Discovering PDEs from Multiple Experiments

Georges Tod[1], **Gert-Jan Both**[1], **Remy Kusters**[1,2]
[1]Université de Paris, INSERM U1284, CRI, 75004 Paris, France
[2]IBM Research Paris-Saclay, 91400 Orsay, France
{firstname.lastname}@cri-paris.org

Abstract

Automated model discovery of partial differential equations (PDEs) usually considers a single experiment or dataset to infer the underlying governing equations. In practice, experiments have inherent natural variability in parameters, initial and boundary conditions that cannot be simply averaged out. We introduce a randomised adaptive group Lasso sparsity estimator to promote grouped sparsity and implement it in a deep learning based PDE discovery framework[1]. It allows to create a learning bias that implies the a priori assumption that all experiments can be explained by the same underlying PDE terms with potentially different coefficients. Our experimental results show more generalizable PDEs can be found from multiple highly noisy datasets, by this grouped sparsity promotion rather than simply performing independent model discoveries.

1 Introduction

A classical approach to perform model discovery is by sparse regression and consists in finding ξ such that, $u_t = \Theta \cdot \xi$, where u_t is the time derivative of the field u. Each column of Θ is a candidate term of the underlying PDE, typically a combination of polynomial and spatial derivative functions (e.g. u, u_x, uu_x). Usually, ξ is identified based on a single experiment consisting of n samples of the field u, see [1, 2, 3, 4, 5, 6]. However, in practice observations of the same experiment might lead to some measurement differences. For instance, it might not be possible: (1) to fix identically the initial and/or boundary conditions of several experiments, (2) some of the parameters might not be controllable and will vary. As a result, there will be experimental results which exhibit natural variability. Let us denote q the number of experiments at hand from which we would like to find the underlying partial differential equations. If we perform q individual model discoveries as described earlier, the data across experiments will not be leveraged, loosing an opportunity for learning. To leverage the data across experiments, the q libraries could be stacked $\Theta = [\Theta_1, \Theta_2, \dots, \Theta_q]^T$ and then a single sparse regression could reveal ξ, see [2, 7]. However, this approach is not adapted if the coefficients of the underlying PDEs vary from an experiment to another. In this paper, a more general approach to this is proposed: by promoting sparsity group-wise we leverage the data across experiments and are able to handle varying coefficients across datasets. In [8], a group sequentially thresholded ridge regression was proposed to infer parametric PDEs from a single dataset. In [9] a group Iterative Hard Thresholding algorithm is used to enforce conservation laws and impose symmetries. In [10] a group Lasso is used it to try to infer the law of gravitation from experimental ball drops.

Furthermore, experimental data might contain high noise levels and it is known that pure sparse regression based model discovery cannot handle noise levels above 5 %, see [2, 4]. For such cases, deep learning model discovery frameworks, that combine deep neural networks, automatic differentiation with sparse regression excel, see [5, 7, 6].

[1] Data, code and results shared on: https://github.com/georgestod/multi_deepmod.

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

Contributions (1) we introduce a randomised adaptive group Lasso to promote grouped sparsity to discover the common underlying PDE from multiple experiments. The group adaptive Lasso [11] is known to have better asymptotic properties than the group Lasso [12] in terms of model selection consistency. The additional randomisation we propose leads empirically to better results, especially once integrated in a stability selection loop - (2) we implement the latter into the deep learning discovery framework DeepMod (see note 1) to perform discoveries from highly noisy datasets. Overall, our experimental results show for the first time, more generalizable PDEs can be found from multiple highly noisy datasets, by promoting grouped sparsity rather than simply performing independent model discoveries. Finally, our results suggest sharing layers across neural networks within DeepMod does not necessarily improve the inference performance.

