Knowledge-based Residual Learning

Guanjie Zheng[‡*], Chang Liu[‡], Hua Wei[†], Porter Jenkins[†] Chacha Chen[†], Tao Wen[⋆], Zhenhui Li[†]

[‡]Shanghai Jiao Tong University, [†]The Pennsylvania State University, [⋆]Syracuse University

{gjzheng,only-changer}@sjtu.edu.cn, {hzw77,prj3,cjc6647,jessieli}@psu.edu, twen08@syr.edu

Abstract

Small data has been a barrier for many machine learning tasks, especially when applied in scientific domains. Fortunately, we can utilize domain knowledge to make up the lack of data. Hence, in this paper, we propose a hybrid model KRL that treats domain knowledge model as a weak learner and uses another neural net model to boost it. We prove that KRL is guaranteed to improve over pure domain knowledge model and pure neural net model under certain loss functions. Extensive experiments have shown the superior performance of KRL over baselines. In addition, several case studies have explained how the domain knowledge can assist the prediction.

1 Introduction

Training with *small* number of data samples has always been one of the biggest challenge in machine learning. In many scientific domains, such as physics and environmental science [Karpatne *et al.*, 2017b], sharing or obtaining data is often at a high cost. How to learn an accurate machine learning model from small data attracts many researchers.

In literature, people have tried to tackle this challenge in two major groups of approaches. The first group is *transferring knowledge from other machine learning models* to the target machine learning model. In this group, frequently used approaches include transfer learning [Pan and Yang, 2009], multi-task learning [Ranjan *et al.*, 2019], and meta-learning [Finn *et al.*, 2017]. This group of approaches require data from the source domain or from other tasks in order to train machine learning models. It does not work for the case where there is a only one task with limited training data.

Another group of methods *use domain expertise* to tackle small data challenge. Frequently-used approaches include curating data representations [Khandelwal *et al.*, 2015], changing loss functions [Karpatne *et al.*, 2016a], and combining domain knowledge model with machine learning model [Ajay *et al.*, 2019]. These approaches do not necessarily compete with each other and can be used at the same time.

[*]Partial work was done when Guanjie Zheng was at The Pennsylvania State University.

In this paper, we focus on how to combine domain models with the machine learning models. We assume that there is a known domain knowledge model $y = \rho(x)$ modeling the relation between features x and response variable y. We investigate this direction because from our collaborations with researcher from different disciplines such as environmental science and social science, we find that these researchers often know some simple model based on the scientific principles (e.g., physical laws or chemical reactions). Those domain knowledge models capture the correlations between features and target variable under the ideal assumptions. But the real world often violate the ideal assumptions, though the principles stay true. The actual models are often more complicated than the ones that domain experts have.

Different from the previously mentioned methods, we propose a surprisingly simple yet effective method to combine domain knowledge model and machine learning model. The key idea of our method is to treat the domain knowledge model as a weak learner and use the machine learning model (neural network is used) to learn the residual from the domain knowledge model. The idea is inspired by the boosting methods [Chen and Guestrin, 2016] and residual learning methods [He *et al.*, 2016]. The key difference here is that our weaker learner is not a machine learning model, but a domain knowledge model derived by domain principles.

We further prove that the proposed hybrid model, when choosing neural network as the machine learning model, is guaranteed to outperform a pure neural net model and a pure domain knowledge model. We have conducted comprehensive experiments on real datasets from a variety of domains and have shown the superior performance of the proposed hybrid residual model. Case studies have shown how this method can boost the performance and maintain important domain properties. It demonstrates that our proposed hybrid residual model can help researchers in other disciplines to tackle their small data challenge.

2 Related Work

Combining domain knowledge model and machine learning model has become increasingly popular recently [Karpatne *et al.*, 2017a; Wagner and Rondinelli, 2016].

The most intuitive way is to *use one component to help the other*. (1) [Chapelle and Li, 2011] use massive data to

calibrate the domain knowledge model. (2) Domain knowledge can also be used in data preprocessing and postprocessing [Khandelwal *et al.*, 2015]. However, these two categories of methods need a large amount of data for calibration or training. Hence, they do not apply to the small data problem mentioned before. In addition, they assume the domain model to be precise, which is usually not true.

