SPIGAN: A GENERATIVE ADVERSARIAL NETWORK SUPERVISED BY SPARSE IDENTIFICATION TO LEARN GOVERNING EQUATIONS FROM SCARCE DATA

Shen Yue[*1] **and Chen Yu**[†1]
[1]Department of Human and Engineered Environmental Studies
Graduate School of Frontier Sciences
The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba, 277-8561, Japan

ABSTRACT

This work introduces SPIGAN to discover governing partial differential equations from scarce data for nonlinear spatiotemporal systems, which leverages the strengths of GAN-based neural networks for physics embedding, automatic differentiation, and data generation, along with the advantages of sparse identification in identifying key derivative terms in an end-to end manner. The efficacy and robustness of SPIGAN are demonstrated with Burgers' equation under different parameter settings. The ablation tests further highlight the advantages of SPIGAN over original sparse identification or GAN approaches.

Keywords Physics-informed machine learning · Generative adversarial networks · Equation learning

1 Introduction

Constructing a coherent description of natural laws using differential equations poses a significant challenge across various scientific fields. The prevailing approach to modeling complex dynamical systems relies heavily on ordinary and/or partial differential equations (ODEs, PDEs) that govern the system's behaviors. Traditionally, in order to analyze the underlying dynamics of the concerned phenomenon, scientists would derive these equations from the first principle or just intuitively write down mathematical equations despite that a reasonable amount of data has already been obtained. However, these classical approaches have shown limited efficacy and slow progress. Fortunately, the availability of rich observational datasets offers an alternative avenue for distilling underlying equations from data. Recent advancements in machine learning (ML) have significantly accelerated the field of equation learning (EQL), which aims to automatically and directly extract interpretable analytic equations from big data.

There has been vast research on data-driven discovery of dynamical systems which involves a variety of approaches. Recently, an impressive breakthrough made by Brunton et al.[1, 2] called sparse identification of nonlinear dynamics (SINDy) has been successfully applied to uncovering the dynamic dependencies in spatio-temporal data generated by numerical simulations of dynamic models[3, 4, 5, 6]. However, when it comes to real-case data-driven discovery of ODEs/PDEs, the SINDy framework faces a critical bottleneck which arises from its strong dependence on the quality and quantity of measurement data. Due to the high costs associated with data collection, generation, and labeling, as well as data quality issues such as missing data, it is often the case that the collected data fail to meet these requirements. In such instances, we encounter data with sparsely and irregularly evaluated temporal or spatial intervals. The direct impact of scarce data on SINDy is most noticeable when numerical differentiation is estimated using traditional algorithms like finite differences. Since the estimated differentiation is subsequently used as inputs for

[*]shenyue@s.h.k.u-tokyo.ac.jp
[†]chen@edu.k.u-tokyo.ac.jp

sparse regression, the inaccuracy of this stage significantly influences the overall performance of SINDy. Recently, some scholars [7] have tried to resort to physics-informed neural networks to reduce noise and obtain robust derivatives, which calls for further exploration with other network structures.

To solve this problem, we introduce a novel approach called SPIGAN (SINDY with Physics-informed GAN) to uncover the governing PDEs of nonlinear spatiotemporal systems using scarce data. Our approach harnesses the combined strengths of GANs for robust representation learning, automatic differentiation for precise derivative calculation, and data generation to address the inherent limitations of existing methods that struggle with data scarcity, while maintaining the ability of inferring ODEs/PDEs in a explicit form from SINDy. SPIGAN allows us to overcome the fundamental challenges associated with scarce data and offers a more effective solution for discovering the underlying dynamics of complex systems.

2 Methods

2.1 Sparse Identification Pipeline

Consider a modeled system of the ODE form

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X}(t) = \boldsymbol{F}(\boldsymbol{X}(t)) \tag{1}$$

where the vector $\boldsymbol{X}(t) = [x_1(t), x_2(t), \cdots, x_n(t)]$ denotes the states of system at time t, and the function $\boldsymbol{F}(\boldsymbol{X}(t))$ describes the dynamics of the system. A key assumption here is that the function $\boldsymbol{F} = \boldsymbol{F}(\boldsymbol{X}, t)$ consists of only a few terms, making the sparse regression methodologies applicable in consideration of both efficiency and accuracy.

