Machine Learning for Discovering PDEs

A literature survey

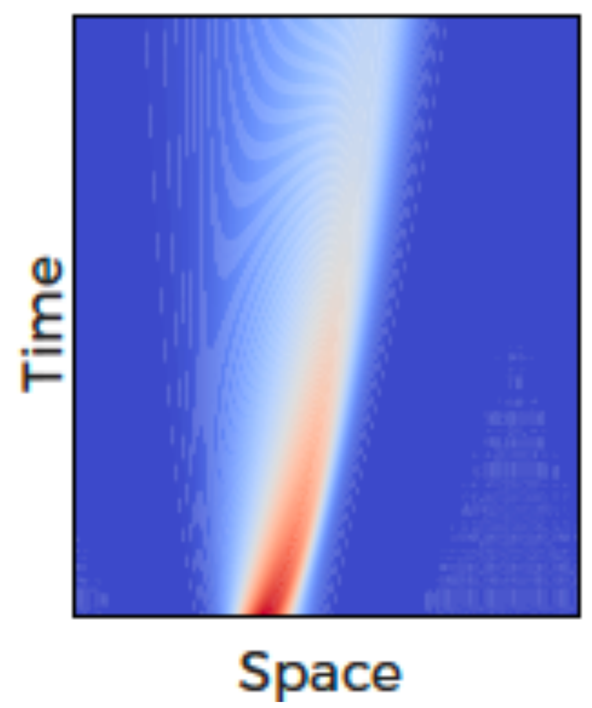
Aman Jalan (05/20/21)

Outline

1. Problem overview and SINDy recap
2. PDENets
3. ML adaptations to SINDy
4. Reinforcement Learning
5. References

Problem Overview and SINDy Recap

Problem Overview



Spatio-temporal Dataset

Burgers equation

$$\partial_t u = u u_x - 0.1 u_{xx}$$

Symbolic PDE

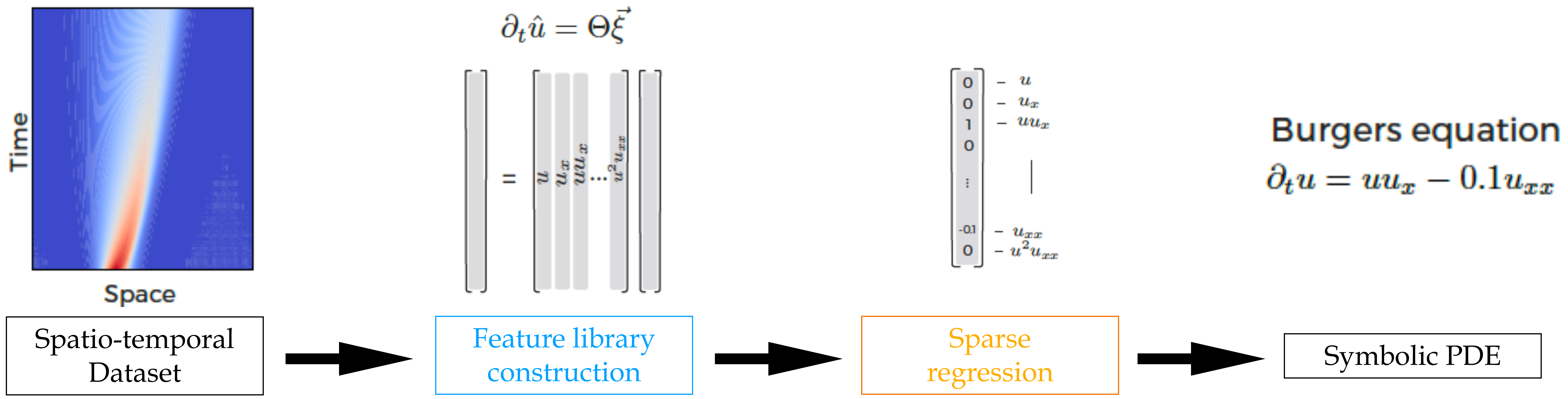
Why is it important?

Facilitates a variety of innovations for characterizing high-dimensional data generated from experiments or observations

Why ML?

Neural network's ability to learn complex non-linear functions provides unique benefits while interpreting underpinning physics

