

Optimization Methods for Parameter Identification in Settings with Only Partial Knowledge

Ph.D. Defense

Aleksei Sholokhov

Tuesday 6th June, 2023



Plan of the Defense

Plan of the Defense

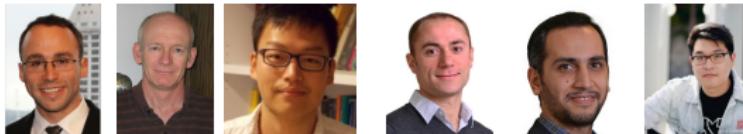


Sparse Relaxed Regularized Regression for Linear Mixed-Effect Models

Together with Aleksandr Aravkin, James Burke, Damian Santomauro, and Peng Zheng

- ▶ "A Relaxation Approach to Feature Selection for Linear Mixed Effects Models"
submitted to Journal of Computational and Graphical Statistics (JCGS)
- ▶ "Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models"
submitted to Journal of Computational Optimization and Applications (COAP)
- ▶ "pysr3: A Python Package for Sparse Relaxed Regularized Regression"
Journal of Open-Source Software (JOSS), 2023, ICCOPT 2022, SIAM OPT 2023

Plan of the Defense



Sparse Relaxed Regularized Regression for Linear Mixed-Effect Models

Together with Aleksandr Aravkin, James Burke, Damian Santomauro, and Peng Zheng

- ▶ "A Relaxation Approach to Feature Selection for Linear Mixed Effects Models"
submitted to Journal of Computational and Graphical Statistics (JCGS)
- ▶ "Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models"
submitted to Journal of Computational Optimization and Applications (COAP)
- ▶ "pysr3: A Python Package for Sparse Relaxed Regularized Regression"
Journal of Open-Source Software (JOSS), 2023, ICCOPT 2022, SIAM OPT 2023

PINODE: Physics-Informed Neural ODEs

Together with Hassan Mansour, Saleh Nabi, and Yuying Liu

- ▶ "Physics-Informed Koopman Network"
SIAM ADS 2023, arXiv:2211.09419
- ▶ "Physics-Informed Neural ODE: Embedding Physics into Models using Collocation Points"
Submitted to Nature Special Issue on Physics-Informed Machine Learning

Plan of the Defense



Sparse Relaxed Regularized Regression for Linear Mixed-Effect Models

Together with Aleksandr Aravkin, James Burke, Damian Santomauro, and Peng Zheng

- ▶ "A Relaxation Approach to Feature Selection for Linear Mixed Effects Models"
submitted to Journal of Computational and Graphical Statistics (JCGS)
- ▶ "Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models"
submitted to Journal of Computational Optimization and Applications (COAP)
- ▶ "pysr3: A Python Package for Sparse Relaxed Regularized Regression"
Journal of Open-Source Software (JOSS), 2023, ICCOPT 2022, SIAM OPT 2023

PINODE: Physics-Informed Neural ODEs

Together with Hassan Mansour, Saleh Nabi, and Yuying Liu

- ▶ "Physics-Informed Koopman Network"
SIAM ADS 2023, arXiv:2211.09419
- ▶ "Physics-Informed Neural ODE: Embedding Physics into Models using Collocation Points"
Submitted to Nature Special Issue on Physics-Informed Machine Learning

Single-Pixel Imaging with Reduced-Order Models

Together with J. Nathan Kutz, Steven Brunton, Joshua Rapp, Hassan Mansour, and Saleh Nabi

- ▶ "Single pixel imaging of spatio-temporal flows using differentiable latent dynamics" *in preparation*

Plan of the Defense



Sparse Relaxed Regularized Regression for Linear Mixed-Effect Models

Together with Aleksandr Aravkin, James Burke, Damian Santomauro, and Peng Zheng

- ▶ ["A Relaxation Approach to Feature Selection for Linear Mixed Effects Models"](#)
submitted to Journal of Computational and Graphical Statistics (JCGS)
- ▶ "Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models"
submitted to Journal of Computational Optimization and Applications (COAP)
- ▶ "pysr3: A Python Package for Sparse Relaxed Regularized Regression"
Journal of Open-Source Software (JOSS), 2023, ICCOPT 2022, SIAM OPT 2023

PINODE: Physics-Informed Neural ODEs

Together with Hassan Mansour, Saleh Nabi, and Yuying Liu

- ▶ "Physics-Informed Koopman Network"
SIAM ADS 2023, arXiv:2211.09419
- ▶ ["Physics-Informed Neural ODE: Embedding Physics into Models using Collocation Points"](#)
Submitted to Nature Special Issue on Physics-Informed Machine Learning

Single-Pixel Imaging with Reduced-Order Models

Together with J. Nathan Kutz, Steven Brunton, Joshua Rapp, Hassan Mansour, and Saleh Nabi

- ▶ ["Single pixel imaging of spatio-temporal flows using differentiable latent dynamics"](#) *in preparation*

MSR3 – Sparse Relaxed Regularized Regression for Linear Mixed-Effect Models

Research Objectives and Plan of the Talk

Objectives:

- ▶ Extend $\mathcal{SR}3$ relaxation to linear mixed-effects models - $\mathcal{MSR}3$ (see¹).
- ▶ Develop theoretical foundations for it (see²).
- ▶ Implement it as a scikit-learn-compatible Python package – `pysr3` (see³).

Plan of the Talk:

1. Proximal Gradient Descent (PGD) for Linear Mixed-Effects Models (LMEs)
2. $\mathcal{MSR}3$ – PGD on $\mathcal{SR}3$ -relaxation for LMEs
3. $\mathcal{MSR}3$ -fast – practical algorithm for feature selection

¹Sholokhov, J. V. Burke, et al., A Relaxation Approach to Feature Selection for Linear Mixed Effects Models.

²Aravkin et al., Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models.

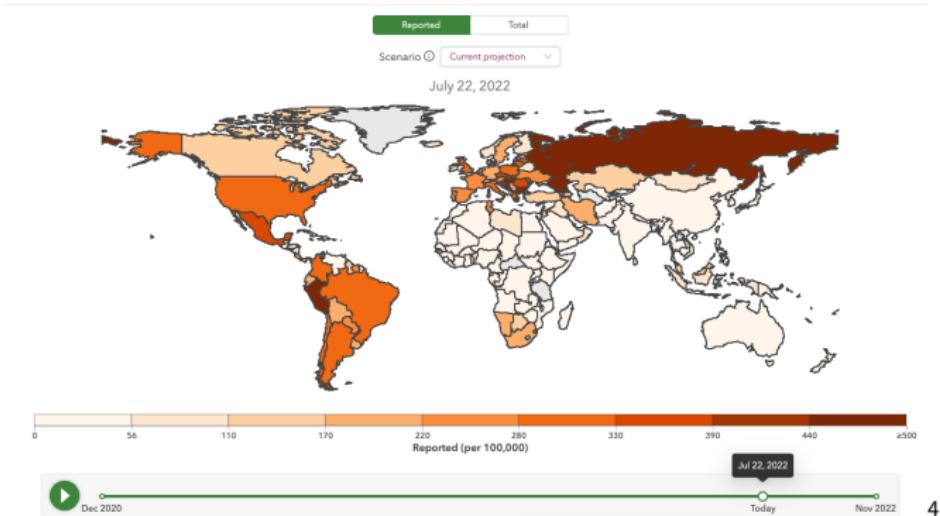
³Sholokhov, Zheng, and Aravkin, "pysr3: A Python Package for Sparse Relaxed Regularized Regression".



Mixed-Effect Models

Mixed-effect models

- ▶ Used for analyzing **combined data** across a range of **groups**.
 - ▶ **Borrow strength** across groups to estimate key statistics.
 - ▶ Use covariates to separate the **population variability** from the **group variability**.



⁴Picture is taken from covid19.healthdata.org

Linear Mixed-Effect (LME) Models

Dataset: m groups (X_i, Z_i, y_i) , $i = 1, \dots, m$, each has n_i observations

- ▶ $X_i \in \mathbb{R}^{n_i \times p}$ – group i design matrix for fixed features
- ▶ $Z_i \in \mathbb{R}^{n_i \times q}$ – group i design matrix for random features
- ▶ $y_i \in \mathbb{R}^{n_i}$ – group i observations
- ▶ $u_i \in \mathbb{R}^q$ – random effects
- ▶ $\Gamma \in \mathbb{R}^{q \times q}$ – covariance matrix of random effects, often $\Gamma = \text{diag}(\gamma)$
- ▶ $\Lambda_i \in \mathbb{R}^{n_i \times n_i}$ – covariance matrix for observation noise

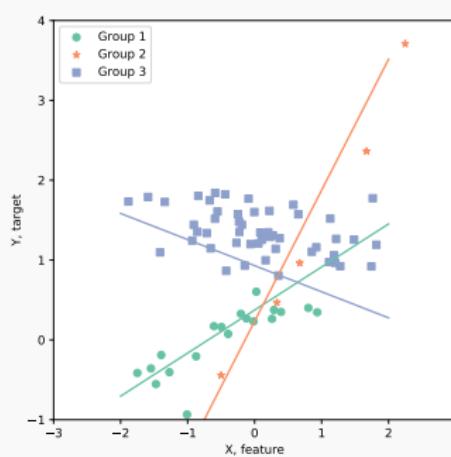
Model:

$$y_i = X_i\beta + Z_i u_i + \varepsilon_i$$

$$\varepsilon_i \sim \mathcal{N}(0, \Lambda_i)$$

$$u_i \sim \mathcal{N}(0, \Gamma)$$

Unknowns: β , u_i , γ , sometimes Λ_i .



Negative Log-Likelihood for Mixed-Effect Models

Optimization problem:

$$\mathcal{FS} - \mathcal{LME} = \min_{\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}_+^q} \mathcal{L}(\beta, \gamma) + R(\beta, \gamma) \quad (1)$$

Where \mathcal{L} :

$$\begin{aligned} \mathcal{L}(\beta, \gamma) = & \sum_{i=1}^m \frac{1}{2} (y_i - X_i \beta)^T (Z_i \Gamma Z_i^T + \Lambda_i)^{-1} (y_i - X_i \beta) + \\ & + \frac{1}{2} \log \det (Z_i \Gamma Z_i^T + \Lambda_i), \quad \Gamma = \text{diag}((\gamma)) \end{aligned} \quad (2)$$

- ▶ $\mathcal{L}(\beta, \gamma)$ is smooth on its domain, quadratic w.r.t. β and $\bar{\eta}$ -weakly-convex w.r.t. γ .
- ▶ $R(\beta, \gamma)$ is closed, proper, with easily computed *prox operator*

Regularization

$R(\beta, \gamma)$ is closed, proper, with easily computed *prox operator*

$$\text{prox}_{\alpha R + \delta_{\mathcal{C}}}(\tilde{\beta}, \tilde{\gamma}) := \underset{(\beta, \gamma) \in \mathcal{C}}{\operatorname{argmin}} R(\beta, \gamma) + \frac{1}{2\alpha} \|(\beta, \gamma) - (\tilde{\beta}, \tilde{\gamma})\|_2^2, \quad (3)$$

where $\mathcal{C} := \mathbb{R}^p \times \mathbb{R}_+^q$

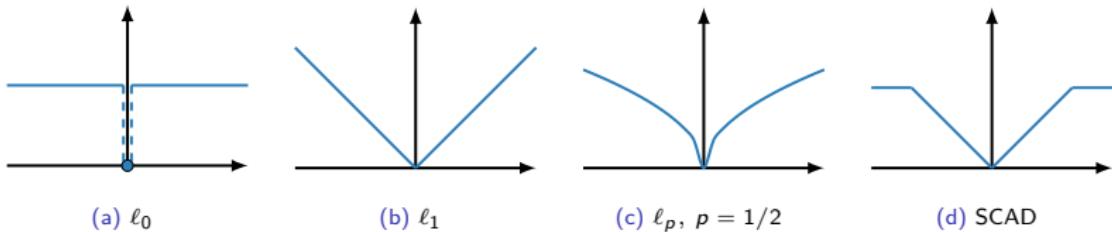


Figure: Four commonly-used regularizers which promote sparsity

Proximal Gradient Descent for Feature Selection in MLE

Let

$$x := [\beta, \gamma], \quad \mathcal{C} := \mathbb{R}^p \times \mathbb{R}_+^q \quad (4)$$

Optimization problem $\mathcal{FS} - \mathcal{LME}$:

$$\min_{x \in \mathcal{C}} \mathcal{L}(x) + R(x) \quad (5)$$

Proximal Gradient Descent for Feature Selection in MLE

Let

$$x := [\beta, \gamma], \quad \mathcal{C} := \mathbb{R}^p \times \mathbb{R}_+^q \quad (4)$$

Optimization problem $\mathcal{FS} - \mathcal{LME}$:

$$\min_{x \in \mathcal{C}} \mathcal{L}(x) + R(x) \quad (5)$$

1 **Algorithm:** PGD for standard LMEs

2 $\beta^+ \leftarrow \beta_0, \quad \gamma^+ \leftarrow \gamma_0, \quad \alpha \leftarrow 1/L$ // Initialization
3 $x^+ = [\beta^+, \gamma^+];$
4 **while** making progress **do**
5 $| \quad x^+ \leftarrow \text{prox}_{\alpha^{-1}R+\delta_{\mathcal{C}}} (x^+ - \alpha \nabla_x \mathcal{L}(x^+))$ // PGD iterations
6 **end**
7 **return** $x^+ = [\beta^+, \gamma^+]$

Proximal Gradient Descent for Feature Selection in MLE

```
1 Algorithm: PGD for standard LMEs
2  $\beta^+ \leftarrow \beta_0, \gamma^+ \leftarrow \gamma_0, \alpha \leftarrow 1/L$                                 // Initialization
3  $x^+ = [\beta^+, \gamma^+];$ 
4 while making progress do
5   |  $x^+ \leftarrow \text{prox}_{\alpha^{-1}R + \delta_C}(x^+ - \alpha \nabla_x \mathcal{L}(x^+))$           // PGD iterations
6 end
7 return  $x^+ = [\beta^+, \gamma^+]$ 
```

Basic Assumptions for the PGD Algorithm⁵:

1. R is a closed proper convex function
2. \mathcal{L} is closed and proper, $\text{dom } \mathcal{L}$ convex, $\text{dom } R \subset \text{int}(\text{dom } \mathcal{L})$, and \mathcal{L} is L -smooth over $\text{int}(\text{dom } \mathcal{L})$. [x].
3. The problem has an optimal solution⁶ with an optimal value \mathcal{L}^*

Algorithm converges with backtracking⁷.

⁵ Beck, First-Order Methods in Optimization, Theorem 10.15.

⁶ Aravkin et al., Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models.

⁷ J. Burke and Engle, "Line Search and Trust-Region Methods for Convex-Composite Optimization".

