma}^0)$ for any $(\beta,\gamma) \in \mathbb{R}^p \times \mathbb{R}^q_{++}$. For $\omega \geq 0$ and $\epsilon \geq 0$, define $\mathfrak{D}(\omega,\epsilon) := \widetilde{\mathcal{V}}(\eta,\mu,\gamma_{\max} + \epsilon\mathbf{1}, \zeta+\epsilon, \omega+\epsilon)$ and set

$$\hat{u}_{\eta,\mu} := u_{\eta,\mu} + \delta_{\mathfrak{D}(\bar{\omega},\bar{\epsilon})} \quad \text{and} \quad \widehat{R} := R + \delta_{\mathfrak{D}(\bar{\omega},0)}, \tag{2.5.26}$$

where $\bar{\epsilon} > 0$, $\widetilde{\mathcal{V}}$ is defined in Lemma 16 and

$$\bar{\omega} := 1 + t_0 \max\left\{ \nabla u_{\eta,\mu}(\tilde{\beta},\tilde{\gamma}) \;\middle|\; (\tilde{\beta},\tilde{\gamma}) \in \mathcal{V}(\eta,\mu,\gamma_{\max},\zeta) \right\}.$$

Observe that all iterates of Algorithm 3 lie in the set $\mathcal{V}(\eta,\mu,\gamma_{\max},\zeta)$ since it is a descent algorithm. Moreover, since the prox operator is nonexpansive (e.g., see (Beck, 2017, Theorem 6.42(a)) or (Rockafellar and Wets, 2009, Theorem 12.19)), all of the points tested in the backtracking line search in Algorithm 3 must also lie in the set $\widetilde{\mathcal{V}}(\eta,\mu,\gamma_{\max},\zeta,\bar{\omega})$ by construction. Therefore, the iterates of Algorithm 3 are identical to those obtained when the algorithm is applied to $\hat{u}_{\eta,\mu}$ with $\widetilde{R} := R + \delta_{\mathfrak{D}(\bar{\omega},0)}$. That is, we can assume that the Algorithm 3 is being applied to $\hat{u}_{\eta,\mu}$. Observe that $\hat{u}_{\eta,\mu}$ is closed and proper, $\operatorname{dom}\hat{u}_{\eta,\mu} = \mathfrak{D}(\bar{\omega},\bar{\epsilon})$ is convex, and $\operatorname{dom}\hat{u}_{\eta,\mu} = \mathfrak{D}(\bar{\omega},\bar{\epsilon})$ has nonempty interior (by Lemma 16(3)) with $\operatorname{dom}\widetilde{R} \subset \operatorname{int}(\operatorname{dom}\hat{u}_{\eta,\mu})$ since $\bar{\epsilon} > 0$. In addition, the final statement of Lemma 16 tells us that there is an $L_{(\eta,\mu,\gamma_{\max},\zeta)} > 0$ such that $\hat{u}_{\eta,\mu}$ is $L_{(\eta,\mu,\gamma_{\max})}$-smooth over $\operatorname{int}(\operatorname{dom}\hat{u}_{\eta,\mu})$. Hence, Assumptions (A)-(C) are satisfied by $\hat{u}_{\eta,\mu}$ and $\widetilde{R}$ and so the convergence properties in (Beck, 2017, Theorem 10.15) hold for Algorithms 1 and 2 applied to $u_{\eta,\mu}$ and R under the assumptions of Theorem 7. By applying these observation to (Beck, 2017, Theorem 10.15), we obtain the following convergence result.

Theorem 17 (Convergence of Algorithms 3 and 5)**.** *Let the assumptions of Theorem 7 hold, and let $\Phi_{\eta,\mu}$ be as defined in (2.5.1). Let $\{(\tilde{\beta}^k,\tilde{\gamma}^k)\}$ be a sequence generated by either Algorithm 3 or 5 with parameters $\theta \in (0,1)$, $\tau \in (0,1)$, $\eta > 0$, $\mu > 0$, $\epsilon_{Tol} = 0$, $t_0 > 0$, and $\gamma_{max} > \tilde{\gamma}^0$. Then, given $\zeta > u_{\eta,\mu}(\tilde{\beta}^0,\tilde{\gamma}^0)$ there is an $L_{(\eta,\mu,\gamma_{max},\zeta)} > 0$ such that $\nabla u_{\eta,\mu}$ is $L_{(\eta,\mu,\gamma_{max},\zeta)}$-smooth*

over $\widetilde{\mathcal{V}}((\eta, \mu, \gamma_{max}, \zeta, 1)$. *In Algorithm 5, replace* $L_{\eta,\mu}$ *with* $L_{(\eta,\mu,\gamma_{max},\zeta)}$ *and set*

$$M := \begin{cases} \alpha(1 - \alpha\frac{L_{(\eta,\mu,\gamma_{max},\zeta)}}{2}), & \text{in Algorithm 5,} \\ \frac{2t_0\theta^2(1-\tau)}{\max\{2\theta(1-\tau), t_0 L_{(\eta,\mu,\gamma_{max},\zeta)}\}}, & \text{in Algorithm 3,} \end{cases} \quad \text{and}$$

$$r := \begin{cases} \alpha, & \text{in Algorithm 5,} \\ t_0, & \text{in Algorithm 3.} \end{cases}$$

Then either $\tilde{\gamma}^k > \gamma_{max}$ *after a finite number of iterations and the algorithms terminate, or the following hold:*

1. *The sequence* $\Phi_{\eta,\mu}(\tilde{\beta}^k, \tilde{\gamma}^k)$ *is nondecreasing. In addition,*

$$\Phi_{\eta,\mu}((\tilde{\beta}^k, \tilde{\gamma}^k))\tilde{\beta}^{k+1}, \tilde{\gamma}^{k+1}) < \Phi_{\eta,\mu}(\tilde{\beta}^k, \tilde{\gamma}^k)$$

   if and only if $(\tilde{\beta}^k, \tilde{\gamma}^k)$ *is not a stationary point of* (2.3.14).

2. $\left\|(\tilde{\beta}^k, \tilde{\gamma}^k) - \text{prox}_{r\widetilde{R}}((\tilde{\beta}^k, \tilde{\gamma}^k) - r\nabla u_{\eta,\mu}(\tilde{\beta}^k, \tilde{\gamma}^k))\right\| \to 0$ *with*

$$\min_{i=0,1,\ldots,k} \left\|(\tilde{\beta}^k, \tilde{\gamma}^k) - \text{prox}_{r\widetilde{R}}((\tilde{\beta}^k, \tilde{\gamma}^k) - r\nabla u_{\eta,\mu}(\tilde{\beta}^k, \tilde{\gamma}^k))\right\| \leq \frac{\sqrt{\Phi_{\eta,\mu}(\tilde{\beta}^0, \tilde{\gamma}^0) - \Phi_{\eta,\mu}^{\mathsf{OPT}}}}{\sqrt{M(k+1)}},$$

   where $\Phi_{\eta,\mu}^{\mathsf{OPT}} := \inf \Phi_{\eta,\mu}$.

3. *All limit points of the sequence* $\{(\tilde{\beta}^k, \tilde{\gamma}^k)\}$ *are stationary points of problem* (2.3.14).

Proof. As observed prior to the statement of the theorem, both Algorithm 3 and 5 behave as if they were applied to the the functions $\hat{u}_{\eta,\mu}$ and $\widehat{R}$ defined in (2.5.26). It was also shown that the functions $\hat{u}_{\eta,\mu}$ and $\widehat{R}$ satisfy the Assumptions (A)-(C) required by (Beck, 2017, Theorem 10.15). Hence, the consequences of (Beck, 2017, Theorem 10.15) hold. By translating the notion of (Beck, 2017, Theorem 10.15) to that of this paper, we obtain the result. □

2.5.4 A Hybrid Algorithms for Feature Selection in Mixed Effects Models

In the previous section we established the convergence properties of the PGD algorithm applied to the function $\Phi_{\eta,\mu}$ for fixed values of η and μ. In subsection 2.5.2, two consistency results are established for the relaxed problem (2.3.14). Theorem 8 shows that, for fixed $\mu \geq 0$, every limit point of solutions to (2.3.14) as $\eta \uparrow \infty$ is a solution to (2.5.10), while Theorem 9 tells us that every limit point of solutions (2.5.10) as $\mu \downarrow 0$ is a solution to the variable selection problem (2.3.1). These results suggest a range of numerical approachs to obtaining approximate solutions to the target problem (2.3.1). The issue of foremost concern is the method for approximating solutions to (2.3.15) since the accuracy in this approximation determines the accuracy in both

$u_{\eta,\mu}$ and $\nabla u_{\eta,\mu}$. To address this concern, we view the algorithm from an interior point perspective where every point on the *central path* is a solution to the optimization problem (2.3.15) defining $u_{\eta,\mu}$ for the associated value of the homotopy parameter μ. An approximate solution is then considered acceptable if it is sufficiently close to the central path where proximity to the central path is measured in terms of the notion of the *neighborhood* of the central path , e.g. see Wright (1997b). Due to the convexity of the optimization problems (2.3.15), this is an efficient algorithm for approximating $u_{\eta,\mu}$ to high accuracy.

The next issue we addressed is the method for initializing and adjusting the parameter η. This is particularly significant since the initial value of η must be chosen to assure the convexity of the problems in (2.3.15). Lemma 5 gives us guidance in this regard, but the necessary computations to obtain a lower bound on η can be arduous, and, in general, produce a wildly pessimistic lower bound. For this reason, we take a somewhat different approach by proposing a variable metric strategy for solving the optimization problems in (2.3.15). In this approach, we replace the Hessian matrix $\nabla^2 \mathcal{L}_{\eta,\mu}(\beta,\gamma)$ in the Newton equation

$$G_{\eta,\mu}((\beta,\gamma,v),(\tilde{\beta},\tilde{\gamma})) + \nabla G_{\eta,\mu}((\beta,\gamma,v),(\tilde{\beta},\tilde{\gamma}))[dv, d\beta, d\gamma] = 0$$

by the positive semi-definite approximation

$$\nabla^2 \mathcal{L}(\beta,\gamma) \approx \sum_{i=1}^m S_i^T \begin{bmatrix} X_i^T \\ -Z_i^T \end{bmatrix} \Omega_i(\gamma)^{-1} \begin{bmatrix} X_i & -Z_i \end{bmatrix} S_i$$

which is motivated by the expression for $\nabla^2 \mathcal{L}_{\eta,\mu}(\beta,\gamma)$ given in (2.5.2). That is, we simply drop the negative semi-definite term $-\sum_{i=1}^m \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2}(Z_i^T \Omega_i(\gamma)^{-1} Z_i)^{\circ 2} \end{bmatrix}$. With this modification, the subproblems we solve are strongly convex for all $\eta > 0$. Consequently, the problem of initializing η is less problematic. Our numerical experiments indicate that the performance of the algorithm is robust with respect to η. For this reason, we choose an initial value for η and then leave it fixed over all iterations. Our method for choosing η is described in (Sholokhov et al., 2022a, Section 4, Figure 5). Briefly, we maximize the Baysian Information Criterion (BIC) over a grid of values for η. The resulting BIC response curve shows that the method is robust with respect to the choice of η and choosing $\eta \in [1, 10]$ yields accurate solutions for our selected test problems. Once η is fixed the PGD algorithm can be applied to solve the problem (2.3.14) for decreasing values of μ.

2.6 Software Implementation

To ensure reproducibility of this research, all new algorithms have been implemented as a part of the `pysr3`[11] library (Sholokhov et al. (2023)). This library implements functionality for fitting linear mixed models and selecting covariates. The user interface was designed to be fully compliant with the standards[12] of `sklearn` library to minimize learning time.

[11] Available at https://github.com/aksholokhov/pysr3

[12] https://scikit-learn.org/stable/developers/develop.html

The baseline optimization method of PySR3 is proximal gradient descent (PGD). Each regularizer included in PySR3 can also be used in its relaxed SR3 form. More information about the structure of the library can be found in the documentation[13] as well as in the package's companion paper (Sholokhov et al. (2023)).

2.7 Discussion

In this paper, we developed and implemented a first-order variable selection framework for LMEs that handles convex and nonconvex regularizers. We also showed that the MSR3 relaxation (2.3.7) improves the covariates selection accuracy of a wide group of popular sparsity-promoting regularizers. The fact that the relaxation improves accuracy, rather than just serving as a means to numertical efficiency, is very interesting and deserves future study.

Since the LME relaxation does not have a closed form, we used an interior method to evaluate the requisite value function. We also developed a more efficient version of the algorithm (MSR3-Fast) that interleaved interior point iterations with updates of the auxiliary variables, and this method was chosen for the open source library `pysr3`. Numerical experiments on synthetic data showed that the MSR3 approach for variable selection extends regions of hyper-parameter values where the highest accuracy is achieved, making it easier for information criteria to select the best model. The variable selection library for the accelerated method MSR3-Fast is much faster than currently available software, and allows the MSR3 approach to be easily applied to a range of regularizers that have computationally efficient prox operators.

The main analytic limitations of the proposed method stem from a lack of an analytical representation of the value function in the MSR3 relaxation for LMEs (2.3.7). However, the MSR3 framework (Algorithm 2) incorporates global variational information about the likelihood $\mathcal{L}$ into the PGD algorithm whereas the standard application of the PGD algorithm (Algorithm 1) only uses a local linear approximation to $\mathcal{L}$ at each iteration. This difference reveals itself in both the increased speed and accurate of the MSR3 approach on this class of problems. In contrast to SR3 in linear regression settings, where the CG method can be efficiently used to evaluate the value function (see e.g. Baraldi et al. (2019)), the nonlinear optimization problem required for LMEs is more difficult. Although the use of Hessian information makes each iteration computationally efficient, it limits the size of the problems to which the method can be applied. On the other hand, switching to first-order methods for the inner problem inside the relaxation may be prohibitively slow. A potential path to balance these limitations is to develop efficient upper-bounding models for the value function that can be evaluated more efficiently.

The suggested methodology can be expanded to a wider class of models. In particular, one can extend MSR3 to the setting of non-linear mixed-effect models or generalized linear mixed models, which are known to be challenging setups for covariate selection tasks. Both of these

[13] https://aksholokhov.github.io/pysr3/

problem classes face require optimizing highly nonlinear objective functions that arise when we consider marginal likelihoods. The SR3 approach may allow new avenues for more efficient strategies, analogous to what was done here for LMEs.

Chapter 3

PINODE: Physics-Informed Neural Ordinary Differential Equations

3.1 Introduction

Edit: Write a big-picture intro to machine learning for physics

Edit: Replace mentions of paper with work

3.1.1 Related Work

Edit: Split original intro into prior works paragraphs: DL for Ph, ROMs, PIML

Edit: Add physics-informed ML overview (PINN, DeepONet, NFO)

Forecasting the behavior of a large-scale real-world system directly from first principles often requires solving highly-nonlinear governing equations such as high-dimensional ordinary differential equations (ODEs) or partial differential equations (PDEs). High-fidelity simulations of such dynamical systems can become intractable, especially if an online control algorithm requires multiple forecasts per second using a low-powered embedded device Rowley and Dawson (2017); Lucia et al. (2004); Benner et al. (2015). A situation like this arises, for example, when a smart heating, ventilation, and air conditioning (HVAC) system attempts to optimize the temperature distribution of the air in a room using only partial measurements Farahmand et al. (2016); Nabi et al. (2022). At the time of writing this paper, such systems are incapable of real-time complex simulations, but they can already run low-dimensional pre-trained models, which invites the development of high-quality reduced order models (ROMs) Otterness et al. (2017). Therefore, ROMs are essential for enabling the design optimization, uncertainty propagation, predictive modeling, and control for such dynamical systems Brunton and Kutz (2022); Kutz et al. (2016); Rowley and Dawson (2017); Jones et al. (2020)

In order to enable control of high dimensional dynamical systems, a ROM training method needs to identify a low-dimensional manifold along with dynamics on the manifold that together yield high-accuracy predictions and long-term stability Ahmed et al. (2021); Noack et al. (2011). Most traditional ROMs are projection-based, e.g. dynamic mode decomposition (DMD) Kutz et al. (2016); Tu et al. (2013) and proper orthogonal decomposition (POD) Holmes et al. (2012), which transform the trajectories of a high-dimensional dynamical system into a suitable, and in some sense optimal, low-dimensional subspace. This projection leads to truncation of higher order modes and parametric uncertainties, which result in large prediction errors over time due to the deterioration of the basis functions (spatial modes) Benner et al. (2015). One challenge for POD methods is their intrusive nature, i.e. requiring access to the solver codes. To overcome this, operator inference approaches Qian et al. (2020); Peherstorfer and Willcox (2016) utilize SVD-based model reduction and exploit lifting to fit the latent space dynamics data into polynomial, typically quadratic, models. These models, however, are (i) limited in representation power (up to quadratic, e.g. for lift and learn approach) and (ii) require a custom-tailored SVD-based optimization technique.

In a thrust to overcome these challenges, significant effort has been invested into developing autoencoder-based reduced-order models, as a popular nonlinear ROM technique, which can yield both accurate and stable ROMs Lee and Carlberg (2020); Gin et al. (2021); Champion et al. (2019); Kim et al. (2019). In practice, however, autoencoder-based ROMs require datasets that densely cover a hypothetical infinite dimensional phase portrait of the dynamical system. Moreover, the large demand for training data significantly limits the use of such models in physics applications where the data can be expensive to obtain.

Another severe challenge of utilizing ROMs comes from their poor out-of-distribution performance Fries et al. (2022); Cranmer et al. (2020); Gin et al. (2021), especially when it is fundamentally impossible for a practitioner to obtain data that covers the entire distribution of possible data inputs. For example, in HVAC applications, one may collect data from a room with two windows but not from a room for every possible number of windows. In atmospheric LiDAR applications, we may conduct experiments on a certain terrain but we can never conduct experiments on all sorts of terrains Nabi et al. (2020). In such situations embedding the knowledge of physics into a model becomes necessary to improve the extrapolation performance, and for which several approaches have recently been proposed. For instance, the seminal works Bongard and Lipson (2007); Schmidt and Lipson (2009) have tried to determine the underlying structure of a nonlinear dynamical system from data using symbolic regression. Recently, Cranmer et al. Cranmer et al. (2020) employed symbolic regression in conjunction with graph neural network (GNN), while encouraging sparse latent representation, to extract explicit physical relations. They showed that the symbolic expressions extracted from the GNN generalized to out-of-distribution data better than the GNN itself. However, symbolic regression also suffers from excessive computational costs, and may be prone to overfitting.

Another example of incorporating physics in ROMs is the use of parametric models at the latent space, e.g. by using the sparse identification of nonlinear dynamics (SINDy) Brunton et al. (2016); Champion et al. (2019). For instance, Fries et al. (2022); He et al. (2022) used a chain-rule based loss that ties latent-space derivatives to the observable-space derivatives for simultaneous training of the auto-encoder and the latent dynamics. However, such loss is highly sensitive to noise in the data, especially when evaluating time-derivatives with finite differences is requiredDelahunt and Kutz (2022). Collocation-based enforcement of the physics, i.e. projection of the candidate functions in the governing equations to enforce the chain rule instead of finite difference, could address such numerical difficulties. Recently, Liu et al. Liu et al. (2022) used an auto-encoder architecture and Koopman theory to demonstrate that combining autoencoders with enforcing linear dynamics in the latent space may result in an interpretable ROM. However, linearity may not be expressive enough for complex dynamics with multiple basins of attraction Page and Kerswell (2019). Finally, recent works on NeuralODE (NODE) Chen et al. (2018b); Rackauckas et al. (2020) show a way to fit an arbitrary non-linear model (e.g. a network) as a latent space dynamics model, significantly extending the set of models for the latent dynamics that one can train efficiently.