SINDy Recap

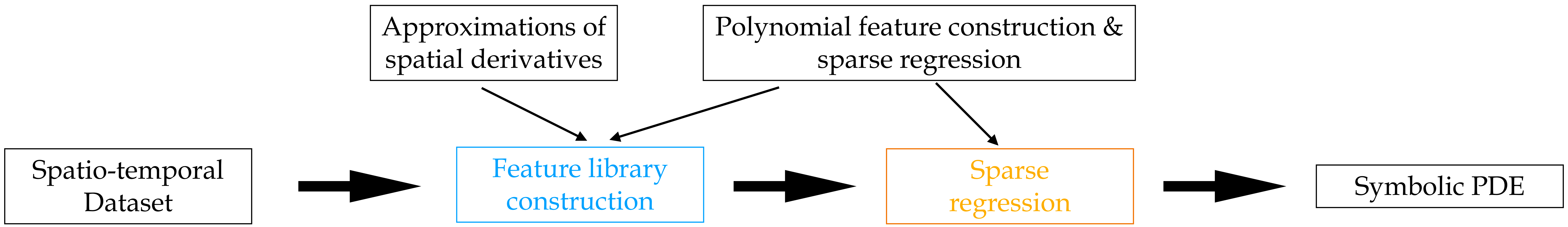


Variants

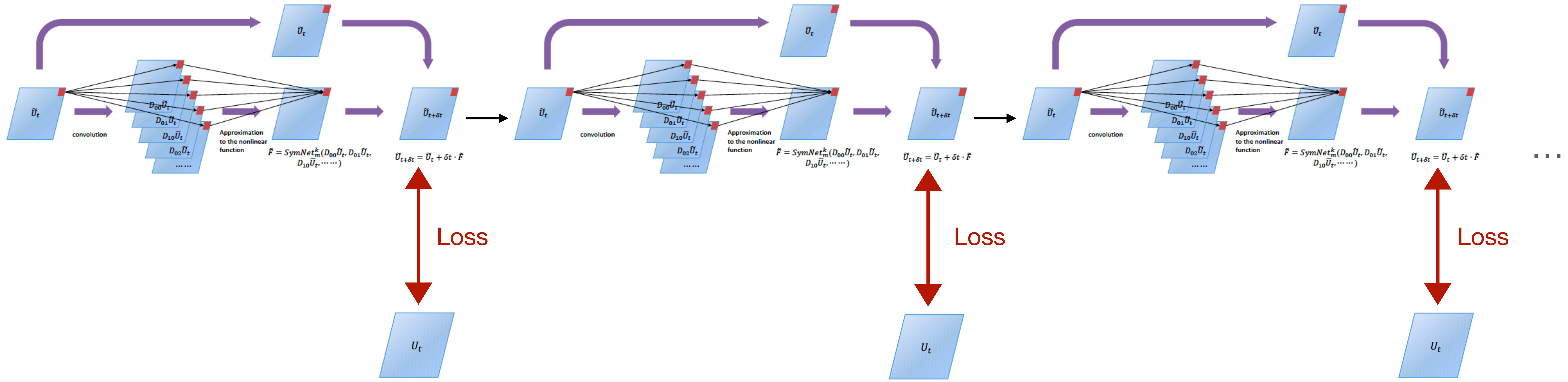
- Sparse regression and adaptive feature generation for the discovery of dynamical systems **[Kulkarni et al. (2019)]**
- Data-driven identification of parametric PDEs **[Rudy et al. (2019)]**
- Data-driven discovery of PDEs **[Rudy et al. (2017)]**
- Robust low-rank discovery of data-driven PDEs **[Li et al. (2020)]**
- Weak SINDy for PDEs **[Messenger and Bortz (2020)]**

PDENets

PDENets


Approximations of spatial derivatives
Polynomial feature construction & sparse regression
Spatio-temporal Dataset
Feature library construction
Sparse regression
Symbolic PDE

PDENets



Novelty

- Relates order of approximation of the spatial derivative with the order of sum rules of the filter.
- Able to learn these filters by imposing sparsity constraints from this relationship

Advantages

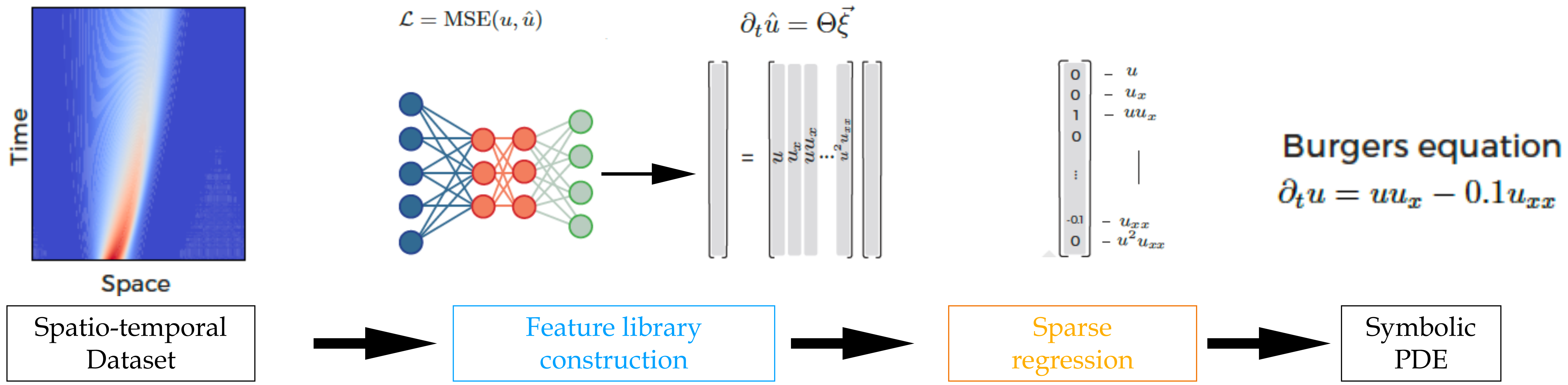
- Outperforms SINDy in terms of accuracy
- Feature libraries constructed are more memory efficient and cheaper to compute compared to SINDy

Pitfalls

- Lot of training data required to learn CNN filters and parameters of the Symbolic NN

