



Theory-inspired machine learning—towards a synergy between knowledge and data

Johannes G. Hoffer¹ · Andreas B. Ofner¹ · Franz M. Rohrhofer¹ · Mario Lovrić^{1,2} · Roman Kern^{1,3} · Stefanie Lindstaedt^{1,3} · Bernhard C. Geiger^{1,4}

Received: 10 December 2021 / Accepted: 4 February 2022
© The Author(s) 2022

Abstract

Most engineering domains abound with models derived from first principles that have been proven to be effective for decades. These models are not only a valuable source of knowledge, but they also form the basis of simulations. The recent trend of digitization has complemented these models with data in all forms and variants, such as process monitoring time series, measured material characteristics, and stored production parameters. Theory-inspired machine learning combines the available models and data, reaping the benefits of established knowledge and the capabilities of modern, data-driven approaches. Compared to purely physics- or purely data-driven models, the models resulting from theory-inspired machine learning are often more accurate and less complex, extrapolate better, or allow faster model training or inference. In this short survey, we introduce and discuss several prominent approaches to theory-inspired machine learning and show how they were applied in the fields of welding, joining, additive manufacturing, and metal forming.

Keywords Theory-inspired machine learning · Theory-guided data science · Machine learning · Artificial intelligence · Welding · Joining · Additive manufacturing · Metal forming · Structural mechanics

1 Introduction

While early approaches to artificial intelligence (AI) were mostly rule-based and thus relied exclusively on expert knowledge, digitization and the advent of deep learning have triggered an era of purely data-driven modeling where the domain experts' knowledge appears to have lost its importance. Recently—since purely data-driven modeling is approaching its limits in some application domains—researchers have started to turn back to AI's roots to combine existing expert knowledge and data in new and promising ways. The scientific communities have realized that not only classical theory-driven models or simulations need to be augmented with available data from

measurements and digitization campaigns, but that also AI algorithms need to be adapted to incorporate knowledge from the respective application domains.

In this short survey, which expands on the Portevin Lecture given by the corresponding author at the 2021 International Conference of the International Institute of Welding (IIW), we will introduce and discuss different approaches of how such domain knowledge can be included in data-driven AI or machine learning models (Section 3). We will subsume these approaches under the umbrella of *theory-inspired machine learning*, contrasting it from *machine learning*, which predominantly refers to the process of obtaining models exclusively from data. Before presenting these approaches, we will highlight the main features, advantages, and limitations of purely theory-driven and purely data-driven models, respectively, and show that combining these two paradigms has the potential to improve the trade-offs between accuracy, computational complexity, and data requirements of the respective models (Section 2).

There exist several surveys covering theory-inspired machine learning, both general [1, 2] and domain-specific. Examples of the latter include surveys in turbulence modeling [3], computational fluid dynamics [4], civil engineering [5], chemical engineering [6], earth observation [7],

Johannes G. Hoffer, Andreas B. Ofner, and Franz M. Rohrhofer contributed equally to this work. The order is alphabetical.

Recommended for publication by Commission XIV - Education and Training

✉ Bernhard C. Geiger
geiger@ieee.org

Extended author information available on the last page of the article.

chemical, petroleum, and energy systems [8], material science [9], and heat transfer modeling [10]. We take inspiration from these surveys and structure our manuscript similarly as [1, 7, 9]. Specifically, we categorize approaches to theory-inspired machine learning based on how theory and data interact (e.g., theory selects model class, theory regularizes learning), rather than based on how theory- and data-driven models are connected (parallel, in series, subsystems, etc.).

The selection of presented approaches cannot be exhaustive and thus remains at least partially subjective. For one, we focus only on ways how existing theory can be utilized to improve data-driven models, namely via data preprocessing or feature engineering (Section 3.1), model selection (Section 3.2) and regularization (Section 3.3). We thus neglect information flowing in the opposite direction, i.e., we do not consider how theory-driven models can benefit from the increasing amounts of available data. As such, we do not cover data-driven parameterization of theory-driven models or defect modeling, in which data-driven models are used to compensate for overly coarse theoretical approximations. Further, we omit discussions about substituting only parts of a theory-driven model by a data-driven one. Rather, we consider these data-driven submodels as special cases of surrogate models, which we treat in Section 4. There is also a growing body of literature on the topic of *hybrid* or *grey-box models*, which contain theory- and data-driven components, the former often implemented via numerical solvers. While we do not discuss approaches that rely on numerical solvers as critical components, we argue that theory-inspired machine learning is a way of obtaining such hybrid models, for example, by utilizing a known functional relationship to preprocess the data prior to data-driven modeling. Finally, we briefly discuss settings in which prior knowledge is incomplete and may only encompass knowledge of cause-effect relationships (Section 5). Such settings recently received a lot of attention in the field of machine learning, and we believe that they can be put to good use in many application domains.

Our manuscript does not claim to be a complete treatment of the emerging topic of theory-inspired machine learning and hybrid modeling. Rather, it is intended as an introduction from which the interested reader can move forward. To assist the reader in this endeavor, the manuscript builds on several examples for theory-inspired machine learning from the fields of welding and joining, additive manufacturing, and metal forming. This simultaneously illustrates the presented approaches with practical applications and suggests how the existing literature can be categorized based on the concepts introduced in this survey.

2 Theory- vs. data-driven modeling

To discuss the fundamental differences between theory- and data-driven modeling, let us consider a simple physical phenomenon that we wish to study. The theory-driven model for this physical phenomenon may be the differential equation as depicted in Fig. 1. This differential equation is characterized by the nonlinear operator F and parameterized by a set of parameters, which we collect in the vector θ . We further assume that a forcing function $u(t)$ influences the phenomenon. We are interested in the trajectory of a quantity x describing this phenomenon. In other words, we are interested in solving the differential equation

$$\frac{dx(t)}{dt} = F(x(t); \theta) + u(t) \quad (1)$$

for a known initial condition $x(0)$ and for all t in a given time period \mathcal{T} , the computational domain.

The theory-driven nature of this model is characterized by the fact that it is deduced by a theoretical understanding of the phenomenon under investigation, i.e., F is derived from existing (physical) theories. It is an inherently causal model, in the sense that the forcing function causes changes in the quantity of interest and not vice-versa. However, not for every phenomenon the existing theory is sufficiently evolved, and even if it is, modeling all aspects of a phenomenon in their full details may be impractical or exhibit prohibitive computational complexity. Thus, often the true operator F is replaced by an approximation, highlighting the fundamental trade-off between accuracy and model complexity. Finally, in many cases the parameterization θ of the model is not deducible from existing theories.