In this paper, we employ autoencoders to perform nonlinear model reduction along with NODE

in the latent space to model complex and nonlinear dynamics. We choose Neural ODEs in the latent space dynamics representation because of their ability to model highly non-linear dynamics, which is especially important when applications limit the size of the latent space dimension. Our goal is to reduce the demand for training data and improve the overall forecasting stability under challenging training conditions. To that end, we build on ideas from classical collocation methods of numerical analysis to embed knowledge from a known governing equation into the latent-space dynamics of a ROM, as described in Section 2. In Section 3, we show that the addition of our physics-informed loss allows for exceptional data supply strategies that improves the performance of ROMs in data-scarce settings, where training high-quality data-driven models is impossible. We demonstrate that such an approach not only reduces the need for large training data-sets and produces highly-accurate and long-term stable models, but also leads to the discovery of more compact latent spaces, which is especially important for applications in compressed sensing and control.

Edit: Add Koopman example to the intro

Edit: Add compressive sensing application to the intro

3.2 Methods

Reduced-Order Model with Non-Linear Latent Dynamics We consider an autonomous dynamical system on a finite space $\mathcal{X} \subseteq \mathbb{R}^n$ given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t)). \tag{3.2.1}$$

In real-world applications, it is often expensive to solve equation (3.2.1) directly because $x(t)$ can be very high-dimensional. However, a variety of works provided both theoretical Holmes et al. (2012) and empirical Noack et al. (2011); Chen et al. (2021) evidence that many physical systems evolve on a manifold $\mathcal{Z} \subseteq \mathbb{R}^m$ of a lower dimension $m << n$. In that space, the dynamics evolve according to a (generally unknown) function $\boldsymbol{h}(\boldsymbol{z})$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{z}(t) = \boldsymbol{h}(\boldsymbol{z}(t)) \tag{3.2.2}$$

We call the space $\mathcal{X}$ an observable space, and $\mathcal{Z}$ a latent space. When an invertible mapping $\psi : \mathcal{Z} \to \mathcal{X}$ between the observable and the latent spaces is known, one can predict the dynamics of the system $\boldsymbol{x}$ at a future time T by projecting the initial condition $\boldsymbol{x}(0)$ into the latent space, integrating the dynamics in the latent space, and mapping the resulting trajectory back to the

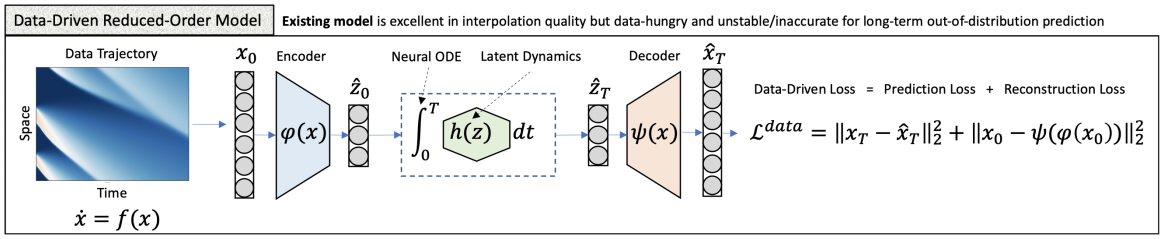


Figure 3.2.1: Illustration of the autoencoer structure with neural ODE in the latent space. The data-driven part of the loss function aims to minimize a sum of two objectives: the prediction loss and the reconstruction loss. The prediction loss minimizes the difference between the data trajectories and their model predictions to ensure temporal consistency of the latent space dynamics. The reconstruction loss ensures accurate reconstruction of individual snapshots, ensuring that the autoencoder behaves as an invertible mapping on all snapshots.

observable space:

$$\begin{aligned} \boldsymbol{z}(0) &= \psi^{-1}(\boldsymbol{x}(0)) \\ \boldsymbol{z}(T) &= \boldsymbol{z}(0) + \int_0^T \boldsymbol{h}(\boldsymbol{z}(t))dt \\ \boldsymbol{x}(T) &= \psi(\boldsymbol{z}(T)) \end{aligned} \tag{3.2.3}$$

When $m << n$ we refer to the triplet $(\psi, \psi^{-1}, \boldsymbol{h})$ as a Reduced-Order Model (ROM) of $\boldsymbol{f}$. It is often the case that for a given system $\boldsymbol{f}$, there exists no ROM $(\psi, \psi^{-1}, \boldsymbol{h})$ such that the relation (3.2.3) holds exactly. In this case, we seek an *approximation* ROM $(\psi_{\theta^*}, \phi_{\theta^*}, h_{\theta^*})$ that minimizes the difference between the data $x(t)$ and the prediction $\hat{x}(t)$ over a chosen class of models $(\psi_\theta, \phi_\theta, h_\theta)$ parameterized by θ.

Multiple real-world applications necessitate using ROMs instead of integrating the relation (3.2.1) directly. For example, integrating (3.2.1) may be computationally intractable especially on platforms with limited computing capability such as embedded and autonomous devices. For instance, in an HVAC system, solving (3.2.1) means solving a Navier-Stokes equation on a fine grid in real time, which exceeds the computing capabilities of current-generation appliances. On the other hand, integrating (3.2.3) may be cheap when $m << n$. Finally, even when solving (3.2.1) is possible in real time (e.g. by utilizing a remote cluster), executing control over the resulting model, which is an end-goal for an HVAC system, may still be intractable. Indeed, executing control requires *multiple* evaluations of (3.2.1) for *each* iteration of control even for the most efficient algorithms known to date Duriez et al. (2017).

Architecture In this work we model ψ, ψ^{-1}, and $\boldsymbol{h}$ with fully-connected neural networks ψ_θ, ϕ_θ, and h_θ, respectively. Specifically, the pair (ψ, ψ^{-1}) is modelled with an auto-encoder $(\psi_\theta, \phi_\theta)$, and $\boldsymbol{h}$ is modelled with a fully-connected network h_θ. Figure 3.2.1 visualizes the architecture of the model.

Data-Driven Loss Similar to prior works Takeishi et al. (2017); Morton et al. (2019); Gin et al. (2021), we define a *data-driven loss* $\mathcal{L}_{data}$ as a sum of reconstruction and prediction losses. The former ensures that ϕ_θ and ψ_θ are inverse mappings of each other, whereas the latter matches the model's predictions to the available data, as illustrated on Figure 3.2.1.

Formally, for a given set of trajectories $\boldsymbol{x}_i$, $i \in [1 \dots k]$, where each trajectory $\boldsymbol{x}_i \in \mathbb{R}^{n \times p}$ is a set of p snapshots that correspond to the recorded states of the system for p time-steps, t_j, $j \in [1, \dots, p]$, the loss function $\mathcal{L}_\theta^{data}$ is defined as:

$$\mathcal{L}_\theta^{data} = \frac{1}{2\sigma^2} \sum_{i=1}^{k} \left[\frac{\omega_1}{p} \sum_{j=1}^{p} \left\| \boldsymbol{x_i}(t_j) - \psi_\theta(\phi_\theta(\boldsymbol{x_i}(t_j))) \right\|^2 + \right. \tag{3.2.4}$$

$$\left. + \frac{\omega_2}{p} \sum_{j=1}^{p} \left\| \psi_\theta \left(\phi_\theta(\boldsymbol{x_i}(t_1)) + \int_{t_1}^{t_j} h(z(t)) dt \right) - \boldsymbol{x_i}(t_j) \right\|^2 \right] \tag{3.2.5}$$

where σ is the standard deviation of the observation noise. We note that each trajectory $\boldsymbol{x}_i$ may be captured over its own time-frame and may use a distinct, possibly non-uniform, step-size, in which case the loss function should be modified accordingly[1]. To simplify the notation, without loss of generality, in the rest of the paper we assume that all trajectories are recorded over the same time-frame with the same uniform step-size. To forecast the behavior of the system in the latent space, we apply the technique of Neural Ordinary Differential Equations (Neural ODEs or NODEs)Chen et al. (2018b), which utilizes the adjoint sensitivity method to back-propagate the gradients through the integral in (3.2.4). Neural ODEs have demonstrated a better ability to model highly non-linear dynamics compared to linear models when the dimensionality of the dynamics variable is limited. This is especially useful in applications where the size of the latent space dimension needs to be small Lee and Carlberg (2020); Gin et al. (2021); Champion et al. (2019); Kim et al. (2019).

Physics-Informed Loss In their recent work, Liu et al. Liu et al. (2022) proposed a method for utilizing knowledge of the governing equations $d\boldsymbol{x}/dt = \boldsymbol{f}(\boldsymbol{x})$ as a finite-dimensional approximation of Koopman eigenfunctions for linear latent dynamics. To extend this approach to the non-linear regime, we note that for a true mapping ϕ the following holds:

$$\frac{d\boldsymbol{z}(\boldsymbol{x}(t))}{dt} = \frac{d\boldsymbol{z}}{d\boldsymbol{x}} \frac{d\boldsymbol{x}}{dt} = \nabla \phi(\boldsymbol{x}(t))^T \boldsymbol{f}(\boldsymbol{x}(t)) \tag{3.2.6}$$

On the other hand, by the definition of ψ and $\boldsymbol{h}$ we have that

$$\frac{d\boldsymbol{z}(\boldsymbol{x}(t))}{dt} = \boldsymbol{h}(\phi(\boldsymbol{x}(t)) \tag{3.2.7}$$

Combining Equations (3.2.6) and (3.2.7) we get that

$$\boldsymbol{h}(\phi(\boldsymbol{x}(t)) = \nabla \phi(\boldsymbol{x})^T \boldsymbol{f}(\boldsymbol{x}) \tag{3.2.8}$$

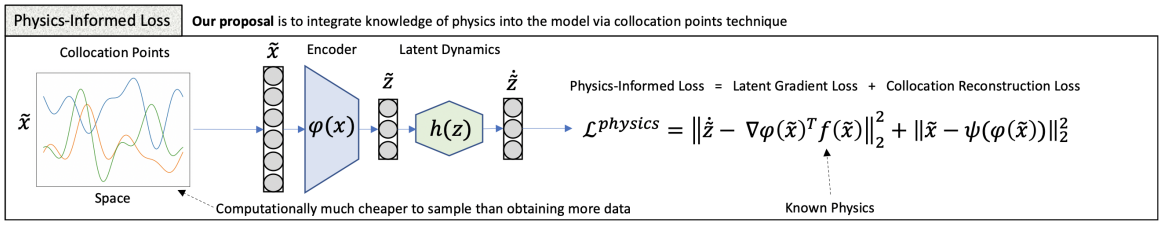


Figure 3.2.2: The physics-informed loss function compares gradient fields in the current latent space with what a correctly-learned field should be in this latent space on set of collocation points.

Equation (3.2.8) links the dynamics $\boldsymbol{h}(\boldsymbol{z})$ and the encoder $\phi(\boldsymbol{x})$ with the known equation $\boldsymbol{f}(\boldsymbol{x})$ and is true for all $z \in \mathcal{Z}$ and $x \in \mathcal{X}$. Hence, as shown on Figure 3.2.2, knowledge of $\boldsymbol{f}$ can be assimilated into the model by evaluating Equation (3.2.8) on a set of N carefully sampled points $\bar{\boldsymbol{x}}_i \in \mathcal{X}$, $i \in [1, \ldots, N]$:

$$\mathcal{L}_\theta^{physics} = \sum_{i=1}^{N} \left[\frac{\omega_3}{N} \left\| h_\theta(\phi_\theta(\bar{\boldsymbol{x}}_i)) - \nabla\phi_\theta(\bar{\boldsymbol{x}}_i)\boldsymbol{f}(\bar{\boldsymbol{x}}_i) \right\|^2 + \frac{\omega_4}{N} \left\| \bar{\boldsymbol{x}}_i - \psi_\theta(\phi_\theta(\bar{\boldsymbol{x}}_i)) \right\| \right] \tag{3.2.9}$$

We refer to the points $\bar{\boldsymbol{x}}_i$ as *collocation points*.

Collocation Points We define a collocation as pair $(\bar{\boldsymbol{x}}, \boldsymbol{f}(\bar{\boldsymbol{x}}))$. collocation points are samples from the space $\mathcal{X} \times Im_f(\mathcal{X})$, and they should satisfy three conditions, ordered by importance:

1. **Simplicity**: $\boldsymbol{f}(\bar{\boldsymbol{x}}_j)$ should be computationally cheap to evaluate. It is especially important for PDE systems, where $\boldsymbol{f}$ may involve high-order derivatives.
2. **Representativeness**: $\bar{\boldsymbol{x}}_j$ should cover the space of states where one aims to improve the model's performance or stability. Collocation points that a model might encounter and that are not represented by data snapshots are the best candidates.
3. **Feasibility**: $\bar{\boldsymbol{x}}_j \in \mathcal{X}$. In other words, x_j should be an attainable state of the system. Collocation points outside of $\mathcal{X}$ may downgrade the performance of the autoencoder by forcing it to be an invertible function on a domain outside of $\mathcal{X}$.

Thus, an optimal sampling procedure for collocation points $\bar{\boldsymbol{x}}_j$ is domain-specific and should be designed given a particular system $\boldsymbol{f}$ and available data $\boldsymbol{x}_i$. We show examples of how these conditions can be implemented for real systems in the following sections.

The above definition of collocation points is not to be confused with a classic notion of collocation points for finding numerical solutions for differential equations Fornberg (1998); Trefethen and Bau (2022). The classic notion refers to a set of points in time $[t_0, t_0 + c_1 h, t_0 + c_2 h, \ldots, t_0 + h]$,

[1]The implementation is affected only in evaluating the integral in (3.2.4). This part is handled by `torchdiffeq` Chen et al. (2018a) library, which supports non-uniform time-frames within a batch

$0 < c1 < c2 < \cdots < 1$ which are chosen to obtain an optimal local interpolant of a solution of a differential equation for a time-period between t_0 and $t_0 + h$. For example, s collocation points for Runge-Kutta methods are defined to provide an optimal Gauss-Legendre interpolant of order s; the coefficients $c_1, \ldots, c_s$ come from a respective Butcher table. In contrast, we define collocation points as pairs $(\bar{\boldsymbol{x}}, \boldsymbol{f}(\bar{\boldsymbol{x}}))$ which are examples of mapping $x \to f(x)$. Our definition is built around solving an *inverse* problem of approximating $\dot{x} = f(x)$ with $f_\theta(x)$ and follows a recent work Liu et al. (2022) which develops upon a definition from Raissi and Karniadakis (2018) with the difference being the sample space: instead of sampling from the spatiotemporal domain we sample them from an appropriate function space.

Combined Loss Function We train the model by optimizing a sum of the physics-informed loss (3.2.9) and the data-driven loss (3.2.4):

$$\min_\theta \left[\mathcal{L}_\theta^{physics} + \mathcal{L}_\theta^{data} \right] \tag{3.2.10}$$

When $\omega_1 = \omega_2 = 0$ we have $\mathcal{L}_\theta^{data} = 0$, so we say that the model is (purely) ***Physics-Informed***. Similarly, when $\omega_3 = \omega_4 = 0$ we have $\mathcal{L}_\theta^{physics} = 0$ and we say that the model is (purely) ***Data-Driven***. When $\omega_i \neq 0$, $\forall i$, we say that the model is ***Hybrid***.

The coefficients ω_i are hyper-parameters which need to be tuned using a validation dataset. However, in all experiments of this paper we set ω_i to be either 0 or 1, and we balance $\mathcal{L}_\theta^{physics}$ and $\mathcal{L}_\theta^{data}$ the choice of samples in a batch of training data. Specifically, we set the number of collocation points per batch N_{batch} to be equal to the number of trajectories per batch k_{batch} times the number of time-steps$T$: $N_{batch} = Tk_{batch}$. In this way both $\mathcal{L}_\theta^{physics}$ and $\mathcal{L}_\theta^{data}$ represent the loss for Tk_{batch} snapshots of the system, providing on average a similar contribution of information to the overall loss function. More laborious approaches of hyper-parameter tuning did not yield sufficient systematic advantage to justify the labour compared to this simple strategy.

We use a `pytorch` Paszke et al. (2019a) implementation of the Adam algorithm Kingma and Ba (2014) for optimization. To evaluate $\nabla_\theta \mathcal{L}_\theta^{physics}$ and $\nabla_\theta \mathcal{L}_\theta^{data}$ we use `torchdiffeq` Chen et al. (2018a) – a `pytorch`-compatible implementation of the Neural ODE framework.

To the best of our knowledge, this is the first framework that combines non-linear latent-dynamics (Neural ODE), autoencoders, and a physics-informed loss term (3.2.9). Thus, we call our framework *Physics-Informed Neural ODE*, or PINODE.

3.3 Experiments

The experiments section is organized as follows. First, to illustrate the ideas behind the framework we study its performance on a high-dimensional ODE – a lifted Duffing oscillator. We show how a non-linear latent dynamics $\boldsymbol{h}(\boldsymbol{z})$ overcomes the limitations of DMD and Koopman

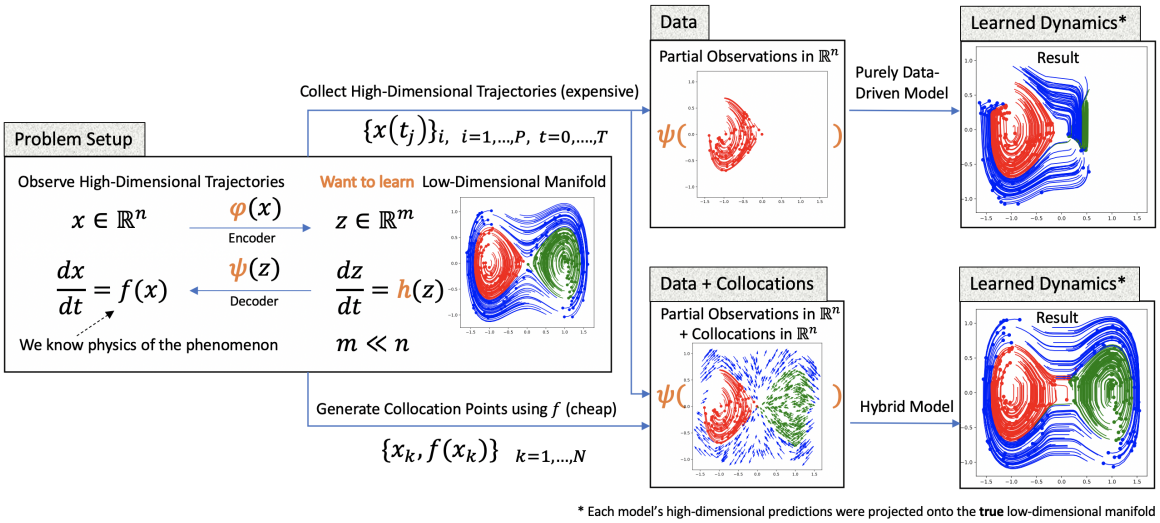


Figure 3.2.3: We use a toy example – a Lifted Duffing Oscillator – to show that it is possible to "fill the gaps" in data with collocation points. Specifically, the Hybrid model is able to learn the dynamics of two additional basins of attraction that were not represented in the dataset. As shown in the top-rightmost frame, without the collocation points the model does not infer the dynamics in the unseen regions correctly.

networks from Liu et al. (2022) by handling multiple basins of attraction within one model. We also show that using physics-informed loss is sufficient for reconstructing the behaviour for basins of attraction that are not represented by the data. Finally, we demonstrate that a purely data-driven model may be highly-accurate in the short-term and highly unstable in the long-term, even when the data is abundant, and show that the physics-informed approach improves long-term stability of such models by multiple orders of magnitude.