2 Methods

Sparse regression based model discovery An approach to discover PDEs from q experiments is to repeat q times a method used to perform discoveries from single experiments. Following [6] a randomised adaptive Lasso to select the terms of the PDE can be used for i^{th} experiment,

$$\hat{\tilde{\xi}}_i = \arg\min_{\tilde{\xi}} \Big(\frac{1}{2n} ||\partial_t u_i - \tilde{\Theta}_i \tilde{\xi}_i||_2^2 + \lambda \sum_{j=1}^{p} \frac{||\tilde{\xi}_{i,j}||_1}{W_{i,j}} \Big) \tag{1}$$

where $\tilde{\Theta}_{i,j} = \Theta_{i,j}/\hat{w}_{i,j}$, $\tilde{\xi}_{i,j} = \hat{w}_{i,j} \cdot \xi_{i,j}$, $\hat{w}_{i,j} = 1/|\hat{\xi}_{i,j}|^{\gamma}$, with $\gamma = 2$ and $W_{i,j}$ is randomly selected from a beta distribution, $w \sim \beta(1,2)$ to promote weights close to 0. While this randomised version[2] of the adaptive Lasso is less common than its deterministic counter part, it is known to perform better in terms of variable selection consistency [13, 6]. Similar observationsas in [6] appear with grouped variables, see Appendix A.

Grouped sparsity By assuming the underlying PDE terms are the same across multiple experiments, with potentially different coefficients, grouped sparsity allows to leverage the data across the experiments. The previous estimator can be generalised (inspired by [12]) to grouped variables,

$$\hat{\tilde{\xi}} = \arg\min \Big(\frac{1}{2n} ||\partial_t u - \tilde{\Theta} \tilde{\xi}||_F^2 + \lambda \sum_{g=1}^{p} \frac{||\tilde{\xi}_g||_2}{W_g} \Big) \tag{2}$$

where $\tilde{\Theta}_{i,g} = \Theta_{i,g}/\hat{w}_{i,g}$, $\tilde{\xi}_{i,g} = \hat{w}_{i,g} \cdot \xi_{i,g}$, $\hat{w}_{i,g} = 1/|\hat{\xi}_{i,g}|^{\gamma}$ and W_g is randomly selected from w. If there is one element per group, the randomised adaptive group Lasso[3] becomes the randomised adaptive Lasso. The amount of regularisation parametrized by λ is found automatically using stability selection with some error control, see Appendix B.

Deep learning based model discovery In practice, experimental data contains not only natural variability but also noise. It is known that pure sparse regression based model discovery cannot handle high noise levels (>5 %) - for such cases, deep learning model discovery frameworks excel see [5, 7] . Here, we extend DeepMod [5] to handle multiple experiments (see note 1). The framework combines a function approximator of u, typically a deep neural network which is trained with the following loss,

$$\mathcal{L} = \sum_{i=1}^{q} \Big(\underbrace{\frac{1}{n} ||u_i - \hat{u}_i||_2^2}_{\mathcal{L}_{MSE}} + \underbrace{\frac{1}{n} ||\partial_t \hat{u}_i - \Theta_i(\hat{\xi}_i \cdot M_i)||_2^2}_{\mathcal{L}_{reg}} \Big) \tag{3}$$

The first term $\mathcal{L}_{MSE}$ learns the data mapping $(x,t) \to \hat{u}_i$, while the second term $\mathcal{L}_{reg}$ constrains the function approximator to solutions of the partial differential equation given by $\partial_t u_i, \Theta_i$ and $(\hat{\xi}_i \cdot M_i)$. The terms to be selected in the PDEs are determined using a mask M derived from the result of the randomised adaptive Lasso/group Lasso,

$$M_i = \begin{cases} 1 & \text{if } \tilde{\xi}_j \in S^{\Lambda^*}_{\text{stable}} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

[2] A lose connection can be made here with the transformation from decision trees to random forests: the randomisation will select features randomly.