The SINDy pipeline begins by collecting all the time series data and building them into a matrix form $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, the superscript of which represents m time points and n dimensions. Numerical methods (e.g., the finite difference method) are then applied to calculate the discrete derivatives $\boldsymbol{X}_t \in \mathbb{R}^{m\times n}$ as an estimation of $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X}(t)$. The third step of the EQL pipeline requires the construction of a nonlinear library $\boldsymbol{\Theta}(\boldsymbol{F}, \boldsymbol{Q}) \in \mathbb{R}^{m\times(N+N_q)}$ as the candidates of explanatory variables for the original data and their time derivatives. Each column of the library matrix corresponds to a specific candidate term in the right hand side of the governing equation. The column vector $\boldsymbol{Q} \in \mathbb{R}^{m\times N_q}$ contains additional terms such as the known constants or other external interacting variables. If we assume that $\boldsymbol{\Theta}(\boldsymbol{F}, \boldsymbol{Q})$ is an over-complete library, for example, the ODE evolution can be expressed with a sparse vector of coefficients $\xi \in \mathbb{R}^{(N+N_q)\times n}$ as follows

$$\boldsymbol{X}_t = \boldsymbol{\Theta}(\boldsymbol{F}, \boldsymbol{Q})\xi \tag{2}$$

Each nonzero entry of the sparse vector ξ_{ij} corresponds to the existence of term represented by the ith column of $\boldsymbol{\Theta}$ in the right hand side of the differential equation $\frac{\mathrm{d}}{\mathrm{d}t}x_j(t)$. The least absolute shrinkage and selection operator (LASSO) regression or other sparse regression methods are commonly applied to Eq. (2) for determining the active nonlinearities.

2.2 SINDy with Physics-informed GAN

Physics supervision is integrated into the adversarial learning framework. Specifically, the physics residuals are used to compute a physics consistency score η for each prediction, indicating the likelihood of the prediction being physically consistent. The physics consistency scores are incorporated into the discriminator as additional inputs, such that the discriminator not only distinguishes between real and fake samples like the original GAN architecture but also using the physics supervision provided by the consistency scores. The physics consistency scores of a prediction $\hat{y}$ regarding to the $k-$th physical constraint is defined using the following equation:

$$\eta_k = \exp\left(-\lambda||R^{(k)}(x, \hat{y})||^2\right) \tag{3}$$

where $R^{(k)}(x, \hat{y})$ is the physics residual calculated by the $k-$th ODE/PDE given x and the predicted value $\hat{y}$. λ is a hyperparameter that controls the weight of the physics consistency term.

To increase the training volume and balance the data distribution, the physics-informed GAN allows the use of unlabeled data instances for training both the generator and discriminator models. In this context, labeled data consists of pairs (x_{u_i}, y_{u_i}) for $i = 1, 2, \cdots, N_u$, where the input-output values are known in advance. Unlabeled data, on the other hand, consists of input points x_{f_j} sampled from the domain of the dynamic system being investigated, with unknown corresponding output values. The training objective of physics-informed GANs is modified accordingly as follows:

$$Loss_{\mathscr{G}} = \frac{1}{N_u}\sum_{i=1}^{N_u}\mathscr{D}(x_{u_i}, \hat{y}_{u_i}, \eta_{u_i}) + \frac{1}{N_f}\sum_{j=1}^{N_f}\mathscr{D}(x_{f_j}, \hat{y}_{f_j}, \eta_{f_j}) \tag{4}$$

$$
\begin{aligned}
Loss_{\mathscr{D}} = &- \frac{1}{N_u}\sum_{i=1}^{N_u}\log(\mathscr{D}(x_{u_i}, \hat{y}_{u_i}, \eta_{u_i})) - \frac{1}{N_u}\sum_{i=1}^{N_u}\log(1-\mathscr{D}(x_{u_i}, y_{u_i}, 1)) \\
&- \frac{1}{N_f}\sum_{j=1}^{N_f}\log(\mathscr{D}(x_{f_j}, \hat{y}_{f_j}, \eta_{f_j})) - \frac{1}{N_f}\sum_{j=1}^{N_f}\log(1-\mathscr{D}(x_{f_j}, \hat{y}_{f_j}, 1))
\end{aligned} \tag{5}
$$

where $\hat{y}_{u_i} = \mathscr{G}(x_{u_i}, z_{u_i}), \hat{y}_{f_j} = \mathscr{G}(x_{f_j}, z_{f_j})$, z_{u_i} and z_{f_j} are the sampled noise.