At the other end of the spectrum are data-driven models (Fig. 2). Assuming that we wish to study the same physical phenomenon of interest, suppose that we have access to a large dataset \mathcal{D} of observations. Specifically, suppose we have observed the same phenomenon for (potentially) different parameters θ , different forcing functions $u(t)$, and different initial conditions $x(0)$, yielding different

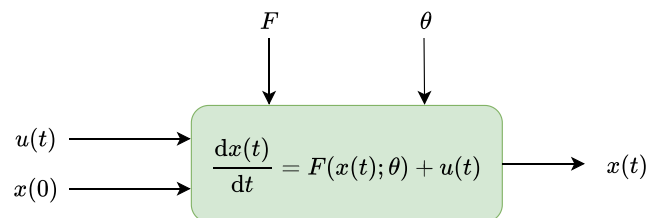
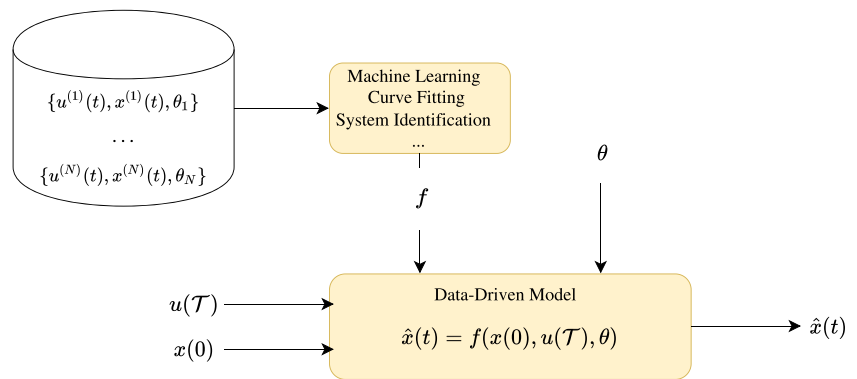


Fig. 1 A simple differential equation representing a theory-driven model

Fig. 2 A purely data-driven model relies on a set of training data and does not regard the data-generating process or its physical reality



trajectories $x(t)$ on (potentially) different computational domains \mathcal{T} . I.e., we have access to a dataset¹

$$\mathcal{D} = \{(\theta_i, x^{(i)}(0), u^{(i)}(t), x^{(i)}(t)), t \in \mathcal{T}_i\}_{i=1, \dots, N} \quad (2)$$

where i indexes the separate observations. Data-driven modeling now aims at learning a mapping between the elements influencing a quantity of interest (which are called *features* in machine learning) and the quantity of interest (which is called the *target*). In other words, we are interested in finding and/or parameterizing a function f such that

$$\hat{x}(t) = f(x(0), u(\mathcal{T}), \theta) \quad (3)$$

is close to $x(t)$ in some well-defined sense, where $x(t)$ is obtained by solving (1) and where $u(\mathcal{T})$ denotes the entire trajectory of the forcing function. In data-driven modeling, this task is often solved by minimizing a distance function between $x(t)$ and $\hat{x}(t)$ over the parameters ψ of the function f , where the distance is computed on the available (training) dataset \mathcal{D} :

$$\min_{\psi} \sum_{i=1}^N d\left(x^{(i)}(\mathcal{T}_i), f(x^{(i)}(0), u^{(i)}(\mathcal{T}_i), \theta_i; \psi)\right) \quad (4)$$

In (4), f is taken from a specific model class \mathcal{F} . For example, if f is a linear model, then ψ are its coefficients; if f is a neural network model, then ψ are its architectural parameters, weight matrices, and bias terms. Whether one refers to the process of determining model class \mathcal{F} and parameters ψ as machine learning, curve fitting, or system identification is immaterial, in all cases we refer to the resulting model as data-driven due to its dependence on \mathcal{D} .

The very nature of these data-driven models is that they model associative relationships rather than causative ones. Essentially, it is equally possible to parameterize a function \tilde{f} that maps the trajectory $x(t)$ and the parameter vector θ

to the forcing function $u(t)$ —although the accuracy of the solution to this inverse problem may be much smaller than for the forward problem, especially if the inverse problem does not allow a functional description. Furthermore, while theory-driven modeling is very structured, data-driven modeling is often a trial-and-error process, requiring testing several model classes or parameterizations in an iterative and exploratory manner. Furthermore, some model classes (such as neural networks) require large datasets \mathcal{D} to effectively learn their parameters ψ and, once learned, are considered black boxes lacking interpretability. Finally, data-driven models lack guarantees for physical consistency: If we select a parameterization θ far from the range covered in the dataset \mathcal{D} , then the solution $\hat{x}(t)$ provided by the data-driven model may not only be inaccurate, but even *unphysical* in the sense of violating fundamental physical laws. While the fact that data-driven models rarely extrapolate well outside of the range of training data is known as *lack of generalization* in the machine learning community, this shortcoming becomes much more severe when applying data-driven models in domains governed by physical laws.

These drawbacks of purely theory-driven and purely data-driven models call for action. Theory-inspired machine learning, hybrid or grey-box modeling, and theory-guided data science are umbrella terms for a variety of approaches to combine the benefits of theory- and data-driven modeling, mitigating their respective shortcomings. Data can be used to parameterize theory-driven models, to improve their accuracy by modeling their deficiencies, or to replace (parts of) theory-driven models for computational speedup. Insights from theory can help in selecting the model class for the data-driven model f or in preprocessing the data such that the parameters of f can be learned from less data. Finally, incorporating theory into data-driven models may guarantee (or at least improve) physical consistency and add inherent interpretability. Thus, combining the powers of theory- and data-driven models has the potential to achieve better trade-offs in terms of accuracy, computational complexity, the amounts of

¹Note that we require that all tuples $(\theta_i, x^{(i)}(0), u^{(i)}(t), x^{(i)}(t))$ in \mathcal{D} are distinct. However, we do not require that all elements of the tuple are distinct. For example, the dataset may comprise only a single parameterization $\theta_i = \theta$, but different initial conditions $x^{(i)}(0)$ and forcing functions $u^{(i)}(t)$.

required data, physical consistency, and interpretability, cf. [10, Fig. 3].

