

Covariate Selection Techniques for Linear Mixed-Effects Models with Applications to Population Health Problems

Aleksei Sholokhov

Committee

Aleksandr Aravkin (co-chair),
Nathan Kutz (co-chair),
James Burke,
Jonathan Wakefield (GSR)

A thesis proposal submitted in fulfillment
of the requirements for the degree of

Doctor of Philosophy

Department of Applied Mathematics
University of Washington

Last modified: Thursday 6th April, 2023

Abstract

Linear Mixed-Effects models are widely used technique for modeling clustered data, such as cohort studies, longitudinal data analysis, and meta-research. In this work we address an important practical problem of simultaneous selection of fixed and random effects in this type of models. Our approach is based on the Relax-and-Split methodology, which uses a relaxation technique together with partial minimization to efficiently solve non-convex problems. We use it to relax the constraints on the maximal number of included covariates and then apply an interior point method to obtain a solution. The performance of the resulting estimator is studied on synthetic data, as well as on two real-data applications: covariates selection for modeling the contact rate for COVID-19, and the estimation of burden of anxiety and depression disorders as results of bullying. These experiments show that the proposed method is advantageous in terms of execution time and selection quality. In addition, several theoretical questions are discussed as a part of future work that would contribute to understanding of convergence conditions of Relax-and-Split based methods in general.

Contents

1 MSR3	7
1.1 Introduction	8
1.2 Linear Mixed-Effects Models: Notation and Fundamentals	10
1.2.1 Prior Work on Feature Selection for Mixed-Effects Models	12
1.3 Algorithms for Feature Selection	14
1.3.1 Variable Selection via Proximal Gradient Descent	14
1.3.2 Variable Selection via MSR3	16
1.3.3 Relaxation and Efficient Algorithms: MSR3 and MSR3-Fast	21
1.4 Verifications	22
1.4.1 MSR3 for Covariate Selection	22
1.4.2 Experiments on Real Data	26
1.5 Theoretical Analysis	27
1.6 Software Implementation	27
1.7 Discussion	27
2 PINODE	29
2.1 Introduction	30
2.2 Method	32
2.3 Experiments	36
2.3.1 Lifted Duffing Oscillator	37
2.3.2 Burger's equation	39
2.3.3 Training in Low-Data Regime with Collocations	41

2.3.4	Robustness to Noise in Low-Data Regime	43
2.4	Discussion and Conclusions	45
2.5	Application to Compressive Sensing	46
2.5.1	Introduction	46
2.5.2	Method	46
2.5.3	Experiments	48
2.5.4	Discussion and Conclusion	48
A	Appendix	57
A.1	Acknowledgement	57
A.2	Derivatives of Marginalized Log-likelihood for Linear Mixed Models	57
A.3	Description of Datasets	59
A.3.1	GBD Bullying Data	59
A.3.2	COVID-19 Contact Rate Forecasting Data	61

Todo list

■ Edit: Add a paragraph summarizing results from the theory paper	10
■ Edit: Extend on the package summary using JOSS paper	10
■ Edit: Add more methods for feature selection Buscemi and Plaia (2019) ; Müller et al. (2013)	13
■ Edit: Add more ICs (e.g. Muller)	14
■ Edit: Make section "Evaluating proximal operators" using Appendix	15
■ Edit: Add other optimization methods for FS in LMEs (e.g. Newton)	16
■ Search: Add table of cross-references between optimization methods and regularizers	16
■ Search: Add table with rates of convergence wherever possible	16
■ Ask Jim What would be the best way to introduce theoretical results to my thesis? .	27

Chapter 1

MSR3

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

1.1 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 implemented in an open source library that makes it easy to use and to compare different methods.

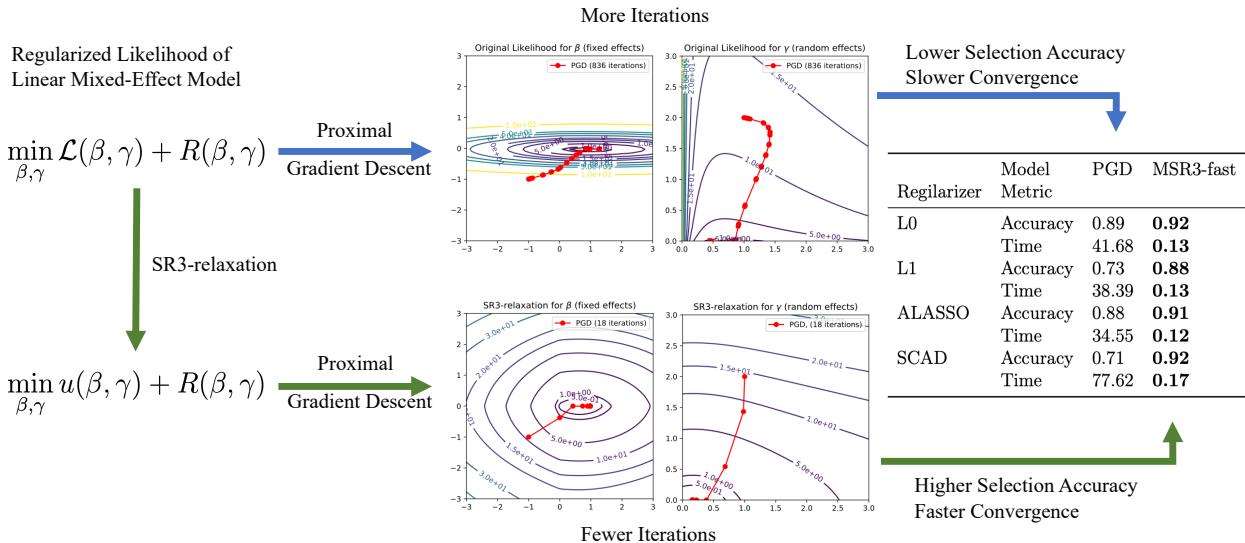


Figure 1.1.1: Selection of fixed and random effects for LME likelihoods \mathcal{L} using ‘regularization-agnostic’ framework and its SR3 extension using four regularizers. SR3 relaxation accelerates algorithmic converge (middle panel), and gives better robustness and improved performance on synthetic problems across regularizers (right panel)

In this work we develop 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. We provide proximal operators for commonly used regularizers and show how to apply the PGD method to the nonconvex LME setting. In particular we apply the PGD method to four regularizers, including the ℓ_0 regularizer, which does not admit local quadratic approximations and has not been used before for LME variable selection strategies.

We also develop a new meta-approach that can improve the performance of LME selection methods for any regularizer. Specifically, we extend sparse relaxed regularized regression (SR3) framework ([Zheng et al. \(2019\)](#)) to the LME setting. In linear regression, SR3 accelerates and improves the performance of regularization strategies by introducing auxiliary variables that decouple the accuracy and sparsity requirements for the model coefficients. We develop a conceptual and algorithmic approach necessary to extend the SR3 concept to LME. This development is necessary because the LME problem is nonlinear, nonconvex, and includes constraints on variance parameters. We show that the new approach yields superior results in

terms of specificity and sensitivity of feature selection, and is also computationally efficient.

Edit: Add a paragraph summarizing results from the theory paper

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 1.4.2. **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.

Edit: Extend on the package summary using JOSS paper

1.2 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 $n = n_1 + n_2 + \dots + n_m$. 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

$$\begin{aligned} Y_i &= X_i\beta + Z_i u_i + \varepsilon_i, \quad i = 1 \dots m \\ u_i &\sim \mathcal{N}(0, \Gamma), \quad \Gamma \in \mathbb{S}_+^q \\ \varepsilon_i &\sim \mathcal{N}(0, \Lambda_i), \quad \Lambda_i \in \mathbb{S}_{++}^{n_i} \end{aligned} \tag{1.2.1}$$

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}_+^ν and \mathbb{S}_{++}^ν 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 i th 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., $\Gamma = \text{Diag}(\gamma)$, $\gamma \in \mathbb{R}_+^s$.

Defining group-specific error terms $\omega_i = Z_i u_i + \varepsilon_i$, we get a compact formulation that re-casts (1.2.1) as a correlated noise model:

$$Y_i = X_i\beta + \omega_i, \quad \omega_i \sim \mathcal{N}(0, \Omega_i(\Gamma)), \quad \Omega_i(\Gamma) = Z_i \Gamma Z_i^T + \Lambda_i. \tag{1.2.2}$$

For brevity, we refer to $\Omega_i(\Gamma)$ as just Ω_i . The reformulation (1.2.2) yields the following marginalized negative log-likelihood function of a linear mixed-effects model ([Patterson and Thompson, 1971](#)):

$$\mathcal{L}_{ML}(\beta, \Gamma) := \sum_{i=1}^m \frac{1}{2} (y_i - X_i \beta)^T \Omega_i^{-1} (y_i - X_i \beta) + \frac{1}{2} \ln \det \Omega_i. \quad (1.2.3)$$

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

$$\min_{\beta, \Gamma} \mathcal{L}_{ML}(\beta, \Gamma) \quad \text{s.t.} \quad \Gamma \in \mathbb{S}_+^q. \quad (1.2.4)$$

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 $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ is defined as*

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

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

$$\text{lev}_\alpha f = \{x : f(x) \leq \alpha\}.$$

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

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

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

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

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

The negative log likelihood (1.2.4) 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 $\bar{\eta}$ computed in ([Aravkin et al., 2022](#), Section 5.1). The expected value of the posterior mode β given Γ has the closed form representation

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

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

$$\min_{\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}_+^q} \mathcal{L}(\beta, \gamma) := \mathcal{L}_{ML}(\beta, \text{Diag}(\gamma)) \quad (1.2.5)$$

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 (1.2.5) and, more generally, (1.2.4) follows from the work of [Zheng et al. \(2021\)](#). Standalone proofs for the existence of minimizers are developed in ([Aravkin et al., 2022](#), Theorem 1), and extended to the presence of regularizers in ([Aravkin et al., 2022](#), 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 (1.2.5)), following the meta-analysis use-case ([Zheng et al., 2021](#)) that is widely used 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 1.4.

1.2.1 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. 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:

$$\min_{\theta} f(\theta) \quad \text{s.t.} \quad \|\theta\|_0 \leq k \quad (1.2.6)$$

where $\|\theta\|_0$ denotes the number of nonzero entries in θ , see panel (c) of Figure 1.2.1.

