
Physics-informed Generative Adversarial Networks for Sequence Generation with Limited Data

Anonymous Author(s)

Abstract

We consider the problem of sequence generation with limited data. We use physics informed generative adversarial networks (PI-GAN) to tackle this problem. PI-GAN integrates a transition function module in the generator part that can iteratively construct the sequence with only one initial point as the input. The model is fully auto-regressive, with the predictive distribution of each point conditioned on the previous points. When applied to real-world scenarios, we show that our model outperforms various baselines.

1 Introduction

Sequential data widely exists in the real world, such as the trajectories of animals, the spread of a disease, or physical kinetics data. The sequence generation problem can be formulated as follows. Let $\{s_i\}_{i=1}^N$ be a set of N training examples. Each example is a sequence where $s_i = \langle s_{i,1}, s_{i,2}, \dots, s_{i,T_i} \rangle$. The goal is that given an initial point $s_{j,1}$, the model could generate the whole sequence x_j as $s_j = \langle s_{j,1}, s_{j,2}, \dots, s_{j,T_j} \rangle$. For example, as shown in Fig 1, in a damped pendulum system, variables q and p changes with time t . The model aims to predict the sequence of $\langle q, p \rangle$, given only an initial data point.

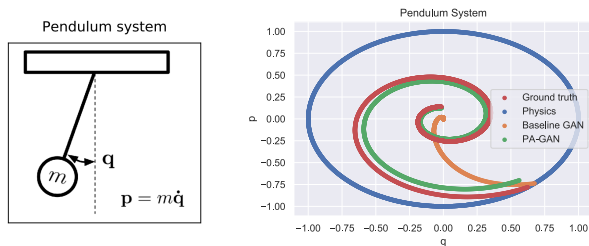


Figure 1: An example of sequence generation in a pendulum system. The variables q and p correspond to position and momentum coordinates. The physics-based model (blue) represents an ideal friction-free curve of (q, p) . However, as there exist frictions, the ground truth (red) shows a spiral shape. By comparison, the PI-GAN (green) learns to predict a more accurate sequence of (q, p) than the baseline model (orange).

Sequence data generation has received much research attention [11]. On one hand, many different neural models have been proposed, including variational autoencoder [12], recurrent neural nets [10, 32, 16, 28], and Generative Adversarial Network (GAN) [8]. On the other hand, these models are used to generate different categories of sequences, including images [12], text [11], music audios [23, 8], and markov chains [26, 14]. In this paper, we focus on generating a special type of sequence data, scientific-process data. Different from the previously mentioned sequence data types, these data are usually generated according to a backend physical generative process (e.g., physical laws). Hence, instead of using pure data-driven models as in previous studies, we propose to incorporate the physics model into the modeling process.

Recently, there are an increasing number of studies trying to incorporate physics into machine learning models. These studies can be categorized as below. (1) **Physics-constraint machine learning.** Some of them [24, 19, 18, 27] use the physics-based prior knowledge as regularization or constraints. Other works [9, 7, 21] use the physics constraint to filter out some data samples according to the scientific knowledge. The main drawback of these works is that physical laws is not always aligned with real-world observations. Different from these methods, our proposed model assumes there are two independent components in the state transition, observed scientific process and unobserved stochastic process. Hence, our proposed model utilize two separate cells (transition cell and scientific model cell) to model them correspondingly. (2) **Physics-informed machine learning.** These works improve the learning process more generally and efficiently. [2, 22, 3, 29, 20, 17] design hybrid-models, which concatenate or stack the data-driven models and scientific models together to map from the input to the output. [5, 30] design structured interactive graph neural networks using the relational inductive bias to reason about the relationships between objects in physical systems. Their applications are limited to graph structured problems. Physics informed neural networks (PINN) [25, 31] incorporate the physical laws in the loss function of the neural nets. However, PINN require the prior information exists employing known governing laws expressed by partial differential equations. Different from these hybrid-models, our model uses a GAN-like structure, which will push the generator to learn a transition model that satisfies both the data and the scientific model.

In this paper, we propose to incorporate physics into data-driven models to tackle the sequence generation problem with limited data. Motivate by the adversarial training process of GAN, we propose to encode physics priors into GAN framework. By constructing an additional physics-based discriminator upon the original GAN framework, we impose a weak supervision signal of physics prior into the model. Moreover, to iteratively generate sequences, we explicitly model the transition function inside the generator. The neural networks parameterized transition function takes the current state as input and predicts the next state. The model can generate a whole sequence by iteratively constructing the samples. PI-GAN forms a novel adversarial framework with incorporated prior knowledge. We demonstrate the effectiveness of our model on a variety of tasks.