3 Approaches for theory-inspired machine learning

In the following sections, we will discuss several approaches to theory-inspired machine learning, i.e., to how domain knowledge can be used to improve data-driven models. For elaborations on how theory-driven models can benefit from data, we refer the reader to other surveys on this topic [1–8, 10].

3.1 Theory-inspired feature engineering

As mentioned in Section 2, data-driven models are obtained by minimizing a certain optimization objective, evaluated on a dataset \mathcal{D} , over the parameters ψ of a function f that should eventually model the relationship of interest, cf. (3). If we have prior knowledge about general properties of this relationship, we can utilize this knowledge to prepare the data such that the data-driven model can be learned more effectively (Fig. 3). For example, suppose that $x(t)$ depends in a highly nonlinear fashion on θ , while the dependence on $u(t)$ and $x(0)$ is much simpler. Now suppose further that we have knowledge about the nonlinear behaviour on θ . Then, rather than directly minimizing (4), one may turn to finding the parameters ψ of a function f by modelling

$$\hat{x}(t) = f(x(0), u(\mathcal{T}), g(\theta)) \quad (5)$$

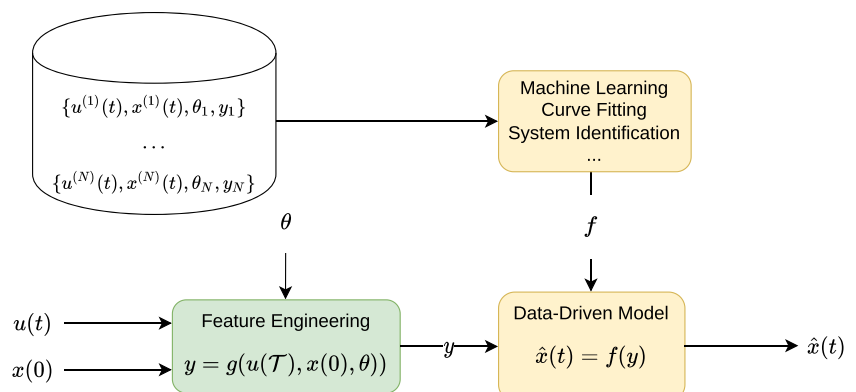
where the function g is chosen based on our knowledge about the nonlinear behavior. Capturing this nonlinear behaviour upfront allows us to choose a less complex model class (see also Section 3.2 below) and simultaneously eases the task of data-driven modeling.

Preprocessing data to simplify data-driven modeling is often referred to as *feature engineering*. While feature engineering also makes use of unsupervised techniques

such as dimensionality reduction or clustering, *theory-inspired* feature engineering utilizes domain knowledge to preprocess data. Both unsupervised and theory-inspired approaches to feature engineering are standard in traditional machine learning. However, the successes of deep learning rely to some extent on the capabilities of neural networks to learn their own features, allowing them to be applied without any pre- or postprocessing. While still successful, the resulting data-driven model is usually more complex than necessary and less interpretable than desired. To give a concrete example, the authors of [11] investigated the problem of clustering patterns in electronic end-of-line tests in the semiconductor industry. Patterns in these tests allow the engineer to detect deviations in the manufacturing process and to react accordingly. A convolutional variational auto-encoder (e.g., [12]) was designed to automatically extract features useful for subsequent pattern classification. Despite its satisfactory performance, the model remained a black box. Interpreting the tests as images, however, allowed the authors of [11] to utilize an interpretable set of features capturing well the structures that constitute the observed test patterns. After linear dimensionality reduction, the resulting features allowed a clustering performance comparable to that obtained from the convolutional variational auto-encoder, but with much lower complexity and much higher interpretability. As a second example, the authors of [13] aimed for a surrogate model (see Section 4) for the energy of carbon crystal structures. While the energy landscape is highly complex, the authors achieved excellent results by performing nonlinear regression based on physically meaningful features extracted from the crystal structures, such as average bond lengths, angular and radial density distributions, and the average number of nearest neighbors.

Theory-inspired features can also improve the generalization performance of machine learning models. For example, there is a class of neural networks that can be used to solve systems of partial differential equations on regular meshes (e.g., by approximating derivatives with

Fig. 3 Theory-inspired feature engineering. Theoretical insights into both the phenomenon under study and the selected class for the data-driven model and its learning algorithm can help preprocess the data accordingly



predefined, non-trainable convolutional filters). The authors of [14] used an elliptic transform as theory-inspired feature engineering, so that these methods can be applied to also to irregular domains. As a second example, the authors of [15] explored generalizable surrogate models for the structural analysis of 3D trusses (structures of connected triangles as in bridges). By using features that encode different geometries, the resulting models generalized better across geometries and outperformed neural network models trained on individual geometries.

Theory-inspired feature engineering has also been employed quite naturally in the fields of welding and manufacturing, e.g., for weld quality assessment. Instead of directly using acoustic emission measurement data for the machine learning model input, the authors of [16] proposed a physics-based step to produce meaningful features such as absolute signal energy or the centroid frequency of the signal. In [17], the authors suggest to detect abnormal heat using a heat transfer model, the parameters of which are fitted to the data and subsequently used for outlier detection (e.g., via isolation forests). This method, combining off-the-shelf outlier detection with theory-inspired features, has the potential to reduce testing time by 43%. Theory-inspired features were also utilized in modeling a steel-sheet galvanizing production line [18]. These features included anode voltage (resistance), calculated using Kirchhoff's laws by summing resistances over the dynamic system which includes anode voltage, electrolyte, steel voltage, and other factors. Using these theory-inspired features in training data-driven machine learning models improved the predictions on the test set. Similarly, the authors of [19] used theory-inspired features for the design of new alloys and showed that transforming data through prior physico-chemical knowledge can create more accurate machine learning models for prediction of transformation temperatures. The improvement was explained by the introduction of mathematical nonlinearities given by, e.g.,