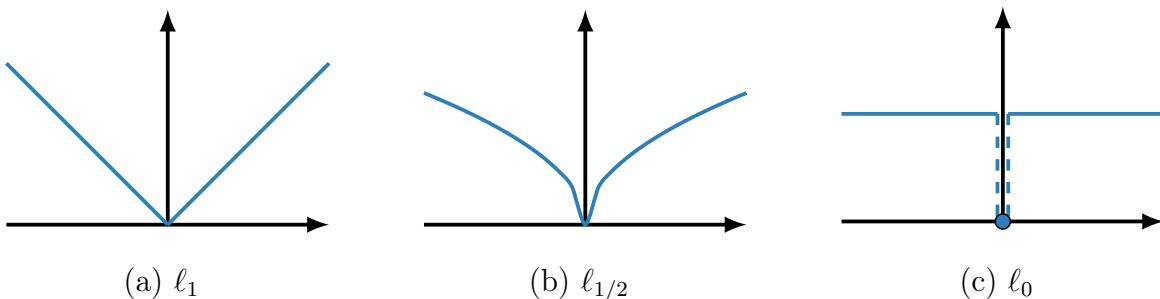


Figure 1.2.1: Common convex and non-convex regularizers used for feature selection.

The constraint in (1.2.6) 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 1.2.1.

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) reparametrized 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`¹ (Groll and Tutz (2014)) and `lmmLasso`² (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.

Edit: Add more methods for feature selection Buscemi and Plaia (2019); Müller et al. (2013)

To better compare methods, we need to consider the regularization parameter λ and how it is tuned. The output of a shrinkage model critically depends on the tuning parameter λ . The entire range of possible 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)):

$$AIC := 2 \mathcal{L}(\hat{\theta}) + 2\alpha_n(p + q) \quad (1.2.7)$$

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);

¹<https://rdrr.io/cran/glmmLasso/man/glmmLasso.html>

²<https://rdrr.io/cran/lmmlasso/>

Ibrahim et al. (2011b)) use a BIC-type information criterion:

$$BIC := 2\mathcal{L}(\hat{\theta}) + \log(n)(p+q). \quad (1.2.8)$$

BIC performs well in practice, but does not have theoretical guarantees (Schelldorfer et al. (2011)).

Edit: Add more ICs (e.g. Muller)

1.3 Algorithms for Feature Selection

We approach feature selection by adding a regularizer to model (1.2.5):

$$\min_x \mathcal{L}(x) + R(x) + \delta_{\mathcal{C}}(x), \quad (1.3.1)$$

where $x = (\beta, \gamma)$, $\mathcal{C} := \mathbb{R}^p \times \mathbb{R}_+^q$, $R : \mathbb{R}^P \times \mathbb{R}_+^q \rightarrow \overline{\mathbb{R}}_+ := \mathbb{R}_+ \cup \{+\infty\}$ is a lower semi-continuous (lsc) regularization term, and $\delta_{\mathcal{C}}$ is the convex indicator function

$$\delta_{\mathcal{C}}(x) := \begin{cases} 0, & x \in \mathcal{C} \\ +\infty, & x \notin \mathcal{C}. \end{cases}$$

By (Aravkin et al., 2022, Theorem 2), solutions to (1.3.1) always exist when R has compact lower level sets. The most common regularizers are separable taking the form

$$R(x) = \sum_{i=1}^p r_i(x_i), \quad (1.3.2)$$

with typical choices for the component functions r_i given in Table 1.3.1.

1.3.1 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 (1.3.1). The proximal operator for $\alpha R + \delta_{\mathcal{C}}$ is defined as the mapping

$$\text{prox}_{\alpha R + \delta_{\mathcal{C}}}(z) := \underset{y \in \mathcal{C}}{\operatorname{argmin}} \ R(y) + \frac{1}{2\alpha} \|y - z\|_2^2,$$

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 (1.3.2), we have

$$\text{prox}_R(z) = (\text{prox}_r(z_1), \dots, \text{prox}_r(z_q)).$$

Regularizer	$r(x)$, $x \in \mathbb{R}$	$\text{prox}_{\alpha r}(z)$
LASSO (ℓ_1) (Tibshirani (1996a))	$ x $	$\text{sign}(z)(z - \alpha)_+$
A-LASSO (Fan and Li (2001))	$\bar{w} x $, $\bar{w} \geq 0$	$\text{sign}(z)(z - \alpha\bar{w})_+$
SCAD (Fan (1997))	$\begin{cases} \sigma x , & x \leq \sigma \\ \frac{-x^2 + 2\rho\sigma x - \sigma^2}{2(\rho-1)}, & \sigma < x < \rho\sigma \\ \frac{\sigma^2(\rho+1)}{2}, & x > \rho\sigma \end{cases}$	$\begin{cases} \text{sign}(z)(z - \sigma\alpha)_+, & z \leq \sigma(1+\alpha) \\ \frac{(\rho-1)z - \text{sign}(z)\rho\sigma\alpha}{\rho-1-\alpha}, & \sigma(1+\alpha) < z \\ \leq \max(\rho, 1+\alpha)\sigma & \\ z, & z > \max(\rho, 1+\alpha)\sigma \end{cases}$
$\delta_{\ x\ _0 \leq k}$ (ℓ_0 ball)	$\begin{cases} 0, & \#\{ x_i \neq 0\} \leq k \\ \infty, & \text{otherwise} \end{cases}$	keep k largest $ x_i $, set the rest to 0

Table 1.3.1: Proximal operators for commonly used sparsity-promoting regularizers.

Edit: Make section "Evaluating proximal operators" using Appendix

Table 1.3.1 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 $\text{prox}_{\alpha R + \delta_C}(z)$.

Theorem 1 (prox for bounded γ). *We consider modified regularizers $r(\gamma)$ from the Table 1.3.1 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

$$\text{prox}_{(\alpha r + \delta_{[0, \bar{\gamma}]})}(\gamma_i) = \begin{cases} \text{prox}_{\alpha r}(\gamma_i), & 0 \leq \gamma_i < \bar{\gamma} \\ \bar{\gamma}, & \gamma_i \geq \bar{\gamma} \\ 0, & \text{otherwise} \end{cases}$$

2. For LASSO, A-LASSO we have for all i that

$$\text{prox}_{(\alpha r + \delta_{[0, \bar{\gamma}]})}(\gamma_i) = \begin{cases} \text{prox}_{\alpha r}(\gamma_i), & 0 \leq \gamma_i < \bar{\gamma} + \alpha \\ \bar{\gamma}, & \gamma_i \geq \bar{\gamma} + \alpha \\ 0, & \text{otherwise} \end{cases}$$

3. For $R(\cdot) = \delta_{\text{lev}_{\|\cdot\|_0}(k)}$ the prox $\text{prox}_{\alpha R + \delta_C}(\gamma)$ 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 1 is provided in Appendix ???. 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 ???. In practice, α is computed using a line-search, since the available estimate for L is very conservative.

```

1  $x = x_0$ ,  $\alpha < \frac{1}{L}$ , where  $\mathcal{L}$  is  $L$ -Lipschitz
2 while not converged do
3   |  $x^+ = \text{prox}_{\alpha R + \delta_C}(x - \alpha \nabla \mathcal{L}(x))$ ;
4 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 regularizer R . Algorithm 1 locates first-order stationary points under weak assumptions, in particular neither the objective nor the regularizer need to be convex (Beck, 2017; Attouch et al., 2013).

Edit: Add other optimization methods for FS in LMEs (e.g. Newton)

Search: Add table of cross-references between optimization methods and regularizers

Search: Add table with rates of convergence wherever possible

1.3.2 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 (1.3.1) we introduce auxiliary parameters designed to discover the fixed and random features:

$$\min_{x,w} \mathcal{L}(x) + R(w) + \delta_C(x) + \kappa_\eta(x - w), \quad (1.3.3)$$

where κ_η penalizes deviations between x and w , and also guarantees that the objective is convex with respect to the γ components of x for sufficiently large η :

$$\kappa_\eta(\beta, \gamma) = \frac{\eta}{2} \|(\beta, \gamma)\|^2, \quad (1.3.4)$$

with $\eta \geq \bar{\eta}$ where $\bar{\eta}$ is the weak convexity constant computed in (Aravkin et al., 2022, Section 5.1). As $\eta \uparrow \infty$, the extended objective (1.3.3) converges in an epigraphical sense to the original

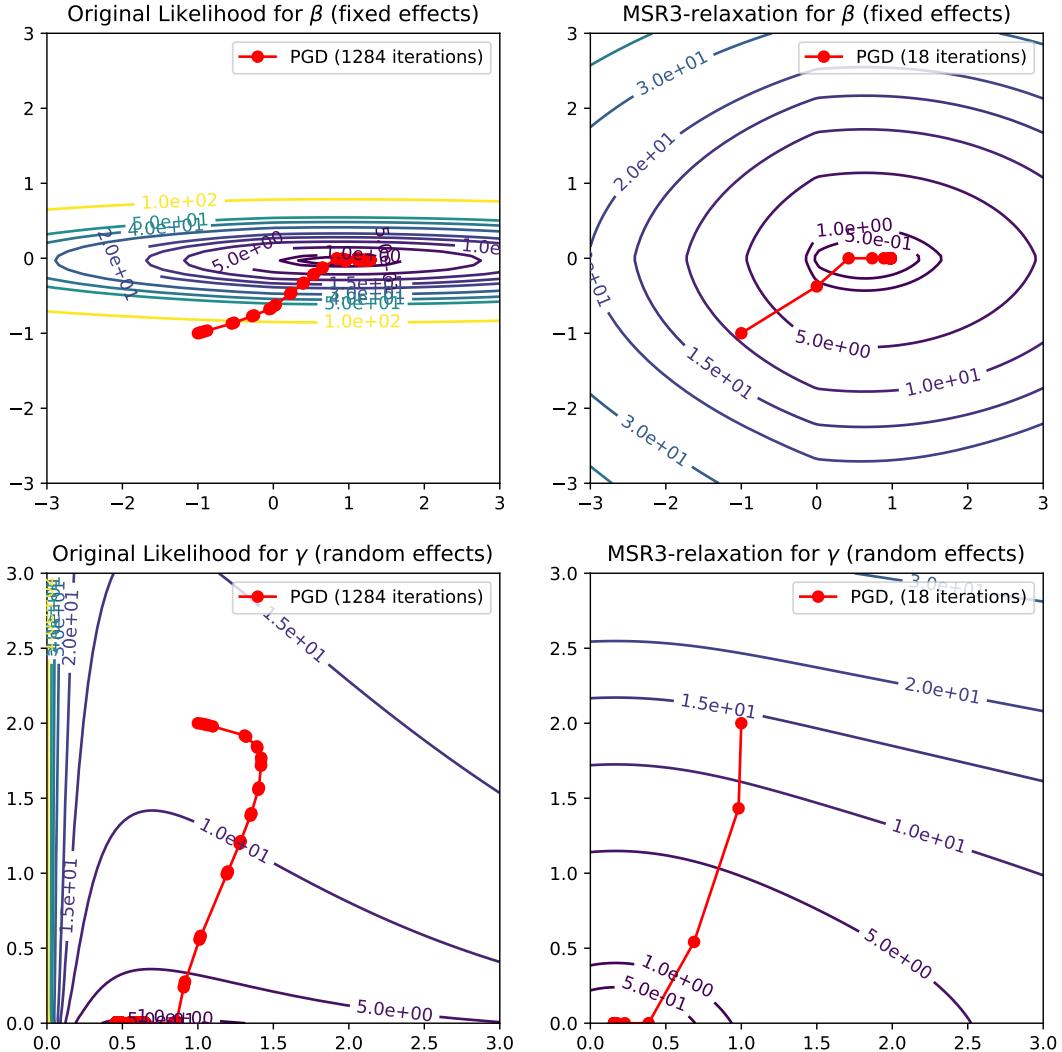


Figure 1.3.1: Proximal Gradient Descent (PGD) for (1.3.6) converges far faster than for (1.3.1), because the SR3-value function u_η yields more spherical level-sets than original likelihood for both convex components β (first row) and non-convex components γ (second row).

objective (1.3.1). However, feature selection accuracy does not require this continuation, and a fixed modest value can be used (Zheng et al., 2019), e.g. $\eta = 1$.

To understand the algorithm and logic behind the objective (1.3.3), we define a value function $u_\eta(w)$ and the solution set $S_\eta(w)$:

$$\begin{aligned} u_\eta(w) &= \min_x \mathcal{L}(x) + \delta_{\mathcal{C}}(x) + \kappa_\eta(x - w) \\ S_\eta(w) &= \operatorname{argmin}_x \mathcal{L}(x) + \delta_{\mathcal{C}}(x) + \kappa_\eta(x - w). \end{aligned} \tag{1.3.5}$$

Substituting (1.3.5) into (1.3.3) the problem transforms into the problem of optimizing a regularized value function:

$$\min_w u_\eta(w) + R(w) \quad (1.3.6)$$

Here we have transformed the original regularized likelihood (1.3.1) through relaxation and partial minimization to obtain an equivalent problem (1.3.6) for w with the same regularizer. The value function u_η encapsulates global variational information on the function $\mathcal{L}(x) + \delta_C(x)$ 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 (1.3.6). Indeed, in ([Aravkin et al., 2022](#), Section 5) it is shown that u_γ is well-defined, differentiable, and Lipschitz continuous, with

$$\nabla u_\eta(w) = \nabla_w k_\eta(x - w)|_{x=S_\eta(w)} = \eta(w - S_\eta(w)). \quad (1.3.7)$$

Our empirical studies indicate that (1.3.6) has advantages over (1.3.1) from an optimization perspective since $u_\eta(w)$ 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 Figure 1.3.1, we plot the level-sets of $\mathcal{L}(x) + \|x\|_1$ (left column) and $u(w) + \|w\|_1$ (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 value function $u_\eta(w)$ which yields the iteration of the form

$$w^+ = \text{prox}_{\alpha^{-1}R}(w - \alpha \nabla u_\eta(w)) \quad (1.3.8)$$

Because of the results stated above, all components of the iteration (1.3.8) are well-defined. The equivalence of Algorithm 2 and (1.3.8) is established in the following lemma, which extends the relationship studied by [Zheng et al. \(2019\)](#) to the case of $x = (\beta, \gamma)$.

Lemma 2 (Equivalence of Algorithms). *Algorithm 2 is equivalent to (1.3.8).*

Proof. Substituting (1.3.7) into (1.3.8), we see that the iteration (1.3.8) is equivalent to the alternating minimization scheme outlined in the Algorithm 2. \square