Next, we study the framework's performance on Burgers' equation. We show that (i) the non-linear latent dynamics model yields more compact latent space representations than its linear counterpart for the same accuracy; (ii) the compact latent space representations allow for more stable long-term predictions; (iii) in the presence of significant noise in the data, the use of collocation points improves stability by providing an extra source of information that is noise-free, and (iv) in certain scenarios, training *only* on collocation points yields *better* models than training on data, even when a vast amount of data is available. The last observation shows that the contribution of the physics-informed loss (3.2.9) may surpass that of the data-based loss (3.2.4), especially when the data is severely limited or noisy.

3.3.1 Lifted Duffing Oscillator

A Duffing oscillator is a dynamical system $d\boldsymbol{z}/dt = \boldsymbol{h}(\boldsymbol{z})$ such that

$$\begin{aligned} \frac{dz_1}{dt} &= z_2 \\ \frac{dz_2}{dt} &= z_1 - z_1^3 \end{aligned} \tag{3.3.1}$$

A phase portrait for 300 randomly sampled trajectories from this system is visualized on Figure 3.2.3, left frame. Depending on the total energy, each trajectory always stays in one of three regions: the left lobe, the right lobe, or the outer area, visualized in red, green, and blue, respectively. To create a synthetic high-dimensional system that retains this property, we lift the Duffing trajectories into a higher-dimensional space by applying an invertible transformation $\mathcal{A}(\boldsymbol{z})$:

$$\boldsymbol{x} := \mathcal{A}(\boldsymbol{z}) = A\boldsymbol{z}^3, \quad A \in \mathbb{R}^{128\times 2}, \quad A_{ij} \sim_{i.i.d.} \mathcal{N}(0,1) \tag{3.3.2}$$

Hence, for this system $z \in \mathcal{Z} = \mathbb{R}^2$ and $\boldsymbol{x} \in \mathcal{X} = \text{span}\{A_{:,1}, A_{:,2}\} \subseteq \mathbb{R}^{128}$. We treat $\mathcal{X}$ as an observable space, in which the dynamical system (3.3.1) obeys the following:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}) = \nabla((A^TA)^{-1}A^T\boldsymbol{x}^{1/3})^T\boldsymbol{h}((A^TA)^{-1}A^T\boldsymbol{x}^{1/3}) \tag{3.3.3}$$

Thus, we created a high-dimensional dynamical system with multiple basins of attraction for which the dynamics $\boldsymbol{f}$ are known.

For the experiment, we generate 6144 trajectories $\boldsymbol{x}_i$, $t = [0, 1]$, $\Delta t = 0.1$, all taken from the left lobe region (in red). We also sample 50000 collocation points $\bar{\boldsymbol{x}}_j$ from the right (green) and the outer (blue) regions each by sampling $\bar{\boldsymbol{z}}_j \in U([-3/2, 3/2] \times [-1, 1])$ and then applying the transformation (3.3.2). For this example the conditions for collocation points discussed in Section 3.2 are trivially satisfied.

We train two PINODE models: a Data-Driven model that only uses the trajectories, and a Hybrid model that uses both trajectories and collocation points. The models share the same architecture and training parameters that are detailed in Appendix A.8.1. After training, we invert the mapping (3.3.2) to project the models' high-dimensional predictions for unseen initial conditions onto the true low-dimensional manifold; those are visualized in Figure 3.2.3.

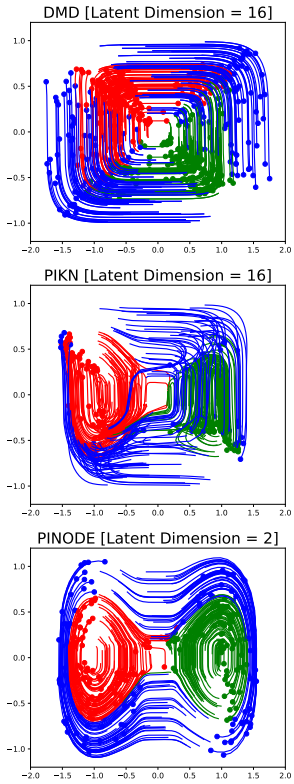


Figure 3.3.1: Non-linearity in the latent dynamics and the autoencoder employed in teh PINODE Hybrid model are important for accurate long-term extrapolation. The DMD model and PIKN Hybrid model were unable to extrapolate the dynamics from collocation points.

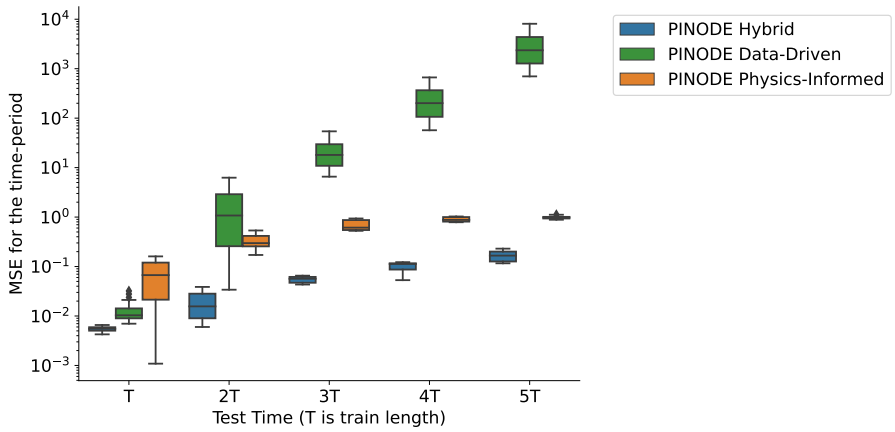


Figure 3.3.2: Box plots of the prediction error for three PINODE models: Data-Driven, Physics-Informed, and Hybrid. The time is measured in multiples of the training time period, i.e. $x = 3T$ refers the time-range between two and three training time-periods away.

We make two observations from the results displayed in Figure 3.2.3.
First, a purely data-driven model is unable to extrapolate outside
its training region using only the data from that region. This observation is consistent with the conclusions from related works Gin et al. (2021) that neural networks interpolate well but struggle with extrapolation tasks. Second, we see that collocation points provided enough extra information for the model to predict nearly perfectly in regions from which no trajectories were provided. This observation suggests that one can use collocation points to "cover the gaps" in data and improve the extrapolation accuracy of the model.

The ability of Neural ODE to model nonlinear dynamics in the latent space is demonstrated in Figure 3.3.1. The figure shows a comparison between the Hybrid PINODE model, the Hybrid PIKN model Liu et al. (2022), and DMD, all of which have been trained using the same dataset. PIKN differs from PINODE in that it uses linear latent dynamics $\frac{dz}{dt} = Lz$, where L is a finite-dimensional approximation of the Koopman operator, instead of a general non-linear dynamics operator $\frac{dz}{dt} = h_\theta(z)$. For PIKN, we set $z \in \mathbb{R}^{16}$, an 8 times expansion of the dimension of the true manifold. We observe in Figure 3.3.1 that PIKN is unable to extrapolate the dynamics to unseen areas correctly using the collocation points: eventually, all trajectories "collapse" onto the same attractor. It can also be seen that DMD shows even worse performance which could be attributed to its linear model reduction.

In the next experiment, we show that collocation points stabilize long-term predictions of the model even when data from all parts of the space are available. To illustrate, we generate a dataset of 6144 trajectories (2048 trajectories per red, green, and blue area) and 50000 collocation points uniformly distributed among all three lobes. We train three models: Data-Driven, Physics-Informed, and Hybrid versions of PINODE. The relative performance of the three models is evaluated in Figure 3.3.2, where the x-axis represents the test time-horizon as multiples of the training trajectory length T. The y-axis shows box plots of the prediction mean squared error (MSE) corresponding to 300 unseen trajectories within the specific period. For

example, $x = 2T$ represents the time-period $[2T, 3T)$, and the y-axis shows the distribution of the prediction errors within the period $[2T, 3T)$. Figure 3.3.2 shows that the performance of the Data-Driven model degrades quickly when the forecasting time-period increases despite its excellent performance when forecasting within its training time-period. The Physics-Informed model starts with modest performance over the training time horizon but maintains a stable performance when forecasting far ahead. The Hybrid model, in its turn, combines both near-term accuracy with long-term stability, yielding the best results over each time period.

3.3.2 Burgers' equation

We now study the performance of our framework on Burgers' equation with $[-\pi, \pi]$-periodic boundary conditions:

$$\begin{aligned} &u_t + uu_x = \nu u_{xx} \\ &u(-\pi, t) = u(\pi, t), \quad \forall t \in [0, T] \end{aligned} \tag{3.3.4}$$

where u_t, u_x, and u_{xx} represent partial derivatives in time, the first, and second spatial derivatives, respectively. Burgers' equation is a PDE occurring in applications in acoustics, gas and fluid dynamics, and traffic flows Burgers (1948). When ν is significantly smaller than one, the system exhibits strong non-linear behaviour and is called "advection-dominated", otherwise when ν is large the system is called "diffusion-dominated". In the case of the former, linear projection methods such as POD become inaccurate as the true solution space has a slow decaying Kolmogorov n-width, manifesting itself in slow decaying singular values Peherstorfer (2022). Therefore, in this section we focus on the advection-dominated Burgers' equation for which we set $\nu = 0.01$.

To generate trajectories, we discretize the spatial domain $[-\pi, \pi]$ into 128 grid-points, and solve Equation 3.3.4 for $t \in [0, 2]$ with $\Delta t = 0.1$ using a spectral solver Trefethen (2000). To generate a diverse set of initial conditions we sum the first 10 harmonic terms with random coefficients:

$$u(x, 0) = \frac{1}{10} \sum_{k=1}^{10} a_k \cos(kx) + b_k \sin((k+1)x), \quad a_k, b_k \sim \mathcal{N}(0, 1) \tag{3.3.5}$$

To generate collocation points we use the same family of functions as we used for the initial conditions in Equation (3.3.5), and additionally randomize the presence of individual frequencies in the sum:

$$\bar{u}(x) = \frac{1}{10} \sum_{k=1}^{10} p_k a_k \cos(kx) + q_k b_k \sin((k+1)x), \quad a_k, b_k \sim \mathcal{N}(0, 1), \quad p_k, q_k \sim Be(1/2). \tag{3.3.6}$$

We choose this family of collocation points to meet the conditions (3.2). First, this family is representative of the state space $\mathcal{X} \times Im_f(\mathcal{X})$ in the region of interest (moving wave-fronts). Second, (3.3.6) is a smooth set of functions that does not contain unattainable states. Finally, and more importantly, the values u_x and u_{xx} and, consequently u_t can be computed analytically, which makes it especially cheap to sample large numbers of collocation points.

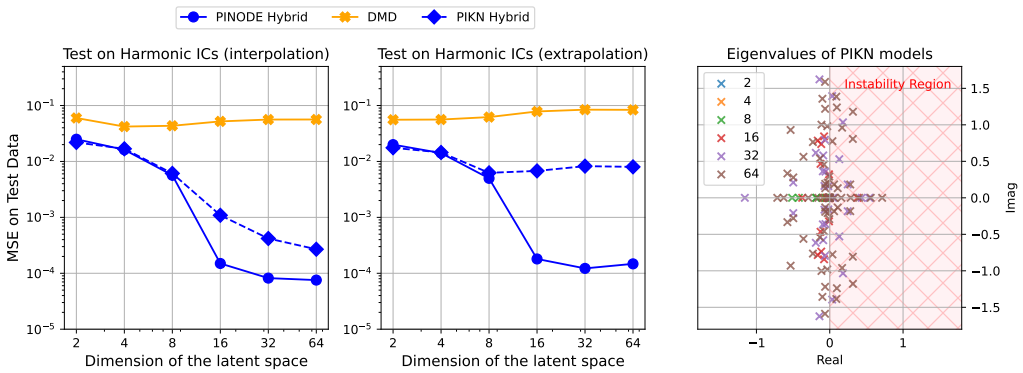


Figure 3.3.3: PINODE Hybrid model utilized the latent space dimension 5 times more efficiently in terms of MSE than PIKN Hybrid model when modelling low-viscosity (highly-nonlinear) Burgers' equation (left frame). The difference in performance grows to x100, when forecasting two times farther than the training period (central frame). PIKN suffers from long-term instability due to the presence of eigenvalues with positive real part in the latent dynamics matrix (right frame). In this frame we plot all the eigenvalues of the latent-space matrix for each PIKN model from frames 1-2. The legend in the right frame refers to the dimension of the latent space used by the corresponding PIKN model.

3.3.3 Compressibility of the Latent Space

In Section 3.3.1, we showed that a non-linear finite-dimensional latent dynamics model can be necessary for building a compact ROM for the high-dimensional lifted Duffing system. That is *not* necessarily the case for Burgers' equation since there exists the Cole-Hopf transformation that linearizes the dynamics for Burgers' equation. However, a latent-space non-linearity can, in principle, be utilized for finding a more compact latent space representation, or for increasing the forecast accuracy for a fixed latent space dimension. In this section, we demonstrate how PINODE can achieve both goals.

For this experiment we generate 16384 trajectories as described in (3.3.5). We also generate 100000 collocation points as described in (3.3.6). The purpose of using such a large amount of data is to allow the trained models to achieve the best performance for the specified latent space dimension. We evaluate the performance of the models on test data with two different time-frames: (1) same as that of training data (*interpolation*), and (2) two times longer than that of the training data (*extrapolation*). More details on the experimental setup are provided in Appendix (A.8.4).

In Figure 3.3.3, we compare the performance of the three models: DMD, PIKN Hybrid, and PINODE Hybrid. First, we notice that DMD does not perform well on the test data, despite achieving a training loss ($\sim 10^{-3}$). This observation is consistent with earlier works (Kalur et al. (2021); Kutz et al. (2016)); and illustrates well that a combination of a linear encoder and a linear latent dynamics operator may not be sufficient for modelling highly-nonlinear phenomena. Second, we notice that PINODE achieves better performance for a given latent space dimension

compared to PIKN. For instance, for $m = 16$ (Figure 3.3.3, left pane), PINODE achieves ~ 5 times lower mean squared error than PIKN, which achieves the same performance only when $m = 512$. More importantly, PINODE maintains a low prediction error over a longer-term horizon (extrapolation in time), which is not the case for PIKN (Figure 3.3.3, center pane). This is a consequence of the latent-dynamics matrix ($h(z) = Lz$) of PIKN having eigenvalues with positive real parts, which implies long-term instability (Figure 3.3.3, right pane). Although there has been progress in the literature Kojima and Okamoto (2022), further research is needed to understand (i) how to enforce stability constraints for PIKN, and (ii) why one does not need the same enforcement for PINODE to exhibit stable behaviour.

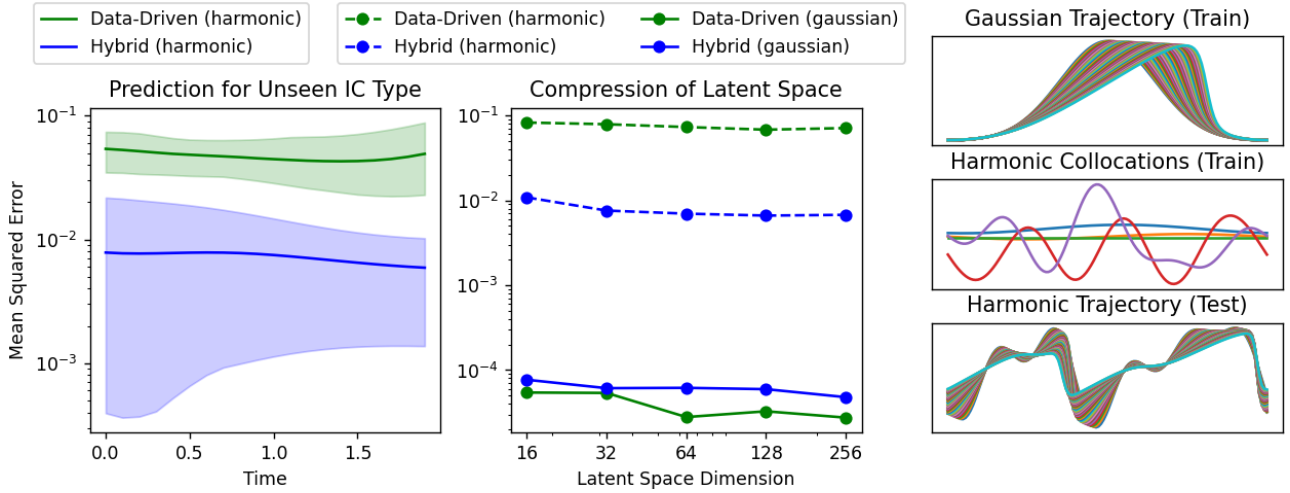


Figure 3.3.4: The right plots give examples of data snapshots used: trajectories with bell-curve ICs (top) and harmonic ICs (bottom). The middle-right pane shows harmonic collocations used by hybrid models in addition to the snapshots. The left plot compares the prediction errors (MSE) of two models, data-driven and hybrid with 128-dimensional latent spaces, on harmonic initial conditions that were not present in the trained data. The shaded regions represent 95% confidence intervals based on 100 test trajectories. The middle plot shows the prediction error (MSE) on each type of test data for a variety of models with different latent-space sizes. We see that the use of harmonic collocations significantly improves the model performance on unseen harmonic ICs without increasing the errors on bell-curve ICs.

However, we observe that PIKN benefits from using physics-informed loss as well. In particular, we show that one can use collocations to improve model's performance on types of initial conditions that are missing in the available training data. To illustrate that we train two models. The data-driven model only uses 1024 trajectories with bell-curve initial conditions (ICs) (we provide an example at the top-right frame of the Figure 3.3.4). The hybrid model additionally observes 80000 harmonic collocations formed by summing first 10 sinusoidal modes with random coefficients (Figure 3.3.4, middle-right frame), for which we evaluate u_t analytically using Equation 3.3.4. Next, we evaluate the performance of both models using unseen trajectories with both harmonic and bell-curve ICs, 100 trajectories each. We observe that the Hybrid model predicts the sinusoidal trajectories 10 times better than the data-driven one (Figure 3.3.4,

left frame, shown for the 128-dimensional latent-space model). Since neither models had any trajectories of that type in its training set we conclude that the difference in performance comes from using harmonic collocations. We also note that better performance of the hybrid model on harmonic ICs does not come at an expense of worse performance on bell-curve ICs, as shown in the central frame of Figure 3.3.4. This evidence suggests that one can improve a model's extrapolation power by supplementing its training with sufficiently diverse set of collocations, especially when additional simulations are expensive to obtain but the collocations are cheap to generate. The details of the network's architecture and training procedure are provided in the Appendix A.8.5.

3.3.4 Training in Low-Data Regime with Collocation Points

In the next experiment, we study the relative efficiency of using collocation points against using data in a low-data regime. It is frequently the case that only a small number of simulations (or measurements) can be obtained for a physical system of interest due to the computational, time, or budget constraints. We would like to compensate the lack of sufficient data with providing collocation points which are considerably cheaper to generate. In this section, we show that, when chosen appropriately, collocation points can be effectively used for training a model in the low-data regime, and their contribution to a model's accuracy may even surpass the contribution of the data.

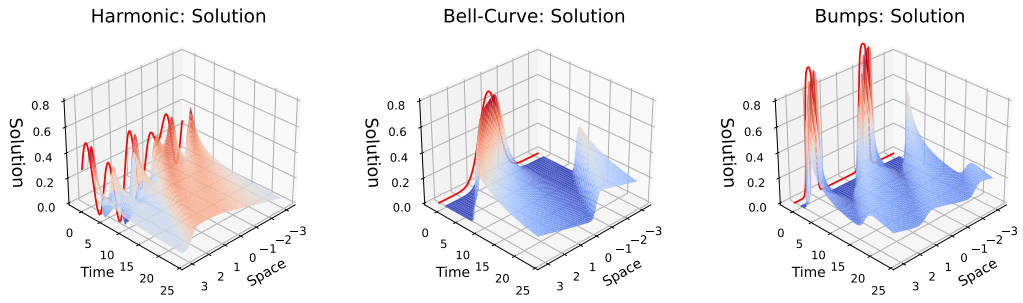


Figure 3.3.5: Examples of "harmonic", "bell-curve", and "bump" initial conditions, as well as the resulting solutions, in columns 1, 2, and 3, respectively.