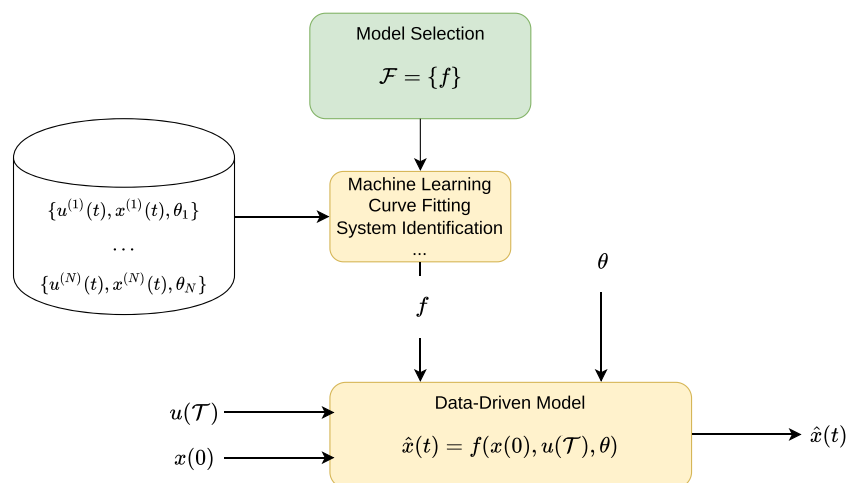
material growth kinetics models which give information on material behavior even in temperature ranges not available in the raw data.

Interesting use cases for theory-inspired feature engineering can also be found in the domain of additive manufacturing (AM). An example is [20], where neural networks are utilized to predict grain structure in deposition processes during AM. Instead of using complex numerical models, the authors trained neural networks to link the thermal data obtained from finite volume simulations (such as temperature gradient and the cooling rate at the liquids temperature) to micro-structure characteristics. In another research paper in AM [21], the authors utilized theory-informed features to predict porosity in selective laser melting. The raw features, being machine and laser settings, are converted to physically meaningful features such as laser energy density in a point of the material powder bed, radiation pressure, and power intensity. The engineered features are used in several nonlinear regression models (support vector regression, Gaussian processes, etc.). A further use case in laser-assisted AM is the prediction of balling defects in [22]. The authors constructed theory-inspired features using 3D, transient, heat transfer, and fluid flow models. The inputs to these theory-driven models are process parameters and material properties, while the outputs are 3D temperature and velocity fields. From these outputs, physically meaningful features are computed (e.g., volumetric energy density or surface tension forces), which were subsequently used in a genetic algorithm to understand the relationship to balling defects.

3.2 Theory-inspired model selection

Another avenue to incorporate prior theoretical knowledge in a data-driven model is via an informed selection of the model class \mathcal{F} (Fig. 4). For example, knowing that the relationship we want to learn is approximately linear or piece-wise constant would suggest to select f from the

Fig. 4 Theory-inspired model selection. Prior knowledge can help selecting the model class \mathcal{F} for the data-driven model. This may reduce the required amount of data



class of linear or decision tree models, respectively. If the relationship is known to be neither linear nor piece-wise constant, then one may resort to nonlinear regression models such as polynomial regression, symbolic regression, or support vector machines, where the prior knowledge about the problem at hand can help selecting the polynomial order, candidate functions for symbolic regression, or appropriate kernel functions.

Theoretical insights about the nature of the data and the problem have further been shown useful for choosing the architecture of neural networks: convolutional neural networks [23] were shown to perform superior on images and industrial time series, recurrent neural networks [24] achieve impressive results for speech signals, and attention mechanisms [25] are now state-of-the-art in natural language processing. Most recently, neural architectures have been developed that are inspired by decision trees and that achieve state-of-the-art performance for tabular data, e.g., [26]. These types of architectural choices are connected with the way how the candidate function f is parameterized (e.g., the class of convolutional neural networks parameterizes f via subsequent convolutions and nonlinear activation functions), and thus influence the *inductive bias* of the model. An appropriately chosen inductive bias helps the optimization algorithm to select a desirable set of locally optimal function parameters ψ more reliably than if the function would be parameterized differently. A concrete example are prior dictionaries [27] in the context of physics-informed neural networks (see Section 3.3), which are analytical or learned functions interacting with the main network and thus enforce optimization constraints (for example, boundary or initial conditions of a system of differential equations).

Prior knowledge can help in selecting the neural architecture also in a more narrow sense, such as choosing kernel sizes and stride parameters for convolutional neural networks or the number of layers and their respective widths for fully connected neural networks. This has been done, for example in the design of a neural classifier for engine knock [28]. There, the authors adjusted the kernel size in the underlying network's initial convolutional layer according to the wavelength of expected vibrations, thus leveraging existing engineering knowledge about the frequency-dependent nature of engine knock. Subsequent Fourier analyses of the trained kernel showed that it indeed amplifies the mentioned target frequencies in the input signal, leading to higher detection accuracy when compared to other parameterized models. The authors of [29] designed a convolutional neural network for fault detection in rotating machines, where the kernels in the initial layers were hand-crafted based on prior knowledge about the fault modes, outperforming classical, uninformed convolutional neural networks. A similar approach was used to predict

the quality of products produced with electrochemical micro-machining [30]. The authors employed a fully connected neural network and assumed that the first layer automatically constructs physically meaningful features (such as current density, void fraction) from the input (voltage, pulse time, etc.). To guide the training process towards this feat, network edges that are inconsistent with the corresponding features were eliminated from the network's first layer, yielding improved performance in all experiments when compared to an exclusively data-driven approach. In other efforts to incorporate theoretical knowledge in machine learning, physics-based constraints have been incorporated in individual layers of Long Short-Term Memory networks [31] to improve generalizability of the presented reduced-order model for fluid flows.