Our proposed framework SPIGAN (SINDY with physics-informed GAN) is based on physics-informed GAN but without the reliance on prior knowledge. In the SPIGAN architecture, the physics residuals are not calculated based on the pre-defined but instead obtained through the SINDy algorithm, resulting the discriminator in SPIGAN to learn the following mapping.

$$
\mathscr{D} : \mathscr{D}(x, y, \eta_{\text{SINDy}}) \rightarrow \Omega \in [0, 1] \tag{6}
$$

3 Results

In our experiments, we deliberately create scarce data by sampling the velocity variable sparsely in the spatial and temporal domains for the Burgers' equation to highlight the challenges faced by the vanilla SINDy approach. Burgers' equation is a fundamental partial differential equation and convection–diffusion equation. In its well-known form, we can write it as follows:

$$
\begin{aligned}
&\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2}, \quad 0 < x < L, \quad 0 < t < T \\
&u(x, 0) = \phi(x), \quad 0 < x < L, \\
&u(0, t) = \zeta_1(t), u(L, t) = \zeta_2(t), \quad 0 < t < T
\end{aligned} \tag{7}
$$

where u, x, t and ν are the velocity, spatial coordinate, time and kinematic viscosity, respectively. $\zeta_1(t), \zeta_2(t)$ and $\phi(x)$ are prescribed boundary conditions depending upon the specific conditions for the problem to be solved. We consider the Burgers' equation under two conditions. In Case 1, we set the Burgers' equation with a small viscosity parameter. The initial condition is set as a sine wave, which leads to the formation of two transmitted waves that meet in the middle of the region ($x = 0.5L$). In Case 2, the initial condition is an activation function. This case demonstrates the behavior of the Burgers' equation in the presence of diffusion, where the velocity field diffuses and smoothens over time. In both of the two cases the boundary condition is set to 0. The dynamics of the two cases are shown in Figure 1, where the color plot represents the velocity $u(x, t)$ at different spatial positions x and time steps t. To simulate the problem of scarce data, we randomly place several sensors at typical time intervals and select $p = 10\%$ of data points from the full data volume, resulting in limited observations of the system's behavior.

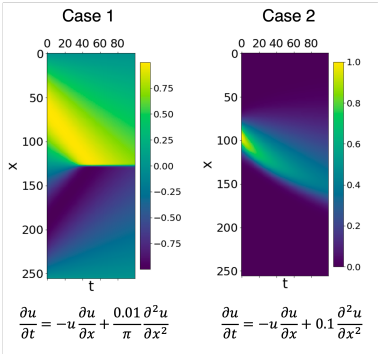

Figure 1: Visualization of ground truth dynamics of Burgers' equation.

3.1 Parameter Settings

The network $\mathscr{G} : \mathscr{G}(x, t, z) \rightarrow u$ is a network with 4 hidden layers with 50 neurons each. The network $\mathscr{D} : \mathscr{D}(x, t, u, \eta) \rightarrow \Omega \in [0, 1]$ is a network with 2 hidden layers with 50 neurons each. To infer the PDE solutions, for every discriminator update, the generator is updated 5 times for all of our SPIGAN. The batch size is set to 256. λ used in Eq. 3 is set to 0.05. We train the baseline models for 20000 epochs, which is common practice in the existing literature. For each training, we sample $N_f = 10000$ unlabeled points uniformly across the input space using latin hypercube sampling (LHS)[8]. This provides a diverse set of unlabeled points to help stabilize the training process. We use the Adam optimizer with a learning rate of 0.0001. The library used in SINDy process is also built based on second order derivatives $\{u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, u\frac{\partial u}{\partial x}, u\frac{\partial^2 u}{\partial x^2}\}$. Additionally, before starting the training process, we standardize the spatio-temporal coordinates of the input data based on the sampled unlabeled points. This standardization helps to balance the range of input data and improve the training performance.

3.2 Discover Burger's Equation under Default Parameter Settings

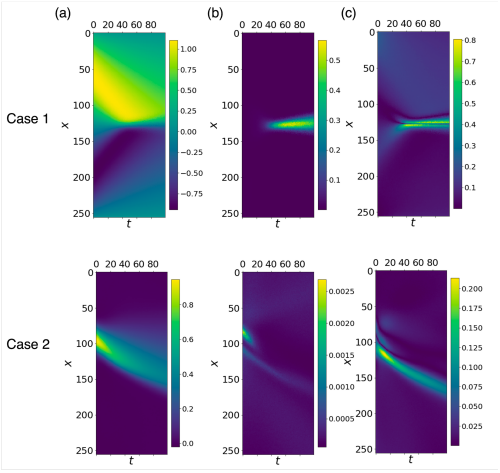

Figure 2: Visualization of (a) generator output, (b) variance and (c) absolute error of SPIGAN for Burgers' Equation under default parameter setting in both two cases.