```

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

```

Algorithm 2: Proximal Gradient Descent for Value Function

While in linear regression setting of of [Zheng et al. \(2019\)](#), Algorithm 2 can be implemented exactly, in the nonlinear case evaluating x^+ requires an iterative algorithm. In particular, we use an interior point method to account for non-negativity of γ . This means that the value function u_η is approximately evaluated, using a Newton iteration with a barrier term. 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

$$\min_{\beta, \gamma} \mathcal{L}(\beta, \gamma) + \kappa_\eta(\beta - \hat{\beta}, \gamma - \hat{\gamma}) \quad \text{s.t.} \quad 0 \leq \gamma. \quad (1.3.9)$$

This problem is well suited for an interior point approach ([Kojima et al., 1991](#); [Nesterov and Nemirovskii, 1994](#); [Wright, 1997](#)). First, the inequality constraint $0 \leq \gamma$ is relaxed using a log-barrier penalty, obtaining a minimization problem for a relaxed objective $\mathcal{L}_{\mu, \eta}$:

$$\min_{\beta, \gamma} \left\{ \mathcal{L}_{\mu, \eta}(\beta, \gamma) := \mathcal{L}(\beta, \gamma) + \kappa_\eta(\beta - \hat{\beta}, \gamma - \hat{\gamma}) - \mu \sum_{i=1}^q \ln(\gamma_i) \right\}. \quad (1.3.10)$$

Here 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. The existence of solutions for the problem (1.3.10) for any positive μ is shown in ([Aravkin et al., 2022](#), Theorem 5), and the convergence of solutions to the MSR3 solution as $\mu \downarrow 0$ is shown in ([Aravkin et al., 2022](#), Theorem 7). Finally, ([Aravkin et al., 2022](#), Theorem 6) shows that the MSR3 relaxation is consistent with respect to the barrier, so that as the MSR3 parameter $\eta \uparrow \infty$, limit points of global solutions to the former are global solutions to the latter. However, in the applications considered here, the empirical studies in Sections 1.3.3 and 1.4.2 indicate that one does not need to make η particularly large in order to accurately identify the correct sparsity pattern.

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

$$\nabla_\gamma \mathcal{L}_{\mu, \eta}(\beta, \gamma) = \nabla_\gamma \mathcal{L}(\beta, \gamma) + \eta(\gamma - \hat{\gamma}) - \mu \text{Diag}(\gamma)^{-1} \mathbf{1} = 0, \quad (1.3.11)$$

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

$$v \odot \gamma - \mu \mathbf{1} = 0, \quad (1.3.12)$$

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 (1.3.10) can

now be written as

$$G_{\mu,\eta}(v, \beta, \gamma) := \begin{bmatrix} v \odot \gamma - \mu \mathbf{1} \\ \nabla_\beta \mathcal{L}(\beta, \gamma) + \eta(\beta - \hat{\beta}) \\ \nabla_\gamma \mathcal{L}(\beta, \gamma) + \eta(\gamma - \hat{\gamma}) - v \end{bmatrix} = 0. \quad (1.3.13)$$

We then apply Newton's method to (1.3.13), that is, in each iteration the search direction $[\Delta v, \Delta \beta, \Delta \gamma]$ solves the linear system

$$\nabla G_{\mu,\eta}(v, \beta, \gamma) \begin{bmatrix} \Delta v \\ \Delta \beta \\ \Delta \gamma \end{bmatrix} = -G_{\mu,\eta}(v, \beta, \gamma). \quad (1.3.14)$$

where

$$\nabla G_{\mu,\eta}(v, \beta, \gamma) = \begin{bmatrix} \text{Diag}(\gamma) & 0 & \text{Diag}(v) \\ 0 & \nabla_{\beta\beta}^2 \mathcal{L} + \eta I & \nabla_{\beta\gamma}^2 \mathcal{L} \\ -I & \nabla_{\gamma\beta}^2 \mathcal{L} & \nabla_{\gamma\gamma}^2 \mathcal{L} + (\eta + \bar{\lambda}) I \end{bmatrix} \quad (1.3.15)$$

and we have used the fact that $v \odot \gamma = \text{Diag}(v)\gamma = \text{Diag}(\gamma)v$. 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

$$\begin{aligned} \text{Positivity: } & \gamma^{(k+1)} > 0, \quad v^{(k+1)} > 0 \\ \text{Sufficient Descent: } & \|G_\mu(v^{(k+1)}, \beta^{(k+1)}, \gamma^{(k+1)})\| \leq 0.99 \|G_\mu(v^{(k)}, \beta^{(k)}, \gamma^{(k)})\|. \end{aligned} \quad (1.3.16)$$

At each iteration the relaxation parameter μ is updated by the formula

$$\mu^{(k+1)} = v^{(k)}{}^T \gamma^{(k)} / q, \quad (1.3.17)$$

where $v^{(k)}{}^T \gamma^{(k)}$ is the duality gap at iteration k . The algorithm terminates when the criteria

$$\begin{aligned} \|G_{\mu,\eta}(v^{(k+1)}, \beta^{(k+1)}, \gamma^{(k+1)})\| & \leq \text{tol} \\ \mu & \leq \text{tol} \end{aligned} \quad (1.3.18)$$

are both satisfied, so the interior point problem is nearly stationary, and closely approximates the original problem (1.3.9). MSR3 is summarized in Algorithm 3, which approximates Algorithm 2

as the tolerance goes to 0. In the numerical experiments, we use $\text{tol} = 10^{-5}$, and accuracy does not change as the tolerance parameter decreases.

```

1  $w = w_0$ 
2 while not converged do
3    $x^+$  satisfies  $\|G_{\mu,\eta}(v^+, x^+)\| \leq \text{tol}$ ,  $\mu \leq \text{tol}$ 
4    $w^+ = \text{prox}_{\alpha^{-1}R}(x^+)$ 
5 end
```

Algorithm 3: MSR3

Positive Approximation of the Hessian For many datasets the weak convexity constant $\bar{\eta}$ can be extremely large and difficult to compute. However, if $\bar{\eta}$ is too small $\nabla_{\gamma\gamma}^2 \mathcal{L}(\beta, \gamma)$ 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., 2022](#), Lemma 3) that

$$\nabla^2 \mathcal{L}(\beta, \gamma) = \sum_{i=1}^m S_i^T \begin{bmatrix} X_i^T \\ -Z_i^T \end{bmatrix} \Omega_i(\gamma)^{-1} \begin{bmatrix} X_i & -Z_i \end{bmatrix} S_i - \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{2}(Z_i^T \Omega_i(\gamma)^{-1} Z_i)^{\circ 2} \end{bmatrix}. \quad (1.3.19)$$

This implies that negative eigenvalues for the Hessian must arise from the Hessian with respect to γ , $\nabla_{\gamma\gamma}^2 \mathcal{L}(\beta, \gamma)$, 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.

1.3.3 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.

```

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

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

1.4 Verifications

1.4.1 MSR3 for Covariate Selection

In this section we compare the numerical performance the feature selection accuracy and 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 1.4.2.