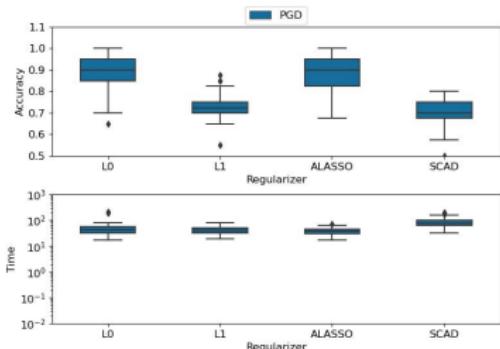
Proximal Gradient Descent for Feature Selection in MLE

1 **Algorithm:** PGD for standard LMEs

```
2  $\beta^+ \leftarrow \beta_0, \gamma^+ \leftarrow \gamma_0, \alpha \leftarrow 1/L$  // Initialization  
3  $x^+ = [\beta^+, \gamma^+];$   
4 while making progress do  
5    $x^+ \leftarrow \text{prox}_{\alpha^{-1}R+\delta_C}(x^+ - \alpha \nabla_x \mathcal{L}(x^+))$  // PGD iterations  
6 end  
7 return  $x^+ = [\beta^+, \gamma^+]$ 
```

Synthetic Benchmark:

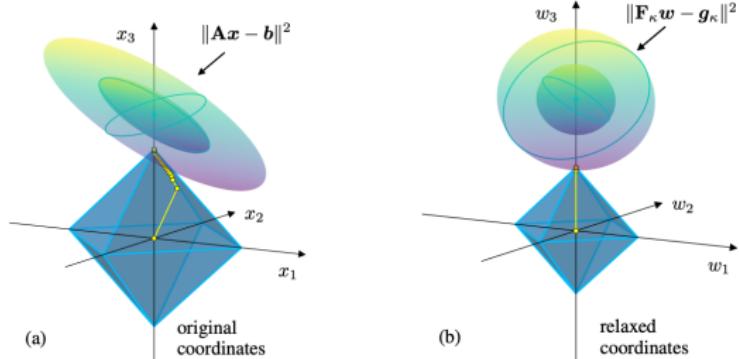
- ▶ 100 randomly-generated problems.
- ▶ $p = q = 20$.
- ▶ $\beta = \gamma = \frac{1}{2}[1, 2, 3, \dots, 10, 0, \dots, 0]$
- ▶ 9 groups from 3 to 15 observations
- ▶ $X_i \sim \mathcal{N}(0, I)^p, Z_i = X_i, \varepsilon_i \sim \mathcal{N}(0, 0.3^2 I)$
- ▶ Golden search for $\lambda \in [0, 10^5]$
- ▶ Final model is chosen to maximize BIC



Sparse Relaxed Regularized Regression ($\mathcal{SR}3$)⁵

Original \rightarrow Decoupled \rightarrow Value Function

$$\min_x f(x) + R(x) \quad \rightarrow \quad \min_{x, w} f(x) + \frac{\eta}{2} \|x - w\|_2^2 + R(w) \quad \rightarrow \quad \min_w v_\eta(w) + R(w)$$



⁵Zheng and Aravkin, "Relax-and-split method for nonconvex inverse problems"

SR3-Relaxation for Mixed-Effect Models ($MSR3$)

Original problem $\mathcal{FS} - \mathcal{LME}$:

$$\min_{x \in \mathcal{C}} \mathcal{L}(x) + R(x) \quad (4)$$

SR3-Relaxation for Mixed-Effect Models ($MSR3$)

Original problem $\mathcal{FS} - \mathcal{LME}$:

$$\min_{x \in \mathcal{C}} \mathcal{L}(x) + R(x) \quad (4)$$

Decoupled problem:

$$\min_{x, w \in \mathcal{C}} \mathcal{L}(x) + \frac{\eta}{2} \|x - w\|_2^2 + R(w) \quad (5)$$

where $w = [\tilde{\beta}, \tilde{\gamma}]$.

SR3-Relaxation for Mixed-Effect Models ($\mathcal{MSR}3$)

Original problem $\mathcal{FS} - \mathcal{LME}$:

$$\min_{x \in \mathcal{C}} \mathcal{L}(x) + R(x) \quad (4)$$

Decoupled problem:

$$\min_{x, w \in \mathcal{C}} \mathcal{L}(x) + \frac{\eta}{2} \|x - w\|_2^2 + R(w) \quad (5)$$

where $w = [\tilde{\beta}, \tilde{\gamma}]$.

By partially minimizing w.r.t. x we get a value function $v_\eta(w)$:

$$v_\eta(w) := \min_{x \in \mathcal{C}} \mathcal{L}(x) + \frac{\eta}{2} \|x - w\|_2^2 \quad (6)$$

so $\mathcal{SR}3$ -formulation (5) becomes

$$\min_{w \in \mathcal{C}} v_\eta(w) + R(w) \quad (7)$$

SR3-Relaxation for Mixed-Effect Models ($MSR3$)

Original problem $\mathcal{FS} - \mathcal{LME}$:

$$\min_{x \in \mathcal{C}} \mathcal{L}(x) + R(x) \quad (4)$$

Decoupled problem:

$$\min_{x, w \in \mathcal{C}} \mathcal{L}(x) + \frac{\eta}{2} \|x - w\|_2^2 + R(w) \quad (5)$$

where $w = [\tilde{\beta}, \tilde{\gamma}]$.

By partially minimizing w.r.t. x we get a value function $v_\eta(w)$:

$$v_\eta(w) := \min_{x \in \mathcal{C}} \mathcal{L}(x) + \frac{\eta}{2} \|x - w\|_2^2 \quad (6)$$

so $\mathcal{SR3}$ -formulation (5) becomes

$$\min_{w \in \mathcal{C}} v_\eta(w) + R(w) \quad (7)$$

NB: $v_\eta(w)$ is smooth on \mathcal{C} and can be evaluated using Interior Point (IP) method

Value Function of $\mathcal{MSR}3$

$\mathcal{MSR}3$ -relaxation replaces the original likelihood \mathcal{L} with a *value function* $v_{\eta,\mu}$:

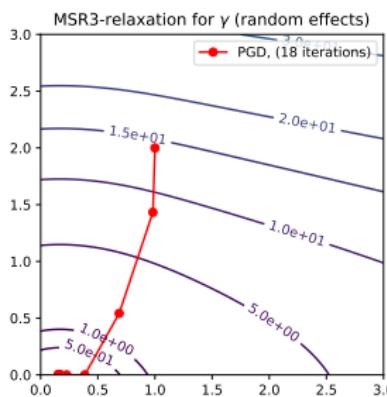
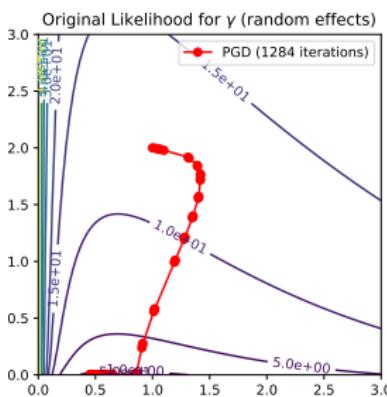
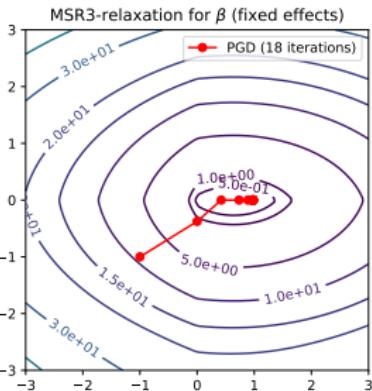
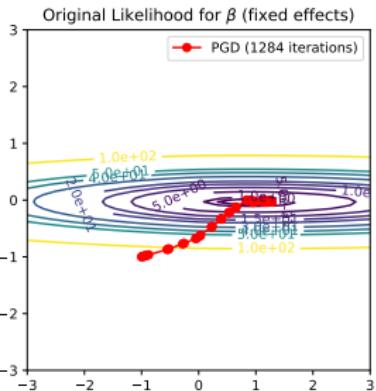
$$v_{\eta,\mu}(w) := \min_{x \in \mathcal{C}} \mathcal{L}(x) + \frac{\eta}{2} \|x - w\|_2^2 + \phi_\mu(x) \quad (8)$$

where *perspective mapping* ϕ_μ replaces $\gamma \geq 0$ with a log-barrier

$$\phi_\mu(x) = \phi_\mu(\gamma) := \begin{cases} -\mu \sum_{i=1}^q \ln(\gamma_i/\mu), & \mu > 0 \\ \delta_{\mathbb{R}_+^q}(\gamma), & \mu = 0 \\ +\infty, & \mu < 0 \end{cases} \quad (9)$$

Value Function of $MSR3$

$$\min_{\beta, \gamma \in C} \mathcal{L}(\beta, \gamma) + R(\beta, \gamma) \quad \text{vs} \quad \min_{\tilde{\beta}, \tilde{\gamma} \in C} v_{\eta, \mu}(\tilde{\beta}, \tilde{\gamma}) + R(\tilde{\beta}, \tilde{\gamma})$$



1 **Algorithm:** PGD for $\mathcal{MSR}3$

```
2  $\tilde{\beta}^+ \leftarrow \tilde{\beta}_0, \quad \tilde{\gamma}^+ \leftarrow \tilde{\gamma}_0, \quad \alpha \leftarrow 1/\eta, \quad \eta > \bar{\eta}$  // Initialization
3  $\tilde{w}^+ := [\tilde{\beta}^+, \tilde{\gamma}^+], \quad x^+ := [\beta, \gamma]$ 
4 while making progress in  $\tilde{w}$  do
5    $x^+ \leftarrow \text{IP on } \mathcal{L}_{\eta, \mu}(x^+, \tilde{w}^+) \text{ s.t. } x^+ \in \mathcal{C} \text{ and } \mu \rightarrow 0$  // IP Iterations
6    $\nabla_{\tilde{w}} v_{\eta, 0}(\tilde{w}^+) \leftarrow \nabla_{\tilde{w}} \mathcal{L}_{\eta, 0}(x^+, \tilde{w}^+)$  // Evaluate Gradient of VF
7    $\tilde{w}^+ \leftarrow \text{prox}_{\alpha^{-1}R + \delta_C}(\tilde{w}^+ - \alpha \nabla_{\tilde{w}} v_{\eta, 0}(\tilde{w}^+))$  // PGD on Value Function
8 end
9 return  $\tilde{w}^+ = [\tilde{\beta}^+, \tilde{\gamma}^+]$ 
```

1 Algorithm: PGD for $\mathcal{MSR}3$

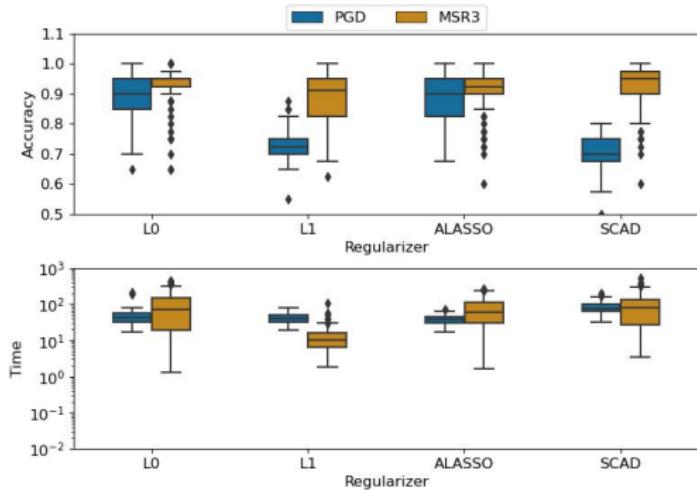
```
2  $\tilde{\beta}^+ \leftarrow \tilde{\beta}_0, \quad \tilde{\gamma}^+ \leftarrow \tilde{\gamma}_0, \quad \alpha \leftarrow 1/\eta, \quad \eta > \bar{\eta}$  // Initialization
3  $\tilde{w}^+ := [\tilde{\beta}^+, \tilde{\gamma}^+], \quad x^+ := [\beta, \gamma]$ 
4 while making progress in  $\tilde{w}$  do
5    $x^+ \leftarrow \text{IP on } \mathcal{L}_{\eta, \mu}(x^+, \tilde{w}^+) \text{ s.t. } x^+ \in \mathcal{C} \text{ and } \mu \rightarrow 0$  // IP Iterations
6    $\nabla_{\tilde{w}} v_{\eta, 0}(\tilde{w}^+) \leftarrow \nabla_{\tilde{w}} \mathcal{L}_{\eta, 0}(x^+, \tilde{w}^+)$  // Evaluate Gradient of VF
7    $\tilde{w}^+ \leftarrow \text{prox}_{\alpha^{-1}R + \delta_{\mathcal{C}}}(\tilde{w}^+ - \alpha \nabla_{\tilde{w}} v_{\eta, 0}(\tilde{w}^+))$  // PGD on Value Function
8 end
9 return  $\tilde{w}^+ = [\tilde{\beta}^+, \tilde{\gamma}^+]$ 
```

Theoretical Results⁶:

1. When $\eta > \bar{\eta}$ the problem has an optimal solution Φ^* (**Theorem 5**)
2. $v_{\eta, \mu}$ is well-defined (**Theorem 5**) and continuously differentiable (**Theorem 10**)
3. When R is 1-coercive $\nabla v_{\eta, \mu}$ is locally \widetilde{L} -continuous (**Theorem 14**)
4. Algorithm converges to a stationary point of $v_{\eta, \mu}$ (**Theorem 15**)
5. As $\mu \rightarrow 0$ (**Theorem 6**) or $\eta \rightarrow \infty$ (**Theorem 7**), cluster points of solutions to $\mathcal{MSR}3$ are FOSPs for $\mathcal{FS} - \mathcal{LME}$

⁶ Aleksandr Aravkin et al. Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models. 2022.
arXiv: 2209.10575 [stat.ME].

MSR3: Results



1 Algorithm: PGD for $\mathcal{MSR}3$

```
2  $\tilde{\beta}^+ \leftarrow \tilde{\beta}_0, \quad \tilde{\gamma}^+ \leftarrow \tilde{\gamma}_0, \quad \alpha \leftarrow 1/\eta, \quad \eta > \bar{\eta}$  // Initialization
3  $\tilde{w}^+ := [\tilde{\beta}^+, \tilde{\gamma}^+], \quad x^+ := [\beta, \gamma]$ 
4 while making progress in  $\tilde{w}$  do
5    $x^+ \leftarrow \text{IP on } \mathcal{L}_{\eta, \mu}(x^+, \tilde{w}^+) \text{ s.t. } x^+ \in \mathcal{C} \text{ and } \mu \rightarrow 0$  // IP Iterations
6    $\nabla_{\tilde{w}} v_{\eta, 0}(\tilde{w}^+) \leftarrow \nabla_{\tilde{w}} \mathcal{L}_{\eta, 0}(x^+, \tilde{w}^+)$  // Evaluate Gradient
7    $\tilde{w}^+ \leftarrow \text{prox}_{\alpha^{-1}R + \delta_C}(\tilde{w}^+ - \alpha \nabla_{\tilde{w}} v_{\eta, 0}(\tilde{w}^+))$  // PGD on Value Function
8 end
9 return  $\tilde{w}^+ = [\tilde{\beta}^+, \tilde{\gamma}^+]$ 
```

$\mathcal{MSR}3$: Algorithm

1 **Algorithm:** PGD for $\mathcal{MSR}3$

```
2  $\tilde{\beta}^+ \leftarrow \tilde{\beta}_0, \quad \tilde{\gamma}^+ \leftarrow \tilde{\gamma}_0, \quad \alpha \leftarrow 1/\eta, \quad \eta > \bar{\eta}$  // Initialization
3  $\tilde{w}^+ := [\tilde{\beta}^+, \tilde{\gamma}^+], \quad x^+ := [\beta, \gamma]$ 
4 while making progress in  $\tilde{w}$  do
5    $x^+ \leftarrow \text{IP on } \mathcal{L}_{\eta, \mu}(x^+, \tilde{w}^+) \text{ s.t. } x^+ \in \mathcal{C} \text{ and } \mu \rightarrow 0$  // IP Iterations
6    $\nabla_{\tilde{w}} v_{\eta, 0}(\tilde{w}^+) \leftarrow \nabla_{\tilde{w}} \mathcal{L}_{\eta, 0}(x^+, \tilde{w}^+)$  // Evaluate Gradient
7    $\tilde{w}^+ \leftarrow \text{prox}_{\alpha^{-1}R + \delta_{\mathcal{C}}}(\tilde{w}^+ - \alpha \nabla_{\tilde{w}} v_{\eta, 0}(\tilde{w}^+))$  // PGD on Value Function
8 end
9 return  $\tilde{w}^+ = [\tilde{\beta}^+, \tilde{\gamma}^+]$ 
```

Key Observation: $\nabla_{\tilde{w}} v(\tilde{w})$ does not need to be evaluated exactly. We only need to come close enough to the central path.