To illustrate the trade-off between data and collocations, we train one model using varying combinations of the number of trajectories vs collocation points in their training datasets. To gauge the extrapolation power of our models, we use trajectories with three types of initial conditions: "harmonic", "bell-curve", and "bumps" (see Figure 3.3.5 for illustrations). We generate 1024 trajectories with "bumps" initial conditions for the training data, and use the harmonic family of initial conditions as described in (3.3.6) for generating the training collocations. We use two test datasets: (1) 100 trajectories with "bump" ICs to assess within-distributuion performance, left frame), and (2) a mix of trajectories with "bump", "bell-curve", and "harmonic" initial conditions, 100 trajectories each, to assess out-of-distribution performance. All test data

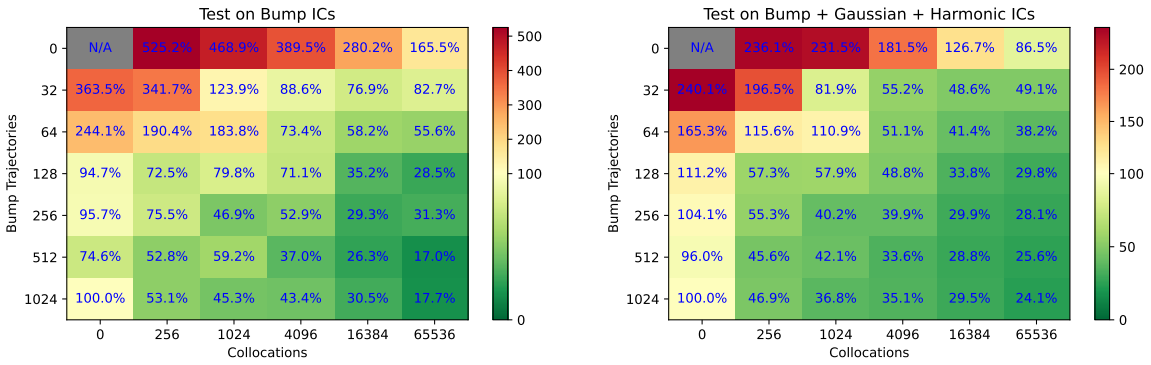


Figure 3.3.6: Comparison of the achievable MSE relative to the full data regime (1024 trajectories). When the data is scarce, collocations-based physics-informed loss improves the forecasting accuracy of ROMs by an average of 5 times lower MSE compared to the data-only regime, as shown in this experiment with Burgers' equation. When other types of initial conditions ("harmonic", "bell-curve") are used, the physics-only model (top-right corner of the right frame) outperformed the most data-rich model in our experiment (bottom-left corner).

trajectories are two times longer than the training trajectories. More details on the experimental setup are provided in Appendix A.8.6. Figure 3.3.6 presents the reconstruction MSE of the test datasets obtained from a PINODE models that were trained on varying combinations of trajectories and collocation points as a percentage of the MSE achievable by a PINODE model that was trained on the full 1024 trajectories alone (no collocations). The PINODE models all use a latent space dimension $m = 16$.

Figure 3.3.6 demonstrates that adding collocation points consistently improves the model performance in our experiments. Moreover, when a sufficient number of collocation points is added in training, the model with fewer training trajectories was always able to outperform the model that was trained on all the available trajectories and no collocations. On average, a collocation-aided model was *5 times better* at both within-distribution and out-of-distribution reconstruction relative to a purely data-driven version of the model. In addition, we noticed that a model that used only collocation points can perform better than a data-rich model, especially when predicting the dynamics of the unseen initial conditions (Figure 3.3.6, right pane, top-right vs bottom-left corner).

We also notice that the Hybrid models yield more stable and accurate predictions, relative to their purely data-driven counterparts, when forecasting far beyond the training time-period. In Figure 3.3.7 we visualize the predictions for a test IC for two models: Data-Driven model from the bottom-left corner of Figure (3.3.6), and a Hybrid model from the bottom-right corner of Figure (3.3.6). The red line separates the time-period of training from the time-period of forecasting. The hybrid model's errors stay below 10^{-2} even when forecasting 10 times farther than what it was trained on. In contrast, the Data-Driven model shows low errors within its training time-region but the forecast errors grow quickly when forecasting beyond that.

Finally, we observe that using collocation points can benefit other models, like DMD and PIKN.

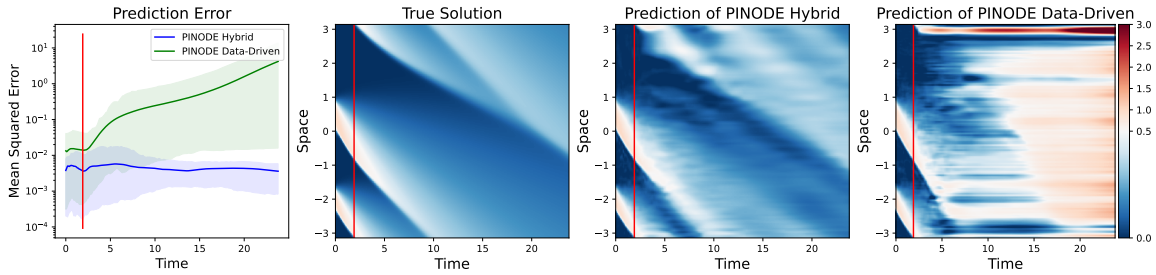


Figure 3.3.7: The first subplot shows the relative error of solving Burgers' equations on 100 test (unseen) initial conditions for two models: PINODE Hybrid and PINODE Data-Driven. Both models interpolate well but a purely data-driven model fails to extrapolate past the training time-horizon (left of the red vertical line). PINODE-Hybrid provides stable long-term predictions that points to its ability to correctly discover the low-dimensional manifold dynamics.

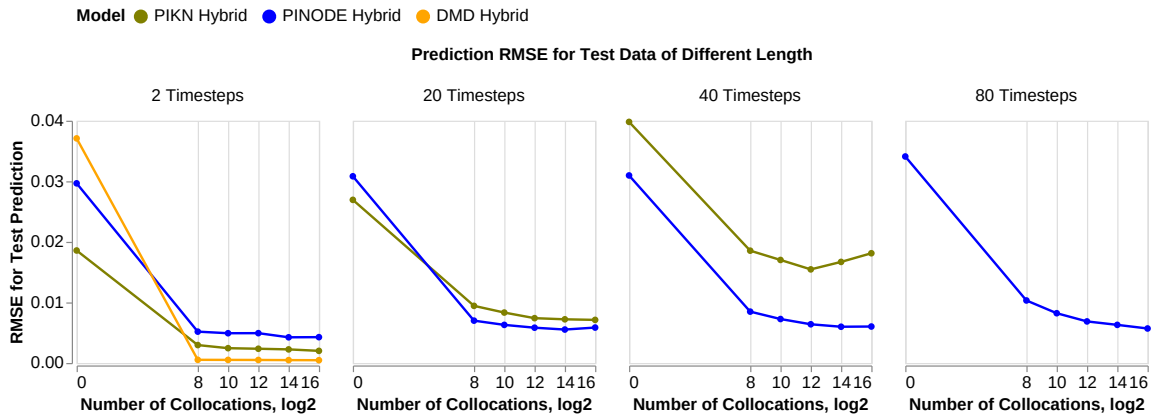


Figure 3.3.8: Collocation points improve results of all three models but they don't fix models' inherent shortcomings like instabilities in linear latent dynamics.

To illustrate, we replicate the experiments from Figure 3.3.6 where the number of trajectories is 256 and with Bump ICs for PINODE, PIKN, and DMD.Figure 3.3.8 shows the root mean squared error (RMSE) for the test data predictions as a function of the number of collocation points that were used in training. The figure illustrates the prediction error for increasing prediction horizons going from left to right, and demonstrates that in all cases, PINODE benefits from the available collocation points. The leftmost panel shows that every model improves its one-step-ahead predictions, with DMD quickly achieving near-optimal performance. However, once the forecast horizon is increased to 20 timesteps ahead (length of the training trajectories) and above, DMD failed to correctly forecast the long-term trajectories and was removed from those figure to improve legibility. The PIKN models improved the one-step-ahead (1st pane) and interpolation performance (2nd pane) by a factor of 4. It also improved the extrapolation performance for 40-steps prediction (3rd pane) but failed to extrapolate for 80 steps (4th pane, removed for legibility). We attribute this behavior of PIKN to the possibility that the latent dynamics operator of PIKN contains positive eigenvalues despite the use of collocation points.

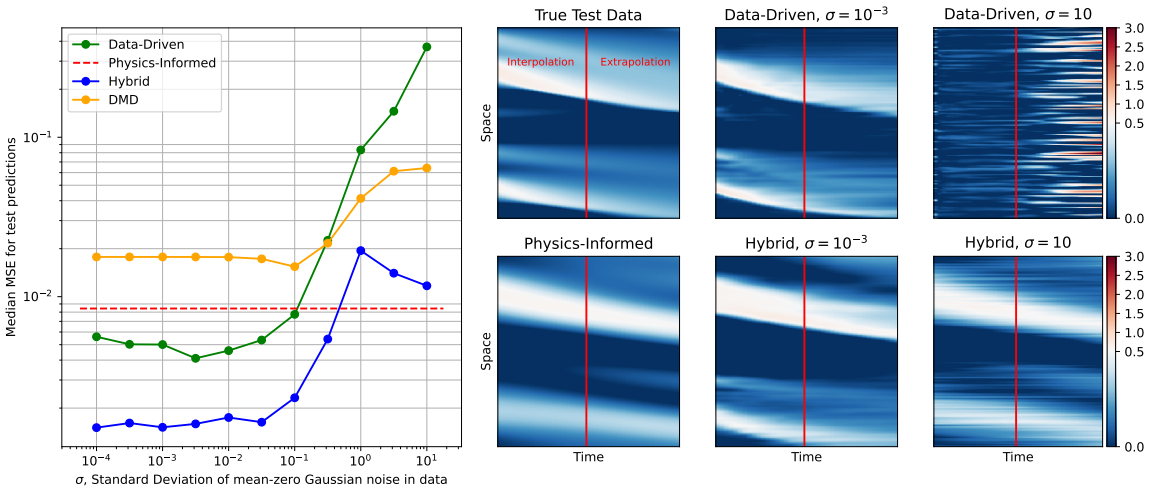


Figure 3.3.9: Physics-informed loss works as a safeguard that prevents unbounded performance drop when quality of the data degrades due to noise. Namely, the solution of the hybrid loss (3.2.10) converges to the solution of the physics-informed loss (3.2.9), when the data-driven loss (3.2.9) becomes uninformative. The performance of purely data-driven methods (Data-Driven, DMD) grows unbounded since these models don't have an alternative noise-independent source of information.

3.3.5 Robustness to Noise in the Low-Data Regime

In this section we show that the use of collocation points improves the ROMs' robustness to noise in the data by providing an alternative, noise-free, source of information.

For this experiment, we use the Burgers' equation dataset containing 1024 trajectories with "bump" initial conditions, and 65536 "harmonic" collocation points as defined in Equation 3.3.6. We then add i.i.d. Gaussian noise to the trajectories, with variance ranging from $\sigma = 10^{-4}$ to $\sigma = 10$. For reference, most of the data values lie between 0 and 1, so a noise level with $\sigma > 1$ dominates the data. We train four models: PINODE Hybrid, PINODE Data-Driven, PINODE Physics-Informed, and DMD. To measure the models' out-of-distribution prediction errors, we use the test dataset with Bump, Gaussian, and Harmonic initial conditions, as described in the previous subsection. The prediction errors are displayed in Figure 3.3.9, left pane. The prediction error of a purely Physics-Informed model (in red) is flat because the collocation points are noise-free.

Figure (3.3.9) shows that in the high noise setting, the error of purely data-driven models (DMD and PINODE Data-Driven) grows unbounded, whereas the performance of the hybrid model converges to the performance of the Physics-Informed model as the noise level increases. We hypothesise that such behavior is due to the second part ($\mathcal{L}_\theta^{data}$) of the combined loss (Eq. 3.2.10) turns into noise, and so its derivative also turns into noise.

$$\nabla \mathcal{L}_\theta = \underbrace{\nabla \mathcal{L}_\theta^{physics}}_{\text{informative}} + \underbrace{\nabla \mathcal{L}_\theta^{data}}_{\text{noise}} \tag{3.3.7}$$

Thus, one can think about optimizing a hybrid model (3.2.10) as about training a Physics-Informed model (3.2.9) using a noisy gradient descent with a fixed-variance noise. From the optimization literature Friedlander and Schmidt (2012); Patel et al. (2021); Shapiro et al. (2021) we know that, under certain conditions, such SGD converges to a neighbourhood of a local minimum of its loss (in this case $\mathcal{L}_\theta^{physics}$) with high probability. So instead of diverging, a hybrid model turns into a Physics-Informed model; where the latter works as a performance safeguard in the high-noise regime. On the right hand-side of Figure (3.3.9), we show an example of the prediction performance of each of the models described above. The data-driven and hybrid models yield visually similar solutions when $\sigma = 10^-3$. However, the former provides inadequate performance when the data is dominated by noise, whereas a hybrid model in this regime produces a solution that is visually similar to the one that the Physics-Informed model produces. A more rigorous analysis of this phenomenon seems possible but lies outside of the scope of this paper.

3.4 Discussion and Conclusions

In this work, we demonstrated how a collocation point-based technique can improve the performance of an emerging class of continuous-time physics-informed neural-network based reduced-order models. First, we demonstrated that the incorporation of collocation points in training data can "cover the gaps" in training trajectories and inform the model about underrepresented basins of attraction. Such an approach alleviates the demand for large volumes of data that is common in network-based models, which is crucial in applications where data is scarce and expensive. Second, the physics-informed loss may work as a safeguard, providing a noise-free source of underlying dynamics. Third, collocation points can stabilize the model's long-term predictions, allowing for accurate forecasting far beyond the training time horizon. Finally, together with using a NODE-based non-linear latent dynamics, adding physics-informed loss leads to the discovery of more compact latent space representations that also yield more accurate models. Simultaneous stability and compactness is especially important if one aims to use models together with compressive sensing and control algorithms. With respect to the computational complexity, we note that adding Tk collocation points to the training imposes less of a computational burden than adding k data trajectories because collocation points do not require computing integrals forward in time as in the case of data trajectories.

One clear limitation of the current work is that the choice of an efficient collocation family is a design decision that a practitioner makes. The authors believe that such decisions can be automated by adopting existing approaches from classic works on numerical approximations of PDEs, which we leave for future research. Another automation that prompts future research is deriving efficient ways of sampling collocation points, possibly via applying modern adaptive learning techniques Subramanian et al. (2022). Finally, although Section 3.3.5 provides some rationale for why one may expect robustness of Hybrid models under noise, the authors believe

that a more rigorous analysis is possible; particularly one that provides conditions under which such robustness is guaranteed.

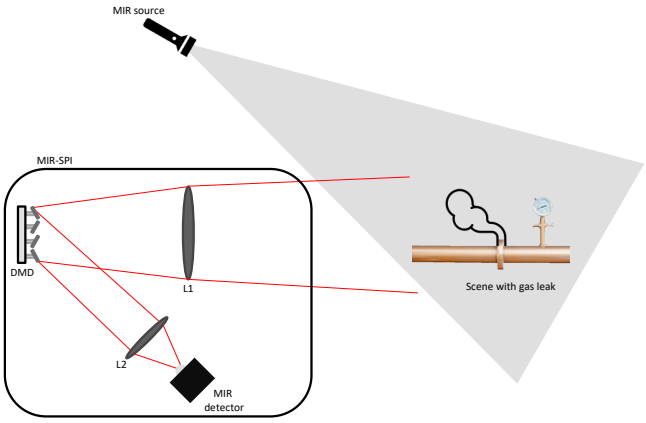


Figure 3.5.1: SPI setup

3.5 Application to Compressive Sensing

3.5.1 Introduction

Prior Work

3.5.2 Method

Compressive Sensing In many applications we can not observe $x(t)$ in real time. Instead, we observe p detectors, and each detector provides a linear combination of a small number of coordinates of $x(t)$ at a time:

$$y(t) = A_t x(t), \quad A_t \in \mathbb{R}^{p \times n} \tag{3.5.1}$$

In particular, we consider a single pixel camera setup where for every time instance t, p acquisitions $y(t)$ are obtained by a high sampling rate photo-detector using the projection matrix A_t. The rows of the matrix A_t correspond to a binary mask pattern that can be encoded using a digital micro-mirror device (DMD) where the incoming light from $x(t)$ is reflected from the DMD array and focused onto the photo-detector. Figure 3.5.1 illustrates an example of the single pixel imaging setup where a gas plume is imaged using a DMD array and a medium infra-red (MIR) photo-detector.

However, it is often possible to have access to a complete state $x(t)$ at the moment of model training. Thus, one can develop a ROM $(\psi_\theta, \phi_\theta, h_\theta)$ using the full state $x(t)$, and then utilize this ROM for real-time compressive sensing applications.

Training Loss Similar to prior works Takeishi et al. (2017); Morton et al. (2019); Gin et al. (2021), we define a *data-driven loss* $\mathcal{L}_{data}$ as a sum of reconstruction and prediction losses. The former ensures that ϕ_θ and ψ_θ are inverse mappings of each other, whereas the latter matches the

model's predictions to the available data. Formally, for a given set of trajectories $\boldsymbol{x}_i$, $i \in [1 \dots k]$, where each trajectory $\boldsymbol{x}_i \in \mathbb{R}^{n \times p}$ is a set of p snapshots that correspond to the recorded states of the system for p time-steps, t_j, $j \in [1, \dots, p]$, the loss function $\mathcal{L}_\theta^{data}$ is defined as:

$$\mathcal{L}^{training}(\theta) = \frac{1}{2\sigma^2} \sum_{i=1}^{k} \left[\frac{\omega_1}{p} \sum_{j=1}^{p} \|\boldsymbol{x_i}(t_j) - \psi_\theta(\phi_\theta(\boldsymbol{x_i}(t_j)))\|^2 + \right. \tag{3.5.2}$$

$$\left. + \frac{\omega_2}{p} \sum_{j=1}^{p} \left\| \psi_\theta \left(\phi_\theta(\boldsymbol{x_i}(t_1)) + \int_{t_1}^{t_j} h(z(t))dt \right) - \boldsymbol{x_i}(t_j) \right\|^2 \right] \tag{3.5.3}$$

where σ is the standard deviation of the observation noise.

To obtain a ROM $(\psi_{\theta^*}, \phi_{\theta^*}, h_{\theta^*})$, we minimize the loss above:

$$\theta^* = \arg\min_\theta \mathcal{L}^{training}(\theta) \tag{3.5.4}$$

We note that each trajectory $\boldsymbol{x}_i$ may be captured over its own time-frame and use a distinct, possibly non-uniform, step-size, in which case the loss function should be modified accordingly[2]. To simplify the notation without loss of generality, in the rest of the paper we assume that all trajectories were recorded over the same time-frame with an equal and uniform step-size.