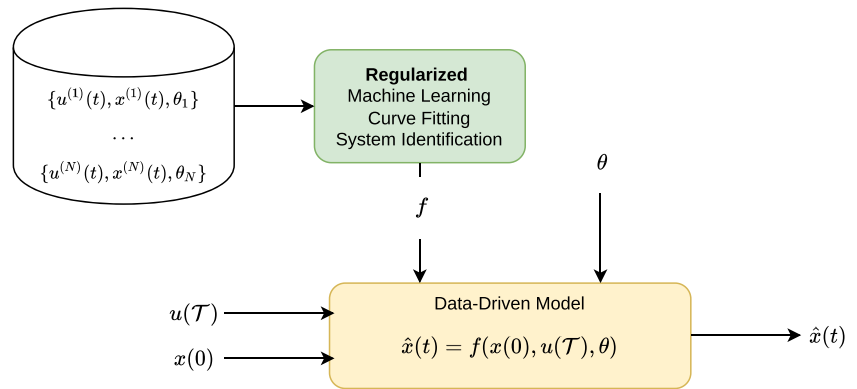
Leveraging special knowledge of welding defects, machine learning methods have also been enhanced in more detailed ways, such as changing the nature of one network layer depending on the training example [32]. Here, a customized pooling function is designed, processing the input image in a distinct way. For weld quality assessment, the authors of [16] utilized their understanding of the welding process to select a sequence model approach, which treats recorded time steps as distinct training examples, while in [33] the underlying task was distributed to multiple submodels dedicated to different subtasks. In the former case, the approach proved to be more stable than more commonly employed methods, while in the latter case the thus selected architecture is characterized by increased interpretability and trust.

3.3 Model regularization via theory

Once a model class \mathcal{F} has been selected, training the model can further benefit from existing domain knowledge. Consider the setting in Fig. 5, where a machine learning, system identification, or curve fitting algorithm is used to find a candidate function f that represents the existing dataset \mathcal{D} .

Very often, the problem of finding the most suitable candidate function f (e.g., of finding the most suitable parameters ψ) within the selected model class is a non-convex optimization problem. Furthermore, especially in the field of deep learning, this problem is often underdetermined, i.e., there are multiple candidate functions f in the model class that fit the data perfectly. In these cases it is necessary to *regularize* the algorithm towards prioritizing certain candidate functions over others. Classical approaches in machine learning penalize the ℓ_2 or ℓ_1 norms of the model parameters, leading to ridge and LASSO regression [34, Sec. 3.1.4] in linear models or weight decay regularization in neural networks [34, Sec. 5.5], respectively. Loosely speaking, these classical approaches prefer

Fig. 5 Model regularization via theory. Domain knowledge can be incorporated into a data-driven model via regularizing the training process. This prioritizes models that are consistent with domain knowledge, or penalizes those that are in conflict with it



simple models over complicated ones, thus formalizing Occam's razor. Regularization can furthermore be seen as a “soft” version of constraining the hypothesis space provided by the model class, which we have discussed in Section 3.2.

Domain knowledge can successfully be used for regularization. By appropriately setting the regularization terms, candidate functions f can be prioritized or penalized that are consistent or in conflict with existing theory. For example, in the field of fluid dynamics, we may not only aim at minimizing some ℓ -norm between the ground truth flow field $x(t)$ and its estimate $\hat{x}(t)$, but we may also regularize f such that the vorticity fields of $x(t)$ and $\hat{x}(t)$ are similar or that (for incompressible fluids) the divergence of $\hat{x}(t)$ is minimized [35]. While these regularizers rely on the availability of ground truth, one can also design regularizers that are based solely on properties of f as suggested by domain knowledge (in the form of algebraic or differential equations). For example, in the domain of lake temperature modeling, neural networks were regularized such that the relationship between water density and depth is monotonic, cf. [36, eq. (3.14)]. Such a *physics-guided neural networks* was also used in [37] to quantify microcrack defects, regularizing the network via approximate mechanistic models. Regularization can also be used to penalize symbolic regression models that violate monotonicity or boundedness constraints [38].

As mentioned in Section 2, the incorporation of domain knowledge has the potential to improve the trade-off between the need of training data and the capability to achieve good generalization performance. Taken to the extreme, proper regularization can obviate the need for (labeled) training data altogether: One example is the work of [39], where a neural network is trained to regress the height of a falling object from a series of images. Rather than providing object heights as ground truth labels, training is based only on time-stamped images and the prior knowledge that the height trajectory of falling objects is parabola. Regularizing training based on this knowledge is here sufficient to allow the neural network to extract the

information of interest (i.e., the object's height) from data that depend on this quantity (i.e., the images). Another class of models, physics-informed neural networks (PINNs), are regularized via a known system of partial differential equations (PDEs) and can dispense with training data altogether [40]. These PINNs have the capability of solving systems of PDEs. In the setting of Fig. 1 without the forcing function $u(t)$, PINNs take the time instances t within the computational domain \mathcal{T} of interest as input and respond with an estimate $\hat{x}(t)$ of the solution of the differential equation. In their original formulation, PINNs are trained by minimizing two kinds of losses: A loss component that accounts for the initial condition $x(0)$ (and, potentially, boundary conditions) which is provided to the PINN as training data, and a loss component that penalizes candidate solutions violating the differential equation $dx(t)/dt = F(x(t); \theta)$. PINNs have also been proposed for inverse problems, where the parameterization θ is learned from the PDE and its solution $x(t)$ [41].

While PINNs are versatile, there have been numerous reports in research showing that standard PINN architectures are often hard to train. Their success and accuracy is problem-specific and typically cannot be determined a-priori. One major failure mode of PINNs is their multi-objective nature, relying on data- and physics-based loss components: During model training several loss components, encoding initial and/or boundary conditions and (sets of) PDEs, compete against each other to meet to overall objective. Failing at minimizing a single objective leaves the overall objective not being fulfilled entirely. As a result, large discrepancies between learned and observed solutions are recorded. Whether an optimization algorithm can find a candidate solution $\hat{x}(t)$ for which all loss components are low is strongly determined by the innate shape of the Pareto front in the multi-objective optimization. System parameters, such as the PDE's parameterization or the computational domain, have a strong impact on the shape of the Pareto front [42]. Scalability is another issue in the use of PINNs. As the system dimension or complexity increases, PINNs tend to be even more difficult to train.

Proper non-dimensionalization of the system under study appears to facilitate optimization. Additionally, several loss weighting techniques have been proposed that deal with the problem at hand. Loss components are either weighted manually or in an adaptive manner based on the history of recorded gradients [43–45]. As mentioned in Section 3.2, another approach are prior dictionaries [27], which implement hard constraints for the boundary conditions and, thus, reduce the number of objectives in the multi-objective optimization. Further modifications of PINNs include X-PINNs [46], which try to break down the system complexity to multiple, smaller and simpler problems, which are solved separately by multiple PINN instances. While X-PINNs show improved accuracy for certain applications, the implementation comes with the cost of computational complexity.