$\mathcal{MSR}3$ -fast: Algorithm

1 **Algorithm:** $\mathcal{MSR}3$ -fast

```
2  $\tilde{\beta}^+ \leftarrow \tilde{\beta}_0, \quad \tilde{\gamma}^+ \leftarrow \tilde{\gamma}_0, \quad \alpha \leftarrow 1/\eta, \quad \eta > \bar{\eta}$  // Initialization
3  $\tilde{w}^+ := [\tilde{\beta}^+, \tilde{\gamma}^+], \quad x^+ := [\beta, \gamma]$ 
4 while making progress do
5   while not close enough to the central path do
6      $| \quad x^+ \leftarrow \text{IP iteration on } \mathcal{L}_{\eta, \mu}(x^+, \tilde{w}^+) \text{ s.t. } x^+ \in \mathcal{C}$  // IP Iterations
7   end
8   Decrease  $\mu$ 
9    $\nabla_{\tilde{w}} v_{\eta, \mu}(\tilde{w}^+) \leftarrow \nabla_{\tilde{w}} \mathcal{L}_{\eta, \mu}(x^+, \tilde{w}^+)$  // Evaluate Gradient
10   $\tilde{w}^+ \leftarrow \text{prox}_{\alpha^{-1}\mathcal{R} + \delta_{\mathcal{C}}}(\tilde{w}^+ - \alpha \nabla_{\tilde{w}} v_{\eta, \mu}(\tilde{w}^+))$  // PGD on Value Function
11 end
12 return  $\tilde{w}^+ = [\tilde{\beta}^+, \tilde{\gamma}^+]$ 
```

Designing an Algorithm

$G_{\nu, \eta}$ encodes both gradient of a Lagrangian (lines 1-2) and the complementarity condition (line 3):

$$G_{\nu, \eta}((\beta, \gamma, \nu), (\tilde{\beta}, \tilde{\gamma})) := \begin{bmatrix} \nabla_\beta \mathcal{L}(\beta, \gamma) + \eta(\beta - \tilde{\beta}) \\ \nabla_\gamma \mathcal{L}(\beta, \gamma) + \eta(\gamma - \tilde{\gamma}) - \nu \\ \nu \odot \gamma - \mu \mathbf{1} \end{bmatrix} \quad (10)$$

We apply Newton method to G while geometrically decreasing μ .

Lemma: For every $(\mu, \eta) \in \mathbb{R}_+ \times \mathbb{R}_{++}$,

$$\begin{aligned} (\hat{\beta}, \hat{\gamma}) &= \operatorname{argmin}_{(\beta, \gamma)} \mathcal{L}_{\eta, \mu}((\beta, \gamma), (\tilde{\beta}, \tilde{\gamma})) \\ &\iff \exists \hat{\nu} \in \mathbb{R}_+^q \text{ s.t. } G_{\nu, \eta}((\beta, \gamma, \hat{\nu}), (\tilde{\beta}, \tilde{\gamma})) = 0 \end{aligned} \quad (11)$$

If $\mu > 0$, then $\hat{\nu} = -\nabla \phi_\mu(\hat{\gamma})$, and if $\mu = 0$, then $\hat{\nu}$ is the unique KKT multiplier associated with the constraint $0 \leq \gamma$.

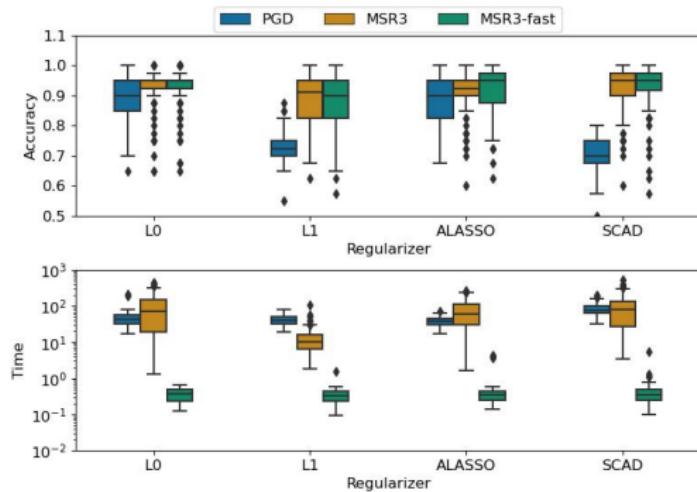
```

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  $\|G_\mu(\beta^+, \gamma^+, v^+)\| > tol$  and progress
do
4    $\beta \leftarrow \beta^+; \gamma \leftarrow \gamma^+; \tilde{\beta} \leftarrow \tilde{\beta}^+; \tilde{\gamma} \leftarrow \tilde{\gamma}^+$ 
5    $[dv, d\beta, d\gamma] \leftarrow \nabla G_\mu((\beta, \gamma, v), (\tilde{\beta}, \tilde{\gamma}))^{-1} G_\mu((\beta, \gamma, v), (\tilde{\beta}, \tilde{\gamma}))$ 
      $\alpha \leftarrow 0.99 \times \min \left( 1, -\frac{\gamma_i}{d\gamma_i}, \forall i : d\gamma_i < 0 \right)$ 
6    $\beta^+ \leftarrow \beta + \alpha d\beta; \gamma^+ = \gamma + \alpha d\gamma; v^+ \leftarrow v + \alpha dv$ 
7   if  $\|\gamma^+ \odot v^+ - q^{-1} \gamma^{+T} v^+ \mathbf{1}\| > 0.5q^{-1} v^{+T} \gamma^+$  then continue;
8   else
9      $\tilde{\beta}^+ = \text{prox}_{\alpha R}(\beta^+); \tilde{\gamma}^+ = \text{prox}_{\alpha R + \delta_{\mathbb{R}_+}}(\gamma^+); \mu = \frac{1}{10} \frac{v^{+T} \gamma^+}{q}$ 
10  end
11  progress = ( $\|\beta^+ - \beta\| \geq tol$  or  $\|\gamma^+ - \gamma\| \geq tol$  or  $\|\tilde{\beta}^+ - \tilde{\beta}\| \geq tol$  or
       $\|\tilde{\gamma}^+ - \tilde{\gamma}\| \geq tol$ )
12  iter += 1
13 end
14 return  $\tilde{\beta}^+, \tilde{\gamma}^+$ 

```

MSR3-fast: Results

- The number of fixed effects p and random effects q is 20.
- $\beta = \gamma = \frac{1}{2}[1, 2, 3, \dots, 10, 0, \dots, 0]$
- 9 groups with sizes [10, 15, 4, 8, 3, 5, 18, 9, 6]
- $X_i \sim \mathcal{N}(0, I)^p$, $Z_i = X_i$, $\varepsilon_i \sim \mathcal{N}(0, 0.3^2 I)$
- Each experiment is repeated 100 times.
- Grid-search for $\eta \in [10^{-4}, 10^2]$, golden search for $\lambda \in [0, 10^5]$
- Final model is chosen to maximize BIC



- + *MSR3-relaxation improves feature selection performance of the original likelihood.*
- + *MSR3-fast optimization accelerates the compute time by $\sim 10^2$.*
- *Initialization of η is problem-specific*

Comparison to Other Libraries

Algorithm	MSR3-Fast (ℓ_1)	glmmLasso ⁷⁸	lmmLasso ⁹¹⁰	PGD (ℓ_1)
Accuracy, %	88	48	66	73
FE Accuracy, %	86	52	47	56
RE Accuracy, %	91	45	84	91
Time, sec	0.19	1.37	11.51	38.39
Iterations, num	34	50	-	7693

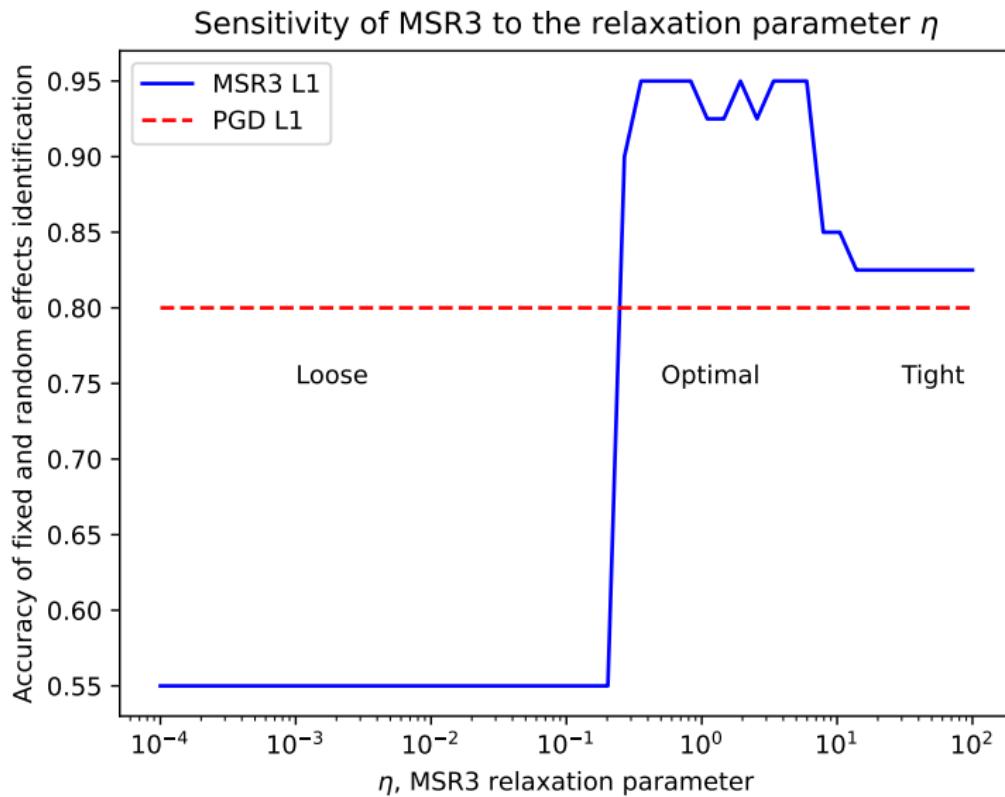
⁷<https://rdrr.io/cran/glmmLasso/man/glmmLasso.html>

⁸Groll and Tutz, "Variable selection for generalized linear mixed models by L 1-penalized estimation".

⁹<https://rdrr.io/cran/lmmlasso/>

¹⁰Schelldorfer, Bühlmann, and DE GEER, "Estimation for high-dimensional linear mixed-effects models using L1-penalization".

Choice of η



ℓ_0 -based Covariate Selection for Bullying Study from GBD

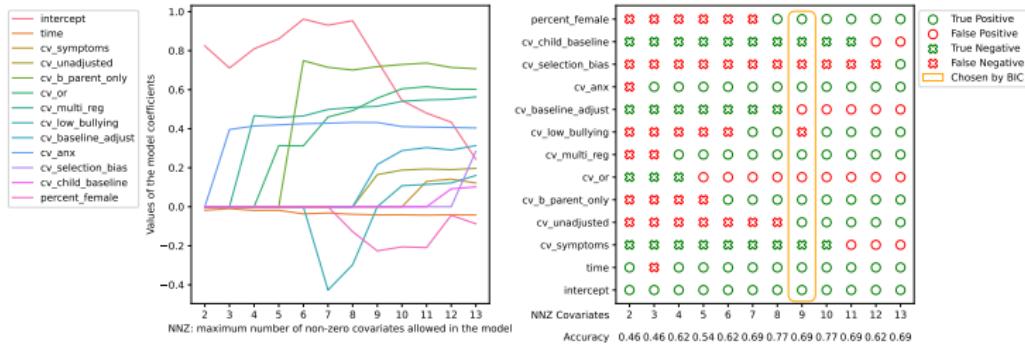


Figure: Fixed and random covariate selection for Bullying dataset¹¹. The model selected 9 covariates, 7 of which were historically significant, and did not select 4 covariates, 1 of which was historically significant.

¹¹ Institute for Health Metrics and Evaluation (IHME). Bullying Victimization Relative Risk Bundle GBD 2020. Seattle, United States of America (USA), 2021.

Software

The screenshot shows the PySR3 documentation website. The left sidebar has a blue header with the PySR3 logo and a search bar. Below it, under 'GETTING STARTED', are links for Quickstart, Installation, Requirements, Usage, and Models Overview. Under 'DEVELOPERS', there are links for Community Guidelines and Modules. The main content area has a dark header with the title 'Quickstart with pysr3'. It includes badges for license (GNU GPLv3), pypi (v0.3.5), build (passing), docs (here), codecov (92%), code quality (A), and JOSS (10.21105/joss.05155). The main text says 'SR3 is a relaxation method designed for accurate feature selection. It currently supports:' followed by a bulleted list of 'Linear Models (L0, LASSO, A-LASSO, CAD, SCAD)' and 'Linear Mixed-Effect Models (L0, LASSO, A-LASSO, CAD, SCAD)'. Below this is a section titled 'Installation' with the note 'pysr3 can be installed via'.

The code is available on GitHub: <https://github.com/aksholokhov/pysr3>¹²

- ▶ All estimators are fully compatible to scikit-learn library.
- ▶ Implements SR3 for linear, generalized-linear, and linear mixed-effect models.
- ▶ Has tutorials, tests, and documentation.

¹² Aleksei Sholokhov, Peng Zheng, and Aleksandr Aravkin. "pysr3: A Python Package for Sparse Relaxed Regularized Regression". In: Journal of Open Source Software 8.84 (2023), p. 5155. < > < > |

Physics-Informed Neural ODE (PINODE): Embedding Physics into Models using Collocation Points

Modeling of Physical Systems

¹³Wikipedia: Navier-Stokes

¹⁴tibco.com

¹⁵COMSOL Simulations

Modeling of Physical Systems

First-Principle Models

- ▶ Require extensive knowledge of the phenomenon
- ▶ Require a lot of compute for simulating large-scale phenomena

$$\begin{cases} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) = \mathbf{f} & \text{in } \Omega \times (0, T) \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times (0, T) \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma_D \times (0, T) \\ \boldsymbol{\sigma}(\mathbf{u}, p) \hat{\mathbf{n}} = \mathbf{h} & \text{on } \Gamma_N \times (0, T) \\ \mathbf{u}(0) = \mathbf{u}_0 & \text{in } \Omega \times \{0\} \end{cases}$$

¹³Wikipedia: Navier-Stokes

¹⁴tibco.com

¹⁵COMSOL Simulations

Modeling of Physical Systems

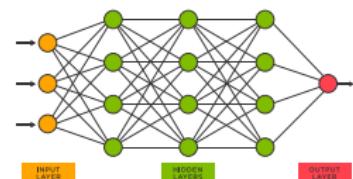
First-Principle Models

- ▶ Require extensive knowledge of the phenomenon
- ▶ Require a lot of compute for simulating large-scale phenomena

$$\begin{cases} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) = \mathbf{f} & \text{in } \Omega \times (0, T) \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times (0, T) \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma_D \times (0, T) \\ \boldsymbol{\sigma}(\mathbf{u}, p) \hat{\mathbf{n}} = \mathbf{h} & \text{on } \Gamma_N \times (0, T) \\ \mathbf{u}(0) = \mathbf{u}_0 & \text{in } \Omega \times \{0\} \end{cases}$$

Data-Driven Models

- ▶ Require a lot of data
- ▶ Often struggle to extrapolate



¹³Wikipedia: Navier-Stokes

¹⁴tibco.com

¹⁵COMSOL Simulations

Modeling of Physical Systems

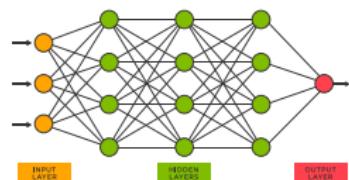
First-Principle Models

- ▶ Require extensive knowledge of the phenomenon
- ▶ Require a lot of compute for simulating large-scale phenomena

$$\begin{cases} \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) = \mathbf{f} & \text{in } \Omega \times (0, T) \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times (0, T) \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma_D \times (0, T) \\ \boldsymbol{\sigma}(\mathbf{u}, p) \hat{\mathbf{n}} = \mathbf{h} & \text{on } \Gamma_N \times (0, T) \\ \mathbf{u}(0) = \mathbf{u}_0 & \text{in } \Omega \times \{0\} \end{cases}$$

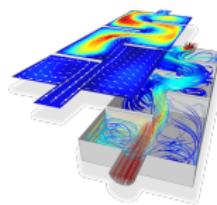
Data-Driven Models

- ▶ Require a lot of data
- ▶ Often struggle to extrapolate



Hybrid Models

- ▶ Incorporate elements of both approaches
- ▶ Supplement data with knowledge or priors



Pictures sources: ¹³, ¹⁴, ¹⁵,

¹³ Wikipedia: Navier-Stokes

¹⁴ tibco.com

¹⁵ COMSOL Simulations

Incorporating Knowledge of Physics into Neural Networks

¹⁶Geiger and Smidt, “e3nn: Euclidean neural networks”.