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

Regularizer	Metric	Model	PGD	MSR3	MSR3-fast
L0	Accuracy	0.89	0.92	0.92	
	Time	41.68	88.54	0.13	
L1	Accuracy	0.73	0.88	0.88	
	Time	38.39	9.13	0.13	
ALASSO	Accuracy	0.88	0.92	0.91	
	Time	34.55	65.19	0.12	
SCAD	Accuracy	0.71	0.93	0.92	
	Time	77.62	84.67	0.17	

Table 1.4.1: 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 ?? of Appendix ??.

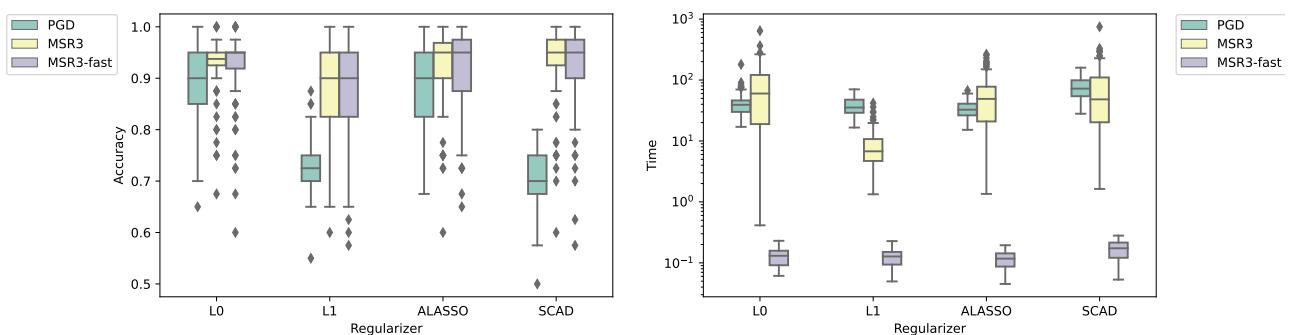


Figure 1.4.1: Feature selection accuracy and execution time in seconds for PGD (Algorithm 1), MSR3 (Algorithm 2), and MSR3-fast (Algorithm 4) with various regularizers. MSR3-Fast has the same accuracy as MSR3 and significantly decreases computation time.

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 begin by setting a log-uniform grid of 20 candidate values for the parameter $\eta \in [10^{-4}, 10^2]$, and then for each value of η on this grid the BIC is optimized using a golden search in $\lambda \in [0, 10^5]$. The final values of η and λ are then chosen to be those that maximize the BIC criterion.

Figure 1.4.2 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 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 of the problem converge to those of the original problem, and thus the minimizer. 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. \(2019\)](#).

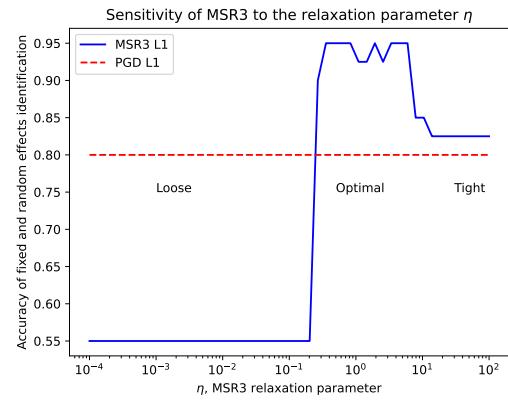


Figure 1.4.2: Dependence of model performance on the relaxation η for a sample problem.

Results. The experimental results are presented in the Table 1.4.1 and Figure 1.4.1. MSR3 improves the selection accuracy of most regularization techniques described in Table 1.3.1, showing a near-perfect performance, while converging two orders of magnitude faster in wall-clock time.

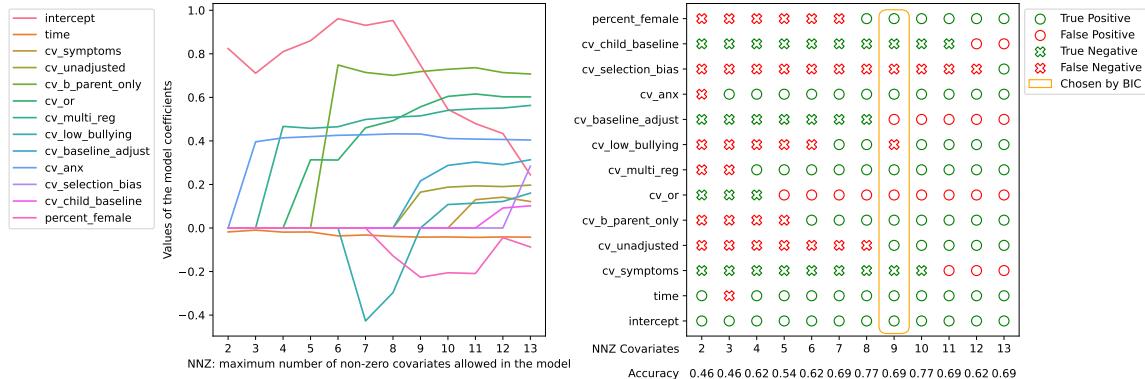
Comparison to `glmmLasso` and `lmmLasso`. We used ([Buscemi and Plaia, 2019](#), Table 3) as a reference for feature selection libraries. Of the 17 entries mentioned, the four libraries

that successfully ran on our synthetic data described above were packages **glmmLasso**³ ([Groll and Tutz \(2014\)](#)), **lmmLasso**⁴ ([Schelldorfer et al. \(2011\)](#)), **fence**⁵ ([Jiang et al. \(2008\)](#)) and **PC0** ([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 **PC0** 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**⁶ ([Groll and Tutz \(2014\)](#)) and **lmmLasso**⁷ ([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 1.4.2. 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 of 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 1.4.2: 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.

Figure 1.4.3: 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.3.1 for covariates description and assessment of significance.



1.4.2 Experiments on Real Data

In this section we validate the MSR3-empowered ℓ_0 -regularized mixed-effect model ($R(x) = \delta_{\|x\|_0 \leq k}$ from Table 1.3.1) 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 age⁸. 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 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.3.1.

³<https://rdrr.io/cran/glmmLasso/man/glmmLasso.html>

⁴<https://rdrr.io/cran/lmlasso/>

⁵<https://rdrr.io/cran/fence/>

⁶<https://rdrr.io/cran/glmmLasso/man/glmmLasso.html>

⁷<https://rdrr.io/cran/lmlasso/>

⁸Institute for Health Metrics and Evaluation (IHME). Bullying Victimization Relative Risk Bundle GBD 2020. Seattle, United States of America (USA), 2021.

The feature selection process is illustrated on Figure 1.4.3. 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 \(2013\)](#).

1.5 Theoretical Analysis

Ask Jim What would be the best way to introduce theoretical results to my thesis?

1.6 Software Implementation

To ensure reproducibility of this research, all new algorithms have been implemented as a part of the `pysr3`⁹ library. This library implements functionality for fitting linear mixed models and selecting covariates. The user interface was designed to be fully compliant with the standards¹⁰ of `sklearn` library to minimize learning time.

1.7 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 (1.3.6) 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 numerical 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

⁹Available at <https://github.com/aksholokhov/pysr3>

¹⁰<https://scikit-learn.org/stable/developers/develop.html>

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 (1.3.6). 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 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 2

PINODE

2.1 Introduction

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 moment of writing this paper such systems are incapable of real-time complex simulations, but they already can run low-dimensional pre-trained models, thus inviting the development of high-quality reduced order models (ROMs) Otterness et al. (2017) . ROMs, therefore, are essential for enabling design optimization, uncertainty propagation, predictive models, and control for such dynamical systems Brunton and Kutz (2022); Kutz et al. (2016); Rowley and Dawson (2017); Jones et al. (2020)

To be suitable for control, a ROM training method needs to find a low-dimensional manifold and a dynamics on it that together yield both 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 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 then 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, they require datasets that densely cover a hypothetical infinite dimensional phase portrait of the dynamical system. Large demand for 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 one 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 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 data, especially when evaluating time-derivatives with finite differences is required Delahunt and Kutz (2022). Collocation-based enforcement of the physics, i.e. projection of the candid 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 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 at the latent space to model complex and nonlinear dynamics. Our goal is to reduce their demand for data and improve overall forecasting stability under challenging training conditions. To achieve that 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. Then, in Section 3, we show that 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