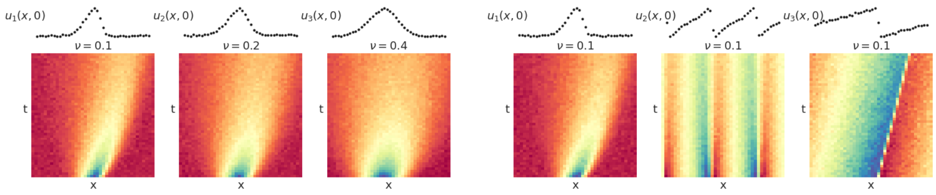
[3] Implementation wise, the group Lasso python library MuTaR is used, see [14].

where j is the index of a potential term and $S^{\Lambda^*}_{\text{stable}}$ is determined by equation (7). The coefficients $\hat{\xi}$ in front of the potential terms are computed using a Ridge regression on the masked library $(\Theta_i \cdot M_i)$. During training, if $\sum_{i=1}^{q} \mathcal{L}_{MSE}$ on the test set does not vary anymore or if it increases, the sparsity estimator is triggered periodically. As a result, the PDE terms are selected iteratively by the dynamic udpate of the mask M during the training.

3 Experiments

In this section, we first present two cases based on the viscous Burgers' equation: $u_t = \nu u_{xx} - uu_x$. The first case consists in generating three spatiotemporal datasets with an identical initial condition (Dirac delta) and different viscosity parameters $\nu = \{0.1, 0.2, 0.4\}$, see figure 1(a). For the second case we fix $\nu = 1$ and vary the initial conditions (Dirac delta, periodic and step-like functions) see figure 1(b). These datasets are generated from analytical solutions [6], on top of which we add a high noise level of 10% Gaussian white noise. In both cases, 40 points in time and 50 points in time are considered leading to 2000 samples per experiment. In such conditions, the observations of the fields u are coarsely sampled and noisy enough such that it would not be possible to compute accurate enough derivatives for the libraries using classical methods such as finite differences or polynomial interpolation. By leveraging automatic differentiation and physics informed neural networks to compute the libraries, these cases are typically the ones where deep learning based model discovery frameworks excel. We compare the two different methods for promoting sparsity in DeepMod described in the previous sections: the individual sparsity promotion method is performed q times (once per experiment) or the grouped sparsity promotion is performed, see figures 2 and 4. The sparsity patterns that are recovered might be different, see figure 2(a): the individual randomised adaptive Lasso is incapable of recovering the nonlinear advection term for the last experiment. This can be explained physically, as it happens for the experiment with the highest diffusion coefficient ($\nu = 0.4$): diffusion is predominant over nonlinear advection. On the other hand, by promoting grouped sparsity, the ground truth underlying PDE can be recovered and it can be verified that it actually generalises better since $\text{MSE}_{\text{test,group}} = 1.1749e^{-3} < \text{MSE}_{\text{test,individual}} = 1.1761e^{-3}$. We report a similar conclusion for the chaotic Kuramoto-Sivashinsky equation, $\text{MSE}_{\text{test,group}} = 5.51e^{-3} < \text{MSE}_{\text{test,individual}} = 5.59e^{-3}$, see figure 3. The same hyperparameters were used, see Appendix C and 2 × 2000 points are randomly sampled from a numerical solution [2], on top of which we add a high noise level of 20% Gaussian white noise. Since we use NNs to interpolate across samples, we also investigated the impact of sharing NN layers across tasks for cases 1 and 2. Interestingly, sharing layers across neural networks does not help DeepMod finding better coefficient estimates nor improves the variable selection performance - it can actually be the opposite, see figure 4 and Appendix D.

It is important to keep in mind when promoting grouped sparsity, that if a term is important to recover a PDE, it will be forced to be conserved for all the other PDEs. By doing so, we create a learning bias that implies the a priori assumption that all experiments can be explained by the same underlying PDE terms with potentially different coefficients. Obviously, the learning bias can also turn into a *negative learning transfer* if the latter assumption is wrong. A limitation of our approach is reported when the discovery problem does not have unique solutions, see Appendix E.