Despite these problems, PINNs and their variants have successfully been used in fluid mechanics [44, 47], aerodynamics [48, 49], (nano-)optics [50, 51], and medical science [41, 52], to name a few. Furthermore, PINNs have been applied in solid mechanics including additive manufacturing [21, 53], elastodynamics [54–56], and thermal engineering [57]. As concrete example for the latter, PINNs were used in [58] to reduce the need for large datasets when predicting the temperature and melt pool dynamics during metal AM using deep learning methods. In this work, domain knowledge from first physical principles is exploited to physics-inform the learning process, resulting in accurately predicted dynamics with only a moderate amount of labeled data.

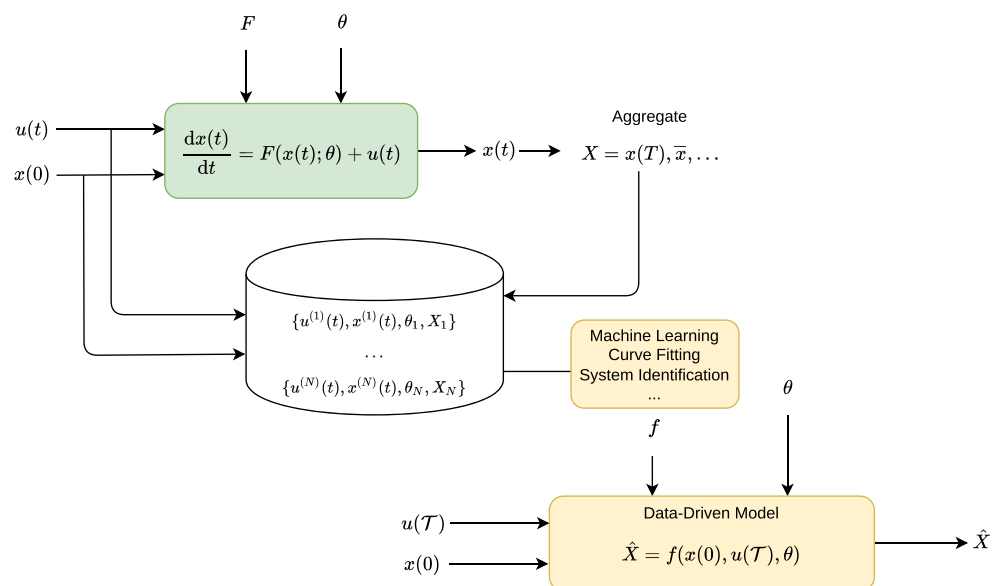
4 Data-driven models replacing costly simulations: (reduced-order) surrogate models

In many scientific disciplines, full-order simulations have prohibitive computational complexity. Examples include computational fluid dynamics as well as multi-physics problems, that often require high-resolution finite element analyses. In these cases, it may be necessary to replace the full-order model simulation by less expensive computations. A classical example is model order reduction, where the full-order model is replaced by a model with a smaller state space, e.g., using proper orthogonal decomposition (POD); the smaller model remains being solved by classical solver schemes. While also this approach can benefit from using machine learning (e.g., several POD bases can be learned by applying clustering techniques, thus achieving more accurate fits for individual parameter ranges [59]), in this section our focus is on replacing numerical solvers entirely by a learned model (Fig. 6).

Specifically, let us assume that we have access to a dataset \mathcal{D} of previous simulations of the full-order model as in (2). With this dataset, it is possible to train a data-driven model that encapsulates the relationship between the respective input parameters ($x(0)$, θ , and $u(t)$) and the solution $x(t)$, i.e., the data-driven model is a function f that satisfies

$$\hat{x}(t) = f(x^{(i)}(0), u^{(i)}(\mathcal{T}_i), \theta_i) \approx x^{(i)}(t) \quad (6)$$

Fig. 6 Surrogate modeling. In settings where the full-order simulation of a physical phenomenon is computationally too complex, it may be possible to replace this simulation by a data-driven model that is trained on data from the full-order simulation. If just an aggregate statistic (denoted as X in the figure) is of interest, a *reduced-order* surrogate model suffices



for all $t \in \mathcal{T}_i$ and $i = 1, \dots, N$. If the dataset is sufficiently large and diverse (e.g., the parameters θ_i cover a large area of the parameter space), then we may assume that $\hat{x}(t)$ is a good approximation of the true solution $x(t)$ also for other parameters, initial conditions, and forcing functions. Then, the function f is a surrogate for the full-order simulation. (In this sense, the PINNs discussed in Section 3.3 can be seen also as surrogate models.) Thus, while surrogate modeling requires a one-time investment in the sense of constructing a dataset \mathcal{D} based on full-order simulations, this investment pays off once the model is trained, allowing to substitute the full-order model at least approximately and within well-defined parameter ranges.

The problem of surrogate modeling simplifies if, instead of the entire solution $x(t)$, only some aggregate statistic is of interest. For example, we may be interested in the solution $x(T)$ at a given time T , or at the average of $x(t)$ over a designated time period; if $x(t)$ is a field, we may further be interested in values at specific positions, etc. In this case, data-driven modeling simplifies as the target to be learned has a lower dimensionality. We call this latter scenario *reduced-order* surrogate modeling.

There is a huge body of literature regarding surrogate and reduced-order surrogate modeling, covering various fields of science and using various types of surrogate models. For example, graph neural networks, trained on mesh-based simulations, were used for surrogate modeling in aerodynamics, structural mechanics, and fabric [60]. Tree-based models trained on finite element method (FEM) simulations were used to estimate the biomechanical behavior of breast tissue under compression [61] and the mechanical properties of carbon fiber reinforced plastics [62]. Kernel ridge regression was used to approximate the energy potential of carbon crystal structures to sidestep computationally costly density functional theory computations [13]. Fully connected neural networks, or multi-layer perceptrons, were used as surrogate models for 3D trusses [15], the mechanical behavior of livers [63], for forming load prediction of AZ13 material [64], the grain structure of additively manufactured material [20], and the velocity field and location of neutral point of cold flat rolling [65]. In [66], the authors predict damage development in forged brake discs reinforced with Al-SiC particles from damage maps using neural networks and Gaussian processes. For three-dimensional turbulent flow inside a lid-driven cavity, neural and random forest-based surrogate models were trained on simulation data to predict local errors as a function of coarse-grid local flow features [67].