Another idea is to *integrate domain knowledge and machine learning models*. (1) Many domain-specific machine learning model designs are proposed according to domain properties [Leibo *et al.*, 2017; Mikolov *et al.*, 2010], e.g., RNN structures [Mikolov *et al.*, 2010] for natural language processing according to the sequential property. These methods are incorporating the specific domain properties, rather than the domain knowledge model in our problem. (2) Domain knowledge can also be used to guide the learning process as model initialization [Schrodt *et al.*, 2015], regularization, priors or constraints [Karpatne *et al.*, 2016b; Karpatne *et al.*, 2016a]. These methods are usually sensitive to hyperparameter settings and performance of them vary among datasets. (3) Recently, several hybrid knowledge-data models are proposed [Ajay *et al.*, 2019; Jia *et al.*, 2019], such as using the output of one model (domain knowledge model or machine learning model) as features to the other model [Karpatne *et al.*, 2017b]. Different from all the three categories of methods, we use the domain knowledge model as a weak learner and use a machine learning model (we use neural net) to boost it. This guarantees the superior performance of our model over either pure neural net model or domain knowledge model. We have tested its performance on routing problem in our earlier work [Liu *et al.*, 2021].

Residual Learning and Boosting Our method is related to boosting theory and residual learning theory. So we briefly review these studies. Boosting methods [Chen and Guestrin, 2016] serve as the state-of-the-art for many applications before recent deep learning methods. The success of boosting methods is attributed to their capability in ensembling weak learners and reducing bias [Chawla *et al.*, 2003]. ResNet [He *et al.*, 2016] is one essential advancement of deep learning. It enables multiple paths from input to output and resolves the gradient vanishing problem. It significantly improves the performance of deep learning methods on supervised learning tasks, e.g., image classification [He *et al.*, 2016], and spatial-temporal prediction [Liu *et al.*, 2019]. Recent research has explored the connection between ResNet and boosting theory. [Huang *et al.*, 2017] have demonstrated that ResNet can be regarded as the ensemble of multiple paths from input to output, and has shown ResNet can be learned in a boosting way.

In contrast, we propose a hybrid framework, which incorporates domain knowledge in a residual learning way. The domain knowledge model serves as the skip connection in ResNet, or a weak learner in boosting theory. This connection guarantees the success of the proposed framework.

3 Problem Definition

Our problem definition follows typical regression and classification problem definition. Specifically, we are given a dataset of N data points. Each data point, represented as $(\mathbf{x}_i, y_i)$, is composed of a feature vector $\mathbf{x}_i \in R^d$ and a response value $y_i \in R$ (for regression) or $y_i \in \{0, 1, ..., K-1\}$ (for classification). Domain knowledge model is given as $y^D = \rho(\mathbf{x})$. Our problem can be formulated as follows:

Problem 1 (Domain-Aware-Prediction)**.** *Given a dataset* $\{(\mathbf{x}_1, y_1), (\mathbf{x}_1, y_2), ..., (\mathbf{x}_N, y_N)\}$*, and domain knowledge model* $y^D = \rho(\mathbf{x})$*, the goal is to build a prediction model* $y = \mathcal{F}(\mathbf{x}, \rho(\mathbf{x}))$*, so that the defined loss is minimized.*

4 Method

To tackle the aforementioned challenges of small data, we propose a **K**nowledge-based **R**esidual **L**earning method (KRL). This model is composed of domain knowledge model part and neural net part.

4.1 Combining Domain Knowledge Model and Machine Learning Model

Here, we show two intuitive combinations of domain knowledge and neural network models (as in Figure 1). Some studies [Wilby *et al.*, 1998] stack domain knowledge model output to a neural network to predict the response (Figure 1 (a)). This design enables fine-tuning towards the correct scaling over the domain knowledge model output. Some other studies [Sadowski *et al.*, 2016] preprocess the data by a neural network and feed to domain knowledge model (Figure 1 (b)). This helps de-noise the input and get better prediction y through domain knowledge model. These designs require the special form of the domain knowledge model, and lead to inferior performance when domain knowledge has flaw.

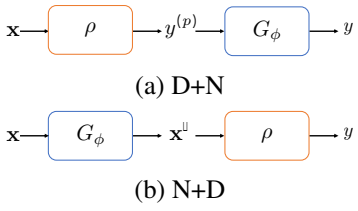


Figure 1: Two ways to combine domain knowledge and neural net.