(a) *case 1: with different parameters ν.*

(b) *case 2: with different initial conditions.*

Figure 1: *Cases 1 and 2 datasets.* Burgers' PDE is solved analytically for (a) different values of the viscosity ν and (b) the initial conditions vary at fixed viscosity. Each case dataset contains 3 × 2000 samples with the addition of 10% Gaussian white noise.

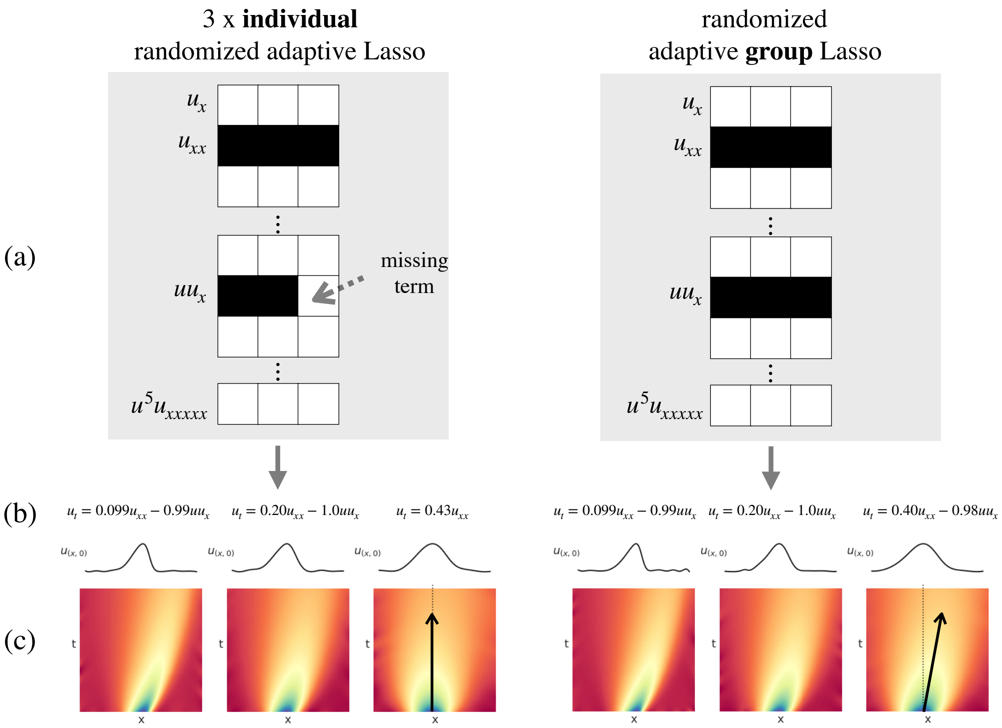

3 x **individual**
randomized adaptive Lasso
randomized
adaptive **group** Lasso
(a)
u_x
u_{xx}
uu_x
u^5u_{xxxxx}
missing term
(b)
$u_t = 0.099u_{xx} - 0.99uu_x$
$u_t = 0.20u_{xx} - 1.0uu_x$
$u_t = 0.43u_{xx}$
$u_t = 0.099u_{xx} - 0.99uu_x$
$u_t = 0.20u_{xx} - 1.0uu_x$
$u_t = 0.40u_{xx} - 0.98uu_x$
$u(x, 0)$
(c)
t
x

Figure 2: *Example of sparsity patterns (a), equations recovered (b) and data interpolations (c) by DeepMod from case 1 dataset.* In (b) the absence of the nonlinear advection term (uu_x) results on the recovery of a PDE with pure diffusive trajectory (c). Using the randomised adaptive group Lasso allows to enforce a learning bias that helps finding the actual ground truth: the Burgers' equation ($u_t = \nu u_{xx} - uu_x$) with varying viscosities ν.