For rapid estimation of forming and cutting forces in hot upsetting and extrusion with given process parameters, the authors of [68] utilized neural network-based surrogates. To obtain training data, they executed FEM simulations modelling the process of hot upsetting and extrusion of

a CK-45 steel axi-symmetric specimen, respectively, to obtain forming forces. The reduced-order surrogates rapidly computed the process load from the coefficient of friction, temperature, velocity, and height-to-diameter ratio for hot upsetting and from die angle, punch velocity, coefficient of friction, and temperature of billet for hot extrusion, respectively and were shown to interpolate well between training parameters. To estimate the forging load in hot upsetting and hot extrusion processes, the authors of [69] used gene expression programming and neural networks. Using FEM simulation data from [68], they showed that the upsetting process was well-approximated by the gene expression programming approach, while for extrusion the neural surrogate model was superior. This connects back to our discussion in Section 2, where we mentioned that data-driven modeling is often an iterative procedure relying trial-and-error, and that it is not always clear which model class will perform best for a given problem setting. From this perspective, comparative studies and similar guidelines provide useful information to the practitioner. An example for such a comparative study in the field of structural analysis can be found in [70], where the authors compared the performance of several neural and classical surrogate models.

Surrogate and reduced-order surrogate models lend themselves to being used for process or design optimization. For example, surrogate models were used in multi-objective optimization to design the shape of textured surfaces with non-Newtonian viscometrics functions [71], and Gaussian processes were used for hydropower Kaplan turbine design [72]. The authors of [73] used two single-layer fully connected neural networks for optimizing the forging process for steel discs (the number of neurons in the hidden layer were selected using a cascade learning procedure [74]). The authors proposed a reduced-order surrogate model mapping from workpiece initial temperature, die temperature, and friction value to flank wear and temperature. The resulting model replaced FEM simulations during sequential approximate optimization. To get appropriate training data, the FEM simulations were executed for points in the feature space deemed important, indicating that domain knowledge can also enter in the selection of training data (see also [13]).

5 Incomplete prior knowledge: causal machine learning

Triggered by multiple advances in the field [75], the topic of causality has generated a lot of interest recently, especially in the machine learning community. Causal models can be seen as being located in between purely theory-driven and purely data-driven models [76], with their exact position

within this spectrum determined by the availability of domain knowledge.

At one end of the spectrum, the physical phenomenon under study is well understood, e.g., its description may be given in the form of a system of differential equations (e.g., (1), see Section 2). Structural Causal Models (SCM, [77]) are built around these equations, but also integrate (unknown) noise factors, allow for explicit modelling of interventions, and distinguish between observable and/or controllable variables. From this perspective, SCMs can be seen to extend the capabilities of the theory-driven model introduced in (1). For example, while our phenomenon under study certainly has an initial condition $x(0)$, we may only be able to determine it with some measurement noise. Similarly, while we may want to influence the phenomenon via a controlled forcing function $u(t)$, we may only be able to set its values to within a limited precision. All these aspects can be included in SCMs. Indeed, it has been shown that ordinary differential equations can be expressed as SCMs under some (stability) assumptions, as illustrated in [78] for damped harmonic oscillators.

Closer to the other end of the spectrum are models where the available domain knowledge only accounts for the presence (or absence) of individual causal relationships. This type of domain knowledge is often represented via causal graphs [79], where nodes in the graph represent variables and directed edges indicate a direct causal relationship. To give an example, the theory-driven model (1) implies that the trajectory of the quantity of interest $x(\mathcal{T})$ is causally affected by the forcing function $u(\mathcal{T})$ and the initial condition $x(0)$, leading to the causal graph depicted in Fig. 7. While the available information in this case is far less than for SCMs, the utility of such models has been shown in a number of applications.

For example, even in the simple setting of a single (unobserved) common cause and two (observed) independent effects, unlabelled data can be used to remove systematic

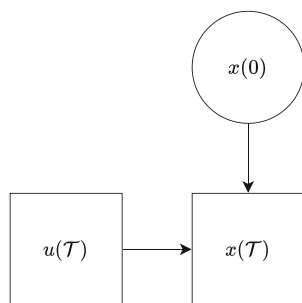


Fig. 7 In settings with incomplete prior knowledge, at least partial knowledge about the cause-effect relationships may be available in the form of a causal graph. In the context of (1), this causal graph indicates that the trajectory $x(\mathcal{T})$ depends on the initial condition $x(0)$ and the trajectory of the forcing function $u(\mathcal{T})$. Boxes indicate quantities that are observable, while the circle indicates that (in this example), the initial condition cannot be observed directly

noise from observations and hence improve the prediction performance. This has been shown exemplary for the detection of exoplanets based on satellite data [80], a task that is traditionally tackled either via theory-driven approaches in combination with simple machine learning methods (cf. Section 3.1), or limited preprocessing and complex machine learning methods (e.g., deep learning) [81].

The direction of causal relationships has been shown to be helpful in assessing the utility of unlabelled data for semi-supervised classification scenarios. Of particular interest is here the anti-causal case where the cause is predicted from the effect, cf. [82, Sec. 3]. Here, the distribution of the cause can be estimated better from unlabelled data if the cause-effect relationship is known [83].

Another advantage of causal models is their ability to make machine learning models robust against changes in the distribution of data, e.g., caused by varying but unknown parameters θ of the phenomenon under study. As we have discussed in Section 2, purely data-driven models do not generalize or extrapolate well outside of the range of training data. Intuitively, knowledge about the causal relationships underlying the data generation process could be used for regularization, such that the resulting model is consistent with these relationships. Indeed, it has been shown in a use case on gene expressions that varying environments and their distribution shifts are even beneficial for obtaining models [84] that generalize better.

Finally, in settings where not even knowledge about cause-effect relationships is available, causal discovery (such as structure learning or cause-effect discovery) can be applied. Successful applications range from economy-related scenarios [85] to indoor localization [86].

6 Discussion and conclusion

Tribal knowledge in machine learning suggests that the success of a data-driven modeling problem depends on (at least) the following ingredients:

- Data (i.e., amount, quality, etc.),
- Modeling assumptions (i.e., what mathematical assumptions do we make about the underlying relationship that we aim to learn),
- Implementation choices (i.e., how do we implement the model numerically; e.g., architectural choices for neural networks),
- Objective function (i.e., based on what quantities do we decide whether learning was successful), and
- Optimization algorithm (i.e., how do we determine from data the parameters of the implemented model such that the objective function is optimized).