2 Method

Problem 1 (Physics-Informed Sequence Generation) *Given N training examples $\{s_i\}_{i=1}^N$, where $s_i = \langle s_{i,1}, s_{i,2}, \dots, s_{i,T_i} \rangle$. The goal is to build a model y that given any initial point $s_{j,1}$, we could iteratively construct the whole sequence s_j as $s_j = \langle s_{j,1}, s_{j,2}, \dots, s_{j,T_j} \rangle$. In addition, we could utilize a physics-based prior knowledge model $T_\rho(\cdot|s)$.*

Generator. As shown in Fig 2a, given a sequence of training samples, $s_t, s_{t+1}, \dots, s_{t+n}$, we formulate the training process as the setting of time series prediction. Given previous ground-truth data points, e.g. s_t , the generator learn to output the state after m steps, \hat{s}_{t+m}^h . We first use a neural network to parameterize the transition function as $T_\theta(\cdot|s)$. We assume that T_θ is relatively easy to sample from and it is a valid transition kernel for any choice of θ , i.e., it satisfies $\int_S T_\theta(s_{t+1}|s_t) ds_{t+1} = 1$. By iteratively applying T_θ for m steps, we could get \hat{s}_{t+m}^d . The transition kernel function is parameterized by a 2-layer MLP. The generator outputs \hat{s}_{t+m}^h as the prediction of s_t after m steps.

Discriminator-1. The discriminator-1 $D_1(x)$ distinguishes the generated samples and the samples given by physics-based model estimations. As is shown in Fig 2a, the discriminator-1 takes input as two pairs: (s_t, \hat{s}_{t+m}^h) as the generated data pair, and (s_t, \hat{s}_{t+m}^p) as the physics-based data pair. Discriminator-1 pushes the generator learns the physics distribution.

Discriminator-2. The discriminator-2 $D_2(x)$ distinguishes the generated samples from real data distribution. As is shown in Fig 2a, the discriminator-2 takes input as two pairs: (s_t, \hat{s}_{t+m}^h) as the generated pair, and (s_t, s_{t+m}) as the real one. Discriminator-2 pushes the generator learns the real distribution.

2.1 Objective Function

In the original *minmax* game setting, training GAN resembles minimizing Jensen-Shannon divergence between the true data distribution P and the approximation model distribution Q . In addition, we can generalize the objective to a whole family of divergences, parameterized by a probability

$0 < \pi < 1$.

$$JS_{\pi}[P|Q] = \pi KL[P|\pi P + (1 - \pi)Q] + (1 - \pi)KL[Q|\pi P + (1 - \pi)Q] \quad (1)$$

The original GAN is trained on balanced samples from the generator and the real data with the probability $\pi = \frac{1}{2}$. By adjusting the sampling probability from the data and generator, we could approximately optimize towards the generalized objective function, Eq. 1.

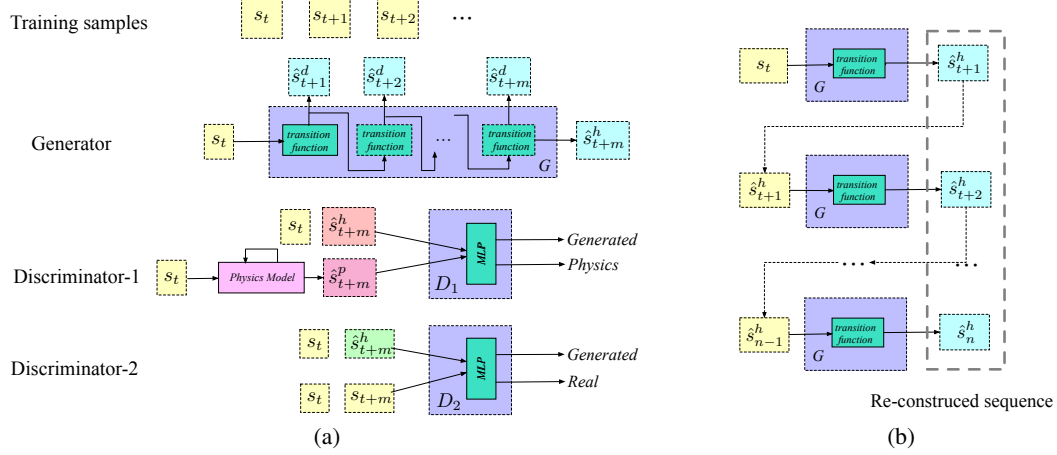


Figure 2: (a) PI-GAN framework. Given training samples s_t, s_{t+1}, \dots , we train the model as follows. First, given s_t , the Generator would generate \hat{s}_{t+m}^h , where m denotes the step size. Then, for the discriminator-1, we use the physics-based model to predict \hat{s}_{t+m}^p after m steps. We feed the physics pair (s_t, \hat{s}_{t+m}^p) and the generated pair (s_t, \hat{s}_{t+m}^h) to the discriminator-1. For the discriminator-2, we feed the fake pair with the real pair (s_t, s_{t+m}) . We iteratively update the generator, discriminator-1, and discriminator-2. (b) Illustration of the testing process. Given one initial start point s_t , the generator iteratively generate the whole sequence