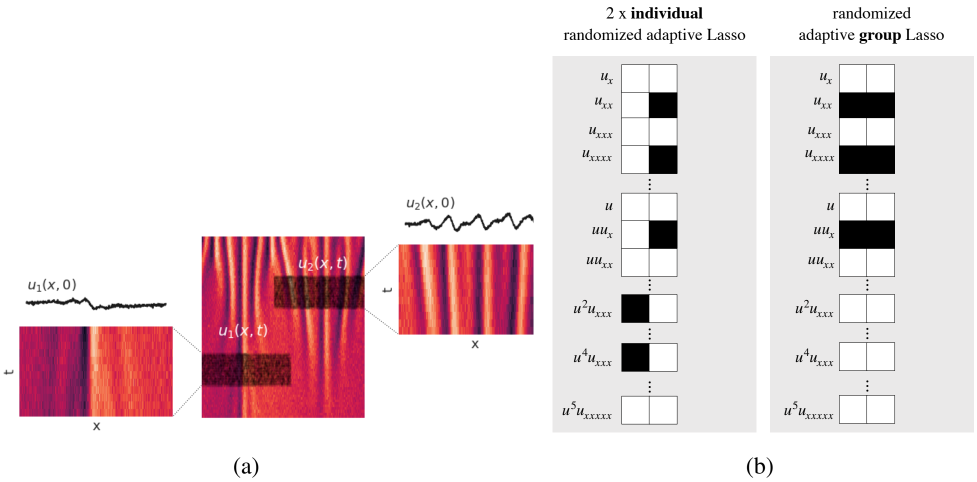


Figure 3: *Case 3 dataset and sparsity patterns recovered by DeepMod.* In this experiment, we extract two datasets (2 × 2000 random samples) in (a) pre-chaotic (u_1) and chaotic (u_2) regimes from a numerical solution [2] of the Kuramoto-Sivashinsky equation ($u_t = -uu_x - u_{xx} - u_{xxxx}$), with 20% Gaussian white noise. (b) The underlying PDE can only be recovered when grouped sparsity is promoted.

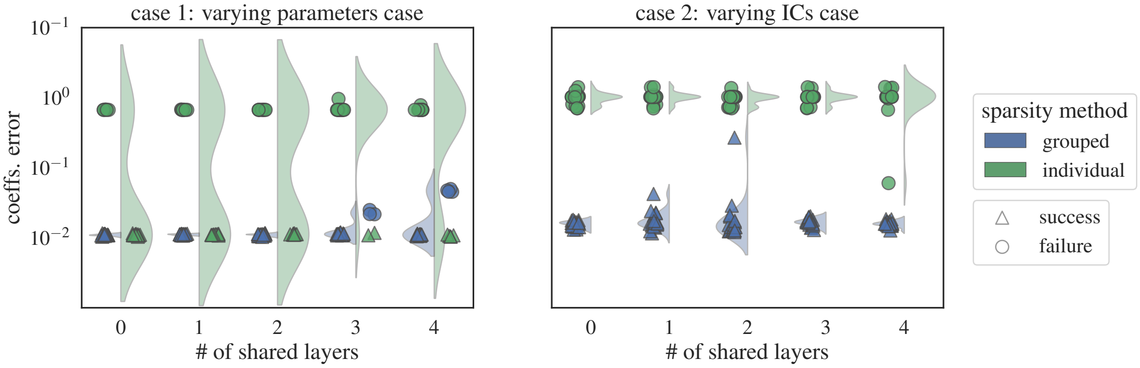


Figure 4: *Performance of multitask task learning in model discovery*. Each experiment is repeated with 20 different initialisation seeds for each NN architecture. By promoting sparsity as a group and without sharing layers across neural networks, DeepMod becomes a sharp estimator of the common underlying PDE from multiple experiments.