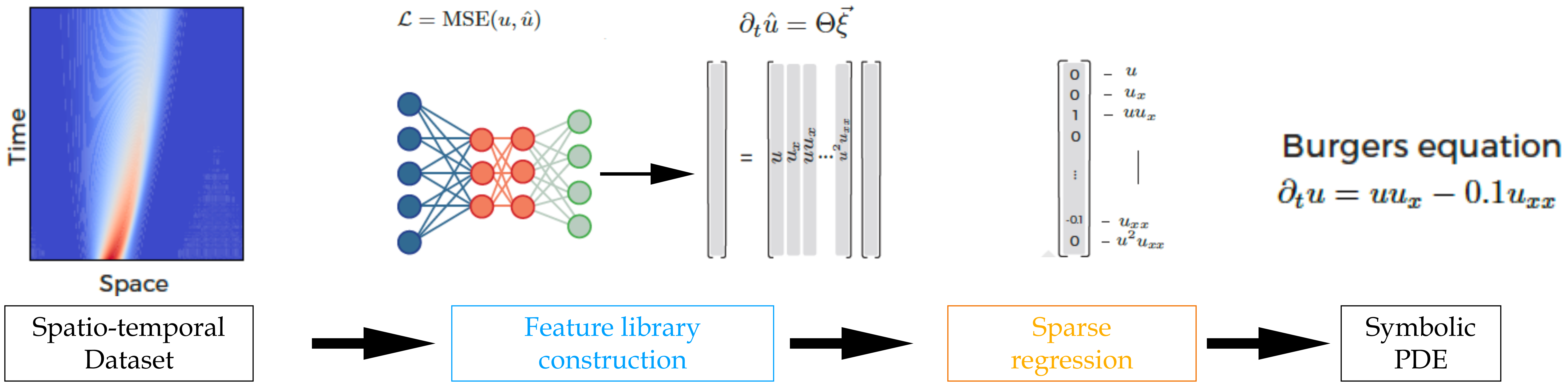
ML adaptations of SINDy: DL-PDE & DeepMOD

DL-PDE

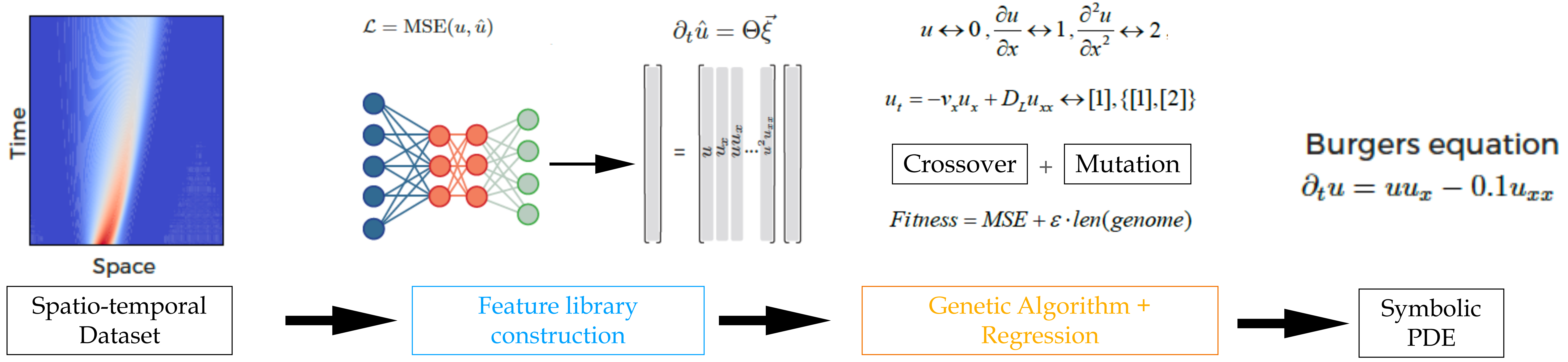


Time
Space
$\mathcal{L} = \mathrm{MSE}(u, \hat{u})$
$\partial_t \hat{u} = \Theta \vec{\xi}$
Burgers equation
$\partial_t u = uu_x - 0.1u_{xx}$
Spatio-temporal Dataset
Feature library construction
Sparse regression
Symbolic PDE