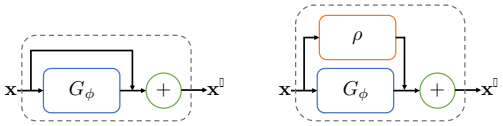


Figure 2: The residual learning unit of ResNet and KRL. G_ϕ: neural layers, ρ: domain knowledge model.

4.2 Residual Learning for Knowledge-Data Hybrid Model

Inspired by the observation that domain knowledge model can capture a rough mapping between feature and response, we propose a residual learning method to combine domain knowledge and neural net model. Generally, one classic residual unit can be defined as

$$\mathbf{x}^{(l)} = H(\mathbf{x}) + G_\phi(\mathbf{x}) \quad (1)$$

where G_ϕ is an arbitrary neural net, and $H(\mathbf{x})$ is a skip connection (e.g., a simpler neural layer). A fully-connected layer is usually added after the final residual unit to map $\mathbf{x}^0$ to y. By setting $H(\mathbf{x})$ as identity mapping, we have the widely-used form $\mathbf{x}^0 = \mathbf{x} + G_\phi(\mathbf{x})$ (as in Figure 2 (a)).

Studies have revealed two reasons that make ResNet successful: (1) Learning the residual value $\mathbf{y}^0 - \mathbf{y}$ is provable better than learning original value $\mathbf{y}$ when G_ϕ satisfies certain criteria and output layer is linear [Shamir, 2018]. (2) The shortcut link $\mathbf{x}$ creates $\mathcal{O}(2^n)$ paths from the input to the output when n residual layers stack together [Veit *et al.*, 2016]. This ensemble significantly increases its robustness and accuracy.

Hence, we propose a simple but effective residual unit with domain knowledge (as in Figure 2 (b))

$$\mathbf{x}^0 = \rho(\mathbf{x}) + G_\phi(\mathbf{x}) \tag{2}$$

where $\rho(\mathbf{x})$ is the domain knowledge model, $G_\phi(\mathbf{x})$ is a neural net model. Intuitively, the domain knowledge model can predict the data reasonably well. Hence, the neural net model will be directed to predict the residual $\mathbf{x}^0 - \rho(\mathbf{x})$. Moreover, theoretically, it is guaranteed to yield superior performance over either pure domain knowledge model or pure deep learning model (shown in the next section).

4.3 Performance Guarantee

Guarantee to outperform domain knowledge model Without loss of generality, we can assume the final output layer as

$$y = \mathbf{w}^T(\rho(\mathbf{x}) + G_\phi(\mathbf{x})) \tag{3}$$

where $\mathbf{x}$ is the output from the previous residual unit, $\mathbf{w}$ is linear weight vector, ρ is the domain knowledge model and G_ϕ is a neural net. Further, we can re-write Eq. (3) as follows by explicitly write out the last fully-connected layer $\mathbf{M}$

$$y = \mathbf{w}^T(\rho(\mathbf{x}) + \mathbf{M}G_\theta(\mathbf{x})) \tag{4}$$

Then, setting $\mathbf{M}$ equal to $\mathbf{0}$, the domain knowledge model is represented as

$$y = \mathbf{w}^T\rho(\mathbf{x}) \tag{5}$$

Next, we will prove that the hybrid model Eq. (4) will perform not worse than the domain knowledge model Eq. (5). Before that, we need to incorporate the following general version corollary from [Shamir, 2018] (this corollary can be proved in the exactly same fashion as the Corollary 1 in [Shamir, 2018] by replacing the identity mapping with more general version, a differentiable function $H(\mathbf{x})$).

Corollary 1. *Suppose we have a function defined as*

$$\Gamma_\psi(\mathbf{a}, \mathbf{B}) \doteq \Gamma(\mathbf{a}, \mathbf{B}, \psi) \doteq E_{\mathbf{x},y}\left[l\left(\mathbf{a}^T\left(H(\mathbf{x}) + \mathbf{B}G_\psi(\mathbf{x})\right), y\right)\right] \tag{6}$$

where l is the defined loss, $\mathbf{a}$, $\mathbf{B}$ are weight vector and matrix respectively, and ψ is the parameters of a neural network. Then, every local minimum of Γ satisfies

$$\Gamma(\mathbf{a}, \mathbf{B}, \psi) \leq \inf_{\mathbf{a}} \Gamma(\mathbf{a}, \mathbf{0}, \psi) \tag{7}$$

if the following two conditions are satisfied: (1) loss $l(\hat{y}, y)$ is twice differentiable and convex in $\hat{y}$; (2) $\Gamma_\psi(\mathbf{a}, \mathbf{B})$, $\bigtriangledown\Gamma_\psi(\mathbf{a}, \mathbf{B})$, and $\bigtriangledown^2\Gamma_\psi(\mathbf{a}, \mathbf{B})$ are Lipschitz continuous in $(\mathbf{a}, \mathbf{B})$.