PI-GAN objective. Similarly, the approximated training objective of PI-GAN can be derived as follows,

$$\begin{aligned} JSD[P_1|Q, P_2|Q] &= \pi[P_1|\pi P_1 + \gamma P_2 + (1 - \pi - \gamma)Q] \\ &\quad + \gamma[P_2|\pi P_1 + \gamma P_2 + (1 - \pi - \gamma)Q] \\ &\quad + (1 - \pi - \gamma)[Q|\pi P_1 + \gamma P_2 + (1 - \pi - \gamma)Q], \end{aligned} \quad (2)$$

where P_1 is the real data distribution, P_2 is the physics distribution, and Q is the approximation model distribution. We sample from the discriminator-1 with probability γ , from the discriminator-2 with probability π , and from the generator with probability $(1 - \pi - \gamma)$.

Initially, we set $\gamma = \pi = \frac{1}{3}$, i.e., sampling from the P_1, P_2 , and Q with probability $\frac{1}{3}$. Gradually, γ decays to zero and the π increases to $\frac{1}{2}$. In this way, we first train the model utilizing the physics-based knowledge. In the end, we update the model towards the true data distribution.

2.2 Training and Testing

Different from classical GAN training and testing procedure, in this section, we would like to elaborate on the modifications of the training and testing of PI-GAN as a sequence generation model. For training, we feed all s_t as ground-truth values. The model learns to predict each of the s_{t+m} accurately given s_t . However, during the testing process, we adopt a sequence generation manner. We only feed the s_0 (the start point) to the model and let the model iteratively construct the whole sequence. Then we evaluate the re-constructed chain with the ground truth. Note the testing process is different from the traditional time series prediction paradigm.

2.3 Model Generalizability

The only required condition in our proposed model is that there is a physics-based model which is in the form of a transition function, i.e. $T_p(\cdot|s)$. Hence, our proposed model could easily adapt

to different tasks. We also show the effectiveness of our proposed model in different domain in section 3.

3 Experiments

3.1 Domains and Datasets

- **Pendulum** [13]. This dataset describes the Hamiltonian dynamics of a pendulum system. The physics-based model is as follows:

$$H = 2mgl(1 - \cos q) + \frac{l^2 p^2}{2m} \tag{3}$$

- **Mass-spring** [13] This dataset describes the Hamiltonian dynamics of a spring system. The analytical equation for this spring system is used as the domain knowledge model, as below:

$$H = 2mgl(1 - \cos q) + \frac{l^2 p^2}{2m} \tag{4}$$

- **Ebola** [1] This dataset contains the total number of probable, confirmed and suspected Ebola cases and deaths in South Africa. The underlined physics-based model is the SIR disease model¹. SIR shows the change of infection rate over time.

3.2 Compared Methods

We compare PI-GAN with the following baselines: (1) **Physics-based Model**: For each of the task, we use the corresponding physics-based model; (2) **LSTM** [4]: Following the recent success of recurrent neural networks, LSTM model has been used to generate the sequences; (3) **GRU** [6]: Recently proposed gated recurrent unit (GRU) has been applied successfully in many domains; and (4) **GAN** [15]: GAN has been one of the most successful generative model in the field. We adopt the classical vanilla GAN achitecture.

Table 1: Overall results. We use the rooted mean squared error (RMSE) as our metric.

	Physics-based Model	LSTM	GRU	GAN	PI-GAN
Pendulum	0.8124	0.4073	0.4079	0.254	0.0714
Spring	0.7801	0.3575	0.348	0.2762	0.189
Ebola	516.2698	961.433	1005.9346	573.91	223.89

3.3 Overall Performance

In order to verify the effectiveness of our proposed method, we conduct experiments on multiple domains and datasets. The results are shown in Table 1. As expected, our proposed method PI-GAN outperforms all the baselines on all datasets. In all these datasets, the deep learning based methods usually achieves better results than physics based method. Further, though other deep learning methods experience unstable results on different datasets, PI-GAN consistently performs the best.