Reconstruction Loss We use the ROM $(\psi_{\theta^*}, \phi_{\theta^*}, h_{\theta^*})$ above to forecast the dynamics based on partial observations in real time. Namely, instead of reconstructing $x(t)$ based on compressive-sensing observations $y(t)$ directly, we first reconstruct the latent dynamics $z(t)$ and then project it to the observable space using the decoder $\psi_{\theta^*}(z)$.

$$\min_{\{z_t\}_{t=1,\dots,T}} \frac{1}{2} \sum_{t=1}^{T} \|y_t - A\psi_{\theta^*}(z_t)\|_2^2 \tag{3.5.5}$$

$$\text{s.t. } \dot{z} = h_{\theta^*}(z) \tag{3.5.6}$$

We integrate the constraint and write it in its Lagrangian form:

$$\min_{\{z_t\}_{t=1,\dots,T}} \mathcal{L}_{\theta^*}^{recon}(z) \tag{3.5.7}$$

where

$$\mathcal{L}_{\theta^*}^{recon}(z) = \frac{1}{2} \sum_{t=1}^{T} \|y_t - A\psi_{\theta^*}(z_t)\|_2^2 + \frac{\lambda}{2} \sum_{t=1}^{T} \left\| z_{t-1} + \int_{t-1}^{t} h_{\theta^*}(z)dz - z_t \right\|_2^2 \tag{3.5.8}$$

[2]The implementation is affected only in evaluating the integral in (3.2.4). This part is handled by `torchdiffeq` Chen et al. (2018a) library, which supports non-uniform time-frames within a batch

where the parameter λ controls the degree on which the compressing sensing algorithm relies on the latent dynamics h_{θ^*} during the signal reconstruction phase. We minimize the loss 3.5.8 using a gradient-based technique, with the gradients obtained using automatic differentiation frameworks.

3.5.3 Experiments

3.5.4 Discussion and Conclusion

Chapter 4

Conclusion

In this work I gave two examples of how partial knowledge can be utilized to improve performance of machine learning models. First, in Section 2.3.2 we showed how complex priors can be embedded into a linear mixed-effects model via a suitable proximal operator, and how the resulting algorithm can be implemented using a proximal gradient descent. We note that a complex combination of priors often yields ill-conditioned problems, and we propose a MSR3 relaxation to address it. We also prove that a solution of such relaxation approaches the solution of the original problem as the relaxation tightens. Next, we showed how partial knowledge of physics can be embedded into deep-learning-based reduced-order models. Namely, we show how to project known physical equations that govern behavior of the system in an observable space into a chosen latent space. In that space one can require that the gradient field should match the projected true dynamics. We show that this methodology leads to identification of more stable latent manifolds, and using such manifolds for forecasting leads to more accurate predictions and lesser sensitivity to noise. We also showed how the reduced-order models described above can be used for compressive sensing. We demonstrated that they can accurately reconstruct the dynamics from partial observations, significantly surpassing current state-of-the-art compressive-sensing methods. We hope that these developments pave the way to higher-quality machine learning models which effectively marry data and the humanity's hard-won expert knowledge.

Perspective Research Directions Due to the limited time and resources we left many promising directions for future exploration. One such direction is extending sparse relaxed regularized regression (SR3) to generalized linear and generalized mixed-effects models. The fundamental difficulty that arises in such extension is the absence of closed-form marginal log-likelihoods. It implies that one would have three nested levels of optimization when they apply an SR3 relaxation on top of such likelihood: evaluating the likelihood, evaluating a value function, and optimizing over the value function. Re-using the recipe that we developed in Section 2.3.2 could help to surmount such obstacle. For example, Algorithm 2 had two nested optimization methods: for evaluating the value function and for optimizing over it. However we showed that one can "blend" both levels of optimization and achieve superior performance results. The same methodology of "blending" could possibly be applied to more than two nested optimization loops to yield algorithms that, in certain scenarios, work faster than non-nested optimization methods.

Our development of physics-informed reduced-order models also invites future work in several promising directions. First, a more systematic way of choosing collocation points would greatly benefit the method. In Section 3.2 we define collocation points and provide criteria for a good choice of a family of colocation points. However, we ultimately leave the reader without a constructive algorithm for identifying candidate families of collocation points suitable for their problem. At the same time, certain differential equations admit families of basis functions that can span their solutions. Such basis functions frequently poses easily-computable derivatives, which makes them perfect candidates for collocation points for PINODE. Bridging the gap between this new methodology and those classic results would undoubtedly yield a fruitful

line of works. Finally, one could utilize PINODE models not only in compressive sensing applications but in any application where one needs to quickly search over a large space of possible simulations. These examples include medical imaging, model discovery, online control, fast simulations, and many others.

Acknowledgements I would like to thank ... (TBD)

Bibliography

Ahmed, S. E., Pawar, S., San, O., Rasheed, A., Iliescu, T., and Noack, B. R. (2021). On closures for reduced order models—a spectrum of first-principle to machine-learned avenues. Physics of Fluids, 33(9):091301.

Aravkin, A., Burke, J., Bell, B., and Pillonetto, G. (2021). Algorithms for block tridiagonal systems: Foundations and new results for generalized kalman smoothing. To appear in 19th IFAC Symposium on System Identification (SYSID 2021).

Aravkin, A., Burke, J., Drusvyatskyi, D., Friedlander, M., and Macphee, K. (2018). Foundations of gauge and perspective duality. SIAM J. on Opt., 28:2406 – 2434.

Aravkin, A., Burke, J., and Friedlander, M. (2013). Variational properties of value functions. SIAM J. on Opt., 23:1689 – 1717.

Aravkin, A., Burke, J., Sholokhov, A., and Zheng, P. (2022a). Analysis of relaxation methods for feature selection in mixed effects models.

Aravkin, A., Burke, J., Sholokhov, A., and Zheng, P. (2022b). Analysis of relaxation methods for feature selection in mixed effects models. https://arxiv.org/abs/2205.06925.

Attouch, H., Bolte, J., and Svaiter, B. F. (2013). Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. Mathematical Programming, 137(1):91–129.

Baraldi, R., Kumar, R., and Aravkin, A. (2019). Basis Pursuit Denoise With Nonsmooth Constraints. IEEE Transactions on Signal Processing, 67(22):5811–5823.

Beck, A. (2017). First-Order Methods in Optimization. MOS-SIAM Series on Optimization. SIAM.

Benner, P., Gugercin, S., and Willcox, K. (2015). A survey of projection-based model reduction methods for parametric dynamical systems. SIAM review, 57(4):483–531.

Bondell, H. D., Krishna, A., and Ghosh, S. K. (2010). Joint Variable Selection for Fixed and Random Effects in Linear Mixed-Effects Models. Biometrics, 66(4):1069–1077.

Bongard, J. and Lipson, H. (2007). Automated reverse engineering of nonlinear dynamical systems. Proceedings of the National Academy of Sciences, 104(24):9943–9948.

Brunton, S. L. and Kutz, J. N. (2022). Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the national academy of sciences, 113(15):3932–3937.

Burgers, J. M. (1948). A mathematical model illustrating the theory of turbulence. Advances in applied mechanics, 1:171–199.

Burke, J. and Engle, A. (2018). Line search and trust-region methods for convex-composite optimization. arXiv:1806.05218.

Buscemi, S. and Plaia, A. (2019). Model selection in linear mixed-effect models. AStA Advances in Statistical Analysis.

Champion, K., Lusch, B., Kutz, J. N., and Brunton, S. L. (2019). Data-driven discovery of coordinates and governing equations. Proceedings of the National Academy of Sciences, 116(45):22445–22451.

Chen, B., Huang, K., Raghupathi, S., Chandratreya, I., Du, Q., and Lipson, H. (2021). Discovering state variables hidden in experimental data. arXiv preprint arXiv:2112.10755.

Chen, F., Li, Z., Shi, L., and Zhu, L. (2015). Inference for mixed models of anova type with high-dimensional data. Journal of Multivariate Analysis, 133:382–401.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018a). Neural ordinary differential equations. Advances in neural information processing systems, 31.

Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018b). Neural ordinary differential equations. In Advances in neural information processing systems, pages 6571–6583.

Chen, Z. and Dunson, D. B. (2003). Random Effects Selection in Linear Mixed Models. Biometrics, 59(4):762–769.

Cranmer, M., Sanchez Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., and Ho, S. (2020). Discovering symbolic models from deep learning with inductive biases. Advances in Neural Information Processing Systems, 33:17429–17442.

Dai, X., Gil, G. F., Reitsma, M. B., Ahmad, N. S., Anderson, J. A., Bisignano, C., Carr, S., Feldman, R., Hay, S. I., He, J., et al. (2022). Health effects associated with smoking: a burden of proof study. Nature Medicine, 28(10):2045–2055.

Delahunt, C. B. and Kutz, J. N. (2022). A toolkit for data-driven discovery of governing equations in high-noise regimes. IEEE Access, 10:31210–31234.

DerSimonian, R. and Laird, N. (1986). Meta-analysis in clinical trials. Controlled clinical trials, 7(3):177–188.

Duriez, T., Brunton, S. L., and Noack, B. R. (2017). Machine learning control-taming nonlinear dynamics and turbulence, volume 116. Springer.

Fan, J. (1997). Comments on "wavelets in statistics: A review" by a. antoniadis. Journal of the Italian Statistical Society, 6(2):131.

Fan, J. and Li, R. (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. Journal of the American Statistical Association, 96(456):1348–1360.

Fan, Y. and Li, R. (2012). Variable selection in linear mixed effects models. The Annals of Statistics, 40(4):2043–2068.

Fan, Y., Qin, G., and Zhu, Z. Y. (2014). Robust variable selection in linear mixed models. Communications in Statistics-Theory and Methods, 43(21):4566–4581.

Farahmand, A.-m., Nabi, S., Grover, P., and Nikovski, D. N. (2016). Learning to control partial differential equations: Regularized fitted q-iteration approach. In 2016 IEEE 55th Conference on Decision and Control (CDC), pages 4578–4585. IEEE.

Fornberg, B. (1998). A practical guide to pseudospectral methods. Number 1. Cambridge university press.

Friedlander, M. P. and Schmidt, M. (2012). Hybrid deterministic-stochastic methods for data fitting. SIAM Journal on Scientific Computing, 34(3):A1380–A1405.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1):1–22.

Fries, W. D., He, X., and Choi, Y. (2022). Lasdi: Parametric latent space dynamics identification. Computer Methods in Applied Mechanics and Engineering, 399:115436.

Ghosh, A. and Thoresen, M. (2018). Non-concave penalization in linear mixed-effect models and regularized selection of fixed effects. AStA Advances in Statistical Analysis, 102(2):179–210.

Gin, C., Lusch, B., Brunton, S. L., and Kutz, J. N. (2021). Deep learning models for global coordinate transformations that linearise pdes. European Journal of Applied Mathematics, 32(3):515–539.

Grant, R. L. (2014). Converting an odds ratio to a range of plausible relative risks for better communication of research findings. Bmj, 348.

Groll, A. and Tutz, G. (2014). Variable selection for generalized linear mixed models by l 1-penalized estimation. Statistics and Computing, 24(2):137–154.

Harville, D. (1976). Extension of the Gauss-Markov Theorem to Include the Estimation of Random Effects. The Annals of Statistics, 4(2):384–395.

He, X., Choi, Y., Fries, W. D., Belof, J., and Chen, J.-S. (2022). glasdi: Parametric physics-informed greedy latent space dynamics identification. arXiv preprint arXiv:2204.12005.

Holmes, P., Lumley, J. L., Berkooz, G., and Rowley, C. W. (2012). Turbulence, coherent structures, dynamical systems and symmetry. Cambridge university press.

Horn, R. and Johnson, C. (1985). Matrix Analysis. Cambridge University Press.

Hui, F. K., Müller, S., and Welsh, A. H. (2017). Joint Selection in Mixed Models using Regularized PQL. Journal of the American Statistical Association, 112(519):1323–1333.

Ibrahim, J. G., Zhu, H., Garcia, R. I., and Guo, R. (2011a). Fixed and random effects selection in mixed effects models. Biometrics, 67(2):495–503.

Ibrahim, J. G., Zhu, H., Garcia, R. I., and Guo, R. (2011b). Fixed and Random Effects Selection in Mixed Effects Models. Biometrics, 67(2):495–503.

IHME (2020). IHME COVID-19 Projections.

IHME COVID-19 Forecasting Team (2020). Modeling COVID-19 scenarios for the United States. Nature Medicine.

Jiang, J., Rao, J. S., Gu, Z., and Nguyen, T. (2008). Fence methods for mixed model selection. The Annals of Statistics, 36(4):1669–1692.

Jones, D., Snider, C., Nassehi, A., Yon, J., and Hicks, B. (2020). Characterising the digital twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology, 29:36–52.

Jones, R. H. (2011). Bayesian information criterion for longitudinal and clustered data. Statistics in Medicine, 30(25):3050–3056.

Kalur, A., Nabi, S., and Benosman, M. (2021). Robust adaptive dynamic mode decomposition for reduce order modelling of partial differential equations. In 2021 American Control Conference (ACC), pages 4497–4502. IEEE.

Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., and Solenthaler, B. (2019). Deep fluids: A generative network for parameterized fluid simulations. In Computer Graphics Forum, volume 38, pages 59–70. Wiley Online Library.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kojima, M., Megiddo, N., Noma, T., and Yoshise, A. (1991). A unified approach to interior point algorithms for linear complementarity problems: A summary. Operations Research Letters, 10(5):247–254.

Kojima, R. and Okamoto, Y. (2022). Learning deep input-output stable dynamics. In Advances in Neural Information Processing Systems.

Kutz, J. N., Brunton, S. L., Brunton, B. W., and Proctor, J. L. (2016). Dynamic mode decomposition: data-driven modeling of complex systems. SIAM.

Lan, L. (2006). Variable Selection in Linear Mixed Model for Longitudinal Data. PhD thesis.

Lee, K. and Carlberg, K. T. (2020). Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. Journal of Computational Physics, 404:108973.

Lescinsky, H., Afshin, A., Ashbaugh, C., Bisignano, C., Brauer, M., Ferrara, G., Hay, S. I., He, J., Iannucci, V., Marczak, L. B., et al. (2022). Health effects associated with consumption of unprocessed red meat: a burden of proof study. Nature Medicine, 28(10):2075–2082.

Lin, B., Pang, Z., and Jiang, J. (2013a). Fixed and random effects selection by REML and pathwise coordinate optimization. Journal of Computational and Graphical Statistics, 22(2):341–355.

Lin, B., Pang, Z., and Jiang, J. (2013b). Fixed and random effects selection by reml and pathwise coordinate optimization. Journal of Computational and Graphical Statistics, 22(2):341–355.

Liu, Y., Sholokhov, A., Mansour, H., and Nabi, S. (2022). Physics-informed koopman network. arXiv preprint arXiv:2211.09419.

Lu, T.-T. and Shiou, S.-H. (2002). Inverses of 2×2 block matrices. Computers and Mathematics with Applications, 43:119–129.

Lucia, D. J., Beran, P. S., and Silva, W. A. (2004). Reduced-order modeling: new approaches for computational physics. Progress in aerospace sciences, 40(1-2):51–117.

Morton, J., Witherden, F. D., and Kochenderfer, M. J. (2019). Deep variational koopman models: Inferring koopman observations for uncertainty-aware dynamics modeling and control. arXiv preprint arXiv:1902.09742.

Müller, S., Scealy, J. L., and Welsh, A. H. (2013). Model selection in linear mixed models. Statistical Science, 28(2):135–167.

Murray, C. J. and Acharya, A. K. (1997). Understanding dalys. Journal of health economics, 16(6):703–730.

Murray, C. J., Aravkin, A. Y., Zheng, P., Abbafati, C., Abbas, K. M., Abbasi-Kangevari, M., Abd-Allah, F., Abdelalim, A., Abdollahi, M., Abdollahpour, I., et al. (2020). Global burden of 87 risk factors in 204 countries and territories, 1990–2019: a systematic analysis for the global burden of disease study 2019. The Lancet, 396(10258):1223–1249.

Nabi, S., Grover, P., and Caulfield, C. (2022). Robust preconditioned one-shot methods and direct-adjoint-looping for optimizing reynolds-averaged turbulent flows. Computers & Fluids, 238:105390.

Nabi, S., Nishio, N., Grover, P., Matai, R., Kajiyama, Y., Kotake, N., Kameyama, S., Yoshiki, W., and Iida, M. (2020). Improving lidar performance on complex terrain using cfd-based correction and direct-adjoint-loop optimization. In Journal of Physics: Conference Series, volume 1452, page 012082. IOP Publishing.

Nesterov, Y. and Nemirovskii, A. (1994). Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics.

Noack, B. R., Morzynski, M., and Tadmor, G. (2011). Reduced-order modelling for flow control, volume 528. Springer Science & Business Media.

Nocedal, J. and Wright, S. (2006). Numerical optimization. Springer Science & Business Media.

Otterness, N., Yang, M., Rust, S., Park, E., Anderson, J. H., Smith, F. D., Berg, A., and Wang, S. (2017). An evaluation of the nvidia tx1 for supporting real-time computer-vision workloads. In 2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), pages 353–364. IEEE.

Page, J. and Kerswell, R. R. (2019). Koopman mode expansions between simple invariant solutions. Journal of Fluid Mechanics, 879:1–27.

Pan, J. and Shang, J. (2018). A simultaneous variable selection methodology for linear mixed models. Journal of Statistical Computation and Simulation, 88(17):3323–3337.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019a). Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019b). Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32, pages 8024–8035. Curran Associates, Inc.

Patel, V., Tian, B., and Zhang, S. (2021). Global convergence and stability of stochastic gradient descent. arXiv preprint arXiv:2110.01663.

Patterson, H. D. and Thompson, R. (1971). Recovery of Inter-Block Information when Block Sizes are Unequal. Biometrika, 58(3):545.

Peherstorfer, B. (2022). Breaking the kolmogorov barrier with nonlinear model reduction. Notices of the American Mathematical Society, 69(5).

Peherstorfer, B. and Willcox, K. (2016). Data-driven operator inference for nonintrusive projection-based model reduction. Computer Methods in Applied Mechanics and Engineering, 306:196–215.

Pinheiro, J. and Bates, D. (2006). Mixed-effects models in S and S-PLUS. Springer science & business media.

Pinheiro, J. C. and Bates, D. M. (2000). Mixed-Effects Models in Sand S-PLUS. Journal of the American Statistical Association, 96(455):1135–1136.

Qian, E., Kramer, B., Peherstorfer, B., and Willcox, K. (2020). Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems. Physica D: Nonlinear Phenomena, 406:132401.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A. (2020). Universal differential equations for scientific machine learning. arXiv preprint arXiv:2001.04385.

Raissi, M. and Karniadakis, G. E. (2018). Hidden physics models: Machine learning of nonlinear partial differential equations. Journal of Computational Physics, 357:125–141.

Razo, C., Welgan, C. A., Johnson, C. O., McLaughlin, S. A., Iannucci, V., Rodgers, A., Wang, N., LeGrand, K. E., Sorensen, R. J., He, J., et al. (2022). Effects of elevated systolic blood pressure on ischemic heart disease: a burden of proof study. Nature Medicine, 28(10):2056–2065.

Reiner, R. C., Barber, R. M., Collins, J. K., Zheng, P., Hay, S. I., Lim, S. S., Murray, C. J. L., and IHME COVID-19 Forecasting Team (2020). Modeling covid-19 scenarios for the United States. Nature medicine.

Rockafellar, R. T. and Wets, R. J.-B. (2009). Variational analysis, volume 317. Springer Science & Business Media.

Rowley, C. W. and Dawson, S. T. (2017). Model reduction for flow analysis and control. Annu. Rev. Fluid Mech, 49(1):387–417.