Then, we can prove the following theorem (please see supplementary material for details).

Theorem 1. *When using squared loss (for regression) and exponential loss or logistic loss (for classification), every local optimum of hybrid model $y = \mathbf{w}^T(\rho(\mathbf{x}) + \mathbf{M}G_\theta(\mathbf{x}))$ will be not worse than domain model $y = \mathbf{w}^T\rho(\mathbf{x})$.*

Guarantee to outperform neural net model By further building connection with boosting theory [Huang *et al.*, 2017], we will prove that KRL outperforms pure neural net model. Here, we use binary classification as an example. For the convenience of derivation, we apply exponential loss following the convention in the boosting theory. This derivation can be extended to logistic loss. The derivation for regression is left to future work.

First, we need to incorporate a corollary about AdaBoost. When using the exponential loss function $L(y, f(x)) = exp(-yf(x))$, the objective of AdaBoost algorithm can be written as

$$(\beta_m, G_m) = \arg\min_{\beta, G} \sum_{i=1}^{N} \exp\left[-y_i\left(f_{m\ \ 1}(x_i) + \beta G(x_i)\right)\right] \tag{8}$$

where m is the index of the weak learner. Then, the following corollary has been proved in [Hastie *et al.*, 2009].

Corollary 2. *With exponential loss function, the AdaBoost algorithm is equivalent to forward stagewise additive modeling, and the classification error is guaranteed to decay when the number of base learners increase. The objective Eq.* (8) *can be achieved by solving the G_m and β_m sequentially.*

Thus, we can have the following corollary.

Corollary 3. *Given G_m, the classifier obtained by the following objective is guaranteed to reduce the loss compared with $f_{m\ \ 1}(x)$.*

$$(\beta_m) = \arg\min_{\beta} \sum_{i=1}^{N} \exp\left[-y_i\left(f_{m\ \ 1}(x_i) + \beta G(x_i)\right)\right] \tag{9}$$

Then, we can prove the following theorem (please see supplementary material for details).

Theorem 2. *The hybrid model $y = \mathbf{w}^T(\rho(\mathbf{x}) + G_\phi(\mathbf{x}))$ is guaranteed to reach a minimum not worse than the pure neural net model $y = \mathbf{w}_1^T G_\phi(\mathbf{x})$.*

5 Experiment

5.1 Experiment Settings

Datasets. We conducted experiments on the following domains, covering regression, classification and information retrieval (IR) problems. Details and citations of the datasets can be found in the supplementary materials.

- **Weather** [Brantley *et al.*, 2008]. This dataset documents the soil formation and weathering process of chemicals in soils. It contains 178 records at various depths. We use the soil depth from the ground surface and time to predict

Table 1: Overall performance comparison. "-" means the method does not apply to this dataset. ″ means the higher the better and # means the lower the better. Note that, some methods perform similar on certain datasets because of the float precision. We show the average results of 5 runs.

Method		Regression (RMSE↓)				Classification (Accuracy↑)	IR (F1↑)
		Weather	Radi	Pend	Spring	Loan	Routing
Domain	DOM	102.48	3.1	1.1	0.47	0.51	0.64
Machine learning	Ridge	368.56	3.99	0.72	0.38	-	-
	XGBoost	464.73	3.79	1.84	0.71	0.54	0.09
	LR	-	-	-	-	0.54	0.22
	NN	746.19	2.93	0.57	0.39	0.55	0.22
	ResNet	720.82	3.52	0.66	0.67	0.54	0.22
Hybrid	LfD	833.84	2.82	0.53	0.55	0.21	
	N+D	975.69	2.96	0.82	0.4	0.51	-
	D+N	113.07	2.91	0.86	0.53	0.55	0.64
	RF-D	104.36	2.45	0.56	0.4	0.55	0.53
	D-Cons	735.48	2.63	0.64	0.38	0.51	0.22
	KRL	**100.48**	**2.23**	**0.47**	**0.35**	**0.57**	**0.76**

chemical concentrations in the soil. For the domain knowledge model, a sigmoid-family analytical equation [Brantley *et al.*, 2008] describing the relation between chemical concentration and depth is utilized.