4 Conclusion

We extend the limitation of relying on a single experiment to infer PDEs and explore how to leverage the information across multiple experiments with different parameters, initial and/or boundary conditions. To do so, we first introduce a randomised adaptive group Lasso to promote grouped sparsity. This creates a learning bias in the model discovery process which assumes there is a common underlying PDE with potentially different coefficients across experiments. Second, once integrated in the deep learning model discovery framework DeepMod, our results show *more generalizable PDEs* can be found from highly noisy datasets, by promoting grouped sparsity rather than simply performing independent model discoveries.

Future work will focus on the extension of the approach to additional spatial dimensions in order to tackle real life experimental applications such as retrieving recovering fluid flow equations in turbulent conditions from particle image velocimetry.

Acknowledgments and Disclosure of Funding

This work received support from the CRI Research Fellowship to attributed to Remy Kusters.

References

[1] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

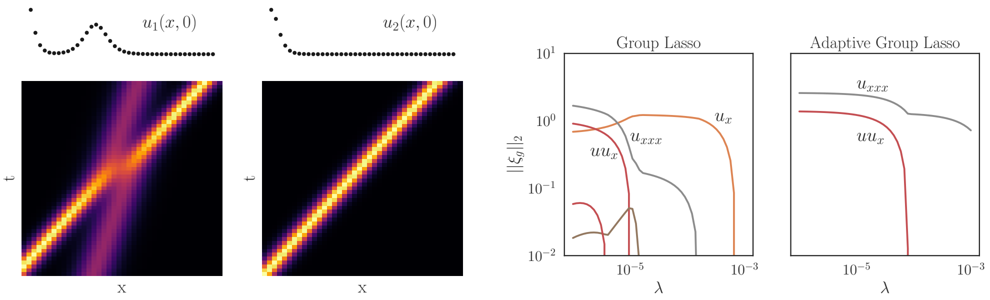
[2] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[3] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446, 2017.

[4] Suryanarayana Maddu, Bevan L. Cheeseman, Ivo F. Sbalzarini, and Christian L. Müller. Stability selection enables robust learning of partial differential equations from limited noisy data. *arXiv preprint arXiv:1907.07810*, 2019.

[5] Gert-Jan Both, Subham Choudhury, Pierre Sens, and Remy Kusters. Deepmod: Deep learning for model discovery in noisy data. *Journal of Computational Physics*, 428:109985, 2021.

[6] Georges Tod, Gert-Jan Both, and Remy Kusters. Sparsistent model discovery. *arXiv preprint arXiv:2106.11936*, 2021.



(a)

(b)

Figure 5: *When the group Lasso fails to recover the Kortweg-de-Vries equation* ($u_t = -6uu_x - u_{xxx}$). From 2×36 terms libraries computed from two analytical solutions with single and double solitons. Group norms of coefficients paths show the group Lasso would not manage to find the KdV equation for any λ, while the Adaptive Group Lasso would.

[7] Zhao Chen, Yang Liu, and Hao Sun. Deep learning of physical laws from scarce data. *arXiv preprint arXiv:2005.03448*, 2020.

[8] Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, 18 (2):643–660, 2019.

[9] Suryanarayana Maddu, Bevan L. Cheeseman, Christian L. Müller, and Ivo F. Sbalzarini. Learning physically consistent mathematical models from data using group sparsity. *arXiv preprint arXiv:2012.06391*, 2020.

[10] Brian M de Silva, David M Higdon, Steven L Brunton, and J Nathan Kutz. Discovery of physics from data: Universal laws and discrepancies. *arXiv preprint arXiv:1906.07906*, 2019.

[11] Fengrong Wei and Jian Huang. Consistent group selection in high-dimensional linear regression. *Bernoulli: official journal of the Bernoulli Society for Mathematical Statistics and Probability*, 16(4):1369, 2010.

[12] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[13] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[14] Hicham Janati. Multi-task regression in python, 2019. URL `https://hichamjanati.github.io/mutar/`.