In the test phase of the SPIGAN framework, we evaluate the performance of the generator by comparing the generated output values with the ground truth values at the coordinates of unlabeled points. In Figure 2(a), we provide visualizations of the generator's output after training the SPIGAN model on the Burgers' equation data. The output is averaged over 50 runs to reduce the influence of noise sampling, and the variance is also shown in Figure 2(b) to indicate the uncertainty of the predictions. We can see that the generator captures the system dynamics of both Case 1 and Case 2 to a large extent. Figure 2(c) illustrates the absolute error within the domain. The final relative L^2-error of the predictions achieved by the SPIGAN model is 0.212 for Case 1 and 0.096 for Case 2. These values are close to those reported in previous GAN-based neural network studies (0.215 for Case 1, no result for Case 2). Though in our setting the volume of labeled data used is much larger than the previous study, our approach requires less reliance on prior knowledge or predefined physics constraints. Similarly, SPIGAN can be used in cases in cases where sensors are placed randomly at several locations. This scenario mimics situations where data scarcity occurs in the spatial domain. The SPIGAN framework preserves the ability to fit and predict within the domain, demonstrating its effectiveness in handling scarce data scenarios.

It should be noticed that, however, the selection of labeled data will affect the performance to some extent. As for the finally inferred equation, we analyze the coefficients of the inferred equation of SPIGAN and compare them with the ground truth equation and the SINDy results in both Case 1 and Case 2. To demonstrate the effect of random sampling of labeled data, the experiment is repeated three times with different selections of labeled data. It can be observed that when the SINDy method is applied to the scarce data, the inferred equations deviate greatly from the ground truth. As the data becomes extremely limited, the coefficients of the inferred equations fluctuate greatly depending on the sampled data, indicating the instability of finite differentiation when facing unequal intervals. Typically, as Burgers' equation contains a second-order derivative in the spatial domain, when data is collected at random locations, the results will be even worse. As Figure 3 shows, at the same time, although there may be some deviation in the coefficients for certain terms, SPIGAN demonstrates better stability in its results. The coefficients of the inferred equation from the three trials generally align with each other, which is a huge advantage over the vanilla SINDy.

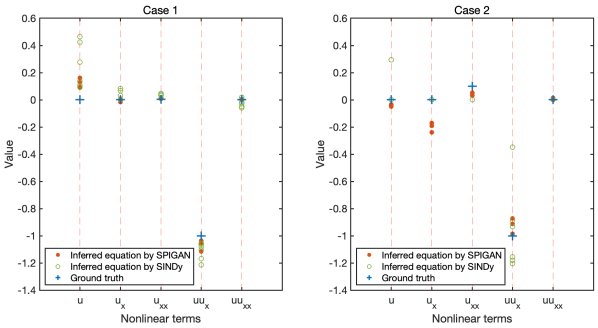

Figure 3: Coefficients of inferred equation of SPIGAN for Burgers' equation.

4 Conclusion

Due to the high costs associated with data collection and limitation of facilities, it is often the case that the collected data will be extremely scarce. Vanilla SINDy will not perform well in these situations. Therefore, we have introduced SPIGAN, a novel GAN-based learning method for discovering physical laws from scarce data. SPIGAN provides an end-to-end solution for Equation Learning (EQL) that integrates data-driven and physics-informed approaches. The experiments on Burger's equation with different boundary conditions have demonstrated the effectiveness of SPIGAN. The generator in SPIGAN is able to recover the system dynamics using only 10% of the data required by vanilla SINDy. The inferred form of the governing equation(s) shows good consistency, albeit with some deviations from the ground truth.

Acknowledgments

This study was supported by the World-leading Innovative Graduate Study Program in Proactive Environmental Studies (WINGS-PES), The University of Tokyo.

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] Markus Quade, Markus Abel, J Nathan Kutz, and Steven L Brunton. Sparse identification of nonlinear dynamics for rapid model recovery. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):063116, 2018.

[3] Jean-Christophe Loiseau and Steven L Brunton. Constrained sparse galerkin regression. *Journal of Fluid Mechanics*, 838:42–67, 2018.

[4] Jean-Christophe Loiseau, Bernd R Noack, and Steven L Brunton. Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *Journal of Fluid Mechanics*, 844:459–490, 2018.

[5] Moritz Hoffmann, Christoph Fröhner, and Frank Noé. Reactive sindy: Discovering governing reactions from concentration data. *The Journal of chemical physics*, 150(2):025101, 2019.

[6] Bhavana Bhadriraju, Abhinav Narasingam, and Joseph Sang-Il Kwon. Machine learning-based adaptive model identification of systems: Application to a chemical process. *Chemical Engineering Research and Design*, 152:372–383, 2019.

[7] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature communications*, 12(1):6136, 2021.

[8] Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.