- **Radi** [TEPCO, 2019]. This dataset has 3,371 concentration samples of multiple radiative chemical analytes in the seawater near Fukushima Daiichi Nuclear Power Station since the nuclear disaster. We use the time as the feature to predict the remaining percentage of the concerned analyte. The radioactive decay rate equation [Bucknell, 2019] (as shown in 10) is included as the domain knowledge model.

$$N = N_0 e^{\lambda t} \tag{10}$$

Here, N and N_0 are the analyte concentration at time t and time $t = 0$ respectively, and λ is the decay constant.

- **Pend** [Greydanus *et al.*, 2019]. This dataset describes the Hamiltonian dynamics of a pendulum system. Hamiltonian mechanics defines coordinates (e.g., position, momentum) to describe the system. For instance, an ideal pendulum system can be described by the following equation [Greydanus *et al.*, 2019].

$$\mathcal{H} = 2mgl(1 - \cos q) + \frac{l^2 p^2}{2m} \tag{11}$$

Here, p, q are the coordinates (position and momentum correspondingly), and $\mathcal{H}$ is the Hamiltonian of this system. Additionally, m, l represent the mass and length of the pendulum respectively, and g is the gravity constant. Then, the system dynamics (w.r.t. time) can be described by the derivative of the coordinates. We use the coordinates to predict the changing rate of these coordinates w.r.t. time.

- **Spring** [Greydanus *et al.*, 2019]. This dataset describes the Hamiltonian dynamics of a spring system. Ideally, this system can be described by the following formula

$$\mathcal{H} = \frac{1}{2}kq^2 + \frac{p^2}{2m}, \tag{12}$$

where m is the mass of the spring, and k is the spring constant. Similar as in Pend, the two coordinates are used to predict the changing rate of the defined coordinates. The analytical equation for this spring system is used as the domain knowledge model.

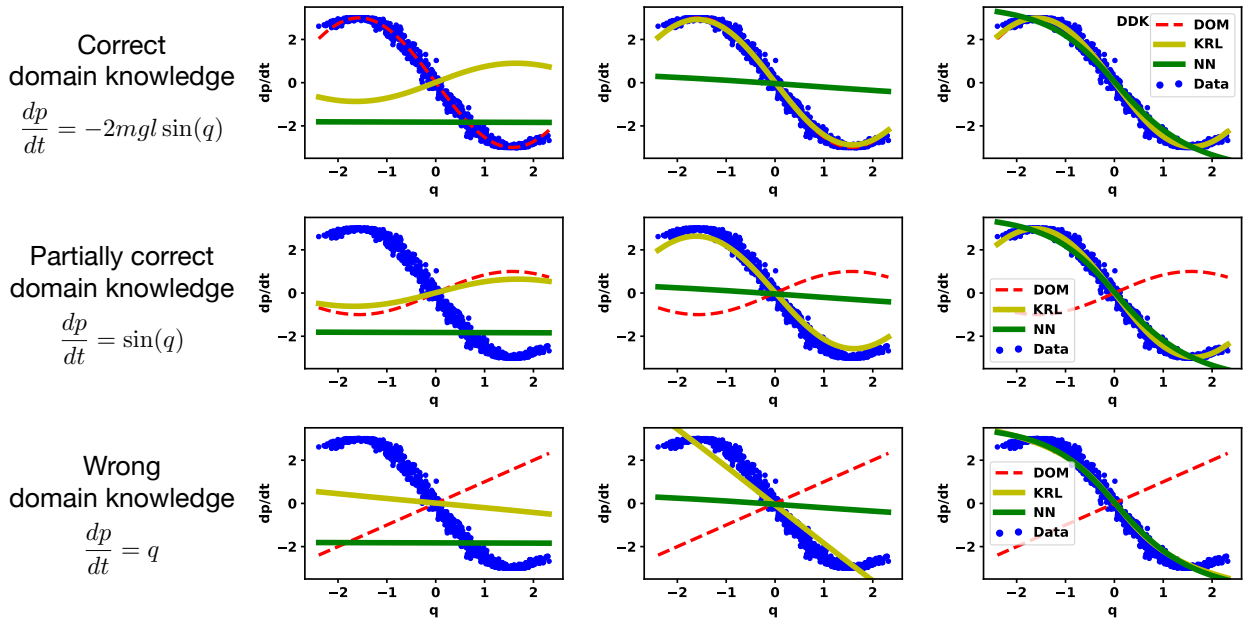
- **Routing** [Moreira-Matias *et al.*, 2013]. This dataset contains 7,734 pieces of taxi trajectories in Porto, Portugal from July 1, 2013 to June 30, 2014. We use the central business area (with 69 intersections) of Porto to build a weighted graph, and then use the origin, destination and departure time information to predict the routes of the vehicles (represented as a vector of 0 and 1 with each bit stand for one road segment). This problem is similar to an information retrieval problem. We use Dijkstra algorithm as the domain knowledge model to find the shortest path. For specific implementation details, please refer to this work [Liu *et al.*, 2021] we finished earlier.