[15] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

A Group Lasso and Adaptive Group Lasso

An illustration of when the group Lasso might not recover the underlying PDEs where the adaptive group Lasso would is given, see see figure 5. We find the irrelevant term u_x is highly correlated with the relevant term u_{xxx}. In that situation, the group Lasso would not manage to find the KdV equation for any λ, see figure 5, while the Adaptive Group Lasso would. An idea of why such a behavior can be found in [6].

B Stability selection for the randomised adaptive Lasso/group Lasso

This appendix extends stability selection with error control from [6] to the randomised adaptive group Lasso. With stability selection, variables are chosen according to their probabilities of being selected with some warranty on the selection error, [13]. The first step consists in finding the probability of a variable k of being selected under some data perturbation: let I_b be one of B random sub-samples of half the size of the training data drawn without replacement. For a given λ an estimation of the probability of k being selected is given by,

$$\hat{\Pi}_k^\lambda = \frac{1}{B}\sum_{b=1}^{B}\begin{cases} 1 & \text{if } f_k^\lambda(I_b) > 0 \\ 0 & \text{otherwise}\end{cases} \tag{5}$$

where $f_k^\lambda(I_b) = ||\hat{\xi}_k^\lambda(I_b)||_1$ for the randomised adaptive Lasso. The extension to the randomised adaptive group Lasso simply requires $f_k^\lambda(I_b) = ||\hat{\xi}_k^\lambda(I_b)||_2$. Second, by computing the probabilities of being selected over a given range of λ's the stability paths can be obtained. That range is initially denoted $\Lambda = [\epsilon\lambda_{max}; \lambda_{max}]$, where ϵ is the path length and λ_{max} is the regularisation parameter where all coefficients $\hat{\xi}$ are null. [13] derive an upper bound on the expected number of false positives $\mathbb{E}(V)$, that can help determining a smaller Λ region where some control on the selection error can be warrantied. By fixing this bound to EV_{max}, the regularisation region becomes,

$$\Lambda^* = \left\{\lambda \in \Lambda \text{ such that, } \mathbb{E}(V) \leq \frac{q_\Lambda^2}{(2\pi_{thr}-1)p} \leq EV_{max}\right\} \tag{6}$$

where π_{thr} is the minimum probability threshold to be selected and q_Λ is the average of selected variables that we propose to approximate by $\hat{q}_\Lambda = \sum_k \hat{\Pi}_k^\lambda$. Finally, the set of stable variables with an upper bound on the expected number of false positives is,

$$S_{\text{stable}}^{\Lambda^*} = \left\{k \text{ such that } \max \hat{\Pi}_k^\lambda \geq \pi_{thr} \text{ for } \lambda \in \Lambda^*\right\} \tag{7}$$

C A single set of hyperparameters

A single set of hyperparameters is used for all cases presented in this work - except for the experiments presented in figure 4 where the NN architectures vary.

Stability selection expected number of false positives upper bound $EV_{max} = 3$, number of resamples $B = 40$ and the minimum probability to be selected $\pi_{thr} = 0.9$.

Library consists of polynomials and partial derivatives up to the fifth order leading to a library size of p =36 potential terms.

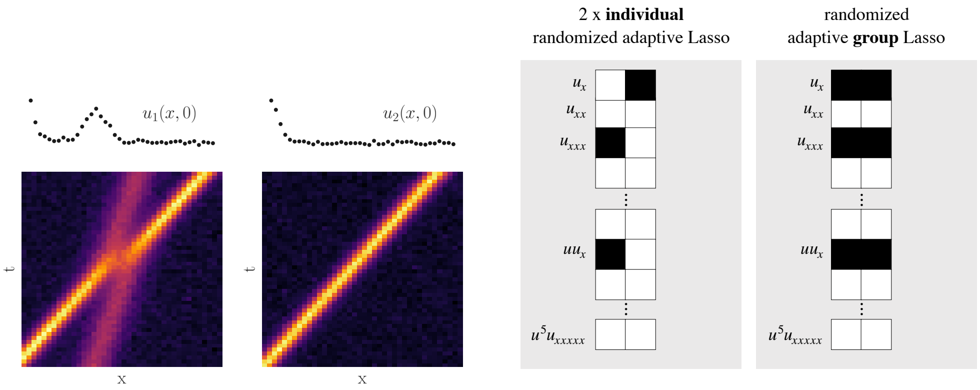
Neural network architecture & optimiser NNs are 4 layers deep with 65 neurons per layer and sinus activation functions with a specific initialisation strategy, see [15]. The NNs are trained by an Adam optimiser with a learning rate of $5 \cdot 10^{-5}$ and $\beta = (0.99, 0.99)$.