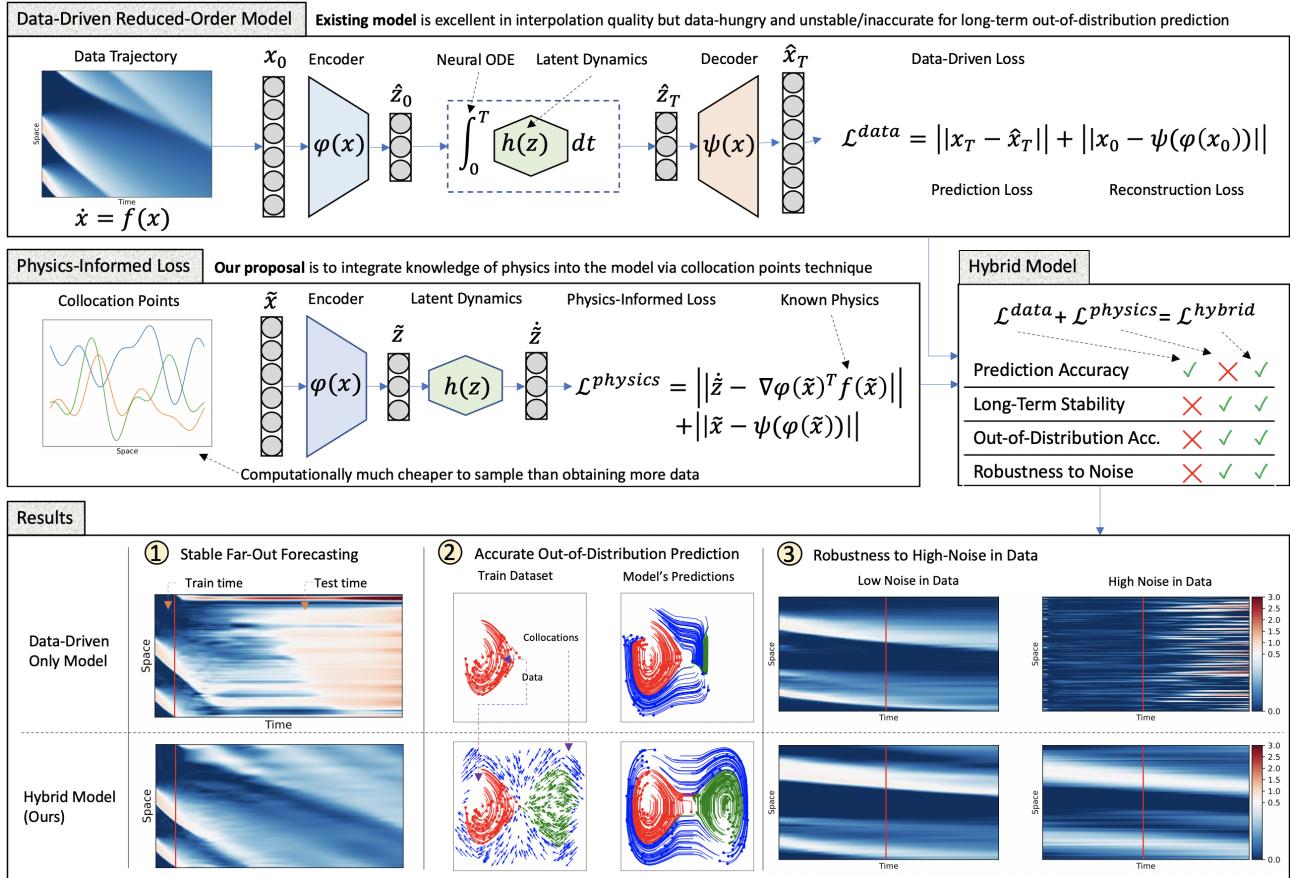


Figure 2.1.1: We propose utilizing a collocation points technique from numerical analysis to transfer knowledge of physics into continuous-time reduced-order models (ROMs). Such physics-informed models yield orders of magnitude more accurate predictions in tasks of far-out forecasting, predictions for out-of-distribution initial conditions, and learning from high-noise data. Good performance on those tasks is crucial for using ROMs in problems of compressive sensing and control.

latent spaces, which is especially important for applications in compressed sensing and control.

2.2 Method

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

$$\frac{d}{dt} \mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)) \quad (2.2.1)$$

In real-world applications it is often expensive to solve the relationship (2.2.1) directly because $\mathbf{x}(t)$ can be very high-dimensional. However, a variety of works provided both theoretical Holmes et al. (2012) and practical 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 $m \ll n$. In that space, the dynamics evolve according to a (generally unknown) function $\mathbf{h}(\mathbf{z})$:

$$\frac{d}{dt}\mathbf{z}(t) = \mathbf{h}(\mathbf{z}(t)) \quad (2.2.2)$$

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

$$\begin{aligned} \mathbf{z}(0) &= \psi^{-1}(\mathbf{x}(0)) \\ \mathbf{z}(T) &= \mathbf{z}(0) + \int_0^T \mathbf{h}(\mathbf{z}(t)) dt \\ \mathbf{x}(T) &= \psi(\mathbf{z}(T)) \end{aligned} \quad (2.2.3)$$

When $m \ll n$ we refer to the triplet $(\psi, \psi^{-1}, \mathbf{h})$ as a Reduced-Order Model (ROM) of \mathbf{f} . It is often the case that for a given system \mathbf{f} there exists no ROM $(\psi, \psi^{-1}, \mathbf{h})$ such that the relation (2.2.3) 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 (2.2.1) directly. For example, integrating (2.2.1) may be computationally intractable especially on platforms with limited computing capability such as embedded and autonomous devices. For instance, in an HVAC system, solving (2.2.1) 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 (2.2.3) may be cheap when $m \ll n$. Finally, even when solving (2.2.1) 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* iterative evaluations of (2.2.1) 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 \mathbf{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 \mathbf{h} is modelled with a fully-connected network h_{θ} . Figure 2.1.1 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* $\mathcal{L}_{\text{data}}$ 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 \mathbf{x}_i , $i \in [1 \dots k]$, where each trajectory $\mathbf{x}_i \in \mathbb{R}^{n \times p}$ 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:

$$\mathcal{L}_\theta^{data} = \frac{1}{2\sigma^2} \sum_{i=1}^k \left[\frac{\omega_1}{p} \sum_{j=1}^p \|\mathbf{x}_i(t_j) - \psi_\theta(\phi_\theta(\mathbf{x}_i(t_j)))\|^2 + \right. \quad (2.2.4)$$

$$\left. + \frac{\omega_2}{p} \sum_{j=1}^p \left\| \psi_\theta \left(\phi_\theta(\mathbf{x}_i(t_1)) + \int_{t_1}^{t_j} h(z(t)) dt \right) - \mathbf{x}_i(t_j) \right\|^2 \right] \quad (2.2.5)$$

where σ is the standard deviation of the observation noise. We note that each trajectory \mathbf{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 accordingly¹. 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.

Physics-Informed Loss In their recent work, [Liu et al. \(2022\)](#) proposed a method for utilizing knowledge of the governing equations $d\mathbf{x}/dt = \mathbf{f}(\mathbf{x})$ 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:

$$\frac{d\mathbf{z}(\mathbf{x}(t))}{dt} = \frac{d\mathbf{z}}{d\mathbf{x}} \frac{d\mathbf{x}}{dt} = \nabla \phi(\mathbf{x}(t))^T \mathbf{f}(\mathbf{x}(t)) \quad (2.2.6)$$

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

$$\frac{d\mathbf{z}(\mathbf{x}(t))}{dt} = \mathbf{h}(\phi(\mathbf{x}(t))) \quad (2.2.7)$$

Combining Equations (2.2.6) and (2.2.7) we get that

$$\mathbf{h}(\phi(\mathbf{x}(t))) = \nabla \phi(\mathbf{x})^T \mathbf{f}(\mathbf{x}) \quad (2.2.8)$$

Equation (2.2.8) links the dynamics $\mathbf{h}(\mathbf{z})$ and the encoder $\phi(\mathbf{x})$ with the known equation $\mathbf{f}(\mathbf{x})$ and is true for all $z \in \mathcal{Z}$ and $x \in \mathcal{X}$. Hence, knowledge of \mathbf{f} can be assimilated into the model by evaluating Equation (2.2.8) on a set of N carefully sampled points $\bar{\mathbf{x}}_i \in \mathcal{X}$, $i \in [1, \dots, N]$:

$$\mathcal{L}_\theta^{physics} = \sum_{i=1}^N \left[\frac{\omega_3}{N} \|h_\theta(\phi_\theta(\bar{\mathbf{x}}_i)) - \nabla \phi_\theta(\bar{\mathbf{x}}_i) \mathbf{f}(\bar{\mathbf{x}}_i)\|^2 + \frac{\omega_4}{N} \|\bar{\mathbf{x}}_i - \psi_\theta(\phi_\theta(\bar{\mathbf{x}}_i))\| \right] \quad (2.2.9)$$

We refer to the points $\bar{\mathbf{x}}_i$ as *collocations*.

¹The implementation is affected only in evaluating the integral in (2.2.4). This part is handled by `torchdiffeq` [Chen et al. \(2018a\)](#) library, which supports non-uniform time-frames within a batch

Collocations We define a collocation as pair $(\bar{\mathbf{x}}, \mathbf{f}(\bar{\mathbf{x}}))$. Collocations are samples from the space $\mathcal{X} \times \text{Im}_f(\mathcal{X})$, and they should satisfy three conditions, ordered by importance:

1. **Simplicity:** $\mathbf{f}(\bar{\mathbf{x}}_j)$ should be computationally cheap to evaluate. It is especially important for PDE systems, where \mathbf{f} may involve high-order derivatives.
2. **Representativeness:** $\bar{\mathbf{x}}_j$ 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:** $\bar{\mathbf{x}}_j \in \mathcal{X}$. 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} where a true mapping ψ operates on.

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

Our definition of collocation points follows recent works [Liu et al. \(2022\)](#), which is a further development of the original definition [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 (2.2.9) and the data-driven loss (2.2.4):

$$\min_{\theta} \left[\mathcal{L}_{\theta}^{physics} + \mathcal{L}_{\theta}^{data} \right] \quad (2.2.10)$$

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

We use a [pytorch](#) [Paszke et al. \(2019\)](#) implementation of 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 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 (2.2.9). Thus, we call our framework *Physics-Informed Neural ODE*, or PINODE.

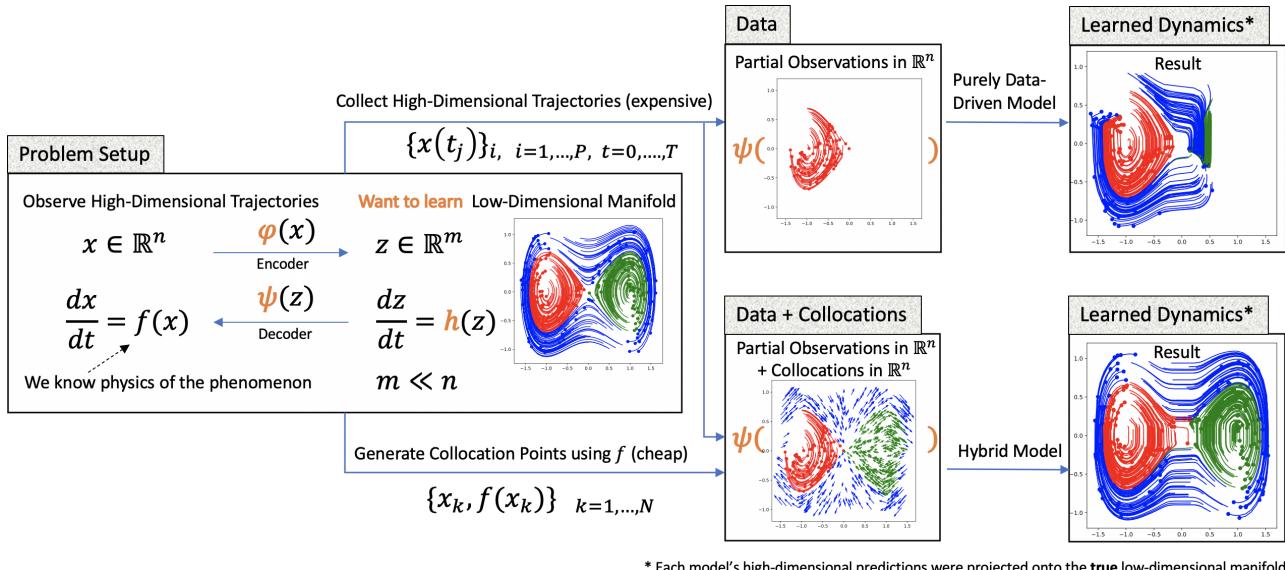


Figure 2.3.1: We use a toy example – a Lifted Duffing Oscillator – to show that it is possible to “fill the gaps” in data with collocations. Namely, Hybrid model from Figure 2.1.1 is able to learn the dynamics of two additional basins of attraction that were not represented in the dataset. As shown in the top-rightmost frame, without the collocations the model does not infer the dynamics in the unseen regions correctly.