- **Loan** [Kaggle, 2019]. This dataset contains 233,154 vehicle loan records. Each record contains features of the borrowers (e.g., employment status, credit history), loan amount, and whether default happened. We aim to predict the default (0 or 1) and utilize the exponential utility function [Kirkwood, 2002] as the domain knowledge. Here, x is a "variable that the economic decision-maker prefers more of" [Wikipedia, 2020], and a is the constant for risk preference.

5.2 Compared Methods

We compare with three groups of methods, domain models (DOM), machine learning models (LR: Logistic Regression, Ridge: Ridge Regression, XGBoost [Chen and Guestrin, 2016], NN: neural net, ResNet [He *et al.*, 2016]) and hybrid models as follows. By default, all neural net models are composed of 3 hidden layers with 32 neurons.

- LfD [Argall *et al.*, 2009]: Learning from demonstration. This method first generates some data from the domain



(a) epoch = 0 (b) epoch = 20 (c) epoch = last

Figure 3: Robustness of KRL w.r.t. domain knowledge models with different levels of correctness on Pend data. For this pendulum system, we have $\frac{dp}{dt} = \quad 2mgl\sin(q)$, where $m = 1$ (mass), $g = 3$ (gravity constant), $l = 1$ (length of the pendulum) (as in [Greydanus *et al.*, 2019]). We adopt different formulas as the domain knowledge models (correct formula, partially correct formula, and wrong formula to train KRL, and compare the trained model against NN and groundtruth data. The three columns describe the trained model at 0th epoch, 20th epoch, and the last epoch correspondingly. .

knowledge model and train a neural net model with both generated data and original data.

- D+N [Wilby *et al.*, 1998]: This method feeds the output from a domain knowledge model to a neural net. This allows fine-tuning of the scale of the output from domain knowledge model.
- N+D [Sadowski *et al.*, 2016]: This method first uses a neural net to map the raw features to the input of the domain knowledge model. The output from the domain knowledge model is the final prediction.
- RF-D [Wang *et al.*, 2017] utilizes a random forest based method to predict the discrepancy between the domain knowledge model output and the observation.
- D-Cons [Karpatne *et al.*, 2016b] uses domain knowledge as a constraint (i.e., an auxiliary loss) to the main loss.
- KRL: We propose a residual learning framework to combine the domain knowledge and the machine learning model. We name the method as KRL (**K**nowledge-based **R**esidual **L**earning).

5.3 RQ1: Overall Prediction Accuracy

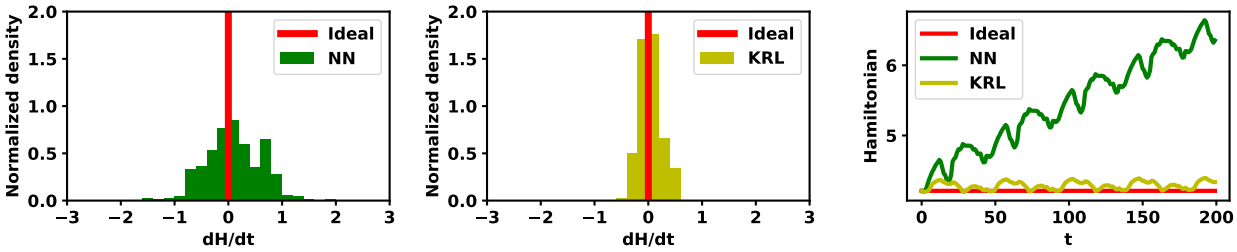
To verify the effectiveness of KRL, we conduct experiments on multiple datasets (results shown in Table 1). As expected, our proposed method KRL outperforms all the baselines on all datasets. Interestingly, the Hybrid group of methods usually achieves better results than other methods. Further, despite the unstable performance of other Hybrid methods, KRL consistently performs the best.