DLGA-PDE

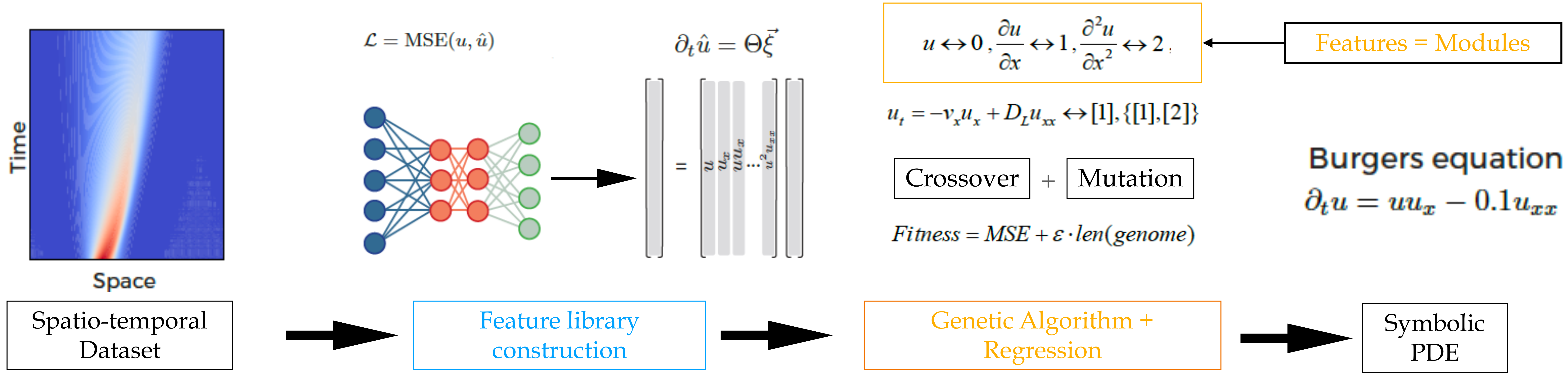

$\mathcal{L} = \mathrm{MSE}(u, \hat{u})$
$\partial_t \hat{u} = \Theta \vec{\xi}$
Time
Space
Burgers equation
$\partial_t u = uu_x - 0.1u_{xx}$
Spatio-temporal Dataset
Feature library construction
Sparse regression
Symbolic PDE

DLGA-PDE

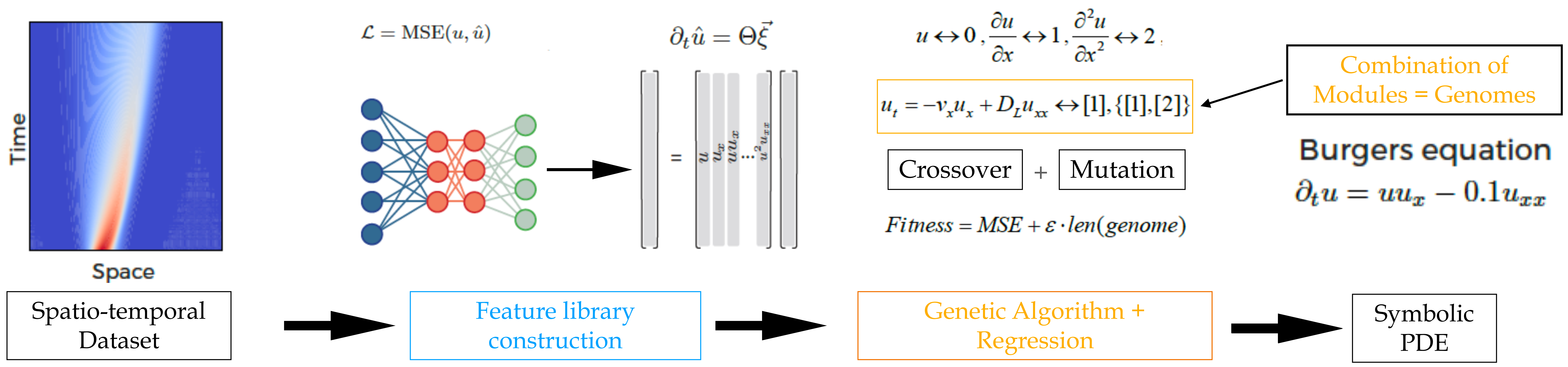


Time
Space
Spatio-temporal Dataset
$\mathcal{L} = \mathrm{MSE}(u, \hat{u})$
$\partial_t \hat{u} = \Theta \vec{\xi}$
Feature library construction
$u \leftrightarrow 0, \frac{\partial u}{\partial x} \leftrightarrow 1, \frac{\partial^2 u}{\partial x^2} \leftrightarrow 2$
$u_t = -v_x u_x + D_L u_{xx} \leftrightarrow [1], \{[1],[2]\}$
Crossover + Mutation
$Fitness = MSE + \varepsilon \cdot len(genome)$
Genetic Algorithm + Regression
Burgers equation
$\partial_t u = u u_x - 0.1 u_{xx}$
Symbolic PDE