2.3 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 $\mathbf{h}(z)$ overcomes the limitations of DMD and Koopman 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 short-term and highly unstable 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 yields more compact latent spaces than its linear counterpart for the same accuracy; (ii) these compact latent spaces allow for more stable long-term predictions; (iii) in the presence of significant noise in data, the use of collocations improves stability by providing an extra source of information that is noise-free, and (iv) in certain scenarios, training *only* on collocations yields *better* models than training on data, even when a vast amount of data is available. The latter shows that the contribution of physics-informed loss (2.2.9) may surpass the one of the data-based loss (2.2.4), especially when the data is severely limited or noisy.

2.3.1 Lifted Duffing Oscillator

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

$$\begin{aligned}\frac{dz_1}{dt} &= z_2 \\ \frac{dz_2}{dt} &= z_1 - z_1^3\end{aligned}\tag{2.3.1}$$

A phase portrait for 300 randomly sampled trajectories from this system is visualized on Figure 2.3.1, 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 $\mathcal{A}(\mathbf{z})$:

$$\mathbf{x} := \mathcal{A}(\mathbf{z}) = A\mathbf{z}^3, \quad A \in \mathbb{R}^{128 \times 2}, \quad A_{ij} \sim_{i.i.d.} \mathcal{N}(0, 1)\tag{2.3.2}$$

Hence, for this system $\mathbf{z} \in \mathcal{Z} = \mathbb{R}^2$ and $\mathbf{x} \in \mathcal{X} = \text{span}\{A_{:,1}, A_{:,2}\} \subseteq \mathbb{R}^{128}$. We treat \mathcal{X} as an observable space, in which the dynamical system (2.3.1) obeys the following:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}) = \nabla((A^T A)^{-1} A^T \mathbf{x}^{1/3})^T \mathbf{h}((A^T A)^{-1} A^T \mathbf{x}^{1/3})\tag{2.3.3}$$

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

For the experiment, we generate 6144 trajectories \mathbf{x}_i , $t = [0, 1]$, $\Delta t = 0.1$, all taken from the left lobe region (in red). We also sample 50000 collocations $\bar{\mathbf{x}}_j$ from the right (green) and the outer (blue) regions each by sampling $\bar{\mathbf{z}}_j \in U([-3/2, 3/2] \times [-1, 1])$ and then applying the transformation (2.3.2). For this example the conditions for collocations discussed in Section 2.2 are trivially satisfied.

We train two PINODE models: Data-Driven model that only uses the trajectories and Hybrid model that uses both trajectories and collocations. The models share the same architecture (Figure 2.1.1) and training parameters that are detailed in Appendix ???. After training, we invert the mapping (2.3.2) to project the models' high-dimensional predictions for unseen initial conditions onto the true low-dimensional manifold; those are visualized in Figure 2.3.1.

We make two observations from the results displayed in Figure 2.3.1. First, a purely data-driven model is unable to extrapolate outside its

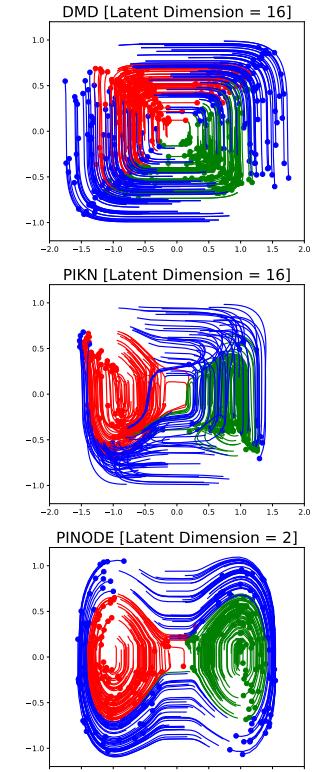


Figure 2.3.2: Non-linearity in latent dynamics and autoencoder is important for long-term extrapolation. PIKN Hybrid model was unable to extrapolate outside its basin of attraction, while PINODE model did it smoothly.

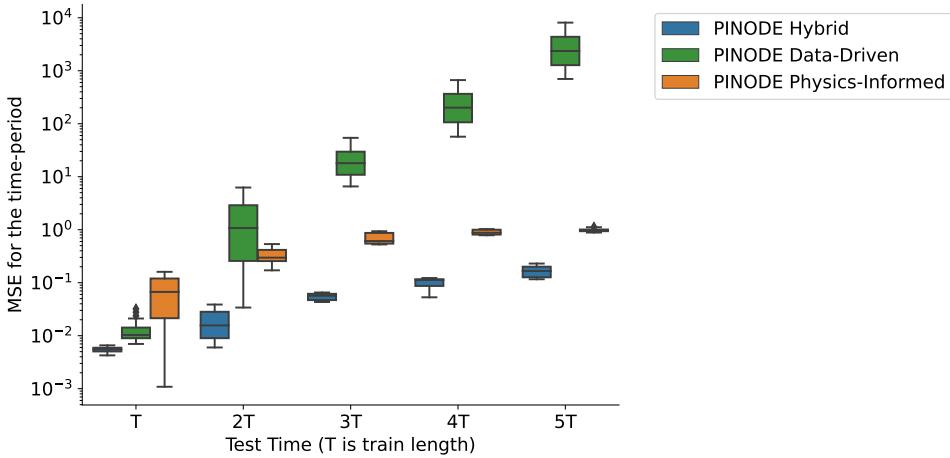


Figure 2.3.3: The time is measured in numbers of training time periods, i.e. $x = 3T$ refers the time-range between two and three training time-periods away. The errors of a purely data-driven model grow exponentially when forecasting multiple times farther in time than they were trained to do. In contrast, purely Physics-Informed model starts with worst interpolation accuracy but maintains nearly the same error when forecasting far out. Hybrid model combines the short-term accuracy of Data-Driven model and long-term stability of Physics-Informed model.

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 collocations 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 collocations to “cover the gaps” in data and improve the extrapolation accuracy of the model.

The non-linearity of the Neural ODE plays a crucial role in modeling the latent dynamics. To illustrate, we train a Hybrid PIKN model [Liu et al. \(2022\)](#) using the same dataset. PIKN differs from PINODE in using a linear latent dynamics $\frac{dz}{dt} = Lz$, where L is a finite-dimensional approximation of Koopman operator, instead of a general non-linear dynamics $\frac{dz}{dt} = h_\theta(z)$. For PIKN we set $z \in \mathbb{R}^{16}$, an 8 times expansion of the dimension of the true manifold. In Figure (2.3.2) we see that PIKN is unable to extrapolate the dynamics to unseen areas correctly using collocations: eventually, all trajectories "collapse" onto the same attractor. DMD shows even worse performance which could be attributed to the linear model reduction.

In the next experiment we show that collocations stabilize long-term predictions of the model even when data from all parts of space is available. To illustrate it we generate a dataset of 6144 trajectories (2048 per red, green, and blue area each) and 50000 collocations uniformly distributed among all three lobes. Next, we train three models: Data-Driven, Physics-Informed, and Hybrid versions of PINODE. Finally, we evaluate their relative performance for ever-

increasing forecast time. In Figure (2.3.3), the x-axis represents test time-period. For example, $x = 2T$ represents prediction errors on a time-period $[2T, 3T]$ where the model was trained on trajectories of length T , and on y-axis we plot the distribution of prediction errors for 300 unseen trajectories within that period $[2T, 3T]$.

The results on Figure (2.3.3) show that the performance of the Data-Driven model degrades quickly when the forecasting time-period increases despite an excellent performance when forecasting within its train time-period. Physics-Informed model starts with modest performance but this performance does not degrade when forecasting far ahead. The Hybrid model, in its turn, combines both near-term accuracy with long-term stability, yielding best results over each time period.

2.3.2 Burger's equation

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

$$\begin{aligned} u_t + uu_x &= \nu u_{xx} \\ u(-\pi, t) &= u(\pi, t), \quad \forall t \in [0, T] \end{aligned} \tag{2.3.4}$$

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 to 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 Burger for which we set $\nu = 0.01$.

To generate trajectories, we discretize the spacial domain $[-\pi, \pi]$ with 128 grid-points, and we solve Equation 2.3.4 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:

$$u(x, 0) = \frac{1}{10} \sum_{k=1}^{10} a_k \cos(kx) + b_k \sin((k+1)x), \quad a_k, b_k \sim \mathcal{N}(0, 1) \tag{2.3.5}$$

To generate collocations we use the same family of functions as we used for the initial conditions in Equation (2.3.5), and we additionally randomize the presence of individual frequencies in the sum:

$$\bar{u}(x) = \frac{1}{10} \sum_{k=1}^{10} p_k a_k \cos(kx) + q_k b_k \sin((k+1)x), \quad a_k, b_k \sim \mathcal{N}(0, 1), \quad p_k, q_k \sim Be(1/2). \tag{2.3.6}$$

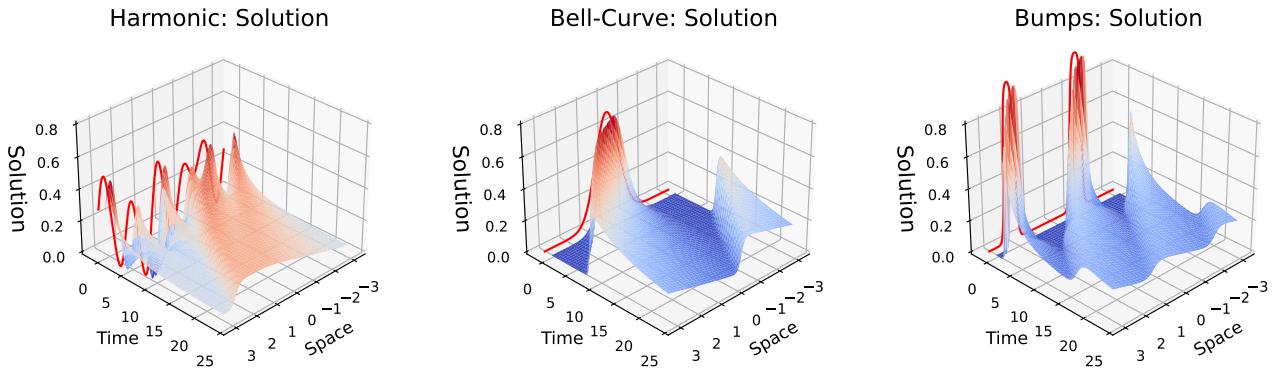


Figure 2.3.4: Examples of "harmonic", "bell-curve", and "bump" initial conditions, as well as the resulting solutions, in columns 1, 2, and 3 respectively.