Schelldorfer, J., Bühlmann, P., and DE GEER, S. V. (2011). Estimation for high-dimensional linear mixed-effects models using l1-penalization. Scandinavian Journal of Statistics, 38(2):197–214.

Schmidt, M. and Lipson, H. (2009). Distilling free-form natural laws from experimental data. science, 324(5923):81–85.

Shapiro, A., Dentcheva, D., and Ruszczynski, A. (2021). Lectures on stochastic programming: modeling and theory. SIAM.

Sholokhov, A., Burke, J., Santomauro, D., Zheng, P., and Aravkin, A. (2022a). A relaxation approach to feature selection for linear mixed effects models. In Preparation.

Sholokhov, A., Burke, J. V., Santomauro, D. F., Zheng, P., and Aravkin, A. (2022b). A relaxation approach to feature selection for linear mixed effects models.

Sholokhov, A., Zheng, P., and Aravkin, A. (2023). pysr3: A python package for sparse relaxed regularized regression. Journal of Open Source Software, 8(84):5155.

Stanaway, J. D., Afshin, A., Ashbaugh, C., Bisignano, C., Brauer, M., Ferrara, G., Garcia, V., Haile, D., Hay, S. I., He, J., et al. (2022). Health effects associated with vegetable consumption: a burden of proof study. Nature Medicine, 28(10):2066–2074.

Subramanian, S., Kirby, R. M., Mahoney, M. W., and Gholami, A. (2022). Adaptive self-supervision algorithms for physics-informed neural networks. arXiv preprint arXiv:2207.04084.

Sugiura, N. (1978). Further analysts of the data by akaike' s information criterion and the finite corrections. Communications in Statistics - Theory and Methods, 7(1):13–26.

Takeishi, N., Kawahara, Y., and Yairi, T. (2017). Learning koopman invariant subspaces for dynamic mode decomposition. Advances in Neural Information Processing Systems, 30.

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

Tibshirani, R. (1996b). Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288.

Trefethen, L. N. (2000). Spectral methods in MATLAB. SIAM.

Trefethen, L. N. and Bau, D. (2022). Numerical linear algebra, volume 181. Siam.

Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., and Kutz, J. N. (2013). On dynamic mode decomposition: Theory and applications. arXiv preprint arXiv:1312.0041.

Vaida, F. and Blanchard, S. (2005). Conditional Akaike information for mixed-effects models. Biometrika, 92(2):351–370.

Vanderbei, R. and Shanno, D. (1999). An interior-point algorithm for nonconvex nonlinear programming. Comp. Opt, and Appl., 13:231–252.

Wang, Z. (2013). Converting odds ratio to relative risk in cohort studies with partial data information. Journal of Statistical Software, 55:1–11.

Wright, S. J. (1997a). Primal-Dual Interior-Point Methods. Society for Industrial and Applied Mathematics.

Wright, S. J. (1997b). Primal-Dual Interior-Point Methods. SIAM.

Xu, P., Wang, T., Zhu, H., and Zhu, L. (2015). Double Penalized H-Likelihood for Selection of Fixed and Random Effects in Mixed Effects Models. Statistics in Biosciences, 7(1):108–128.

Zheng, P., Afshin, A., Biryukov, S., Bisignano, C., Brauer, M., Bryazka, D., Burkart, K., Cercy, K. M., Cornaby, L., Dai, X., et al. (2022). The burden of proof studies: assessing the evidence of risk. Nature Medicine, 28(10):2038–2044.

Zheng, P., Askham, T., Brunton, S. L., Kutz, J. N., and Aravkin, A. Y. (2019). A Unified Framework for Sparse Relaxed Regularized Regression: SR3. IEEE Access, 7:1404–1423.

Zheng, P., Barber, R., Sorensen, R. J., Murray, C. J., and Aravkin, A. Y. (2021). Trimmed constrained mixed effects models: formulations and algorithms. Journal of Computational and Graphical Statistics, pages 1–13.

Zou, H. (2006). The adaptive lasso and its oracle properties. Journal of the American Statistical Association, 101(476):1418–1429.

Zuur, A., Ieno, E. N., Walker, N., Saveliev, A. A., and Smith, G. M. (2009). Mixed effects models and extensions in ecology with R. Springer Science & Business Media.

Appendix A

Appendix

A.1 Acknowledgement

I am extremely grateful to my advisor, Sasha Aravkin, for providing support, guidance, and enthusiasm during my work on this project in the Department of Applied Mathematics and in Institute for Health Metrics and Evaluation.
I would also like to thank Jim Burke for his detailed review of my work which would hopefully lead to fruitful collaboration.
Last, but not least: I am grateful to Damian Santomauro, who introduced me to his work on the consequences of bullying, provided with a dataset, and with a valuable feedback on the performance of the proposed method.

A.2 Derivatives of Marginalized Log-likelihood for Linear Mixed Models

For conciseness, let us define the mismatch $\xi_i = Y_i - X_i\beta$. We also omit the dependence on β, as it's fixed at this point. The loss function 2.2.3 takes the form

$$\mathcal{L}(\gamma) = \sum_{i=1}^{m} \frac{1}{2}\xi_i^T(\Omega_i(\gamma))^{-1}\xi_i + \frac{1}{2}\log\det(\Omega_i(\gamma)). \tag{A.2.1}$$

The derivative of the objective w.r.t γ_j, the j'th diagonal element of the matrix Γ is

$$\frac{\partial \xi_i^T\Omega_i^{-1}\xi_i}{\partial \Gamma_{jj}} = \mathrm{Tr}\left[\left(\frac{\partial \xi_i^T\Omega_i^{-1}\xi_i}{\partial \Omega_i}\right)\frac{\partial \Omega}{\partial \Gamma_{jj}}\right] = \mathrm{Tr}\left[\left(-\Omega_i^{-T}\xi_i\xi_i^T\Omega_i^{-T}\right)^T Z_i\frac{\partial \Gamma}{\partial \Gamma_{jj}}Z_i^T\right] = \tag{A.2.2}$$

where $\frac{\partial \Gamma}{\partial \Gamma_{jj}}$ is a structure matrix, which, in a general case, is equal to a single-entry matrix J^{jj} with jj'th element is equal to 1 and zeroes elsewhere. Substituting this back we get

$$= \mathrm{Tr}\left[\left(-\Omega_i^{-T}\xi_i\xi_i^T\Omega_i^{-T}\right)^T Z_i^j Z_i^{jT}\right] = \tag{A.2.3}$$

where Z_i^j is a j'th column of the matrix Z_i. Making a circular swap we end up with

$$= \mathrm{Tr}\left[-Z_i^{jT}\Omega_i^{-T}\xi_i\xi_i^T\Omega_i^{-T}Z_i^j\right] = -(Z_i^{jT}\Omega_i^{-T}\xi_i)^2 \tag{A.2.4}$$

Similarly,

$$\frac{\partial \log\det\Omega_i}{\partial \Gamma_{jj}} = \mathrm{Tr}\left[\left(\frac{\partial \log\det\Omega_i}{\partial \Omega_i}\right)\frac{\partial \Omega_i}{\partial \Gamma_{jj}}\right] = \mathrm{Tr}\left[\Omega_i^{-1}Z_i^j Z_i^{jT}\right] = Z_i^{jT}\Omega_i^{-1}Z_i^j \tag{A.2.5}$$

Taking into account that Ω_i is symmetric, we have

$$\left[\nabla_\gamma \mathcal{L}(\beta,\gamma)\right]_j = \sum_{i=1}^m -(Z_i^{jT}\Omega_i^{-T}\xi_i)^2 + Z_i^{jT}\Omega_i^{-1}Z_i^j = \tag{A.2.6}$$

or, in vector form

$$= \sum_{i=1}^m \mathrm{diag}\left(Z_i{}^T\Omega_i^{-1}Z_i\right) - (Z_i{}^T\Omega_i^{-T}\xi_i)^{\circ 2} = \tag{A.2.7}$$

where $\circ$ denotes the Hadamard (element-wise) product. Using the Cholesky decomposition $\Omega_i = L_iL_i^T$ we can calculate it more effectively, using only one triangular matrix inversion:

$$= \sum_{i=1}^m \left[\sum_{\text{rows}}\left(L_i^{-1}Z_i\right)^{\circ 2} - [(L_i^{-1}Z_i)^T(L_i^{-1}\xi_i)]^{\circ 2}\right] \tag{A.2.8}$$

Notice, that the loss function (2.2.3) and the optimal β solution (2.2.5) can also be effectively computed using Cholesky:

$$\mathcal{L}(\gamma) = \sum_{i=1}^m \frac{1}{2}\xi_i^T(\Omega_i(\gamma))^{-1}\xi_i + \frac{1}{2}\log\det(\Omega_i(\gamma)) = \sum_{i=1}^m \frac{1}{2}\left\|L_i^{-1}\xi_i\right\|^2 - \sum_{j=1}^k \log\left[L_i^{-1}\right]_{jj} \tag{A.2.9}$$

$$\begin{aligned}\beta_{k+1} = \underset{\beta}{\operatorname{argmin}}\, \mathcal{L}(\beta, \gamma_k) = \left(\sum_{i=1}^{m} X_i^T \Omega_i^{-1} X_i\right)^{-1} \sum_{i=1}^{m} X_i^T \Omega_i^{-1} y_i = \\ = \left(\sum_{i=1}^{m} (L_i^{-1} X_i)^T L_i^{-1} X_i\right)^{-1} \sum_{i=1}^{m} (L_i^{-1} X_i)^T L_i^{-1} y_i\end{aligned} \tag{A.2.10}$$

The Hessian w.r.t. γ also can be found:

$$\begin{aligned}\frac{\partial^2 \mathcal{L}(\beta,\gamma)}{\partial \gamma_j^2} &= \sum_{i=1}^{m} -2({Z_i^j}^T \Omega_i^{-T} \xi_i) \operatorname{Tr}\left[\frac{\partial {Z_i^j}^T \Omega_i^{-T} \xi_i}{\partial \Omega_i} \frac{\partial \Omega_i}{\partial \Gamma_{jj}}\right] + \operatorname{Tr}\left[\frac{\partial {Z_i^j}^T \Omega_i^{-1} Z_i^j}{\partial \Omega_i} \frac{\partial \Omega_i}{\partial \Gamma_{jj}}\right] = \\ &= \sum_{i=1}^{m} 2({Z_i^j}^T \Omega_i^{-T} \xi_i) \operatorname{Tr}\left[\Omega_i^{-1} Z_i^j {\xi_i}^T \Omega_i^{-1} Z_i^j {Z_i^j}^T\right] - ({Z_i^j}^T \Omega_i^{-T} Z_i^j)^2 = \\ &= \sum_{i=1}^{m} 2({Z_i^j}^T \Omega_i^{-T} \xi_i)({Z_i^j}^T \Omega_i^{-1} Z_i^j)({\xi_i}^T \Omega_i^{-1} Z_i^j) - ({Z_i^j}^T \Omega_i^{-T} Z_i^j)^2\end{aligned} \tag{A.2.11}$$

$$\begin{aligned}\frac{\partial^2 \mathcal{L}(\beta,\gamma)}{\partial \gamma_j \partial \gamma_k} &= \sum_{i=1}^{m} -2({Z_i^j}^T \Omega_i^{-T} \xi_i) \operatorname{Tr}\left[\frac{\partial {Z_i^j}^T \Omega_i^{-T} \xi_i}{\partial \Omega_i} \frac{\partial \Omega_i}{\partial \Gamma_{kk}}\right] + \operatorname{Tr}\left[\frac{\partial {Z_i^j}^T \Omega_i^{-1} Z_i^j}{\partial \Omega_i} \frac{\partial \Omega_i}{\partial \Gamma_{kk}}\right] = \\ &= \sum_{i=1}^{m} 2({Z_i^j}^T \Omega_i^{-T} \xi_i) \operatorname{Tr}\left[\Omega_i^{-1} Z_i^j {\xi_i}^T \Omega_i^{-1} Z_i^k {Z_i^k}^T\right] - ({Z_i^j}^T \Omega_i^{-T} Z_i^k)^2 = \\ &= \sum_{i=1}^{m} 2(\xi_i^T \Omega_i^{-T} Z_i^j)({Z_i^j}^T \Omega_i^{-1} Z_i^k)({Z_i^k}^T \Omega_i^{-1} \xi_i) - ({Z_i^j}^T \Omega_i^{-T} Z_i^k)^2\end{aligned} \tag{A.2.12}$$

$$\begin{aligned}\nabla_\gamma^2 \mathcal{L}(\beta,\gamma) = \frac{1}{2} \sum_{i=1}^{m} -(Z_i^T \Omega_i^{-T} Z_i)^{\circ 2} + 2\operatorname{diag}(()\, Z_i^T \Omega_i^{-T} \xi_i)(Z_i^T \Omega^{-1} Z_i) \operatorname{diag}(()\, \xi_i^T \Omega^{-T} Z_i) = \\ = \frac{1}{2} \sum_{i=1}^{m} -(Z_i^T \Omega_i^{-T} Z_i)^{\circ 2} + 2(Z_i^T \Omega_i^{-T} \xi_i)(\xi_i^T \Omega^{-T} Z_i)^T \circ (Z_i^T \Omega^{-1} Z_i)\end{aligned} \tag{A.2.13}$$

A.3 Derivation of Selected Proximal Operators from Table 2.3.1

SCAD For a scalar variable $x \in \mathbb{R}$, SCAD-regularizer is defined as

$$r(x) = \begin{cases} \sigma|x|, & |x| \le \sigma \\ \frac{-x^2 + 2\rho\sigma|x| - \sigma^2}{2(\rho - 1)}, & \sigma < |x| < \rho\sigma \\ \frac{\sigma^2(\rho+1)}{2}, & |x| > \rho\sigma \end{cases} \tag{A.3.1}$$

To evaluate the $\text{prox}_{\alpha r}$ operator we need to solve the following minimization problem:

$$\min_x r(x) + \frac{1}{2\alpha}(x-z)^2 \tag{A.3.2}$$

For $\alpha = 1$, the solution was derived by Fan (1997). Here we extend it for an arbitrary α. To identify the set of stationary points $\{x^*\}$ of a non-smooth function $f(x)$, we the optimality condition

$$0 \in \partial_x f(x^*) \tag{A.3.3}$$

where $\partial_x f(x)$ denotes a sub-differential set of f at the point x. For the prox problem, we get

$$0 \in \frac{1}{\alpha}(x^* - z) + \partial r(x)_{x=x^*} \tag{A.3.4}$$

Since $r(x)$ is piece-wise defined the precise value of $\partial r(x)_{x=x^*}$ will depend on x^*:

1. Let $0 < x^* \leq \sigma$, then we have $\partial r(x)_{x=x^*} = \{x^*\}$ and so

$$x = z - \sigma\alpha, \quad z \in [\sigma\alpha, \sigma + \sigma\alpha] \tag{A.3.5}$$

2. Let $-\sigma\alpha \leq x^* < 0$, then we have $\partial r(x)_{x=x^*} = \{-x^*\}$ and so

$$x = z + \sigma\alpha, \quad z \in [-\sigma - \sigma\alpha, -\sigma\alpha] \tag{A.3.6}$$

3. Let $x^* = 0$, then $\partial r(x)_{x=x^*} = [-1, 1]$, which yields

$$\frac{1}{\alpha}(x^* - z) \in -\sigma[-1, 1] \quad \Rightarrow \quad z \in [-\sigma\alpha, \sigma\alpha] \tag{A.3.7}$$

4. Let $\sigma < x^* < \rho\sigma$, then $r(x)_{x=x^*} = \frac{-x^{*2} + 2\rho\sigma x^* - \sigma^2}{2(\rho-1)}$, which gives us

$$\frac{1}{\alpha}(x^* - z) = \frac{x^* - \rho\sigma}{\rho - 1} \tag{A.3.8}$$

   To ensure that the stationary point is indeed a minimizer, we need to ensure that

$$\frac{1}{\alpha} - \frac{1}{\rho - 1} > 0 \quad \Rightarrow \quad \alpha < \rho - 1. \tag{A.3.9}$$

   Rearranging the terms to express x^* as a function of z we get

$$x^* = \frac{(\rho-1)z - \lambda\rho\sigma}{\rho - 1 - \alpha} \quad \Rightarrow \quad z \in [\sigma + \alpha\sigma, \rho\sigma] \tag{A.3.10}$$

5. Let $-\rho\sigma < x^* < -\sigma$, then, similarly to the previous case, we get

$$\frac{1}{\alpha}(x^* - z) = \frac{x^* + \rho\sigma}{\rho - 1} \tag{A.3.11}$$

   Rearranging the terms to express x in terms of z we get:

$$x^* = \frac{(\rho-1)z + \lambda\rho\sigma}{\rho - 1 - \alpha} \quad \Rightarrow \quad z \in [-\sigma - \alpha\sigma, -\sigma] \tag{A.3.12}$$

6. Finally, when $|x^*| \geq \sigma\rho$ we have $\partial r(x)_{x=x^*} = \{0\}$ and so

$$x^* = z, \quad |z| \geq \sigma\rho \tag{A.3.13}$$

Bundling all six cases together, we have

$$\text{prox}_{\alpha r}(z) = \begin{cases} \text{sign}(z)(|z| - \sigma\alpha)_+, & |z| \leq \sigma(1+\alpha) \\ \frac{(\rho-1)z - \text{sign}(z)\rho\sigma\alpha}{\rho-1-\alpha}, & \sigma(1+\alpha) < |z| \leq \max(\rho, 1+\alpha)\sigma \\ z, & |z| > \max(\rho, 1+\alpha)\sigma \end{cases} \tag{A.3.14}$$

The middle branch is active only when $\rho > 1 + \alpha$. One special case of this is when $\alpha = 1$, and then (A.3.14) recovers the classic result by Fan and Li (2001).

To get $\text{prox}_{\alpha r + \delta_{\mathbb{R}_+}}(z)$ from $\text{prox}_{\alpha r}(z)$ we only need to notice that (1) the minimizer x^* of

$$\min_x r(x) + \delta_{\mathbb{R}_+} + \frac{1}{\alpha}(x - z)^2 \tag{A.3.15}$$

can never be negative, and that (2) when the minimizer x^* is exactly zero we get:

$$\frac{1}{\alpha}(x^* - z) \in -\partial(r(x)|_{x=x^*} + \delta_{\mathbb{R}_+}(x)|_{x=x^*}) \quad \Rightarrow \quad z \in [-\infty, \sigma\alpha] \tag{A.3.16}$$

A-LASSO A-LASSO regularizer is defined as

$$r(x) = w|x| \tag{A.3.17}$$

where $w = 1/|\hat{x}|$ with $\hat{x}$ the solution of a non-regularized problem (Zou (2006)). The derivation of the proximal operator of A-LASSO nearly matches the steps 1, 2, and 3 that of SCAD above. We wish to evaluate

$$\min_x w|x| + \frac{1}{2\alpha}(x - z)^2 \tag{A.3.18}$$

as a function of z. The sub-differential optimality criterion yields

$$0 \in \frac{1}{\alpha}(x^* - z) + w\partial|x| \tag{A.3.19}$$

1. Let $0 < x^*$, then we have $\partial r(x)_{x=x^*} = \{x^*\}$ and so

$$x^* = z - \alpha w, \quad z > \alpha w \tag{A.3.20}$$

2. Let $x^* < 0$, then we have $\partial r(x)_{x=x^*} = \{-x^*\}$ and so

$$x^* = z + \alpha w, \quad z < -\alpha w \tag{A.3.21}$$

3. Let $x^* = 0$, then $\partial r(x)_{x=x^*} = [-1, 1]$, which yields

$$\frac{1}{\alpha}(x^* - z) \in [-w, w] \quad \Rightarrow \quad z \in [-\alpha w, \alpha w] \tag{A.3.22}$$

Combining all cases together we get

$$\text{prox}_{\alpha r}(z) = \text{sign}(z)(|z| - \alpha w)_+ \tag{A.3.23}$$

Finally, $\text{prox}_{\alpha r+\delta_{\mathbb{R}}}(z)$ can be derived by noticing that, in this case, (1) $x^* \geq 0$, and (2) when $x^* = 0$ the sub-differential changes due to the presence of the delta-function:

$$x^* = 0 \implies \frac{1}{\alpha}(x^* - z) \in -([-\alpha w, \alpha w] + [-\infty, 0]) = [-\alpha w, +\infty] \tag{A.3.24}$$

which gives us the condition

$$x^* = z, \quad z \in [-\infty, \alpha w]. \tag{A.3.25}$$

LASSO LASSO is a particular case of A-LASSO above when $w = 1$.