DLGA-PDE

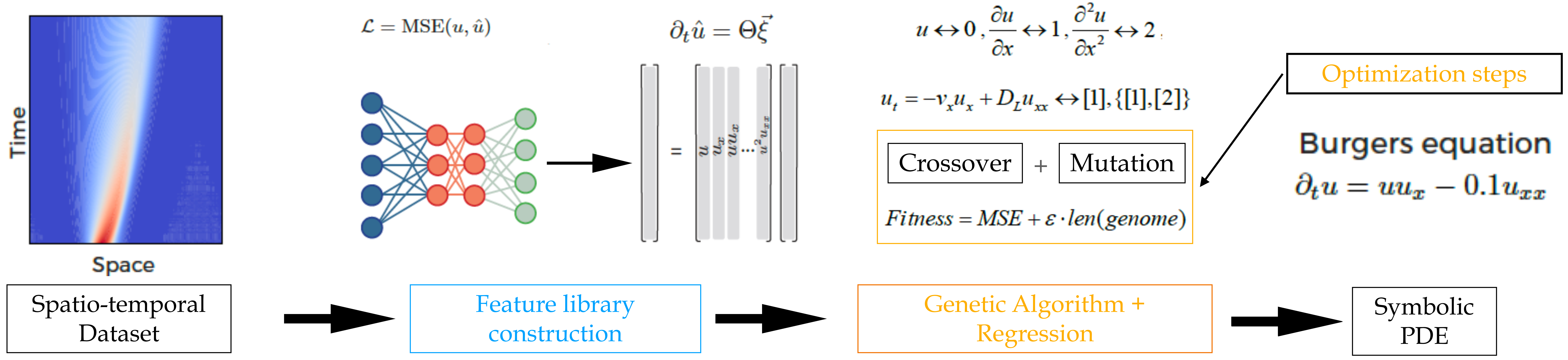

Time
Space
Spatio-temporal Dataset
$\mathcal{L} = \mathrm{MSE}(u, \hat{u})$
$\partial_t \hat{u} = \Theta \vec{\xi}$
Feature library construction
$u \leftrightarrow 0, \frac{\partial u}{\partial x} \leftrightarrow 1, \frac{\partial^2 u}{\partial x^2} \leftrightarrow 2$
Features = Modules
$u_t = -v_x u_x + D_L u_{xx} \leftrightarrow [1], \{[1],[2]\}$
Crossover
+
Mutation
$Fitness = MSE + \varepsilon \cdot len(genome)$
Genetic Algorithm + Regression
Burgers equation
$\partial_t u = uu_x - 0.1u_{xx}$
Symbolic PDE

DLGA-PDE



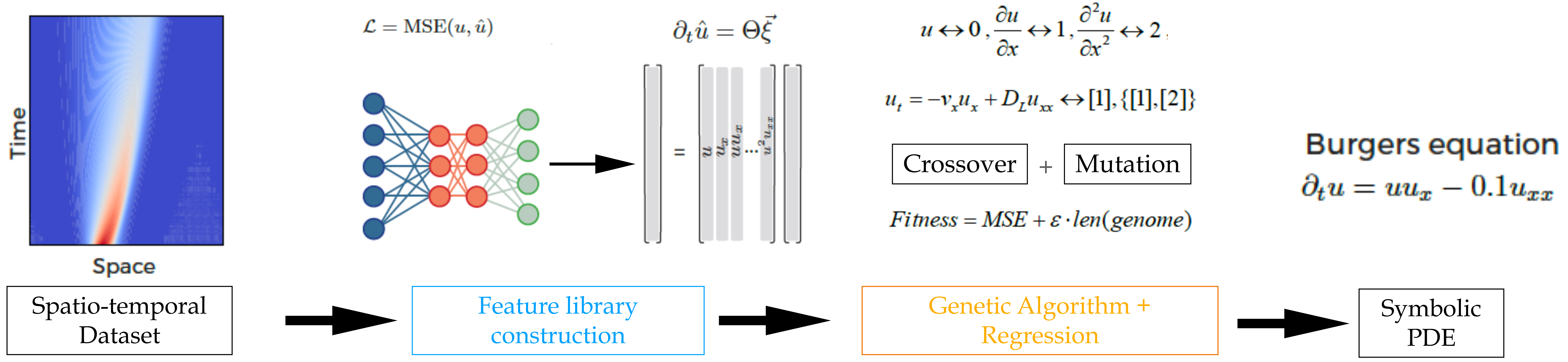
Time
Space
Spatio-temporal Dataset
$\mathcal{L} = \mathrm{MSE}(u, \hat{u})$
$\partial_t \hat{u} = \Theta \vec{\xi}$
Feature library construction
$u \leftrightarrow 0, \frac{\partial u}{\partial x} \leftrightarrow 1, \frac{\partial^2 u}{\partial x^2} \leftrightarrow 2$
$u_t = -v_x u_x + D_L u_{xx} \leftrightarrow [1], \{[1],[2]\}$
Combination of Modules = Genomes
Crossover
+
Mutation
$Fitness = MSE + \varepsilon \cdot len(genome)$
Genetic Algorithm + Regression
Burgers equation
$\partial_t u = u u_x - 0.1 u_{xx}$
Symbolic PDE

DLGA-PDE



Time
Space
Spatio-temporal Dataset
$\mathcal{L} = \mathrm{MSE}(u, \hat{u})$
$\partial_t \hat{u} = \Theta \vec{\xi}$
Feature library construction
$u \leftrightarrow 0, \frac{\partial u}{\partial x} \leftrightarrow 1, \frac{\partial^2 u}{\partial x^2} \leftrightarrow 2$
$u_t = -v_x u_x + D_L u_{xx} \leftrightarrow [1], \{[1],[2]\}$
Crossover
Mutation
$Fitness = MSE + \varepsilon \cdot len(genome)$
Optimization steps
Genetic Algorithm + Regression
Burgers equation
$\partial_t u = u u_x - 0.1 u_{xx}$
Symbolic PDE