¹⁷Finzi et al., “Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data”.

¹⁸Chidester, Do, and Ma, Rotation Equivariance and Invariance in Convolutional Neural Networks.

¹⁹Champion et al., “Data-driven discovery of coordinates and governing equations”.

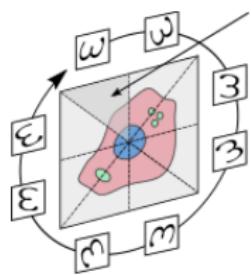
²⁰Schmidt and Lipson, “Distilling free-form natural laws from experimental data”.

²¹Raissi, Perdikaris, and Karniadakis, “Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations”.

²²Rackauckas et al., “Universal differential equations for scientific machine learning”.

Incorporating Knowledge of Physics into Neural Networks

1. **Symmetry Based:** incorporate symmetries and conservation laws as hard constraints into a network: E3NNs¹⁶, LieConv¹⁷, RiCNN¹⁸



¹⁶Geiger and Smidt, "e3nn: Euclidean neural networks".

¹⁷Finzi et al., "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data".

¹⁸Chidester, Do, and Ma, Rotation Equivariance and Invariance in Convolutional Neural Networks.

¹⁹Champion et al., "Data-driven discovery of coordinates and governing equations".

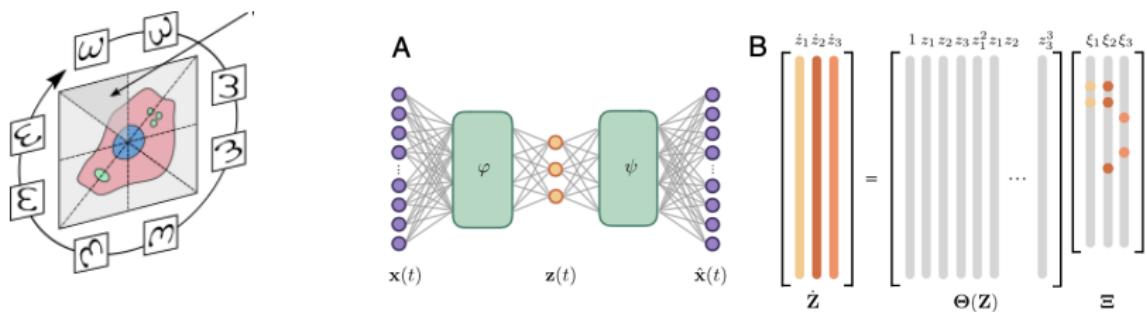
²⁰Schmidt and Lipson, "Distilling free-form natural laws from experimental data".

²¹Raissi, Perdikaris, and Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations".

²²Rackauckas et al., "Universal differential equations for scientific machine learning".

Incorporating Knowledge of Physics into Neural Networks

1. **Symmetry Based:** incorporate symmetries and conservation laws as hard constraints into a network: E3NNs¹⁶, LieConv¹⁷, RiCNN¹⁸
2. **Model Discovery Based:** use a library of terms to discover simple equations: SINDy¹⁹, Genetic Programming²⁰



¹⁶Geiger and Smidt, "e3nn: Euclidean neural networks".

¹⁷Finzi et al., "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data".

¹⁸Chidester, Do, and Ma, Rotation Equivariance and Invariance in Convolutional Neural Networks.

¹⁹Champion et al., "Data-driven discovery of coordinates and governing equations".

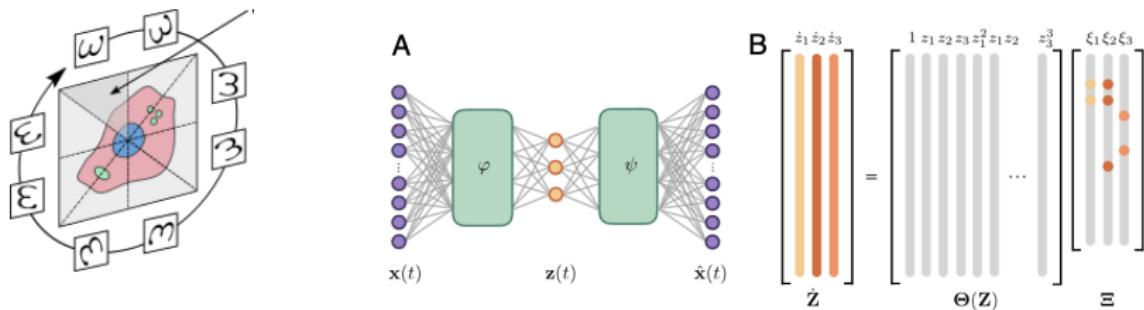
²⁰Schmidt and Lipson, "Distilling free-form natural laws from experimental data".

²¹Raissi, Perdikaris, and Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations".

²²Rackauckas et al., "Universal differential equations for scientific machine learning".

Incorporating Knowledge of Physics into Neural Networks

1. **Symmetry Based:** incorporate symmetries and conservation laws as hard constraints into a network: E3NNs¹⁶, LieConv¹⁷, RiCNN¹⁸
2. **Model Discovery Based:** use a library of terms to discover simple equations: SINDy¹⁹, Genetic Programming²⁰
3. **Equation Based:** incorporate first-principle models to aid training of networks: PINNs²¹, UDEs²², PINODE



¹⁶Geiger and Smidt, "e3nn: Euclidean neural networks".

¹⁷Finzi et al., "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data".

¹⁸Chidester, Do, and Ma, Rotation Equivariance and Invariance in Convolutional Neural Networks.

¹⁹Champion et al., "Data-driven discovery of coordinates and governing equations".

²⁰Schmidt and Lipson, "Distilling free-form natural laws from experimental data".

²¹Raissi, Perdikaris, and Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations".

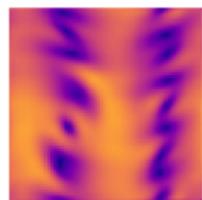
²²Rackauckas et al., "Universal differential equations for scientific machine learning".

Reduced-Order Models (ROMs)

$$x \in \mathbb{R}^n$$

$$\frac{dx}{dt} = f(x)$$

$$x_0$$



Reduced-Order Models (ROMs)

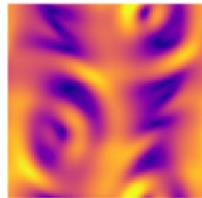
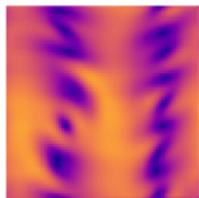
$$x \in \mathbb{R}^n$$

$$\frac{dx}{dt} = f(x)$$

$$x_T = x_0 + \int_0^T f(x) dt$$

x_0

x_T

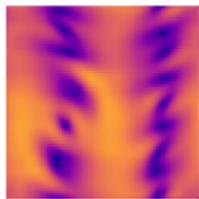


Reduced-Order Models (ROMs)

$$x \in \mathbb{R}^n$$

$$\frac{dx}{dt} = f(x)$$

$$x_0$$

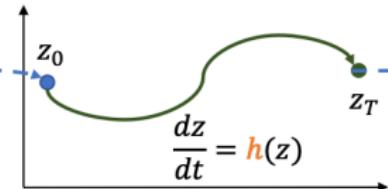


$$z \in \mathbb{R}^m$$

$$m \ll n$$

$$\varphi(x)$$

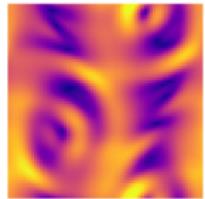
Encoder



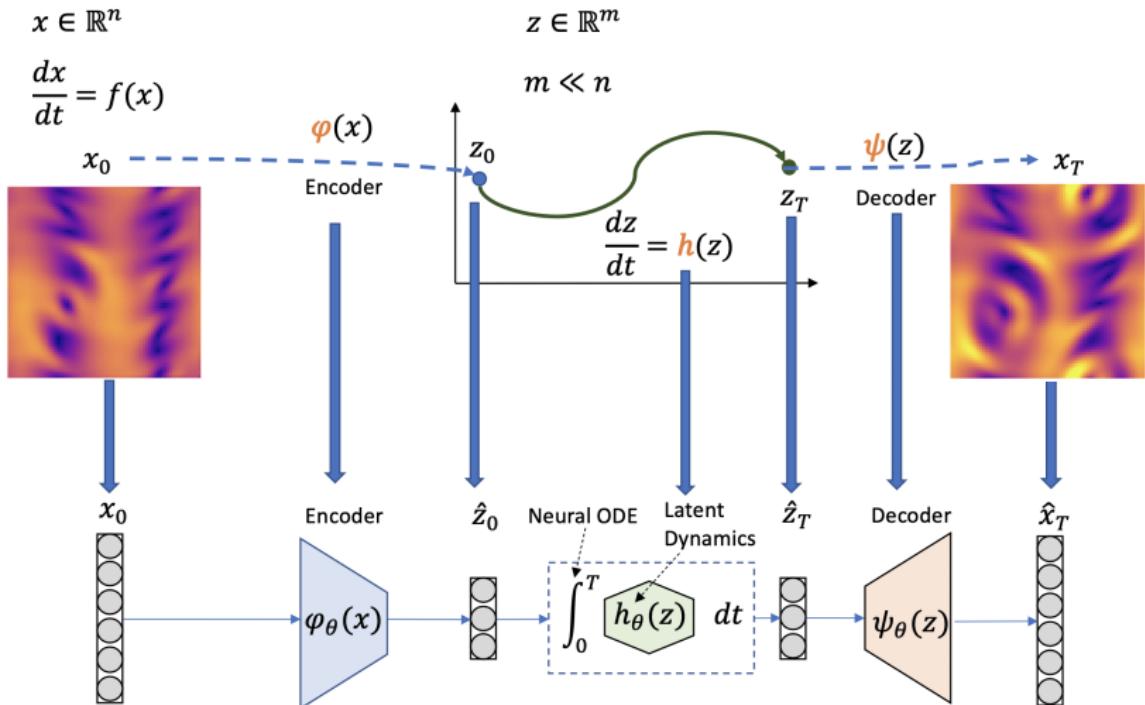
$$\psi(z)$$

Decoder

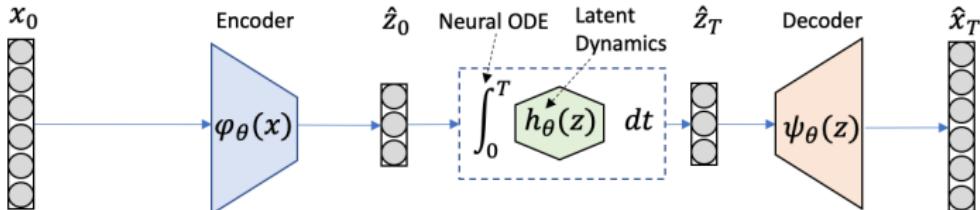
$$x_T$$



Reduced-Order Models (ROMs)



Reduced-Order Models: Data-Driven Loss



Notation:

- ▶ k trajectories $x_i(t_j)$, p timesteps each ($i = 1, \dots, k$, $j = 1, \dots, p$)
- ▶ σ – observation noise variance
- ▶ ω_1, ω_2 – weights (hyper-parameters)

$$\mathcal{L}_\theta^{data} = \frac{1}{2\sigma^2} \sum_{i=1}^k \left[\underbrace{\frac{\omega_1}{p} \sum_{j=1}^p \|x_i(t_j) - \psi_\theta(\phi_\theta(x_i(t_j)))\|^2}_{\text{reconstruction loss}} + \right] \quad (12)$$

$$+ \frac{\omega_2}{p} \sum_{j=1}^p \underbrace{\left\| \psi_\theta \left(\phi_\theta(x_i(t_1)) + \int_{t_1}^{t_j} h(z(t)) dt \right) - x_i(t_j) \right\|^2}_{\text{prediction loss}} \quad (13)$$

Physics-Informed Loss

Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx} \frac{dx}{dt} = \nabla \varphi(x)^T f(x)$$

Physics-Informed Loss

Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx} \frac{dx}{dt} = \nabla \varphi(x)^T f(x) \quad \frac{dz}{dt} = h(\varphi(x))$$

Physics-Informed Loss

Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx} \frac{dx}{dt} = \nabla \varphi(x)^T f(x) \quad \frac{dz}{dt} = h(\varphi(x))$$
$$\mathcal{L}_\theta^{physics} = \sum_{i=1}^N \omega_3 \|\nabla \varphi_\theta(\tilde{x}_i)^T f(\tilde{x}_i) - h_\theta(\varphi_\theta(\tilde{x}_i))\|_2^2 + \omega_4 \|\tilde{x}_i - \psi_\theta(\varphi_\theta(\tilde{x}_i))\|_2^2$$

Physics-Informed Loss

Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx} \frac{dx}{dt} = \nabla \varphi(x)^T f(x) \quad \frac{dz}{dt} = h(\varphi(x))$$
$$\mathcal{L}_{\theta}^{physics} = \sum_{i=1}^N \omega_3 \|\nabla \varphi_{\theta}(\tilde{x}_i)^T f(\tilde{x}_i) - h_{\theta}(\varphi_{\theta}(\tilde{x}_i))\|_2^2 + \omega_4 \|\tilde{x}_i - \psi_{\theta}(\varphi_{\theta}(\tilde{x}_i))\|_2^2$$

The diagram illustrates the derivation of the physics-informed loss function. It shows two intermediate equations at the top: $\frac{dz}{dt} = \frac{dz}{dx} \frac{dx}{dt} = \nabla \varphi(x)^T f(x)$ and $\frac{dz}{dt} = h(\varphi(x))$. Below these, the final loss function is given as $\mathcal{L}_{\theta}^{physics} = \sum_{i=1}^N \omega_3 \|\nabla \varphi_{\theta}(\tilde{x}_i)^T f(\tilde{x}_i) - h_{\theta}(\varphi_{\theta}(\tilde{x}_i))\|_2^2 + \omega_4 \|\tilde{x}_i - \psi_{\theta}(\varphi_{\theta}(\tilde{x}_i))\|_2^2$. Three blue arrows point from these intermediate equations to specific terms in the loss function: arrow 1 points to the first term involving the gradient, arrow 2 points to the second term involving the function h , and arrow 3 points to the third term involving the function ψ .