5.4 RQ2: How the Domain Knowledge Helps

Robust mimicking from domain knowledge model. We investigate a pendulum system [Greydanus *et al.*, 2019] with mass m and length l, and collect the data and model fitting results during the training process. This system is described by Hamiltonian Dynamics, where two variables p, q are defined to represent the position and the momentum of the pendulum respectively. Different models (KRL, NN) are built to capture the mapping between q(position change rate) and dp/dt(time).

When the correct domain knowledge is incorporated (first row in Figure 3), KRL can better predict the response and converge faster (at epoch 20). Specifically, KRL can learn the sine function relation but NN can not (even at the last epoch). *When a partially correct formula is provided* (second row in Figure 3), KRL can still learn the correct response. *Even with a totally wrong formula* (third row in Figure 3), KRL is able to abandon the wrong formula and converge to similar results as pure data-driven model NN. Thus, KRL can utilize the correct information in the domain knowledge model to converge to a better result faster, and avoid mistakes in the domain knowledge model.

Maintaining domain properties. Incorporating domain knowledge model also enables the learned model to maintain domain properties. Following the setting in the previous experiment, we show how the Hamiltonian changes over time in this system. The Hamiltonian of this system is supposed to keep constant (i.e., dH/dt always equals 0) because it is energy loss-free. As shown in Figure 4, in the results recov-



(a) NN (b) KRL (c) Hamiltonian v.s. Time

Figure 4: Illustration of Hamiltonian of the trained model on the Pend dataset. Since there is no energy loss, the Hamiltonian should remain constant when time evolves (i.e., the derivative of Hamiltonian w.r.t. time should equals 0). Figures (a) (b) show the histogram of the derivative of the Hamiltonian w.r.t. time recovered by the NN and KRL correspondingly. The derivative recovered by NN can vary within a range of -2 to 2, while the derivative recovered by KRL is within a much smaller range close to 0. In Figure (c), we let the trained model to control the system for 200 time steps. The Hamiltonian of the system controlled by NN keeps increasing, which violates the physical laws. In contrast, the Hamiltonian of the system controlled by KRL remains stable, close to the ideal case.

ered by NN (Figure 4 (a)) dH/dt can vary in a relatively large range, while in the results recovered by KRL (Figure 4 (b)) dH/dt remains close to 0. In addition, when using the predicted dp/dt and dq/dt to control the system (Figure 4 (c)), the system controlled by KRL successfully keeps the Hamiltonian stable though small variation is observed, while the system controlled by NN experiences an explosion of Hamiltonian, which is impossible in the real world.

6 Conclusion

In this paper, we propose to solve the limited data issue for real world problems by incorporating domain knowledge. We propose a model KRL, and prove that the proposed model has a guaranteed superior performance over the domain knowledge model and pure neural net. Extensive experiments are conducted and have shown the superior performance of KRL over baselines. We have also demonstrated the domain knowledge can help maintaining domain properties.

Our proposed model can be extended to various kinds of problems, and incorporated in different model designs. We believe it will be a general and essential progress in the application of machine learning models for real world problems where data is limited but the mechanisms are complex.

Acknowledgment

This work was supported in part by NSF awards #1652525, #1618448 and #1639150. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[Ajay *et al.*, 2019] 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.

[Argall *et al.*, 2009] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[Brantley *et al.*, 2008] Susan Louise Brantley, J Bandstra, J Moore, and AF White. Modelling chemical depletion profiles in regolith. *Geoderma*, 145(3-4):494–504, 2008.

[Bucknell, 2019] Bucknell. *System Dynamics - Time Constants*, 2019. https://web.archive.org/web/20060617205436/http://www.facstaff.bucknell.edu/mastascu/elessonshtml/SysDyn/SysDyn3TCBasic.htm.