DLGA-PDE



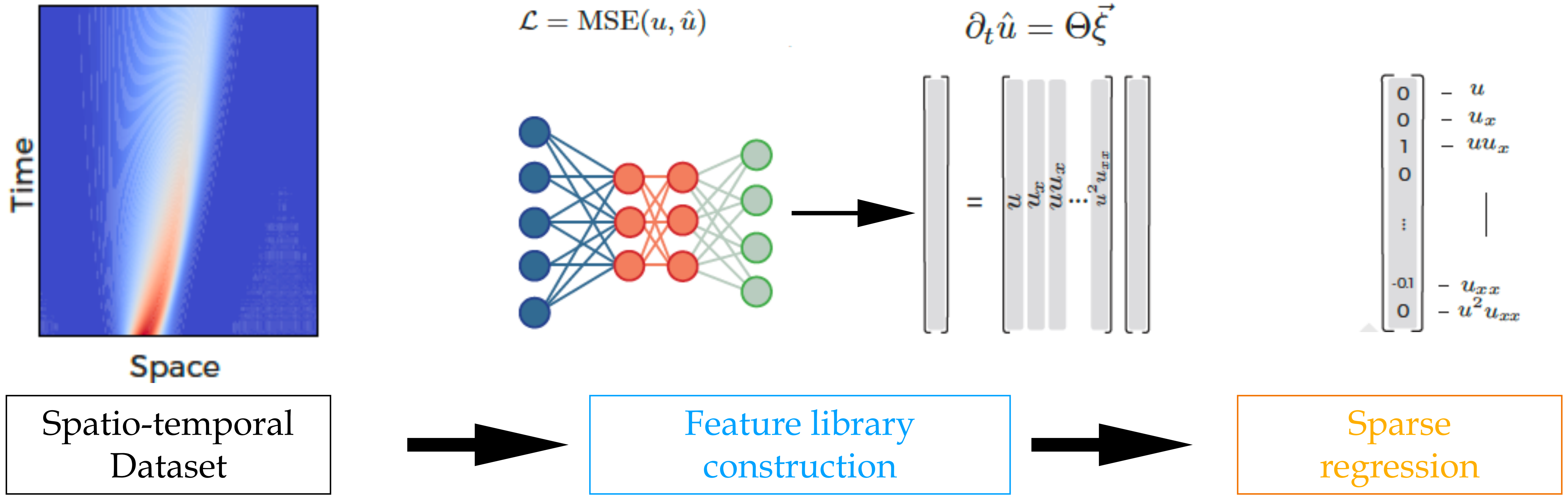
Advantages

- Meta-data can be generated using learned NN - more data available for feature library construction & regression
- Automatic differentiation can be used to compute spatial derivatives instead of FD schemes

Pitfalls

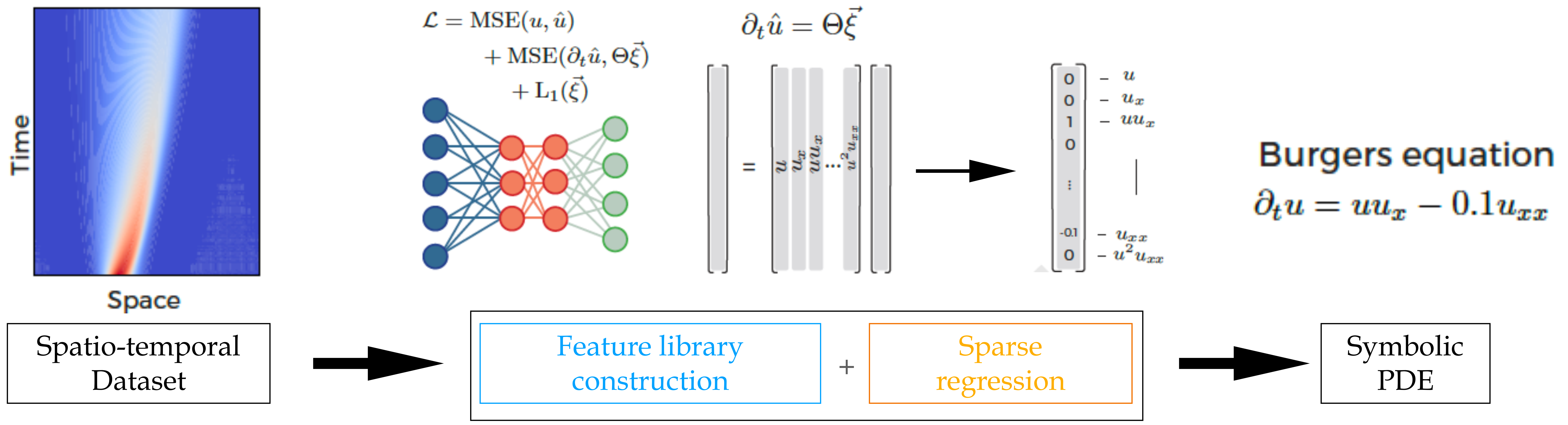
- Prone to over-fitting noisy data with significant loss of accuracy