Physics-Informed Loss

Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx} \frac{dx}{dt} = \nabla \varphi(x)^T f(x) \quad \frac{dz}{dt} = h(\varphi(x))$$
$$\mathcal{L}_{\theta}^{physics} = \sum_{i=1}^N \omega_3 \|\nabla \varphi_{\theta}(\tilde{x}_i)^T f(\tilde{x}_i) - h_{\theta}(\varphi_{\theta}(\tilde{x}_i))\|_2^2 + \omega_4 \|\tilde{x}_i - \psi_{\theta}(\varphi_{\theta}(\tilde{x}_i))\|_2^2$$


1

Let $\dot{x} = f(x)$ be a discretization of $-u_t = u_{yy} + u_{yyyy} + \frac{1}{2}u_y^2$

Physics-Informed Loss

Using chain rule:

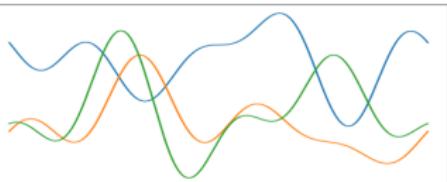
$$\frac{dz}{dt} = \frac{dz}{dx} \frac{dx}{dt} = \nabla \varphi(x)^T f(x) \quad \frac{dz}{dt} = h(\varphi(x))$$
$$\mathcal{L}_{\theta}^{physics} = \sum_{i=1}^N \omega_3 \|\nabla \varphi_{\theta}(\tilde{x}_i)^T f(\tilde{x}_i) - h_{\theta}(\varphi_{\theta}(\tilde{x}_i))\|_2^2 + \omega_4 \|\tilde{x}_i - \psi_{\theta}(\varphi_{\theta}(\tilde{x}_i))\|_2^2$$

1

Let $\dot{x} = f(x)$ be a discretization of $-u_t = u_{yy} + u_{yyyy} + \frac{1}{2}u_y^2$

Then the **collocations** could be:

$$u(y) = \sum_{w=1}^{30} a(w) \sin(2\pi y) + b(w) \cos(2\pi y)$$



- Diverse
- Cheap
- Representative

Collocations: a set of pairs $(\tilde{x}_i, \dot{\tilde{x}}_i)$, where \tilde{x}_i are s.t. $\dot{\tilde{x}}_i$ are cheap to evaluate.

Physics-Informed Loss

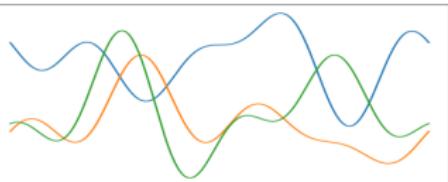
Using chain rule:

$$\frac{dz}{dt} = \frac{dz}{dx} \frac{dx}{dt} = \nabla \varphi(x)^T f(x) \quad \frac{dz}{dt} = h(\varphi(x))$$
$$\mathcal{L}_{\theta}^{physics} = \sum_{i=1}^N \omega_3 \|\nabla \varphi_{\theta}(\tilde{x}_i)^T f(\tilde{x}_i) - h_{\theta}(\varphi_{\theta}(\tilde{x}_i))\|_2^2 + \omega_4 \|\tilde{x}_i - \psi_{\theta}(\varphi_{\theta}(\tilde{x}_i))\|_2^2$$

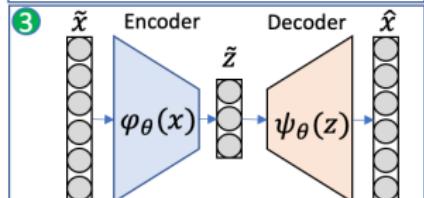
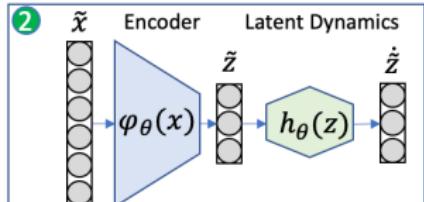
1 Let $\dot{x} = f(x)$ be a discretization of $-u_t = u_{yy} + u_{yyyy} + \frac{1}{2}u_y^2$

Then the **collocations** could be:

$$u(y) = \sum_{w=1}^{30} a(w) \sin(2\pi y) + b(w) \cos(2\pi y)$$



- Diverse
- Cheap
- Representative



Collocations: a set of pairs $(\tilde{x}_i, \dot{\tilde{x}}_i)$, where \tilde{x}_i are s.t. $\dot{\tilde{x}}_i$ are cheap to evaluate.

Physics-Informed, Data-Driven, and Hybrid Models

Values of ω_1 , ω_2 , ω_3 , and ω_4 control the type of the ROM:

- ▶ If $\omega_1 = \omega_2 = 0$ then $\mathcal{L}_\theta^{\text{data}} = 0$. Thus, the model is purely Physics-Informed
- ▶ If $\omega_3 = \omega_4 = 0$ then $\mathcal{L}_\theta^{\text{physics}} = 0$. Thus, the model is purely Data-Driven
- ▶ Otherwise the model is Hybrid

Results: Extrapolation to Unknown Regions

Duffing Oscillator on a low-dimensional (2D) manifold:

$$\begin{aligned}\frac{dz_1}{dt} &= z_2 \\ \frac{dz_2}{dt} &= z_1 - z_1^3\end{aligned}\tag{14}$$

Projection to a high-dimensional (128) space:

$$x := \mathcal{A}(z) = Az^3, \quad A \in \mathbb{R}^{128 \times 2}, \quad A_{ij} \sim_{i.i.d.} \mathcal{N}(0, 1)\tag{15}$$

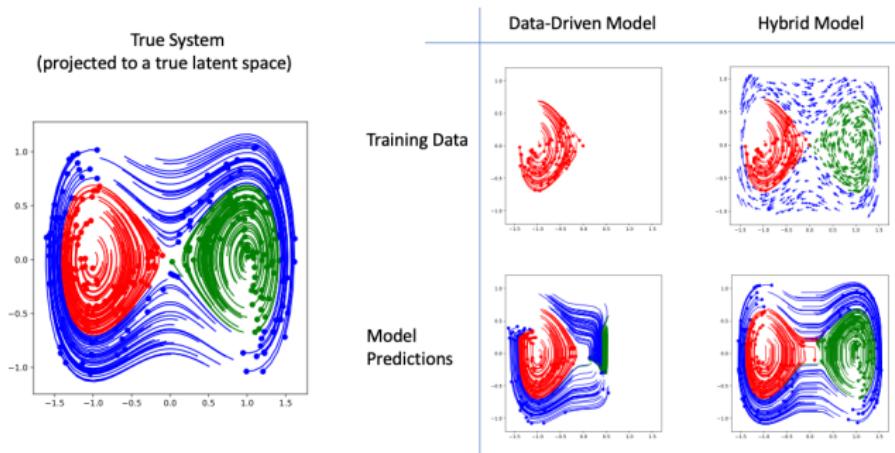
Results: Extrapolation to Unknown Regions

Duffing Oscillator on a low-dimensional (2D) manifold:

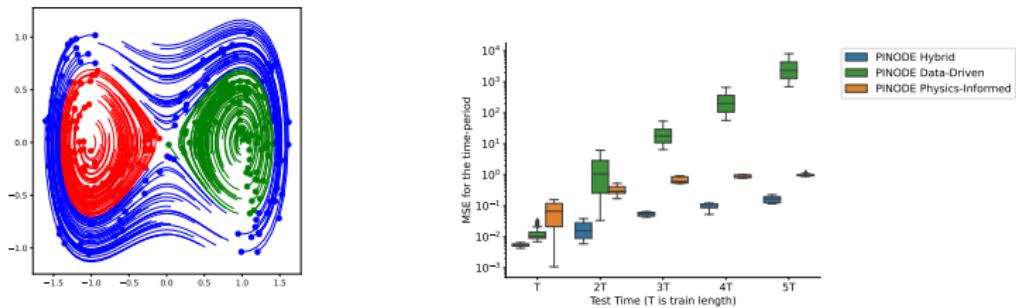
$$\begin{aligned}\frac{dz_1}{dt} &= z_2 \\ \frac{dz_2}{dt} &= z_1 - z_1^3\end{aligned}\tag{14}$$

Projection to a high-dimensional (128) space:

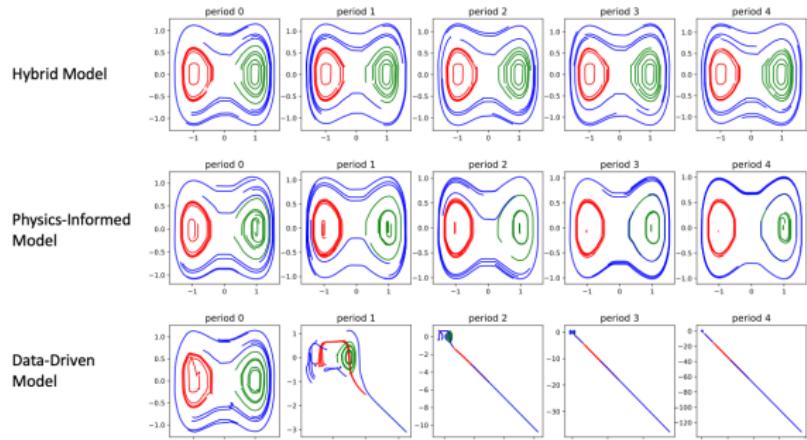
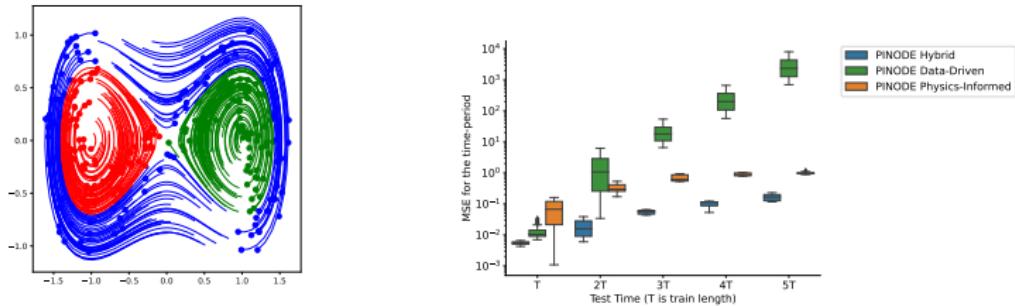
$$x := \mathcal{A}(z) = Az^3, \quad A \in \mathbb{R}^{128 \times 2}, \quad A_{ij} \sim_{i.i.d.} \mathcal{N}(0, 1)\tag{15}$$



Results: Stable Long-Term Predictions



Results: Stable Long-Term Predictions



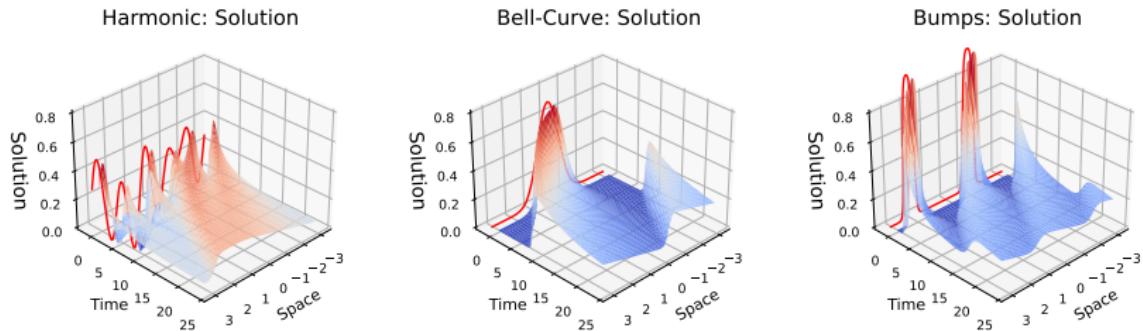
Results: Burgers' Equation

$$\begin{aligned} u_t + uu_x &= \nu u_{xx} \\ u(-\pi, t) &= u(\pi, t), \quad \forall t \in [0, T], \quad \nu = 0.01 \end{aligned} \tag{16}$$

Results: Burgers' Equation

$$u_t + uu_x = \nu u_{xx} \quad (16)$$

$$u(-\pi, t) = u(\pi, t), \quad \forall t \in [0, T], \quad \nu = 0.01$$



ICs and Collocations:

Harmonic:

$$u(x) = \sum_{k=1}^{30} a_k \sin(2\pi x/k)$$

$$+ b_k \cos(2\pi x/k)$$

$$a_k = \alpha_k s_k$$

$$\alpha_k \sim \mathcal{U}(0, 1)$$

$$s_k \sim \mathcal{B}(0.5)$$

Bell-curve:

$$u(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(x-x_0)^2}{2\sigma^2}$$

$$\sigma \sim \mathcal{U}(0, 3)$$

$$x_0 \sim \mathcal{U}(-\pi, \pi)$$

Bumps:

$$u(x) = \frac{a}{1 + \exp^{-k(x-x_0)}}$$

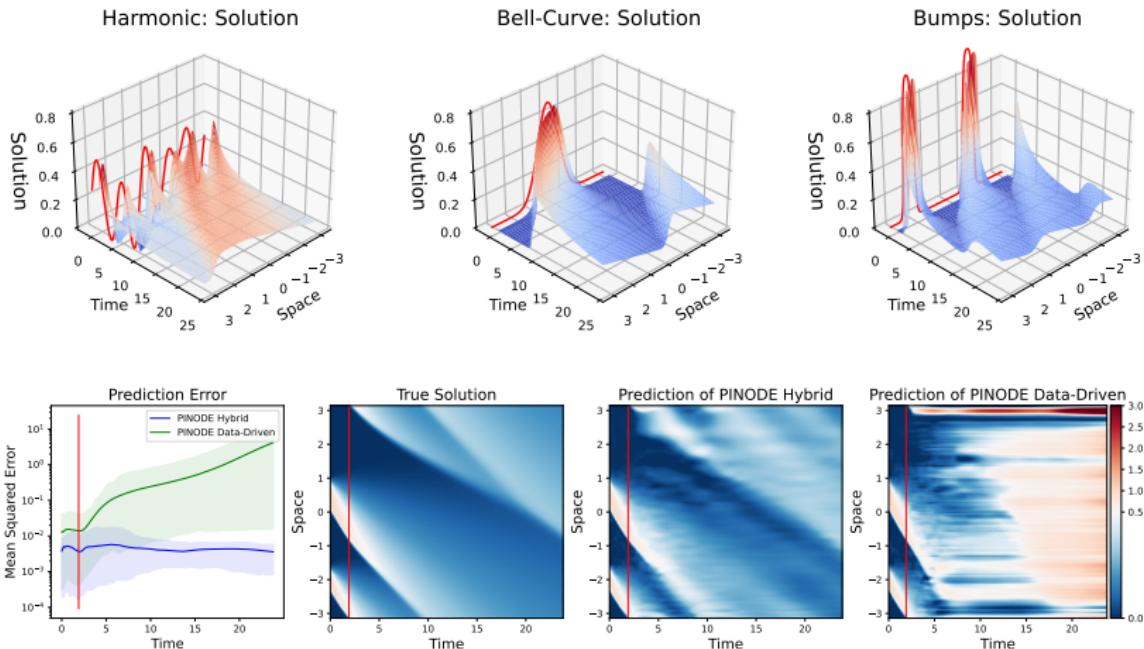
$$- \frac{a}{1 + \exp^{-k(x-x_1)}}$$

$$k \sim \mathcal{U}(5, 20)$$

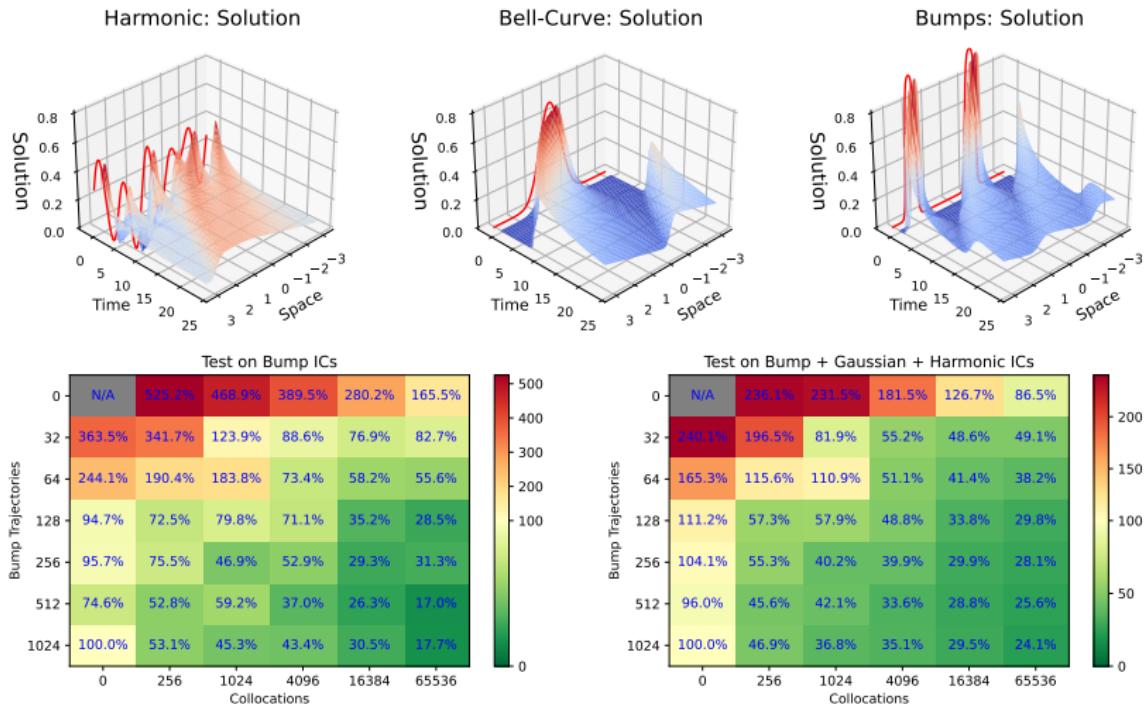
$$x_0, x_1 \sim \mathcal{U}(-\pi, \pi)$$