[Chapelle and Li, 2011] Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257, 2011.

[Chawla *et al.*, 2003] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In *European conference on principles of data mining and knowledge discovery*, pages 107–119. Springer, 2003.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[Greydanus *et al.*, 2019] Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *arXiv preprint arXiv:1906.01563*, 2019.

[Hastie *et al.*, 2009] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Boosting and additive trees. In *The elements of statistical learning*, pages 337–387. Springer, 2009.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Huang *et al.*, 2017] Furong Huang, Jordan Ash, John Langford, and Robert Schapire. Learning deep resnet blocks

sequentially using boosting theory. *arXiv preprint arXiv:1706.04964*, 2017.

[Jia *et al.*, 2019] 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.

[Kaggle, 2019] Kaggle. *L&T Vehicle Loan Default Prediction*, 2019. https://www.kaggle.com/gauravdesurkar/lt-vehicle-loan-default-prediction.

[Karpatne *et al.*, 2016a] 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.

[Karpatne *et al.*, 2016b] 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.

[Karpatne *et al.*, 2017a] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, 2017.

[Karpatne *et al.*, 2017b] 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.

[Khandelwal *et al.*, 2015] 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.

[Kirkwood, 2002] Craig W Kirkwood. Decision tree primer. *available on-line at http://www. public. asu. edu/˜ kirkwood/DAStuff/decisiontrees/index. html*, 2002.

[Leibo *et al.*, 2017] Joel Z Leibo, Qianli Liao, Fabio Anselmi, Winrich A Freiwald, and Tomaso Poggio. View-tolerant face recognition and hebbian learning imply mirror-symmetric neural tuning to head orientation. *Current Biology*, 27(1):62–67, 2017.

[Liu *et al.*, 2019] 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.

[Liu *et al.*, 2021] Chang Liu, Guanjie Zheng, and Zhenhui Li. Learning to route via theory-guided residual network. *arXiv preprint arXiv:2105.08279*, 2021.

[Mikolov *et al.*, 2010] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[Moreira-Matias *et al.*, 2013] Luis Moreira-Matias, Joao Gama, Michel Ferreira, Joao Mendes-Moreira, and Luis Damas. Predicting taxi–passenger demand using streaming data. *TITS*, 2013.

[Pan and Yang, 2009] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[Ranjan *et al.*, 2019] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:121–135, 2019.

[Sadowski *et al.*, 2016] Peter Sadowski, David Fooshee, Niranjan Subrahmanya, and Pierre Baldi. Synergies between quantum mechanics and machine learning in reaction prediction. *Journal of chemical information and modeling*, 56(11):2125–2128, 2016.

[Schrodt *et al.*, 2015] Franziska Schrodt, Jens Kattge, Hanhuai Shan, Farideh Fazayeli, Julia Joswig, Arindam Banerjee, Markus Reichstein, Gerhard Bönisch, Sandra Díaz, John Dickie, et al. Bhpmf–a hierarchical b ayesian approach to gap-filling and trait prediction for macroecology and functional biogeography. *Global Ecology and Biogeography*, 24(12):1510–1521, 2015.

[Shamir, 2018] Ohad Shamir. Are resnets provably better than linear predictors? In *Advances in neural information processing systems*, pages 507–516, 2018.

[TEPCO, 2019] TEPCO. *Decommissioning Plan of Fukushima Daiichi Nuclear Power*, 2019. https://www4.tepco.co.jp/en/nu/fukushima-np/f1/smp/index-e.html#anchor01air.

[Veit *et al.*, 2016] Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in neural information processing systems*, pages 550–558, 2016.

[Wagner and Rondinelli, 2016] Nicholas Wagner and James M Rondinelli. Theory-guided machine learning in materials science. *Frontiers in Materials*, 3:28, 2016.

[Wang *et al.*, 2017] 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.

[Wikipedia, 2020] Wikipedia. *Exponential utility*, 2020. https://en.wikipedia.org/wiki/Exponential_utility.

[Wilby *et al.*, 1998] Robert L Wilby, TML Wigley, D Conway, PD Jones, BC Hewitson, J Main, and DS Wilks. Statistical downscaling of general circulation model output: A comparison of methods. *Water resources research*, 34(11):2995–3008, 1998.