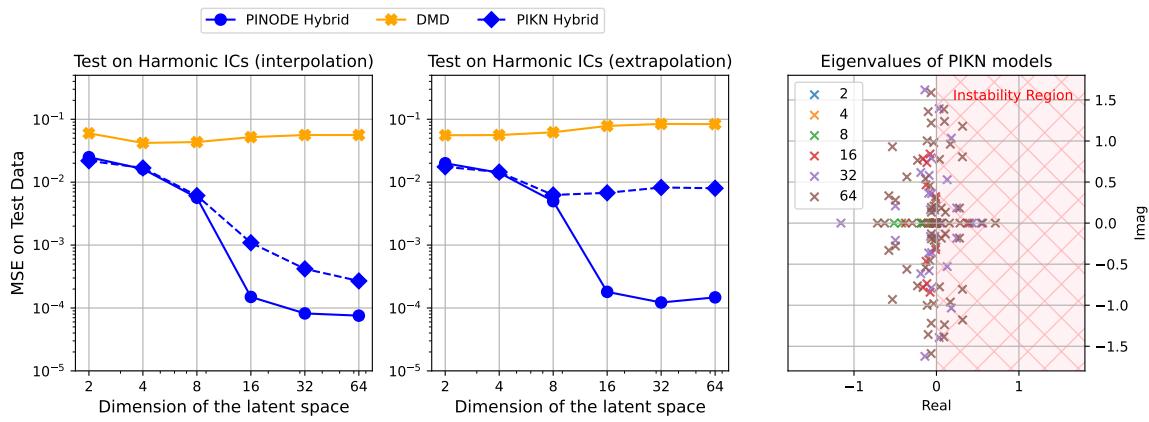


Figure 2.3.5: PINODE Hybrid model utilized the latent space dimension 5 times more efficiently by MSE than PIKN Hybrid model when modelling low-viscosity (highly-nonlinear) Burger's equation (left frame). The difference in performance grows to x100, when forecasting two times farther than the train period (central frame). The reason behind PIKN's long-term instability is the presence of eigenvalues with positive real part in the latent dynamics matrix (right frame).

We choose this family of collocations to meet the conditions (2.2). 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, (2.3.6) 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 amounts of collocations.

Compressibility of the Latent Space

In Section 2.3.1 we showed that a non-linear finite-dimensional latent dynamics model can be a necessity for building a compact ROM for a high-dimensional system. It is *not* the case for Burgers' equations since there is Cole-Hopf transformation that linearize the dynamics. However, a latent-space non-linearity can, in principle, be utilized for finding a more compact latent space, or for increasing the forecast accuracy for the same size of the latent space. In this section we

give an example of how PINODE achieves both goals.

For this experiment we generate 16384 trajectories as described in (2.3.5). We also generate 100000 collocations as described in (2.3.6). We use the exceedingly large amount of data to allow the models to achieve the best performance for the allowed dimension of the latent space. We evaluate the performance of the models on test data with two different time-frames: (1) same as the training data (*interpolation*), and (2) two times longer than training data (*extrapolation*). More details of the experimental setup are provided in Appendix (??).

We compare the performance of three models: DMD, PIKN Hybrid, and PINODE Hybrid. First, we notice that DMD, despite achieving small loss values on training data ($\sim 10^{-3}$), does not perform well on test data. This observation is consistent with earlier works ([Kalur et al. \(2021\)](#); [Kutz et al. \(2016\)](#)); it illustrates well that a combination of a linear encoder and a linear latent dynamics may not be sufficient for modelling highly-nonlinear phenomena. Second, we notice that PINODE achieves better performance within a given latent space budget. For instance, for $m = 16$ (Figure (2.3.5), 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 low error when predicting for a longer-term horizon (extrapolation in time), which is not the case for PIKN (Figure (2.3.5, right 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 (2.3.5), 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.

2.3.3 Training in Low-Data Regime with Collocations

In the next experiment we study relative efficiency of using collocations 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 data with providing collocations which are considerably cheaper to generate. The collocations, however, can be a “weaker” signal for the model than real data, especially when time passes and the shape of the solution moves out of the collocation family’s span. In this section we show that collocations can be effectively used for this goal, when chosen right, and their contribution to a model’s accuracy may even surpass the contribution of the data.

To illustrate the trade-off between data and collocations, we train one model per various combinations of number of trajectories vs collocations in their training sets. To gauge the extrapolation power of our models, we use trajectories with three types of initial conditions: “harmonic”, “bell-curve”, and “bumps” (see Figure (2.3.4) for equations and illustrations). For training, we use 2048 trajectories with “bumps” initial conditions. For collocations, we use

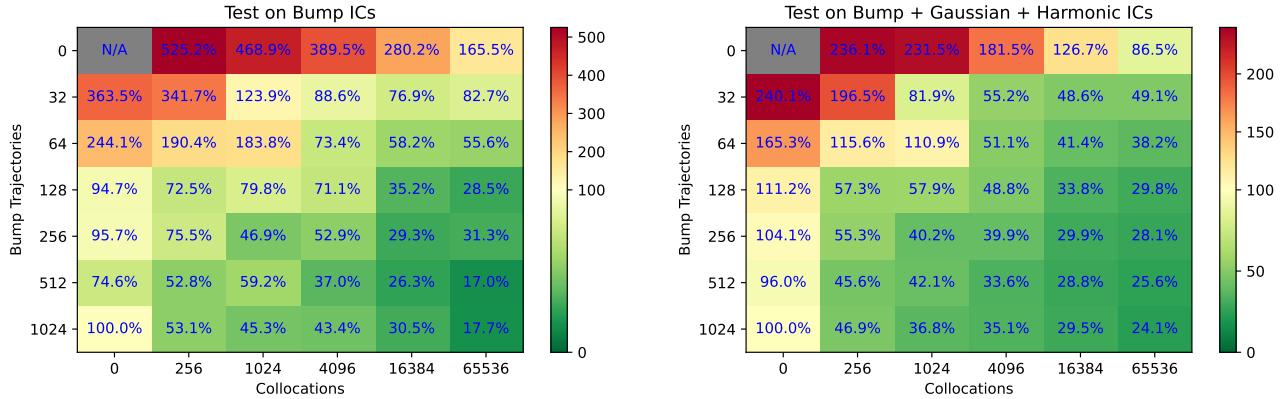


Figure 2.3.6: The figure shows a comparison of the achievable MSE relative to the full data regime (1024 trajectories). When the data is scarce, collocations-based physics-informed loss improves the forecasting accuracy of ROMs by an average of 5 times lower MSE compared to the data-only regime, as shown in this experiment with Burger’s equation. Moreover, 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).

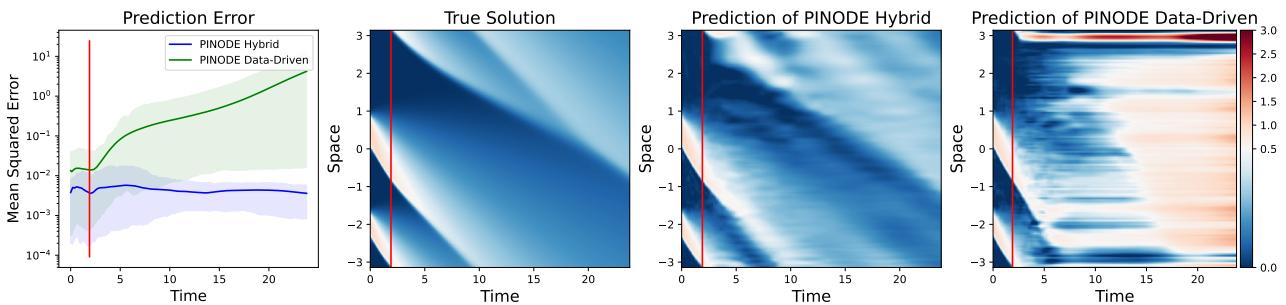


Figure 2.3.7: The first subplot shows the relative error of solving Burger’s equations on 100 test (unseen) initial conditions for two models: PINODE Hybrid and PINODE Data-Driven. Both models interpolate well but a purely data-driven model fails to extrapolate past the time-period that it was trained on (to the left of the red line). PINODE-Hybrid provides stable long-term predictions that indicates a high-quality of a discovered low-dimensional manifold dynamics. An example of such effect is provided on subplots 2-4.

the harmonic family as described in (2.3.6). We use two test datasets: (1) 100 trajectories with “bump” ICs to assess within-distribution performance, left frame), and (2) a mix of trajectories with “bump”, “bell-curve”, and “harmonic” initial conditions, 100 of each, to assess out-of-distribution performance. All test data trajectories were two times longer than the train trajectories. More details on the experimental setup are provided in Appendix ???. We fit one PINODE model with the latent dimension $m = 16$ per each combination of the amount of data and collocations. On Figure (2.3.6) we report its performance on two test datasets relative to the richest data-driven model with no collocations (bottom-left corner).

From the results displayed on Figure (2.3.6) we see that adding collocations always improved the model in our experiments. Moreover, when a sufficient number of collocations is added in training, the model with fewer data snapshots was always able to outperform the model with the maximum amount of data but no collocations. On average, a collocation-aided model was *5 times better* at both within-distribution and out-of-distribution performance relative to a purely data-driven version of the model. In addition, we noticed that a model that used only collocations can perform better than a data-rich model, especially when predicting the dynamics of the unseen initial conditions (Figure (2.3.6), 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 (2.3.7) we visualize the predictions for a test IC for two models: Data-Driven model from the bottom-left corner of Figure (2.3.6), and a Hybrid model from the bottom-right corner of Figure (2.3.6). 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, Data-Driven model shows similarly low errors within its training time-region but the forecast errors grow quickly when forecasting beyond that.

2.3.4 Robustness to Noise in Low-Data Regime

In this section we show that the use of collocations improves the ROM’s robustness when the data is noisy by providing an alternative, noise-free, source of information.