DeepMOD

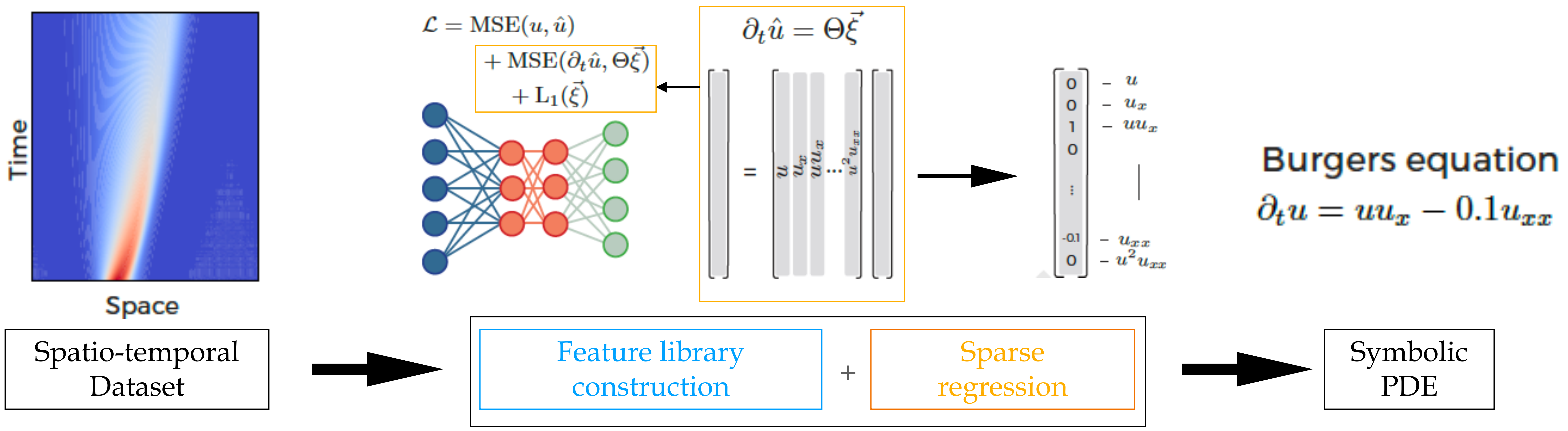


Time
Space
$\mathcal{L} = \mathrm{MSE}(u, \hat{u})$
$\partial_t \hat{u} = \Theta \vec{\xi}$
=
u
u_x
uu_x
$u^2 u_{xx}$
0
0
1
0
-0.1
0
u
u_x
uu_x
u_{xx}
$u^2 u_{xx}$
Spatio-temporal Dataset
Feature library construction
Sparse regression

DeepMOD


Time
Space
Spatio-temporal Dataset
$\mathcal{L} = \text{MSE}(u, \hat{u}) + \text{MSE}(\partial_t \hat{u}, \Theta\vec{\xi}) + \text{L}_1(\vec{\xi})$
$\partial_t \hat{u} = \Theta\vec{\xi}$
Feature library construction
+
Sparse regression
Burgers equation
$\partial_t u = u u_x - 0.1 u_{xx}$
Symbolic PDE

DeepMOD



Novelty

- Learns NN parameters and sparse regression coefficients together
- Training the NN effectively de-noises data and adjusts components of the feature library

Advantages

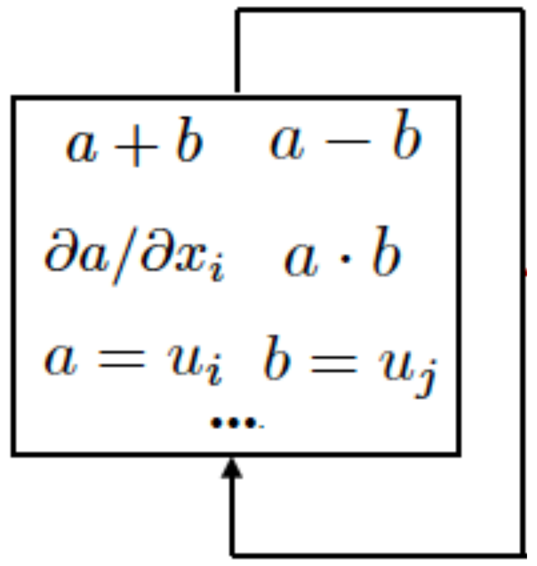
- Avoids over-fitting to noisy data
- Improved accuracy compared to DL-PDE