ℓ_0-regularizer Comparing to its counterparts above, the regularizer $R(x) = \delta_{\|x\| \leq k}(x)$ is non-separable. However, the proximal operator of it can still be evaluated analytically:

$$[\text{prox}_{\alpha R}(z)]_i = \left[\underset{\|x\| \leq k}{\text{argmin}} \frac{1}{2\alpha}\|x - z\|^2\right]_i = \begin{cases} z_i, & i \in \mathcal{I}_k \\ 0, & \text{otherwise} \end{cases} \tag{A.3.26}$$

where $\mathcal{I}_k$ is a set of k largest in their absolute value coordinates of z. To get $\text{prox}_{\alpha R+\delta_{\mathbb{R}_+}}$ we replace $\mathcal{I}_k$ with a set of k largest positive coordinates of z, and set the rest of the coordinates to 0.

A.4 Existence of Minimizers (Theorems 1 and 2)

The key tool to prove existence of minimizers for both the likelihood and the penalized likelihood is the function $f : \mathbb{R}^n \times \mathbb{S}^n_{++} \to \mathbb{R}$ given by

$$f(r, M) := \frac{1}{2}[r^T M^{-1} r + \ln |M|] \,. \tag{A.4.1}$$

If $M = U\text{Diag}\,(\mu)\,U^T$ is the eigenvalue decomposition for M where $U^T U = I$, and $\tilde{r} = U^T r$, then

$$f(r, M) = \frac{1}{2}\left[\sum_{i=1}^{n} \frac{\tilde{r}_i^2}{\mu_i} + \ln(\mu_i)\right]. \tag{A.4.2}$$

For $\rho > 0$, observe that $\frac{\rho^2}{\omega} + \ln(\omega)$ is greater that both $\ln(\omega)$ and $1 + 2\ln(\rho)$ for all $\omega > 0$. Therefore, using the facts $\mu_{\max}(M) = \|M\|$ and $\|\tilde{r}\|_\infty \geq (\|\tilde{r}\|/\sqrt{n}) = (\|r\|/\sqrt{n})$, we have

$$\begin{aligned} f(r, M) &\geq \frac{1}{2}\sum_{i=1}^{n} \max\{\ln \mu_i,\, 1 + 2\ln|\tilde{r}_i|\} \\ &\geq \max\{1 + 2\ln(\|r\|/\sqrt{n}) + \frac{n-1}{2}\ln \mu_{\min}(M),\ \ln\|M\| + \frac{n-1}{2}\ln \mu_{\min}(M)\} \\ &\geq \max\{\ln(\|r\|^2/n), \ln\|M\|\} + \frac{n-1}{2}\ln \mu_{\min}(M), \end{aligned} \tag{A.4.3}$$

where $\mu_{\min}(M)$ and $\mu_{\max}(M)$ are the smallest and largest eigenvalue of M, respectively. We have the following result due to Zheng et al. (2021) modified slightly with an independent proof.

Lemma 18 (Level Compactness of f). *(Zheng et al., 2021, Theorem 1) Let f be as given in* (A.4.1). *Then, given $\rho \in \mathbb{R}$ and $\alpha > 0$, the set*

$$\mathcal{D}_{\rho,\alpha} := \left\{(r, M) \in \mathbb{R}^n \times \mathbb{S}^n_{++} \mid f(r, M) \leq \rho \text{ and } \mu_{min}(M) \geq \alpha\right\}$$

is compact, where $\mu_{min}(M)$ and $\mu_{max}(M)$ are the smallest and largest eigenvalue of M, respectively.

Proof. If $\mathcal{D}_{\rho,\alpha} = \emptyset$, it is compact so we assume it is not empty. Since f is continuous on $\mathcal{D}_{\rho,\alpha}$, we need only show that this set is bounded. The boundedness of this set follows immediately from (A.4.3). Indeed, if $\{(r^k, M_k)\} \subset \mathbb{R}^n \times \mathbb{S}^n_{++}$ diverges in norm then, without loss of generality, either $\left\|r^k\right\| \to \infty$ or $\mu_{\max}(M) = \|M_k\| \to \infty$, or both in which case (A.4.3) tells us that $f(r^k, M_k) \to \infty$. □

Observe that

$$\mathcal{L}_{ML}(\beta, \Gamma) = f(r(\beta), \Omega(\Gamma))$$

where $r : \mathbb{R}^p \to n$ and $\Omega : \mathbb{R}^q \to \mathbb{S}^n$ are the affine transformations

$$\begin{aligned} r(\beta) &:= X\beta - Y, \qquad \text{and} \\ \Omega(\Gamma) &:= \operatorname{Diag}\left(\Lambda_1 + Z_1\Gamma Z_1^T, \ldots, \Lambda_m + Z_m\Gamma Z_m^T\right). \end{aligned}$$

For $i = 1, \ldots, m$, define

$$\omega^i_{\min} := \mu_{\min}(\Lambda_i) + \mu_{\min}(\Gamma)\sigma^2_{\min}(Z_i) \quad \text{and} \quad \omega_{\min} := \min_{i=1,\ldots,m} \omega^i_{\min},$$

where $\mu_{\min}(\Psi)$ and $\sigma_{\min}(\Phi)$ are the smallest eigenvalues and singular-values of Ψ and Φ, respectively. By (Aravkin et al., 2021, Theorem 3.1),

$$0 < \tilde{\alpha} := \mu_{\min}(\Lambda) \leq \omega_{\min} \leq \mu_{\min}(\Omega(\Gamma)) \quad \forall\, \Gamma \in \mathbb{S}^q_+. \tag{A.4.4}$$

Proof for Theorem 1 The bound (A.4.4) tell us that

$$\hat{\mathcal{D}} := \{(r, \Omega(\Gamma)) \mid r \in \mathbb{R}^n,\, \Gamma \in \mathbb{S}^q_+ \text{ and } f(r, \Omega(\Gamma)) \leq \rho\} \subset \mathcal{D}_{\tilde{\alpha},\rho}.$$

In particular, $\mathcal{L}$ is bounded below by (A.4.3). Hence there exists a sequence $\{(\beta^k, \Gamma^k)\} \subset \mathbb{R}^p \times \mathbb{S}^q_+$ such that

$$\mathcal{L}_{ML}(\beta^k, \Gamma_k) \downarrow \inf_{\beta \in \mathbb{R}^p, \Gamma \in \mathbb{S}^q_+} \mathcal{L}_{ML}(\beta, \Gamma).$$

Let $\rho = \mathcal{L}_{ML}(\beta^0, \Gamma_0)$. Since f is continuous on $\hat{\mathcal{D}} \subset \mathcal{D}_{\tilde{\alpha},\rho}$, $\mathcal{D}_{\tilde{\alpha},\rho}$ is compact by Lemma 18, and both $\operatorname{Im}(X)$ and $\operatorname{Im}(\Omega)$ are closed, with no loss in generality there is a $(\bar{\xi}, \overline{\Omega}) \in \operatorname{Im}(r) \times \operatorname{Im}(\Omega) \cap \mathcal{D}_{\tilde{\alpha},\rho}$ such that $(r(\beta^k), \Omega(\Gamma_k)) \to (\bar{\xi}, \overline{\Omega})$. Since $(\bar{\xi}, \overline{\Omega}) \in \operatorname{Im}(r) \times \operatorname{Im}(\Omega)$, there is a $(\bar{\beta}, \overline{\Gamma}) \in \mathbb{R}^p \times \mathbb{S}^q_+$ such that $(\bar{\xi}, \overline{\Omega}) = (r(\bar{\beta}), \Omega(\overline{\Gamma}))$. In addition, since $0 < \tilde{\alpha} \leq \mu_{\min}(\Omega(\Gamma))$ for all $\Gamma \in \mathbb{S}^q_+$, we have $\mathcal{L}_{ML}$ is lsc at $(\bar{\beta}, \overline{\Gamma})$ telling us that $\mathcal{L}(\bar{\beta}, \overline{\Gamma}) = \inf_{\beta \in \mathbb{R}^p, \Gamma \in \mathbb{S}^q_+} \mathcal{L}(\beta, \Gamma)$.

Proof of Theorem 2 Define the affine transformations $\widehat{\Omega} : \mathbb{R}^q \to \mathbb{S}^n$ and $\widehat{\Omega}_i : \mathbb{R}^q \to \mathbb{S}^{n_i}$ by

$$\widehat{\Omega}(\gamma) := \Omega(\text{Diag}\,(\gamma)) \quad \text{and} \quad \widehat{\Omega}_i(\gamma) := \Omega_i(\text{Diag}\,(\gamma)) \quad i = 1, \dots, m. \tag{A.4.5}$$

The existence of a solution follows immediately once the level compactness of $\mathcal{L} + \widehat{R}$ is establinshed. To this end observe that $\mathcal{L}(\beta, \gamma) = \mathcal{L}_{ML}(\beta, \text{Diag}\,(\gamma)) = f(r(\beta), \widehat{\Omega}(\gamma))$ and so (A.4.3) and (A.4.4) tell us that $\mathcal{L}(\beta, \gamma) \geq \frac{n+1}{2} \ln \tilde{\alpha}$. Since $\widehat{R}$ is level compact, it is lower bounded. Therefore, $\mathcal{L} + \widehat{R}$ is bounded below. Let $\rho \in \mathbb{R}$ and $\{(\beta^k, \gamma^k)\} \subset \left\{(\beta, \gamma) \;\middle|\; \mathcal{L}(\beta, \gamma) + \widehat{R}(\beta, \gamma) \leq \rho\right\}$. We need to show that $\{(\beta^k, \gamma^k)\}$ is bounded. If $\left\|(\beta^k, \gamma^k)\right\| \to \infty$, then $\widehat{R}(\beta^k, \gamma^k) \to \infty$. Since $\mathcal{L}(\beta^k, \gamma^k) + \widehat{R}(\beta^k, \gamma^k) \leq \rho$, we must have $\mathcal{L}(\beta^k, \gamma^k) \to -\infty$. But $\mathcal{L}$ is bounded below, hence $\{(\beta^k, \gamma^k)\}$must be bounded, and so $\mathcal{L} + \widehat{R}$ is level compact.

A.5 Lipschitz-constant for Likelihood of a Linear Mixed-Effects Model

Recall that a function $\mathcal{L}(x)$ is called L-Lipschitz smooth when

$$\|\nabla \mathcal{L}(x) - \nabla \mathcal{L}(y)\|_2 \leq L \|x - y\|_2 \tag{A.5.1}$$

To find the Lipschitz-constant of the function $\mathcal{L}_{ML}$ (2.2.3) we will use the fact that $\mathcal{L}(x)$ is L-Lipschitz if and only if $\|\nabla^2 \mathcal{L}(x)\| \leq L$ for any x. Hence, to upper-bound L we need to upper-bound the norms of Hessians. Assume that $\|y_i - X_i\beta\| \leq \rho$ where $\rho > 0$. We get

$$\begin{aligned} \left\|\nabla^2 \mathcal{L}(x)\right\|_2 &= \left\| \begin{bmatrix} \nabla^2_{\beta\beta} \mathcal{L}(\beta, \gamma) & \nabla^2_{\beta\gamma} \mathcal{L}(\beta, \gamma) \\ \nabla^2_{\gamma\beta} \mathcal{L}(\beta, \gamma) & \nabla^2_{\gamma\gamma} \mathcal{L}(\beta, \gamma) \end{bmatrix} \right\| \leq \sum_{i=1}^m \left\| \begin{bmatrix} \frac{\|X_i\|_2^2}{\|\Lambda_i\|_2} & \frac{\rho \|X_i\|_2 \|Z_i\|_2^2}{\|\Lambda_i\|^2} \\ \frac{\rho \|X_i\|_2 \|Z_i\|_2^2}{\|\Lambda_i\|^2} & \frac{\rho \|Z_i\|_2^4}{\|\Lambda_i\|_2^3} \end{bmatrix} \right\| \\ &\leq \sum_{i=1}^m \max\left(\frac{\|X_i\|_2^2}{\|\Lambda_i\|_2}, \frac{\rho \|X_i\|_2 \|Z_i\|_2^2}{\|\Lambda_i\|^2}, \frac{\rho \|X_i\|_2 \|Z_i\|_2^2}{\|\Lambda_i\|^2}, \frac{\rho \|Z_i\|_2^4}{\|\Lambda_i\|_2^3} \right) = L \end{aligned} \tag{A.5.2}$$

The assumption $\|y_i - X_i\beta\| \leq \rho$ is typically enforced artificially by introducing a box-constraint for β. Unfortunately, if the assumption is violated then $\nabla \mathcal{L}$ is not globally Lipschitz on its domain, as noted in Aravkin et al. (2022a). Nonetheless, it is possible to obtain convergence results with the inclusion of a line search or trust region strategy, see e.g. Burke and Engle (2018).

Model	Regularizer Metric	L0	L1	ALASSO	SCAD
PGD	Accuracy	89 (75-95)	73 (68-82)	88 (72-98)	71 (62-78)
	FE Accuracy	88 (70-95)	56 (45-70)	84 (65-100)	53 (45-65)
	RE Accuracy	90 (75-100)	91 (80-100)	92 (80-100)	89 (75-100)
	F1	88 (74-95)	77 (71-83)	88 (74-97)	75 (68-80)
	FE F1	87 (72-95)	67 (62-75)	85 (70-100)	66 (62-72)
	RE F1	89 (74-100)	91 (78-100)	91 (78-100)	88 (74-100)
	Time	47.47 (20.22-78.43)	43.02 (23.02-67.01)	38.68 (20.52-58.26)	87.24 (40.73-160.34)
	Iterations	29662 (20985-43234)	31693 (22361-45603)	28912 (20915-39210)	41724 (26911-69881)
MSR3	Accuracy	92 (75-98)	89 (72-100)	91 (75-98)	92 (75-100)
	FE Accuracy	92 (70-100)	85 (60-100)	91 (70-100)	93 (70-100)
	RE Accuracy	91 (78-95)	92 (75-100)	91 (75-100)	92 (80-100)
	F1	91 (76-97)	89 (73-100)	91 (76-98)	92 (76-100)
	FE F1	92 (75-100)	87 (69-100)	92 (75-100)	93 (75-100)
	RE F1	90 (74-94)	91 (74-100)	90 (73-100)	91 (75-100)
	Time	109.86 (5.49-335.01)	13.74 (3.12-31.69)	81.52 (5.94-232.98)	104.20 (6.46-308.19)
	Iterations	1135 (27-3148)	126 (41-314)	895 (81-2262)	1182 (47-3146)
MSR3-fast	Accuracy	92 (75-100)	88 (68-100)	91 (75-98)	92 (75-100)
	FE Accuracy	92 (65-100)	85 (60-100)	91 (70-100)	94 (75-100)
	RE Accuracy	93 (85-100)	91 (75-100)	92 (75-100)	91 (70-100)
	F1	92 (76-100)	88 (71-100)	91 (75-97)	92 (74-100)
	FE F1	92 (72-100)	87 (69-100)	91 (75-100)	94 (78-100)
	RE F1	92 (82-100)	90 (74-100)	90 (74-100)	90 (71-100)
	Time	0.36 (0.15-0.57)	0.35 (0.15-0.56)	0.45 (0.18-0.55)	0.45 (0.16-0.77)
	Iterations	86 (41-119)	87 (43-123)	115 (45-119)	102 (49-145)

Table A.6.1: Comparison of performance of algorithms

A.6 Detailed Results from Simulation from Table 2.4.1

A.7 Description of Real-World Datasets

A.7.1 GBD Bullying Data

The author acknowledges his colleague and collaborator Damian Santomauro[1] for providing the dataset, the description of its covariates, and the expert assessment of their historical importance in different rounds of GBD study below.

1. `cv_symptoms`

    - 0 = study assesses participants for MDD or anxiety disorders via a diagnostic interview to determine whether they have a diagnosis.

[1] d.santomauro@uq.edu.au, Affiliate Assistant Professor of Health Metrics Sciences, Institute for Health Metrics and Evaluation, University of Washington

- 1 = study uses a symptom scale (e.g., Beck Depression Inventory) and uses an established cut-off on that scale to determine caseness.
- Has not historically been significant.

2. `cv_unadjusted`
   - 0 = RR is adjusted for potential confounders (e.g., SES, etc.)
   - 1 = RR is not adjusted for potential confounders
   - Has been significant in the past.
3. `cv_b_parent_only`
   - 0 = Child is involved in reporting their own exposure to bullying.
   - 1 = Only parent is involved in reporting the child's exposure to bullying
   - This covariate has recently started becoming significant (but not consistently).
4. `cv_or`
   - 0 = estimate is a RR
   - 1 = estimate is an odds ratio (OR)
   - ORs are always larger than RRs. However the magnitude may be very small / insignificant.
5. `cv_multi_reg`
   - 0 = RR is the ratio of the rate of the outcome in persons exposed vs all persons unexposed (including persons exposed to low-threshold bullying victimization)
   - 1 = RRs are estimated via a logistic regression where exposure represented by 3 categories: 1) No exposure, 2) Occasional exposure, 3) Frequent exposure. The RR for occasional exposure will exclude participants with frequent exposure, and the RR for frequent exposure will exclude participants with occasional exposure.
   - Is expected to be significant.
6. `cv_low_threshold_bullying`
   - 0 = uses a 'frequent' exposure frequency threshold for classing someone as exposed to bullying.
   - 1 = uses an 'occasional' exposure frequency threshold for classing someone as exposed to bullying.
   - Has been consistently significant with a strong magnitude.
7. `cv_anx`

- 0 = estimate represents risk for MDD
- 1 = estimate represents risk for anxiety disorders

8. `cv_selection_bias`
   - 0 = $<$ 15% attrition at followup
   - 1 = $\geqslant$ 15% attrition at followup
   - Has been significant in the past
9. `Percent_female`
   - Indicates % of sample in estimate that are female.
10. `cv_child_baseline`
    - Has not been significant in the past.