D Multitask neural networks performance

When sharing layers, the samples used to train for a task are going to be used in the training of other tasks as well, creating another learning bias. To verify more systematically the performance of the promotion of grouped sparsity and multitask neural networks, we vary the methods to promote sparsity and the NN architectures [4], see figure 4.

Performance metrics We introduce two metrics: (1) the result of an experiment is declared as a success, if and only if, the terms of the ground truth underlying PDE are found. In the case of failure, the metric is uninformative about whether an additional term was discovered or a term was missing - (2) the coefficients error metric quantifies the discrepancy between the right hand side coefficients of

[4] To make sure the NN architectures are comparable, the total number of weights and biases $\sim 40k$ are fixed.



(a) (b)

Figure 6: *Case 4 dataset and sparsity patterns recovered by DeepMod.*. In this experiment, we infer the equations from two analytical solutions of the Kortweg-de-Vries equation ($u_t = -6uu_x - u_{xxx}$): with single and double solitons. The dataset consists of 2 × 2000 samples with the addition of 10% Gaussian white noise.

the ground truth PDEs and the discovered ones by computing: $||\xi - \hat{\xi}||_F/||\xi||_F$, where F stands for the Frobenius norm.

Overall the variance of the results is much lower when promoting sparsity as a group. When no layers are shared, the coefficients error is the lowest and the success rate of the discoveries is equal to 1 when the grouped sparsity method is leveraged. Furthermore, the common underlying PDE across experiments with varying ICs can only be found when grouped sparsity is promoted. Interestingly, sharing layers across neural networks does not help DeepMod finding better coefficient estimates nor improves the variable selection performance - it can actually be the opposite.

E Non-unique solutions limitation

In this experiment, we infer the equations from two analytical solutions of the Kortweg-de-Vries equation ($u_t = -6uu_x - u_{xxx}$): with single and double solitons, see figure 6(a). Each dataset consists of 2 × 2000 samples with 10% Gaussian white noise. The resulting sparsity pattern, see figure 6(b), from the individual sparsity promotion is correct: the single soliton solution of the KdV equation is also a solution of the simpler travelling wave equation: $u_t = -c \cdot u_x$, where c is the velocity of the wave. This illustrates the non-uniqueness of solutions of the PDE discovery problem. The promotion of grouped sparsity results in the discovery of a PDE with the 3 terms: u_x, uu_x and u_{xxx}. While we know the single soliton is a solution of a PDE containing those 3 terms, the double soliton solution is not. The spurious term u_x can be discerned by remarking its small magnitude: $u_t = -0.083u_x - 5.9uu_x - 0.99u_{xxx}$ - alternatively, it can also be diagnosed by comparing MSEs on some out of training data: $\text{MSE}_{\text{test,group}} = 5.48e^{-4} > \text{MSE}_{\text{test,individual}} = 5.41e^{-4}$. However, without the addition of noise, the library Θ can be constructed analytically and the randomised adaptive group Lasso does find the KdV equation as the underlying PDE for both datasets. We speculate the origin of the limitation of DeepMod in the case of non-unique solutions, comes from the Ridge regression used to compute the regularisation term of the loss, see equation 3. As a matter of fact, we remark for this experiment only, the results from DeepMod are very sensitive to the regularisation parameter fixed for the Ridge regression.