Results: Burgers' Equation

$$u_t + uu_x = \nu u_{xx} \quad (16)$$
$$u(-\pi, t) = u(\pi, t), \quad \forall t \in [0, T], \quad \nu = 0.01$$



Results: Learning From Collocations



Discussion and Limitations

We showed that:

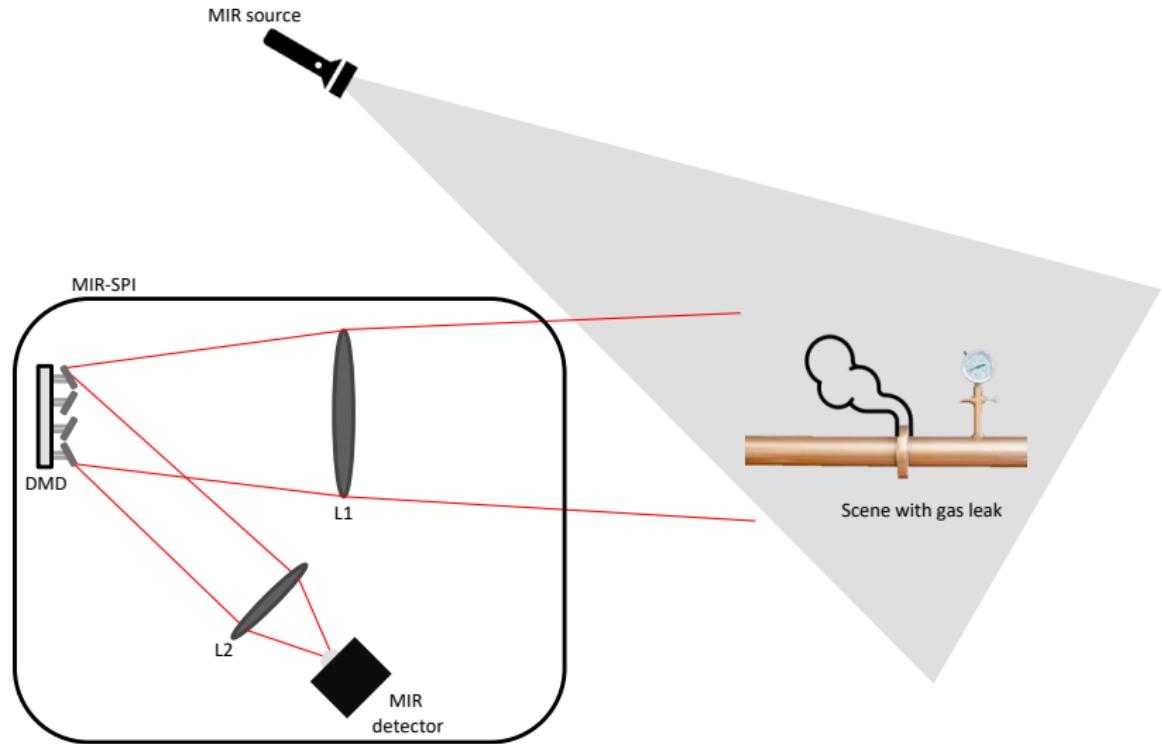
- ▶ Physics-informed loss improves accuracy and forecasting stability of ROMs
- ▶ Collocations can supplement data to improve model's performance for unseen initial conditions.

Limitations:

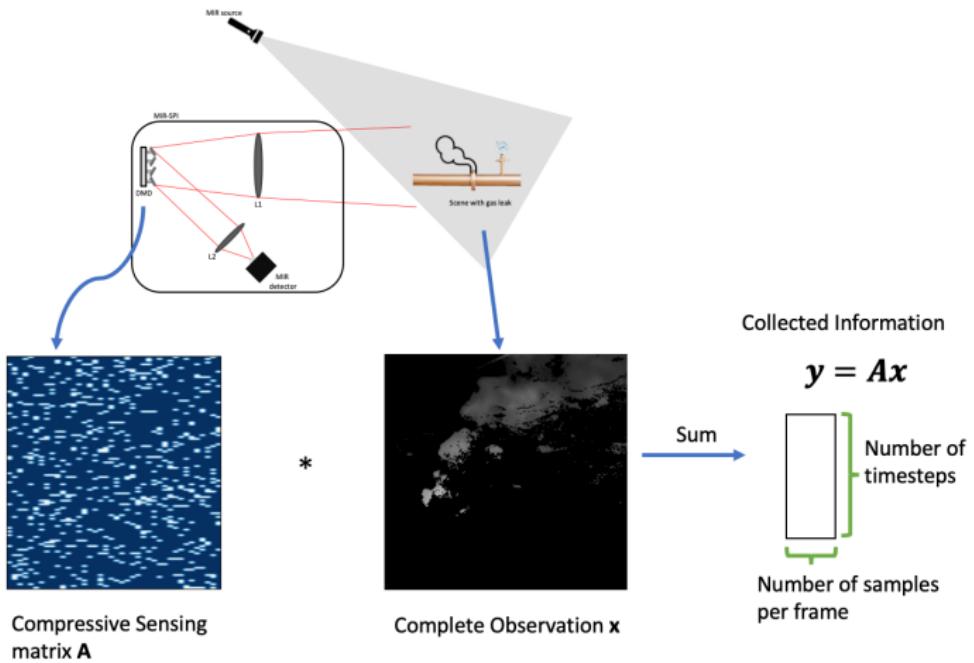
- ▶ Optimal choice of collocations is problem-specific
- ▶ Need a lot of collocations
 - ▶ Seems possible to overcome with smarter sampling techniques

Single pixel imaging of spatio-temporal flows using differentiable latent dynamics

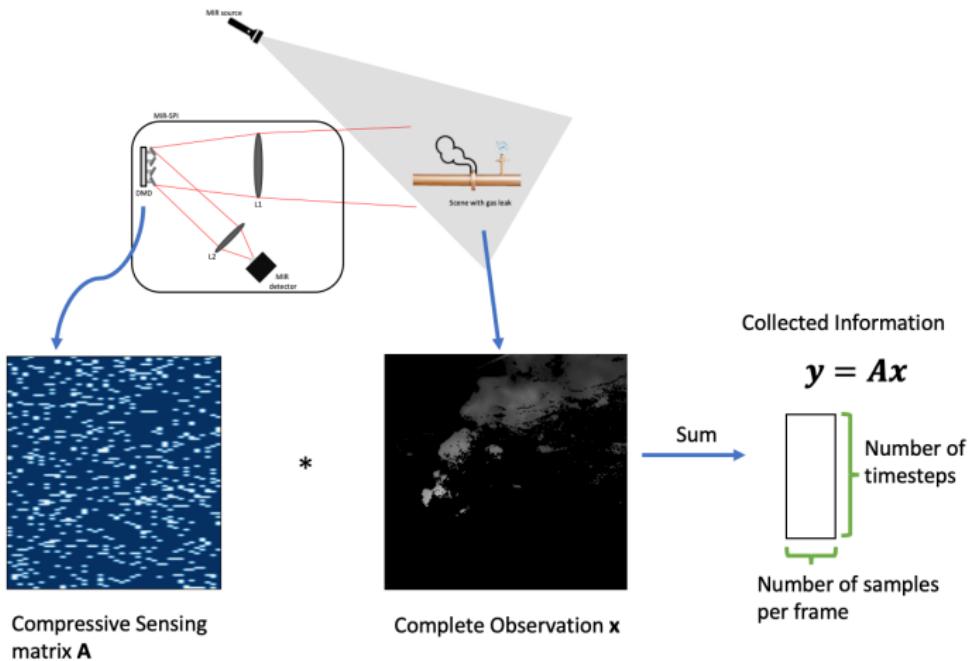
Single-Pixel Imaging



Single-Pixel Imaging



Single-Pixel Imaging

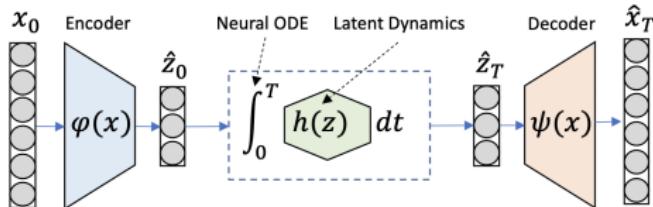


Single-Pixel Image Recovery: find \mathbf{x} using \mathbf{y} and \mathbf{A} .

SPF rate, % = Number of samples per frame / number of pixels per frame * 100

Compressive Sensing with Reduced-Order Models

Offline Step: Train a Data-Driven Reduced-Order Model



Online Step: Reconstruct Complete Observations by Optimizing in Latent Space

$$\mathcal{L}^{recon.}(z) = \|y - A\psi(z)\| + \lambda \left\| z - (z_0 + \int_0^T h(z) dz) \right\|$$

Latent-space representation of the trajectory "What the data tells us the trajectory should be" "What the model thinks the trajectory should be"

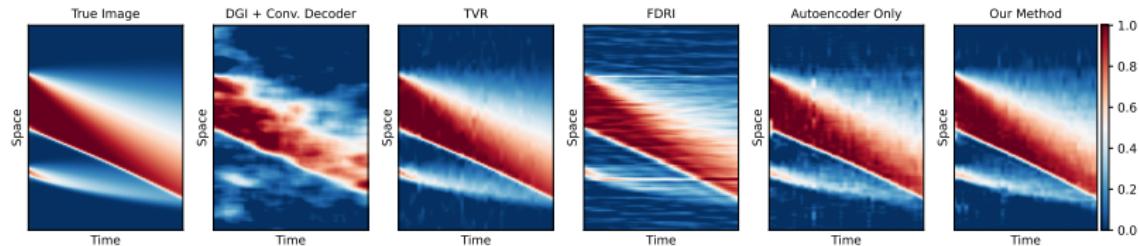
The diagram illustrates the Online Step: Reconstruct Complete Observations by Optimizing in Latent Space. It shows the reconstruction loss function:

$$\mathcal{L}^{recon.}(z) = \|y - A\psi(z)\| + \lambda \left\| z - (z_0 + \int_0^T h(z) dz) \right\|$$

The first term, $\|y - A\psi(z)\|$, represents the Reconstruction Loss. The second term, $\lambda \left\| z - (z_0 + \int_0^T h(z) dz) \right\|$, represents the Compressive Sensing Loss. The term $z_0 + \int_0^T h(z) dz$ is enclosed in brackets with a brace below it, labeled "What the data tells us the trajectory should be". The term z is enclosed in brackets with a brace below it, labeled "What the model thinks the trajectory should be".

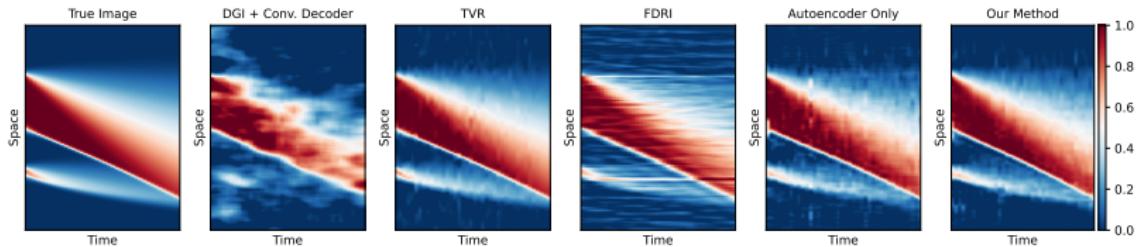
Results: Burger's Equation

Recovery with SPF=25%:

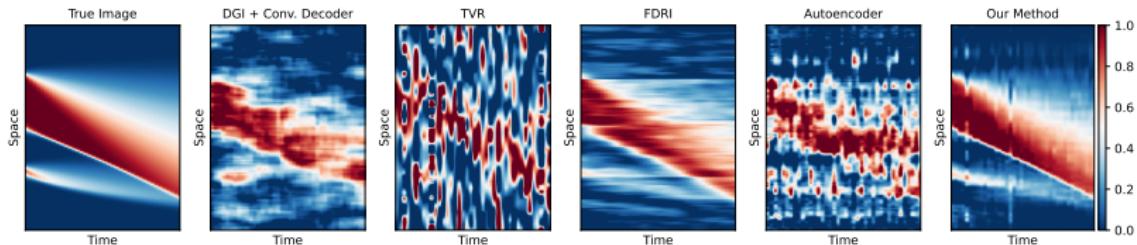


Results: Burger's Equation

Recovery with SPF=25%:

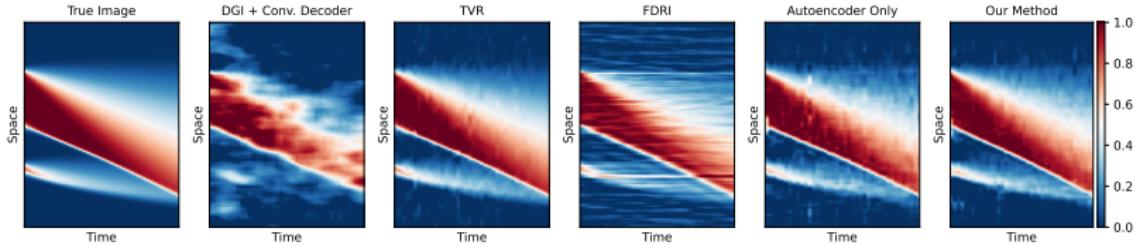


Recovery with SPF=6.25%:

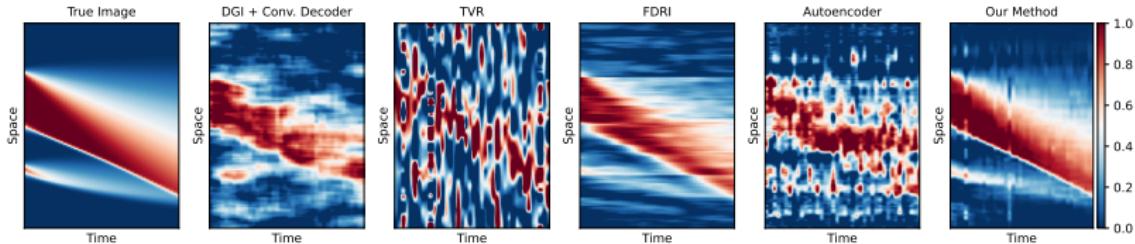


Results: Burger's Equation

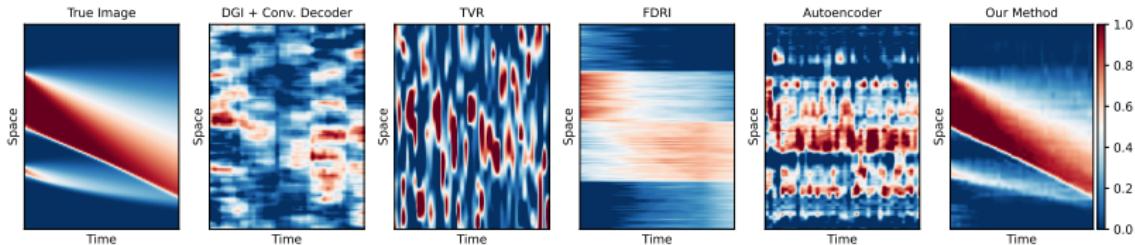
Recovery with SPF=25%:



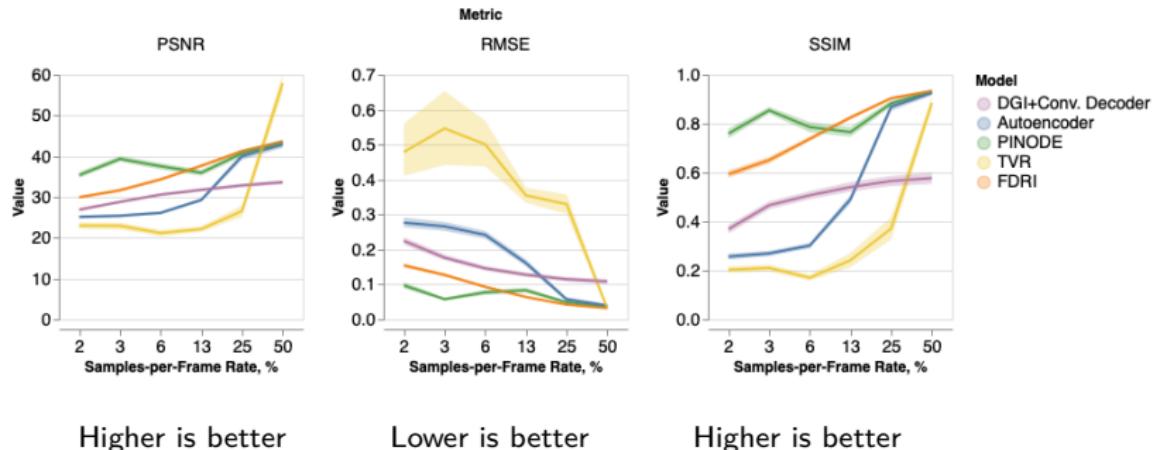
Recovery with SPF=6.25%:



Recovery with SPF=1.5%:



Results: Burger's Equation



Higher is better

Lower is better

Higher is better

Metrics:

- ▶ **PSNR:** Peak Signal-to-Noise Ratio
- ▶ **RMSE:** Residual Mean Square Error
- ▶ **SSIM:** Structural Similarity Index Measure

Results: Kolmogorov Flow

A 2D velocity field $\mathbf{u}(x, y, t)$:

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (17)$$

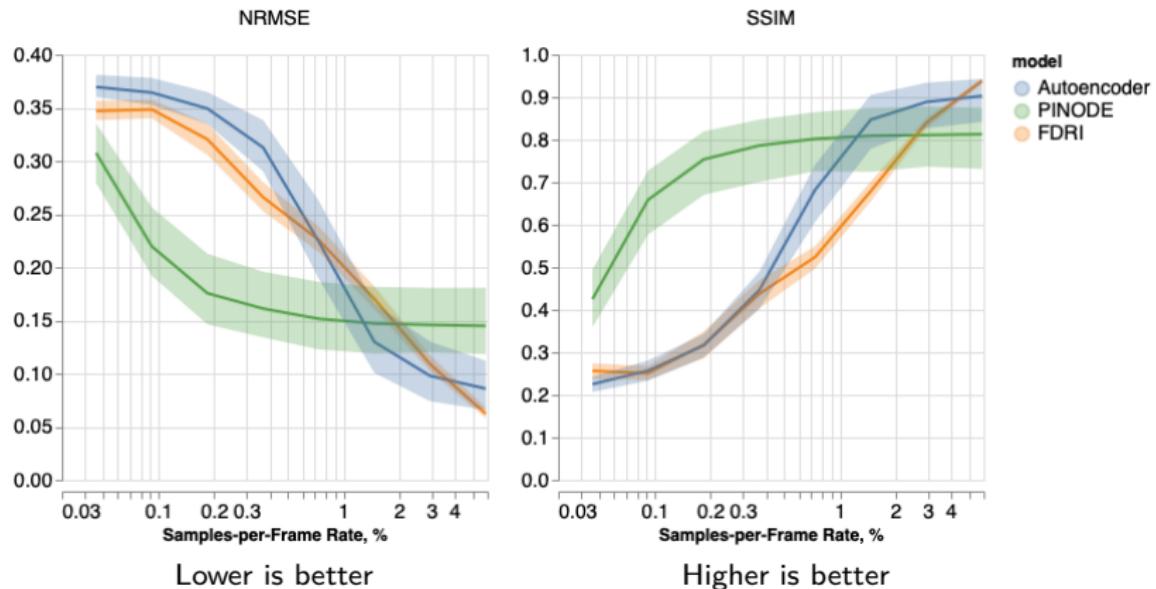
$$\nabla \cdot \mathbf{u} = 0 \quad (18)$$

where

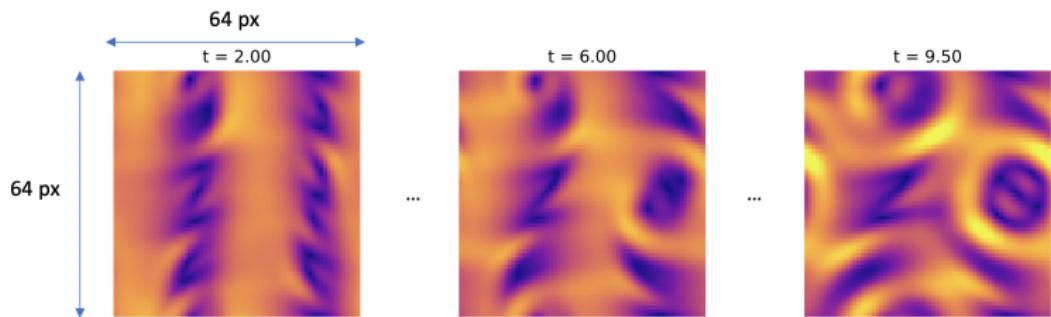
- ▶ p is a 2D pressure field
- ▶ $\mathbf{f} = \alpha \sin(ky)x$ - driving force with amplitude A and wavenumber k ($k = 4$)
- ▶ $\nu = 1/\text{Re}$ is the non-dimensional viscosity. $\text{Re}=40$

Reconstruction with SPF rate = 0.18% (8 SPI samples per a frame of 66×66 pixels):

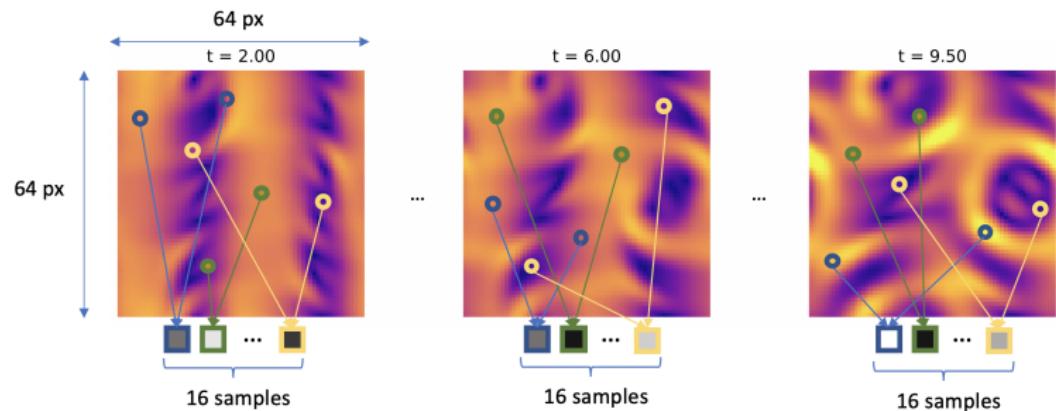
Results: Kolmogorov Flow



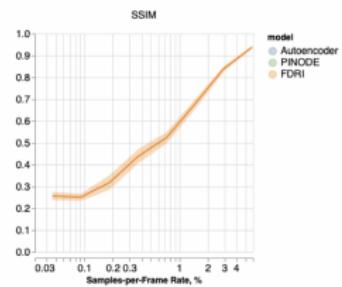
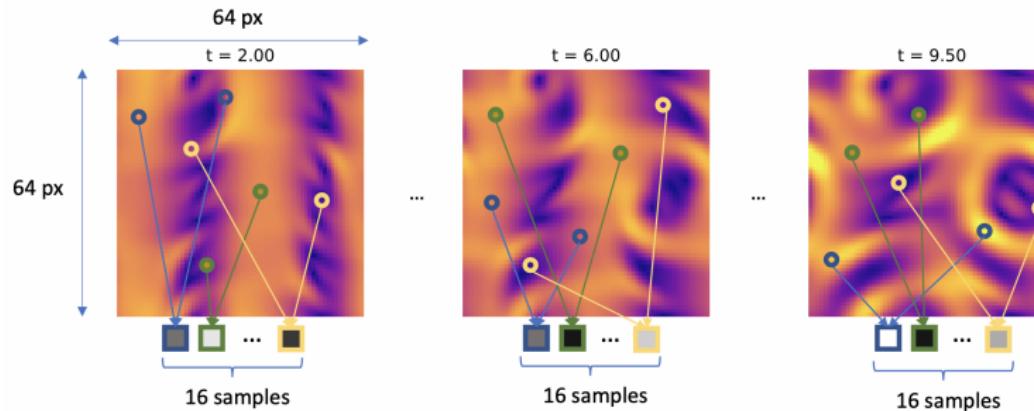
Results: Interpretation



Results: Interpretation



Results: Interpretation



Results: Kolmogorov Flow

A 2D velocity field $\mathbf{u}(x, y, t)$:

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (17)$$

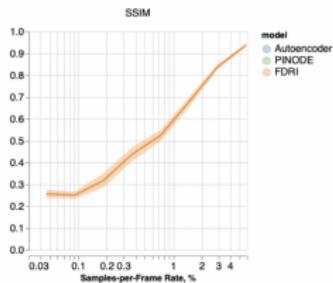
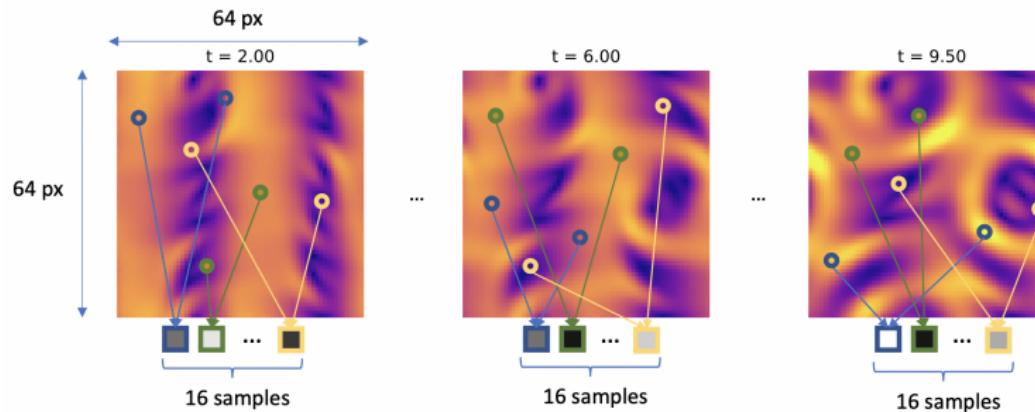
$$\nabla \cdot \mathbf{u} = 0 \quad (18)$$

where

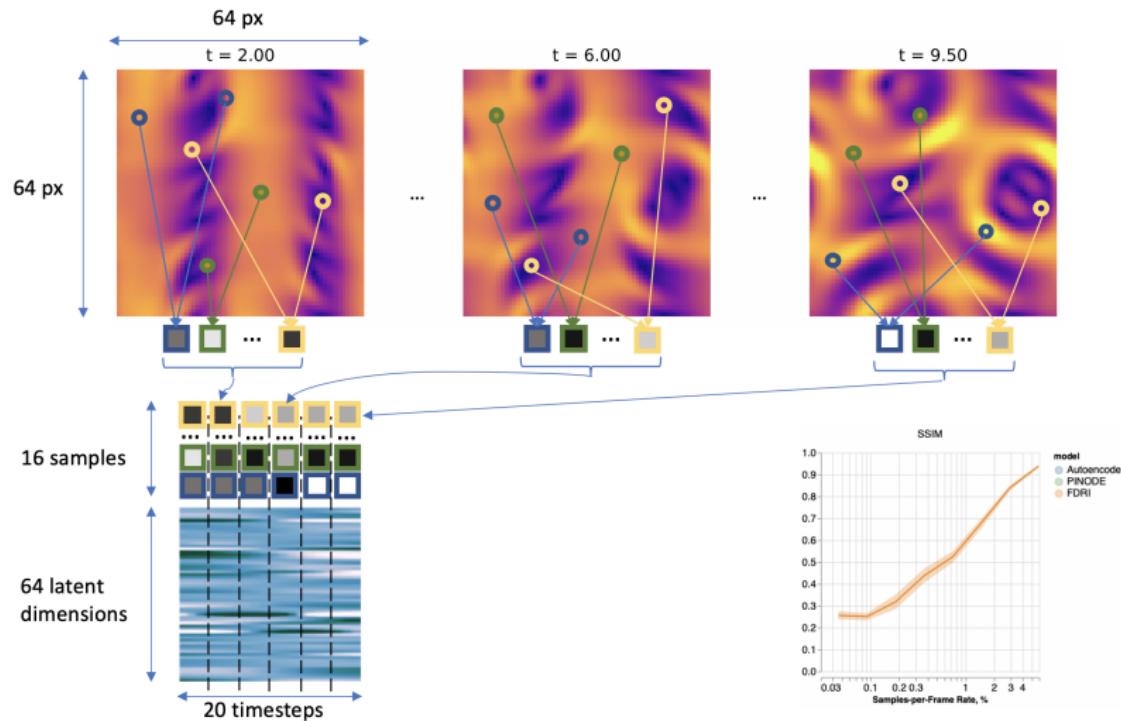
- ▶ p is a 2D pressure field
- ▶ $\mathbf{f} = \alpha \sin(ky)x$ - driving force with amplitude A and wavenumber k ($k = 4$)
- ▶ $\nu = 1/\text{Re}$ is the non-dimensional viscosity. $\text{Re}=40$