A.7.2 COVID-19 Contact Rate Forecasting Data

Table A.7.1: List of locations, number of observations, start and end date for each location for COVID-19 Contact Rate Forecasting data

Location	Obs	Start	End
Malaysia	60	2020-02-27	2020-04-26
Philippines	67	2020-02-21	2020-04-27
Bulgaria	50	2020-03-09	2020-04-27
Croatia	50	2020-03-08	2020-04-26
Czechia	54	2020-03-05	2020-04-27
Hungary	55	2020-03-04	2020-04-27
Poland	56	2020-03-03	2020-04-27
Romania	56	2020-03-03	2020-04-27
Serbia	55	2020-03-04	2020-04-27
Slovakia	32	2020-03-26	2020-04-26
Slovenia	54	2020-03-05	2020-04-27
Estonia	48	2020-03-10	2020-04-26
Latvia	26	2020-04-01	2020-04-26
Lithuania	53	2020-03-05	2020-04-26
Republic of Moldova	48	2020-03-11	2020-04-27
Ukraine	53	2020-03-06	2020-04-27
Japan	68	2020-02-20	2020-04-27

Continued on next page

Table A.7.1: List of locations, number of observations, start and end date for each location for COVID-19 Contact Rate Forecasting data

Location	Obs	Start	End
Republic of Korea	85	2020-02-02	2020-04-26
Austria	62	2020-02-26	2020-04-27
Belgium	65	2020-02-23	2020-04-27
Cyprus	49	2020-03-09	2020-04-26
Denmark	61	2020-02-27	2020-04-27
Finland	53	2020-03-06	2020-04-27
France	63	2020-02-24	2020-04-26
Greece	62	2020-02-26	2020-04-27
Iceland	43	2020-03-15	2020-04-26
Ireland	58	2020-03-01	2020-04-27
Israel	56	2020-03-03	2020-04-27
Luxembourg	58	2020-02-29	2020-04-26
Netherlands	61	2020-02-27	2020-04-27
Norway	62	2020-02-26	2020-04-27
Portugal	58	2020-03-01	2020-04-27
Sweden	63	2020-02-25	2020-04-27
Switzerland	69	2020-02-19	2020-04-27
United Kingdom	70	2020-02-18	2020-04-27
Argentina	56	2020-03-03	2020-04-27
Chile	54	2020-03-05	2020-04-27
Dominican Republic	58	2020-03-01	2020-04-27
Ecuador	50	2020-03-01	2020-04-19
Peru	55	2020-03-04	2020-04-27
Colombia	55	2020-03-04	2020-04-27
Panama	50	2020-03-09	2020-04-27
Egypt	68	2020-02-20	2020-04-27
Iran (Islamic Republic of)	69	2020-02-19	2020-04-27
Turkey	48	2020-03-11	2020-04-27
Puerto Rico	45	2020-03-14	2020-04-27
Alabama	48	2020-03-11	2020-04-27
Alaska	49	2020-03-09	2020-04-26
Arizona	55	2020-03-04	2020-04-27
Arkansas	52	2020-03-07	2020-04-27
California	67	2020-02-21	2020-04-27
Colorado	54	2020-03-05	2020-04-27

Continued on next page

Table A.7.1: List of locations, number of observations, start and end date for each location for COVID-19 Contact Rate Forecasting data

Location	Obs	Start	End
Connecticut	52	2020-03-07	2020-04-27
Delaware	51	2020-03-08	2020-04-27
District of Columbia	54	2020-03-05	2020-04-27
Florida	57	2020-03-02	2020-04-27
Georgia	56	2020-03-03	2020-04-27
Hawaii	43	2020-03-15	2020-04-26
Idaho	50	2020-03-08	2020-04-26
Illinois	55	2020-03-04	2020-04-27

Table A.7.2: List of location-specific coefficients for the R&S-Mixed model fit, as well as RMSEs for three models discussed in the respective chapterCoefficient for `temperature` was set to -674.86. Coefficients for `proportion_over_1k` and `testing_reference` were set to 0.

Location	Intercept	Mobility	RMSE_IHME	RMSE_Dense	RMSE_Sparse
Malaysia	13.94	52.15	5.17	5.00	5.00
Philippines	13.60	29.45	4.20	4.16	4.16
Bulgaria	14.24	123.94	3.42	3.20	3.20
Croatia	13.28	56.84	3.70	3.67	3.67
Czechia	13.77	103.24	2.86	3.07	3.12
Hungary	12.51	28.59	0.73	0.70	0.72
Poland	12.66	39.47	0.64	0.58	0.62
Romania	13.63	80.40	3.59	3.89	4.01
Serbia	13.16	56.92	3.79	3.80	3.84
Slovakia	11.27	-43.53	5.27	4.50	5.01
Slovenia	12.74	42.75	1.46	1.40	1.56
Estonia	14.17	155.26	2.21	2.55	2.73
Latvia	11.82	-14.61	4.94	4.41	4.68
Lithuania	12.96	66.04	3.89	3.86	3.86
Republic of Moldova	15.15	153.32	3.05	2.44	2.44
Ukraine	13.54	103.82	3.06	3.29	3.30
Japan	12.47	35.51	4.21	4.22	4.22
Republic of Korea	12.62	99.62	4.84	4.67	4.70
Austria	13.07	64.96	3.97	3.90	3.93
Belgium	13.44	51.77	3.47	3.39	3.39
Cyprus	12.65	25.21	1.84	0.64	0.66

Continued on next page

Table A.7.2: List of location-specific coefficients for the R&S-Mixed model fit, as well as RMSEs for three models discussed in the respective chapterCoefficient for `temperature` was set to -674.86. Coefficients for `proportion_over_1k` and `testing_reference` were set to 0.

Location	Intercept	Mobility	RMSE_IHME	RMSE_Dense	RMSE_Sparse
Denmark	12.79	47.14	1.79	1.87	1.94
Finland	12.87	73.08	3.53	3.63	3.68
France	12.95	32.79	1.57	1.64	1.66
Greece	12.74	29.97	1.42	1.56	1.56
Iceland	16.35	226.87	5.77	3.09	3.17
Ireland	13.54	57.98	3.98	4.00	4.00
Israel	13.83	66.97	3.46	3.71	3.71
Luxembourg	12.51	21.39	1.47	1.53	1.73
Netherlands	12.90	52.99	1.35	1.46	1.47
Norway	12.22	34.27	1.70	1.64	1.65
Portugal	13.51	52.51	2.47	2.55	2.55
Sweden	12.95	99.37	3.93	3.80	3.90
Switzerland	12.68	66.51	3.79	3.88	3.98
United Kingdom	13.28	51.22	4.70	4.71	4.72
Argentina	13.29	36.93	1.45	1.63	1.67
Chile	13.64	73.48	3.51	3.62	3.63
Dominican Republic	13.78	40.14	2.21	2.23	2.23
Ecuador	15.97	128.02	8.21	8.07	8.07
Peru	12.97	16.00	1.08	0.96	1.18
Colombia	13.88	47.20	3.63	3.63	3.63
Panama	13.12	10.67	0.25	0.27	0.27
Egypt	13.11	41.80	3.89	3.95	3.96
Iran (Islamic Republic of)	12.27	15.23	0.93	1.03	1.05
Turkey	12.60	28.83	0.27	0.18	0.17
Puerto Rico	13.76	45.18	0.68	0.34	0.35
Alabama	12.81	27.02	0.70	0.65	0.84
Alaska	12.24	57.82	3.85	3.98	4.00
Arizona	13.40	66.31	4.00	3.85	4.03
Arkansas	13.32	91.92	4.02	3.87	3.88
California	13.14	54.89	3.81	3.86	3.88
Colorado	12.45	37.13	0.75	1.01	1.03
Connecticut	13.28	69.51	0.59	0.77	0.92
Delaware	13.34	68.28	3.86	3.77	3.83
District of Columbia	13.27	49.45	3.42	3.55	3.56

Continued on next page

Table A.7.2: List of location-specific coefficients for the R&S-Mixed model fit, as well as RMSEs for three models discussed in the respective chapterCoefficient for `temperature` was set to -674.86. Coefficients for `proportion_over_1k` and `testing_reference` were set to 0.

Location	Intercept	Mobility	RMSE_IHME	RMSE_Dense	RMSE_Sparse
Florida	13.34	30.01	0.29	0.34	0.36
Georgia	12.74	19.19	0.45	0.34	0.66
Hawaii	14.97	134.74	3.74	3.34	3.39
Idaho	12.75	92.74	3.81	3.84	3.89
Illinois	12.73	32.57	0.64	0.56	0.70

A.8 Detailed Design of Experiments for PINODE

Hardware All experiments were computed using an `Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20 GHz` equipped with a `Tesla K80` GPU. The computer had `Linux 4.15` installed as an OS.

A.8.1 Lifted Duffing Oscillator: Learning Unseen Basins with Collocations (Figure 3.2.3)

In this experiment we trained two models; both share the same architecture but differ in the input data, as described in Chapter 3.3.1.

Architecture The network consists of two blocks: the autoencoder pair $\phi_\theta(\boldsymbol{x})$, $\psi_\theta(\boldsymbol{z})$, and the latent dynamics $h_\theta(\boldsymbol{z})$, where θ represents the combined weights of all networks.

Both ϕ and ψ were fully-connected networks with three layers, the input-output dimension of 128, the bottleneck-space dimension of 2, and the hidden-layer dimensions of 256. The hidden layers had ReLU activations except for the output layers of ϕ and ψ which had linear activation.

The network h was a fully-connected network with three layers, with the input and output dimension of 2 and the hidden-layer dimensions of 128. All hidden layers had ReLU activation, the output layer had a linear activation.

We used standard network classes of `pytorch`[2] by Paszke et al. (2019b) to implement the networks, and we used a differentiable integrator `torchdiffeq`[3] by Chen et al. (2018b) for evaluating derivatives of the loss function.

[2] https://pytorch.org
[3] https://github.com/rtqichen/torchdiffeq

Data For data **snapshots** we used 6144 trajectories 10 steps long each, with a step-size $dt = 0.1$. For **collocations** we generated a set of 10^5 2-dimensional points $\bar{\boldsymbol{z}}_j \in U([-3/2, 3/2] \times [-1, 1])$, excluded those belonging to the left (red) lobe, and projected them to the observable space $\mathcal{X}$ using the true decoder (3.3.2). We then used those high-dimensional points $\bar{\boldsymbol{x}}_j$ as collocations.

Training We trained the combined model for 400 epochs, with the learning rate of 10^{-4}. All weights ω_i were set to 1 for a hybrid model, and ω_2, ω_4 were set to 0 for a data-driven model.

In each batch we had 64 trajectories and, in case of hybrid models, 640 collocation points. The rationale behind this ratio is that one trajectory contains number-of-steps snapshots of the system. Hence, to balance the amount of information that comes from both sources within a batch, we were taking the number-of-steps more collocations than trajectories for every batch. The same rationale holds for all other instances of training of a hybrid model in this paper.

A.8.2 Lifted Duffing Oscillator: Far-Out Forecasting (Figure 3.3.2).

In this experiment we trained three models, all sharing the same architecture but differ in their loss functions, namely in the coefficients w_i.

Architecture The networks architectures match to the one described in Appendix A.8.1.

Data For data **snapshots** we used 6144 trajectories (2048 for each of the attractors) 10 steps long each, with a step-size $dt = 0.1$. For **collocations** we generated a set of 10^5 2-dimensional points $\bar{\boldsymbol{z}}_j \in U([-3/2, 3/2] \times [-1, 1])$ and projected them to the observable space $\mathcal{X}$ using the true decoder (3.3.2). We then used those high-dimensional points $\bar{\boldsymbol{x}}_j$ as collocations.

Training We trained the combined model for 400 epochs, with the learning rate of 10^{-4}. All weights ω_i were set to 1 for a hybrid model, ω_3, ω_4 were set to 0 for a data-driven model, and $\omega_1 = \omega_2 = 0$ for Physics-Informed model.

A.8.3 Lifted Duffing Oscillator: Role of Non-Linear Latent Dynamics (Figure 3.3.1).

In this experiment we trained three models: DMD, PIKN, and PINODE.

Architecture The PINODE model's architecture matches to the one described in Appendix A.8.1. The PIKN model's architecture is the same except that the latent dynamics $h(z) = Lz$ is linear: it consists of one fully-connected linear layer of width 16 with no bias and

no activation function. DMD model had the latent space of 16; the implementation is faithful to the original works Kutz et al. (2016).

Data The datasets for both trajectories and collocations match to the ones described in Apendix A.8.2

Training We trained the combined model for 400 epochs, with the learning rate of 10^{-4}. All weights ω_i were set to 1 for both PIKN and PINODE. DMD model used no collocations, whereas PIKN and PINODE used both trajectories and collocations.

A.8.4 Burgers' Equation: Compressibility (Figure 3.3.3)

In this section we study how efficiently different ROMs use the same size of the latent space.

Architecture In **PINODE** and **PIKN** models, both ϕ and ψ were fully-connected networks with three layers, the input-output dimension of 128 and the hidden-layer dimensions of 512. The hidden layers had ReLU activation except for the output layers of ϕ and ψ which had linear activation. The size of the latent space was varying from 2 to 512, see the x-axis of Figure (3.3.3). In **PINODE**, the network h was a fully-connected network with three layers with the hidden-layer dimensions of 512. All hidden layers had ReLU activation, the output layer had a linear activation. For **PIKN** the network had one layer of the latent space size with no bias and linear activation. In other words, $h(z) = Az$. For **DMD** we used a classic algorithm from Kutz et al. (2016), and set the number of DMD modes to be equal to the size of the latent dimension.

Data For **snapshots**, we generated 16384 trajectories of the system for our train dataset and 300 trajectories for our test dataset. Each trajectory had 40 time-steps with $dt = 0.1$. We used randomly-generated functions from Equation 3.3.5 as initial conditions for the trajectories. We used only first 20 time-steps of train trajectories for training. We used the first 20 steps of the test trajectories to evaluate *interpolation* performance of the model, and the next 20 time-steps for evaluating *extrapolation* performance. All performance measures on Figure (A.8.4) are based on the test dataset. For **collocations**, we used Equation (3.3.6) to generate 10^5 collocation points.

Training We trained models for 500 epochs, with the learning rate of 10^-4. All weights ω_i were set to 1 for a hybrid model, and ω_3, ω_4 were set to 0 for a data-driven model. Every batch contained 64 trajectories and 1280 (that is, 64*20) collocation points, with the same rationale as in Appendix A.8.1.

A.8.5 Burger's Equation: Compressibility of Linear Latent Space for PIKN

Architecture In the PIKN architecture for Burger's equation, the encoder is implemented as a a feed-forward network with the size of input layer set to be equal 128 (the number of spatial grid-points). It has two hidden layers with 512 neurons. The size of the output layer of the encoder – the size of the latent dimension – varies from 16 to 256 to study the latent space compressibility (see Figure 3.3.4). The decoder's architecture mirrors the architecture of the encoder with the sizes of the inputs and outputs switched. An Adam optimizer has been applied for 500 epochs for training. We use an adaptive learning rate: initially set to 10^{-4}, it keeps decreasing by a factor of 0.5 if no improvement has been made over the recent 20 epochs, until it hits the minimal value of 10^{-7}.
Each network was trained in two different ways. Namely, we set different values for the weights $\omega_1, \omega_2, \omega_3, \omega_4$ in the loss function, leading to two different training regimes:

- $\omega_1 = 0, \omega_2 = 0, \omega_3 = 1, \omega_4 = 1$. These settings lead to a purely data-driven learning; it corresponds to green lines on Figure 3.3.4.
- $\omega_1 = 1, \omega_2 = 1, \omega_3 = 1, \omega_4 = 1$. These settings represent a model that uses both data trajectories and collocations (a hybrid model); it corresponds to blue lines on Figure 3.3.4.

Training We sample 80000 functions $\{u^{(1)}, u^{(2)}, \dots, u^{(80000)}\}$ to use them as collocations. Each function $u^{(j)}$ is a superposition of the Fourier modes that satisfy the periodic boundary conditions. We call them "harmonic" initial conditions, five examples of which are displayed on the middle-right pane of Figure 3.3.4. We used $\sin(kx)$ and $\cos(kx)$ for $k = 0, 1, 2, \dots$ evaluated on $x \in [-\pi; \pi]$-interval as basis functions. The value of k is restricted to be no greater than 10. To evaluate u_t at each collocation point we evaluated the spatial derivatives $\{(u_x^{(i)}, u_{xx}^{(i)})\}_{i=1}^{80000}$ using a numerical spectral method.

For training trajectories, we use simulation data obtained from a solver base on spectral method. We run the simulation with 1024 different initial states. Each initial state of u is a Gaussian (bell-) curve with mean 0 and a randomly generated variance $\sigma \sim U(0.1, 1)$, evaluated on $[-\pi; \pi]$-interval. Then we sample snapshots of the state u with a temporal gap $\Delta t = 0.1$ for 20 steps (i.e. $p = 20$). An example trajectory is displayed on the top-right pane of Figure 3.3.4.

To evaluate and compare the performance of both models we use 200 trajectories: 100 trajectories with harmonic ICs and 100 trajectories with Gaussian ICs generated exactly in the same way as described above. Then we sample snapshots of the state u with a temporal gap $\Delta t = 0.1$ for 20 steps (i.e. $p = 20$).

Results The results are aggregated in Table A.8.1 and visualized on Figure A.8.1.

Latent space	Model	Train Time	Test MSE	Test Harmonic MSE	Test Gaussian MSE
16	Data-Driven	0 hours, 18 minutes	0.04124	0.082426	0.000054
	Hybrid	0 hours, 37 minutes	0.005451	0.010826	0.000077
32	Data-Driven	0 hours, 19 minutes	0.039412	0.078771	0.000054
	Hybrid	1 hours, 24 minutes	0.003815	0.00757	0.000061
64	Data-Driven	0 hours, 21 minutes	0.036496	0.072964	0.000028
	Hybrid	1 hours, 40 minutes	0.003513	0.006964	0.000062
128	Data-Driven	0 hours, 22 minutes	0.033983	0.067933	0.000033
	Hybrid	4 hours, 28 minutes	0.003334	0.006608	0.00006
256	Data-Driven	0 hours, 38 minutes	0.035508	0.070988	0.000028
	Hybrid	6 hours, 11 minutes	0.003396	0.006745	0.000048

Table A.8.1: Performance measures for Burger's experiment depending on the dimensions of the latent space (Figure 3.3.4)

A.8.6 Burgers' Equation: Data-vs-Collocations (Figure 3.3.6)

In this section we study the relative impact of data and collocations as training datasets to the performance of the resulting models.

Architecture In **PINODE** models both ϕ and ψ were fully-connected networks with three layers, the input-output dimension of 128, the latent-space dimension of 16 and the hidden-layer dimensions of 512. The hidden layers had ReLU activation except for the output layers of ϕ and ψ which had linear activation. The network h was a fully-connected network with three layers with the input-output dimension of 16, and the hidden-layer dimensions of 512. All hidden layers had ReLU activation, the output layer had a linear activation.

Data For **snapshots**, we generated 2048 trajectories of the system for our train dataset. Each trajectory had 20 time-steps with $dt = 0.1$. We used randomly-generated functions from Equation 3.3.5 as initial conditions for the train trajectories. For **collocations**, we used Equation (3.3.6) to generate 65536 collocation points. We generated 300 trajectories for our test datasets, 100 per kind of initial conditions from Figure (3.3.5). We used all 40 steps of the test trajectories to evaluate the performance of the models. All performance measures on Figure (A.8.6) are based on this test dataset.

Training We trained models for 500 epochs, with the learning rate of $10^{-}4$. All weights ω_i were set to 1 for a hybrid model, and ω_3, ω_4 were set to 0 for a data-driven model. Every batch contained 64 trajectories and 1280 (that is, 64*20) collocation points, with the same rationale as in Appendix A.8.1.

A.8.7 Burgers' Equation: Robustness to Noise (Figure 3.3.9)

In this section we examine robustness to noise for four models: PINODE (Data-Driven, Physics-Informed, Hybrid). We also add DMD to the comparison as a reference point.

Architecture All PINODE models share the same architecture as described in Appendix A.8.6.

Data We start with the same data as described in Appendix A.8.6. Then we apply 11 different levels of Gaussian noise with the mean 0 and the variances distributed log-uniformly between $[10^{-4}, 10^{1}]$. We only apply noise to the train snapshots; the test snapshots are noise-free.

Training We trained models for 500 epochs, with the learning rate of $10^{-}4$. All weights ω_i were set to 1 for a hybrid model, ω_1, ω_2 , and ω_3, ω_4 were set to 0 for a data-driven model. Every batch contained 64 trajectories and 1280 (that is, 64*20) collocation points, with the same rationale as in Appendix A.8.1.