For this experiment, we use the datasets from the bottom-right cell of Figure (2.3.6), Pane 2. Namely, we use 1024 trajectories with "bump" initial conditions (an example is on Figure (2.3.4)), and we use 65536 "harmonic" collocations as defined in Equation 2.3.6. We then add i.i.d. Gaussian noise to the trajectories, with the 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 2 from the previous section; this dataset consists of three types of noise-free trajectories (see Figure 2.3.4), 100 each. The errors are displayed on Figure (2.3.8), left pane. The error of a purely Physics-Informed model (in red) is flat because the collocations are

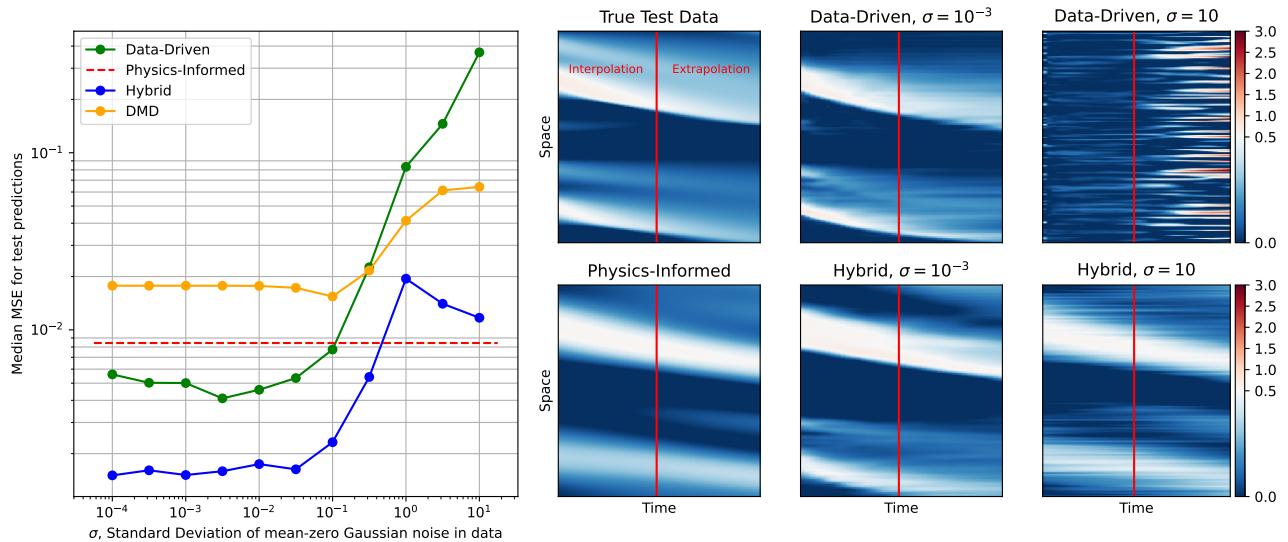


Figure 2.3.8: Physics-informed loss works as a safeguard that prevents unbounded performance drop when quality of the data degrades due to noise. Namely, the solution of the hybrid loss (2.2.10) converges to the solution of the physics-informed loss (2.2.9), when the data-driven loss (2.2.9) becomes 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.

noise-free and thus the same for all noise-levels, so we only train this model once.

On Figure (2.3.8) we see that, when noise is high, the error of purely data-driven models grows unbounded, whereas the performance of the hybrid method converges to the performance of the Physics-Informed model. We hypothesise that it happens because the second part ($\mathcal{L}_\theta^{data}$) of the combined loss (Eq. 2.2.10) turns into noise, and so its derivative also turns into noise.

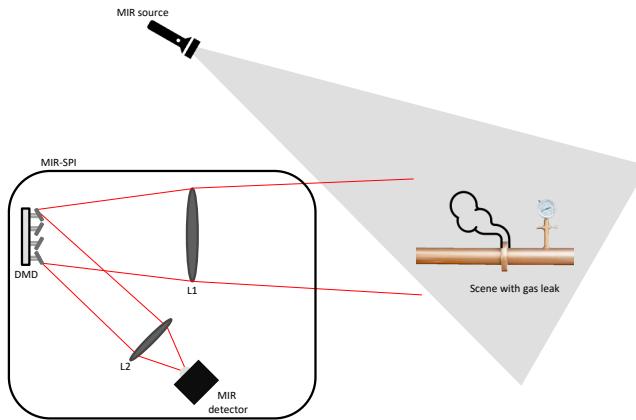
$$\nabla \mathcal{L}_\theta = \underbrace{\nabla \mathcal{L}_\theta^{physics}}_{\text{informative}} + \underbrace{\nabla \mathcal{L}_\theta^{data}}_{\text{noise}} \quad (2.3.7)$$

Thus, one can think about optimizing a hybrid model (2.2.10) as about training Physics-Informed model (2.2.9) 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 minima of its loss (in this case $\mathcal{L}_\theta^{physics}$) with high probability. So instead of diverging, a hybrid model turns into Physics-Informed model; where the latter works as a performance safeguard in the high-noise regime. On the right hand-side of Figure (2.3.8) 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 (which is expected), whereas a hybrid model in this regime produces a solution that is visually similar to the one that Physics-Informed model produces. A more rigorous analysis of this phenomenon seems possible but lies outside of the scope of this paper.

2.4 Discussion and Conclusions

In this work we demonstrated how a collocation point technique can improve the performance of an emerging class of continuous-time neural-network based reduced-order models. First, it can “cover the gaps” in datasets and inform the model about underrepresented basins of attraction. It alleviates the demand for large volumes of data that is common in network-based models, which is crucial in applications where the data is scarce and expensive. Second, the physics-informed loss may work as a safeguard, providing a noise-independent source of truth. Third, collocations can stabilize the model’s long-term predictions, allowing to accurately forecast far beyond the training horizon. Finally, together with using a NODE-based non-linear latent dynamics, adding physics-informed loss leads to discovery of more compact latent spaces that yield more accurate models. Simultaneous stability and compactness is especially important if one aims to use models together with compressive sensing and control algorithms.

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 more efficient way of sampling collocation points, possibly via applying modern adaptive learning techniques [Subramanian et al. \(2022\)](#). Finally, although Section 2.3.4 provides some rationale 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.



2.5 Application to Compressive Sensing

2.5.1 Introduction

Prior Work

2.5.2 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:

$$y(t) = A_t x(t), \quad A_t \in \mathbb{R}^{p \times n} \quad (2.5.1)$$

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 ?? 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 \mathbf{x}_i , $i \in [1 \dots k]$,

where each trajectory $\mathbf{x}_i \in \mathbb{R}^{n \times p}$ 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:

$$\mathcal{L}^{training}(\theta) = \frac{1}{2\sigma^2} \sum_{i=1}^k \left[\frac{\omega_1}{p} \sum_{j=1}^p \|\mathbf{x}_i(t_j) - \psi_\theta(\phi_\theta(\mathbf{x}_i(t_j)))\|^2 + \right. \quad (2.5.2)$$

$$\left. + \frac{\omega_2}{p} \sum_{j=1}^p \left\| \psi_\theta \left(\phi_\theta(\mathbf{x}_i(t_1)) + \int_{t_1}^{t_j} h(z(t)) dt \right) - \mathbf{x}_i(t_j) \right\|^2 \right] \quad (2.5.3)$$

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:

$$\theta^* = \arg \min_{\theta} \mathcal{L}^{training}(\theta) \quad (2.5.4)$$

We note that each trajectory \mathbf{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 accordingly². 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 $\psi_{\theta^*}(z)$.

$$\min_{\{z_t\}_{t=1,\dots,T}} \frac{1}{2} \sum_{t=1}^T \|y_t - A\psi_{\theta^*}(z_t)\|_2^2 \quad (2.5.5)$$

$$\text{s.t. } \dot{z} = h_{\theta^*}(z) \quad (2.5.6)$$

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

$$\min_{\{z_t\}_{t=1,\dots,T}} \mathcal{L}_{\theta^*}^{recon}(z) \quad (2.5.7)$$

where

$$\mathcal{L}_{\theta^*}^{recon}(z) = \frac{1}{2} \sum_{t=1}^T \|y_t - A\psi_{\theta^*}(z_t)\|_2^2 + \frac{\lambda}{2} \sum_{t=1}^T \left\| z_{t-1} + \int_{t-1}^t h_{\theta^*}(z) dz - z_t \right\|_2^2 \quad (2.5.8)$$

²The implementation is affected only in evaluating the integral in (2.2.4). 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 2.5.8 using a gradient-based technique, with the gradients obtained using automatic differentiation frameworks.

2.5.3 Experiments

2.5.4 Discussion and Conclusion

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., Sholokhov, A., and Zheng, P. (2022). 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.

- 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.
- 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.
- 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.
- 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.
- 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.
- 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.

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.

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. (2019). 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](https://arxiv.org/abs/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.
- 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.
- 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](https://arxiv.org/abs/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.
- 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.
- 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. (1997). *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics.
- 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., 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.
- 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 1.2.3 takes the form

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

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

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

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

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

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

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

Similarly,

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

Taking into account that Ω_i is symmetric, we have

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

or, in vector form

$$= \sum_{i=1}^m \text{diag}(Z_i^T \Omega_i^{-1} Z_i) - (Z_i^T \Omega_i^{-T} \xi_i)^{\circ 2} = \quad (\text{A.2.7})$$

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

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

Notice, that the loss function (1.2.3) and the optimal β solution (??) can also be effectively computed using Cholesky:

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

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

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

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

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

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

A.3 Description of Datasets

A.3.1 GBD Bullying Data

The author acknowledges his colleague and collaborator Damian Santomauro¹ for providing the dataset, the description of its covariates, and the expert assessment of their historical importance in different rounds of GBD study below.

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

1. cv_symptoms

- 0 = study assesses participants for MDD or anxiety disorders via a diagnostic interview to determine whether they have a diagnosis.
- 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 = $\geq 15\%$ 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.3.2 COVID-19 Contact Rate Forecasting Data

Table A.3.1: List of locations, number of observations, start and end date for each location for COVID-19 Contact Rate Focecasting 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

Continued on next page

Table A.3.1: 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 Moldova	48	2020-03-11	2020-04-27
Ukraine	53	2020-03-06	2020-04-27
Japan	68	2020-02-20	2020-04-27
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

Continued on next page

Table A.3.1: List of locations, number of observations, start and end date for each location for COVID-19 Contact Rate Forecasting data

Location	Obs	Start	End
Arkansas	52	2020-03-07	2020-04-27
California	67	2020-02-21	2020-04-27
Colorado	54	2020-03-05	2020-04-27
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.3.2: List of location-specific coefficients for the R&S-Mixed model fit, as well as RMSEs for three models discussed in the respective chapter. Coefficient 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

Continued on next page

Table A.3.2: List of location-specific coefficients for the R&S-Mixed model fit, as well as RMSEs for three models discussed in the respective chapter. Coefficient 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
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
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

Continued on next page

Table A.3.2: List of location-specific coefficients for the R&S-Mixed model fit, as well as RMSEs for three models discussed in the respective chapter. Coefficient 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
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
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