4 Conclusion

To summarize, we proposed a novel framework of sequence generation with physics prior encoded GAN. In order to generate sequences, we explicitly model the transition function inside the generator, which is parameterized by a neural network. By utilizing the transition function, the generator is able to predict one state at a time. Ultimately, the generator is able to construct a whole sequence with complex structures. For the future work, we hope to extend the experiment setting to more complicated scenarios with more system dynamics. We would also like to explore beyond the scope of sequence generation.

¹https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology#The_SIR_model

References

- [1] Number of ebola cases and deaths in affected countries. <https://data.humdata.org/dataset/ebola-cases-2014>.
- [2] Anurag Ajay, Jiajun Wu, Nima Fazeli, Maria Bauza, Leslie P Kaelbling, Joshua B Tenenbaum, and Alberto Rodriguez. Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3066–3073. IEEE, 2018.
- [3] Anurag Ajay, Maria Bauza, Jiajun Wu, Nima Fazeli, Joshua B Tenenbaum, Alberto Rodriguez, and Leslie P Kaelbling. Combining physical simulators and object-based networks for control. *arXiv preprint arXiv:1904.06580*, 2019.
- [4] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.
- [5] Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pages 4502–4510, 2016.
- [6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [7] Stefano Curtarolo, Gus LW Hart, Marco Buongiorno Nardelli, Natalio Mingo, Stefano Sanvito, and Ohad Levy. The high-throughput highway to computational materials design. *Nature materials*, 12(3):191–201, 2013.
- [8] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.
- [9] Christopher C Fischer, Kevin J Tibbetts, Dane Morgan, and Gerbrand Ceder. Predicting crystal structure by merging data mining with quantum mechanics. *Nature materials*, 5(8):641, 2006.
- [10] Kratarth Goel, Raunaq Vohra, and JK Sahoo. Polyphonic music generation by modeling temporal dependencies using a rnn-dbn. In *International Conference on Artificial Neural Networks*, pages 217–224. Springer, 2014.
- [11] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.
- [12] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.
- [13] Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *arXiv preprint arXiv:1906.01563*, 2019.
- [14] Yi Hao, Alon Orlitsky, and Venkatadheeraj Pichapati. On learning markov chains. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, page 646–655, Red Hook, NY, USA, 2018. Curran Associates Inc.
- [15] Goodfellow Ian, Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherjil, Courville Aaron, and B Yoshua. Generative adversarial nets. *Advances in neural information processing systems*, 3, 2014.
- [16] Sathish Reddy Indurthi, Dinesh Raghu, Mitesh M Khapra, and Sachindra Joshi. Generating natural language question-answer pairs from a knowledge graph using a rnn based question generation model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 376–385, 2017.
- [17] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan Read, Jacob Zwart, Michael Steinbach, and Vipin Kumar. Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 558–566. SIAM, 2019.
- [18] Anuj Karpatne, Zhe Jiang, Ranga Raju Vatsavai, Shashi Shekhar, and Vipin Kumar. Monitoring land-cover changes: A machine-learning perspective. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):8–21, 2016.

- [19] Anuj Karpatne, Ankush Khandelwal, Xi Chen, Varun Mithal, James Faghmous, and Vipin Kumar. Global monitoring of inland water dynamics: State-of-the-art, challenges, and opportunities. In *Computational Sustainability*, pages 121–147. Springer, 2016.
- [20] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.
- [21] Ankush Khandelwal, Varun Mithal, and Vipin Kumar. Post classification label refinement using implicit ordering constraint among data instances. In *2015 IEEE International Conference on Data Mining*, pages 799–804. IEEE, 2015.
- [22] Ning Liu, Rui Ma, Yue Wang, and Lin Zhang. Inferring fine-grained air pollution map via a spatiotemporal super-resolution scheme. In *Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pages 498–504. ACM, 2019.
- [23] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [24] Jian Pei and Jiawei Han. Constrained frequent pattern mining: a pattern-growth view. *ACM SIGKDD Explorations Newsletter*, 4(1):31–39, 2002.
- [25] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [26] Jiaming Song, Shengjia Zhao, and Stefano Ermon. Generative adversarial learning of markov chains. 2017.
- [27] Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [28] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1017–1024, 2011.
- [29] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data. *Physical Review Fluids*, 2(3):034603, 2017.
- [30] Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video. In *Advances in neural information processing systems*, pages 4539–4547, 2017.
- [31] Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *arXiv preprint arXiv:2003.06097*, 2020.
- [32] Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. Generating chinese classical poems with rnn encoder-decoder. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 211–223. Springer, 2017.