Reconstruction with SPF rate = 0.18% (8 SPI samples per a frame of 66×66 pixels):

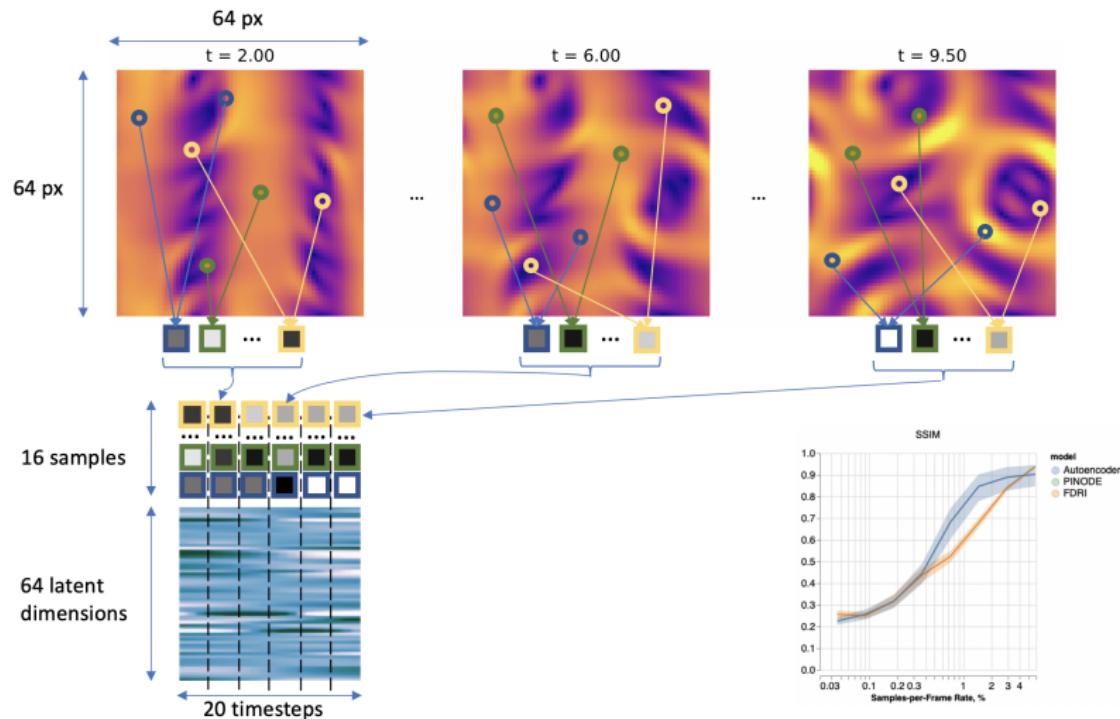
Results: Interpretation



Results: Interpretation



Results: Interpretation



Results: Kolmogorov Flow

A 2D velocity field $\mathbf{u}(x, y, t)$:

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (17)$$

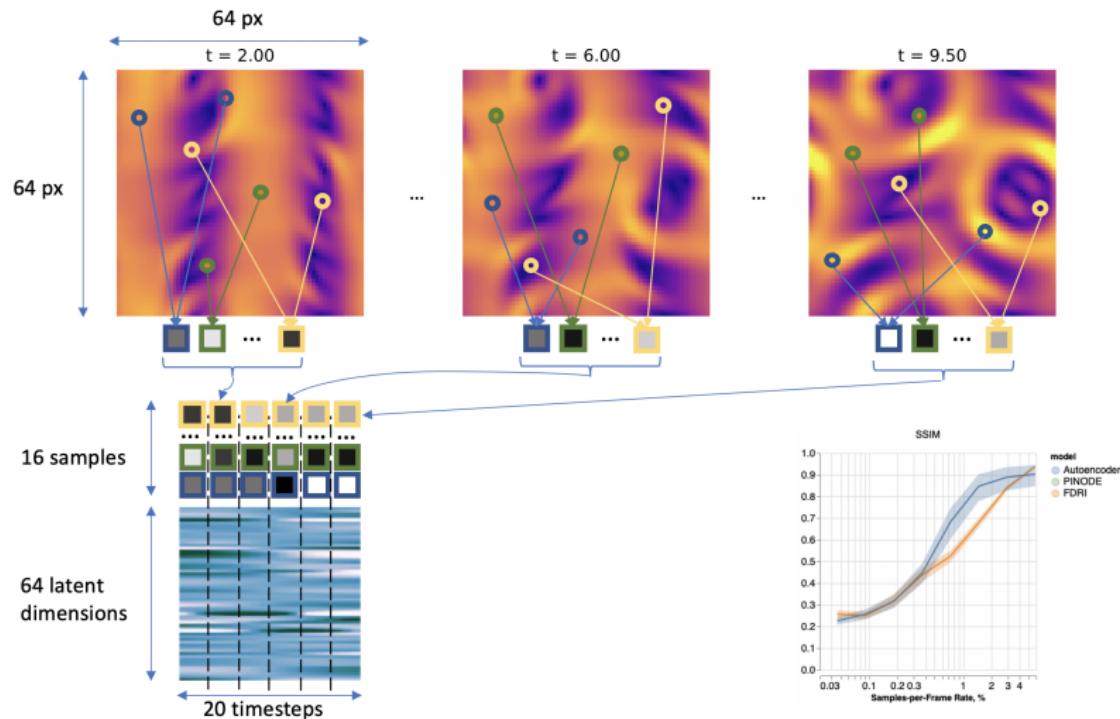
$$\nabla \cdot \mathbf{u} = 0 \quad (18)$$

where

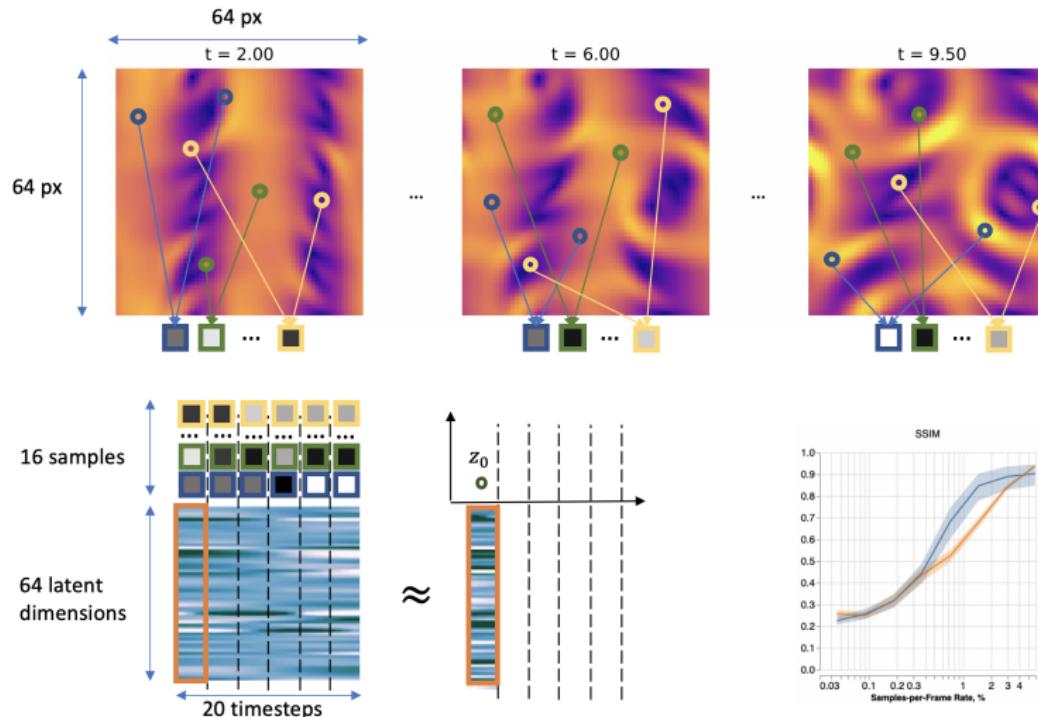
- ▶ p is a 2D pressure field
- ▶ $\mathbf{f} = \alpha \sin(ky)x$ - driving force with amplitude A and wavenumber k ($k = 4$)
- ▶ $\nu = 1/\text{Re}$ is the non-dimensional viscosity. $\text{Re}=40$

Reconstruction with SPF rate = 0.18% (8 SPI samples per a frame of 66×66 pixels):

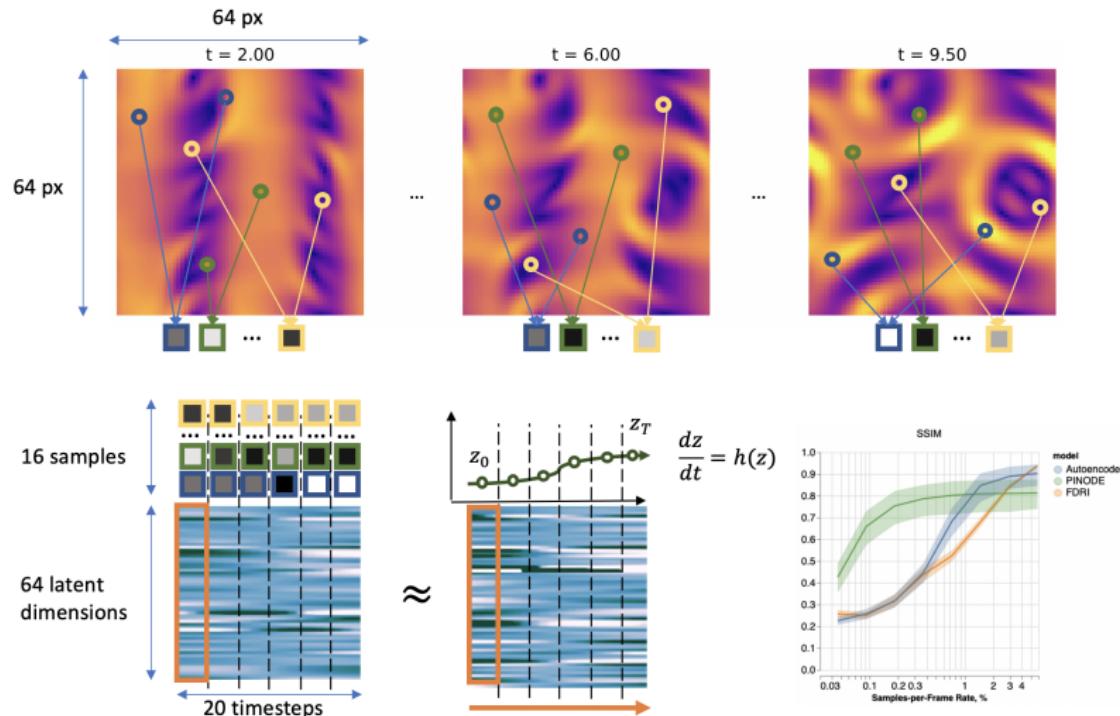
Results: Interpretation



Results: Interpretation



Results: Interpretation



Results: Kolmogorov Flow

A 2D velocity field $\mathbf{u}(x, y, t)$:

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (17)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (18)$$

where

- ▶ p is a 2D pressure field
- ▶ $\mathbf{f} = \alpha \sin(ky)x$ - driving force with amplitude A and wavenumber k ($k = 4$)
- ▶ $\nu = 1/\text{Re}$ is the non-dimensional viscosity. $\text{Re}=40$

Reconstruction with SPF rate = 0.18% (8 SPI samples per a frame of 66×66 pixels):

Results: Gas Monitoring

Gas monitoring problem:

- ▶ 494 240×320 video recordings, 10 steps each
- ▶ 512 samples per frame, SPF rate of 0.67%
- ▶ Background noise

Thank You!



Sparse Relaxed Regularized Regression for Linear Mixed-Effect Models

Together with Aleksandr Aravkin, James Burke, Damian Santomauro, and Peng Zheng

- ▶ "A Relaxation Approach to Feature Selection for Linear Mixed Effects Models"
submitted to Journal of Computational and Graphical Statistics (JCGS)
- ▶ "Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models"
submitted to Journal of Computational Optimization and Applications (COAP)
- ▶ "pysr3: A Python Package for Sparse Relaxed Regularized Regression"
Journal of Open-Source Software (JOSS), 2023, ICCOPT 2022, SIAM OPT 2023

PINODE: Physics-Informed Neural ODEs

Together with Hassan Mansour, Saleh Nabi, and Yuying Liu

- ▶ "Physics-Informed Koopman Network"
SIAM ADS 2023, arXiv:2211.09419
- ▶ "Physics-Informed Neural ODE: Embedding Physics into Models using Collocation Points"
Submitted to Nature Special Issue on Physics-Informed Machine Learning

Single-Pixel Imaging with Reduced-Order Models

Together with J. Nathan Kutz, Steven Brunton, Joshua Rapp, Hassan Mansour, and Saleh Nabi

- ▶ "Single pixel imaging of spatio-temporal flows using differentiable latent dynamics" *in preparation*

References I

-  Aravkin, Aleksandr et al. Analysis of Relaxation Methods for Feature Selection in Mixed Effects Models. 2022. arXiv: 2209.10575 [stat.ME].
-  Beck, Amir. First-Order Methods in Optimization. MOS-SIAM Series on Optimization. SIAM, 2017. ISBN: 9781611974980. DOI: 10.1137/1.9781611974997.
-  Burke, J.V. and A. Engle. "Line Search and Trust-Region Methods for Convex-Composite Optimization". In: arXiv:1806.05218 (2018). DOI: <https://doi.org/10.48550/arXiv.1806.05218>.
-  Champion, Kathleen et al. "Data-driven discovery of coordinates and governing equations". In: Proceedings of the National Academy of Sciences 116.45 (2019), pp. 22445–22451.
-  Chidester, Benjamin, Minh N. Do, and Jian Ma. Rotation Equivariance and Invariance in Convolutional Neural Networks. 2018. arXiv: 1805.12301 [stat.ML].
-  Finzi, Marc et al. "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data". In: 37th International Conference on Machine Learning, ICML 2020 PartF16814 (2020), pp. 3146–3157. arXiv: 2002.12880.
-  Geiger, Mario and Tess Smidt. "e3nn: Euclidean neural networks". In: arXiv preprint arXiv:2207.09453 (2022).
-  Groll, Andreas and Gerhard Tutz. "Variable selection for generalized linear mixed models by L 1-penalized estimation". In: Statistics and Computing 24.2 (2014), pp. 137–154.

References II

-  Rackauckas, Christopher et al. "Universal differential equations for scientific machine learning". In: [arXiv preprint arXiv:2001.04385](#) (2020).
-  Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis. "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations". In: [arXiv preprint arXiv:1711.10561](#) (2017).
-  Schelldorfer, Jürg, Peter Bühlmann, and SARA VAN DE GEER. "Estimation for high-dimensional linear mixed-effects models using l1-penalization". In: [Scandinavian Journal of Statistics](#) 38.2 (2011), pp. 197–214.
-  Schmidt, Michael and Hod Lipson. "Distilling free-form natural laws from experimental data". In: [science](#) 324.5923 (2009), pp. 81–85.
-  Sholokhov, Aleksei, James V. Burke, et al.
A Relaxation Approach to Feature Selection for Linear Mixed Effects Models. [2022. arXiv: 2205.06925 \[stat.ME\]](#).
-  Sholokhov, Aleksei, Peng Zheng, and Aleksandr Aravkin. "pysr3: A Python Package for Sparse Relaxed Regularized Regression". In: [Journal of Open Source Software](#) 8.84 (2023), p. 5155.
-  Zheng, Peng and Aleksandr Aravkin. "Relax-and-split method for nonconvex inverse problems". In: [Inverse Problems](#) 36.9 (2020). ISSN: 13616420. DOI: [10.1088/1361-6420/aba417](https://doi.org/10.1088/1361-6420/aba417).

Appendix: $\mathcal{MSR}3$