Reinforcement Learning

Reinforcement Learning

Reinforcement Learning

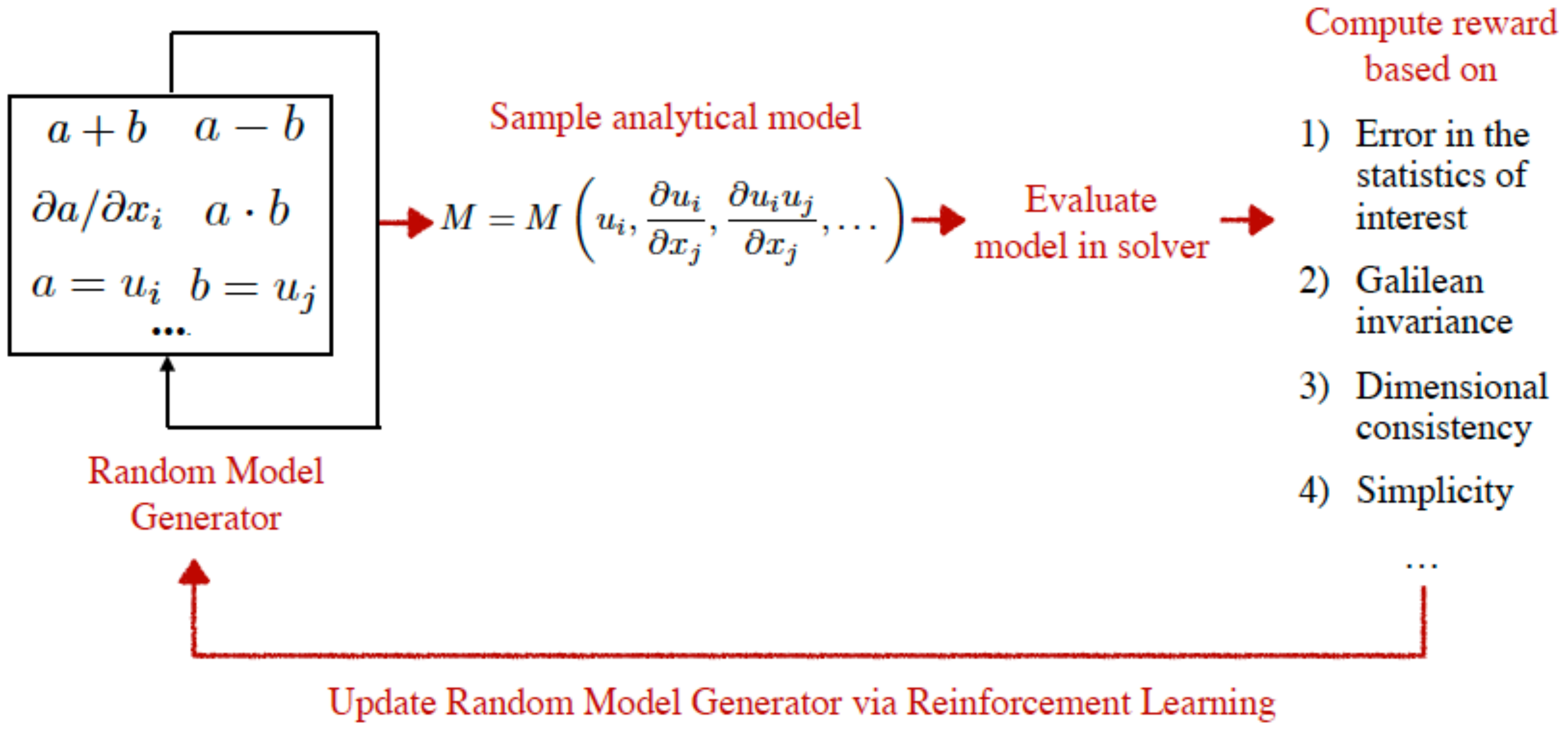
- Outputs mathematical expressions given a probability distribution
- Contains a computational graph to encode expressions in a Domain-specific language (DSL)
- Probability distribution is parameterized using a NN



Random Model Generator

Reinforcement Learning

- Outputs mathematical expressions given a probability distribution
- Contains a computational graph to encode expressions in a Domain-specific language (DSL)
- Probability distribution is parameterized using a NN



$$J(\theta) = \mathbb{E}_{a\sim\pi_\theta(\cdot)}[R(a)]$$

Objective = Average accuracy

Similar approach:

- Automating turbulence modelling by multi-agent reinforcement learning **[Novati et al. (2021)]**
  - Use RL to find coefficients of turbulence models
  - Rewards are computed by checking if statistical properties of DNS are preserved

Summary

1. PDENets
2. ML adaptations to SINDy
3. Reinforcement Learning

References

Kulkarni, Chinmay S., Abhinav Gupta, and Pierre FJ Lermusiaux. "Sparse Regression and Adaptive Feature Generation for the Discovery of Dynamical Systems." *International Conference on Dynamic Data Driven Application Systems*. Springer, Cham, 2020.

Rudy, Samuel, et al. "Data-driven identification of parametric partial differential equations." *SIAM Journal on Applied Dynamical Systems* 18.2 (2019): 643-660.

Rudy, Samuel H., et al. "Data-driven discovery of partial differential equations." *Science Advances* 3.4 (2017): e1602614.

Li, Jun, et al. "Robust Low-Rank Discovery of Data-Driven Partial Differential Equations." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. No. 01. 2020.

Messenger, Daniel A., and David M. Bortz. "Weak SINDy For Partial Differential Equations." *arXiv preprint arXiv:2007.02848* (2020).

Long, Zichao, Yiping Lu, and Bin Dong. "PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network." *Journal of Computational Physics* 399 (2019): 108925.

Long, Zichao, et al. "Pde-net: Learning pdes from data." *International Conference on Machine Learning*. PMLR, 2018.

Xu, Hao, Haibin Chang, and Dongxiao Zhang. "Dl-pde: Deep-learning based data-driven discovery of partial differential equations from discrete and noisy data." *arXiv preprint arXiv:1908.04463* (2019).

Xu, Hao, Haibin Chang, and Dongxiao Zhang. "DLGA-PDE: Discovery of PDEs with incomplete candidate library via combination of deep learning and genetic algorithm." *Journal of Computational Physics* 418 (2020): 109584.

Both, Gert-Jan, et al. "DeepMoD: Deep learning for Model Discovery in noisy data." *Journal of Computational Physics* 428 (2021): 109985.

Bassenne, Maxime, and Adrián Lozano-Durán. "Computational model discovery with reinforcement learning." *arXiv preprint arXiv:2001.00008* (2019).

Thank you!