Theory and domain knowledge can influence the selection of any of these ingredients, and in this small survey we presented several approaches how this influence can be exerted: Theory can assist selecting or even engineering appropriate features for the subsequent machine learning algorithm (data and modeling assumptions), it can help selecting the model class (modeling assumptions and implementation choices), or regularize model training to ensure consistency with established theory (objective function). Further, we have shown that theory-driven models are often used to generate training data for data-driven modeling, and that the resulting data-driven models can successfully step in for the often computationally costly theory-driven models.

Of course, the distinction between the presented approaches can sometimes be difficult. For example, structural causal models as discussed in Section 5 can be seen as a generalized framework to incorporate data into fully developed theory-driven models, while causal graphs can be used for theory-inspired model selection or regularization. As another example, consider [29], which proposed hand-crafting the initial layers of a convolutional neural network based on prior knowledge about the failure modes of rotating machinery. On the one hand, this can be seen as theory-inspired model selection. On the other hand, since the first layers are thus not learnable, these hand-crafted convolutional kernels can be interpreted as generating theory-inspired features for the subsequent network layers. This resonates with the fact that also the ingredients of a machine learning algorithm are strongly dependent on each other, and that in some cases modeling choice, objective function, and optimization algorithm turn out to be the different sides of the same coin, cf. [87].

Further, note that the presented approaches are not mutually exclusive. Different approaches can indeed be combined, e.g., theory can assist both model selection and feature engineering (e.g., [16]) or surrogate models can be designed based on theory-inspired features [13, 20]. PINNs can be seen as surrogate models that are trained exclusively using theory-inspired regularization, and if initial and boundary conditions are implemented via prior dictionaries, the PINN architecture is furthermore selected by theory. Indeed, theory and domain knowledge can influence the selection of any of the ingredients mentioned above, and one can expect that the performance of the resulting models will be the better the more ingredients are theory-inspired. We are thus convinced to see theory-inspired machine learning and hybrid modeling on the rise, heading towards an all-encompassing synergy between knowledge and data.

Funding Open access funding provided by Graz University of Technology. The work of Johannes G. Hoffer and Bernhard C. Geiger was partially supported by the project BrAIN. BrAIN – Brownfield Artificial Intelligence Network for Forging of High Quality Aerospace

Components (FFG Grant No. 881039) is funded in the framework of the program ‘TAKE OFF’, which is a research and technology program of the Austrian Federal Ministry of Transport, Innovation and Technology.

The authors further received financial support from the Austrian COMET – Competence Centers for Excellent Technologies – Programme of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology, the Austrian Federal Ministry for Digital and Economic Affairs, and the States of Styria, Upper Austria, Tyrol, and Vienna for the COMET Centers Know-Center and LEC EvoLET, respectively. The COMET Programme is managed by the Austrian Research Promotion Agency (FFG).

Declarations

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Karpatne A, Atluri G, Faghmous JH, Steinbach M, Banerjee A, Ganguly A, Shekhar S, Samatova N, Kumar V (2017) Theory-guided data science: a new paradigm for scientific discovery from data. *IEEE Trans Knowl Data Eng* 29(10):2318–2331. <https://doi.org/10.1109/TKDE.2017.2720168>
- Karniadakis G, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L (2021) Physics-informed machine learning. *Nat Rev Phys* 3:422–440. <https://doi.org/10.1038/s42254-021-00314-5>
- Beck A, Kurz M (2021) A perspective on machine learning methods in turbulence modeling. *GAMM-Mitteilungen* 44(1):e202100,002. <https://doi.org/10.1002/gamm.202100002>
- Brunton SL (2021) Applying machine learning to study fluid mechanics. *arXiv:2110.02083v1 [physics.flu-dyn]*
- Vadyala SR, Betgeri SN, Matthews JC, Matthews E (2021) A review of physics-based machine learning in civil engineering. *arXiv:2110.04600*
- Schweidtmann AM, Esche E, Fischer A, Kloft M, Repke JU, Sager S, Mitsos A (2021) Machine learning in chemical engineering: a perspective. *Chemie Ingenieur Technik* 93(12):2029–2039. <https://doi.org/10.1002/cite.202100083>
- Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, Prabhat (2019) Deep learning and process understanding for data-driven earth system science. *Nature* 566:195–204
- Zendehboudi S, Rezaei N, Lohi A (2018) Applications of hybrid models in chemical, petroleum, and energy systems: a systematic review. *Appl Energy* 228:2539–2566. <https://doi.org/10.1016/j.apenergy.2018.06.051>

9. Wagner N, Rondinelli JM (2016) Theory-guided machine learning in materials science. *Front Mater* 3:28. <https://doi.org/10.3389/fmats.2016.00028>
10. Hughes MT, Kini G, Garimella S (2021) Status, challenges, and potential for machine learning in understanding and applying heat transfer phenomena. *J Heat Transfer* 143(12):120,802. <https://doi.org/10.1115/1.4052510>
11. Santos T, Schrunner S, Geiger BC, Bluder O, Zernig A, Kaestner A, Kern R (2019) Feature extraction from analog wafermaps: a comparison of classical image processing and a deep generative model. *IEEE Trans Semicond Manuf* 32(2):190–198. <https://doi.org/10.1109/TSM.2019.2911061>
12. Kingma DP, Welling M (2014) In: *Proc. Int. Conf. on Learning Representations (ICLR)*. Banff
13. Rohrhofer FM, Saha S, Cataldo SD, Geiger BC, von der Linden W, Boeri L (2021) Importance of feature engineering and database selection in a machine learning model: a case study on carbon crystal structures. Technical report: arXiv:2102.00191 [cond-mat.mtrl-sci]
14. Gao H, Sun L, Wang JX (2021) PhyGeoNet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. *J Comput Phys* 428:110,079. <https://doi.org/10.1016/j.jcp.2020.110079>. <https://www.sciencedirect.com/science/article/pii/S0021999120308536>
15. Nourbakhsh M, Irizarry J, Haymaker J (2018) Generalizable surrogate model features to approximate stress in 3D trusses. *Eng Appl Artif Intell* 71:15–27
16. Asif K, Zhang L, Derrible S, Indacochea JE, Ozevin D, Ziebart B (2020) Machine learning model to predict welding quality using air-coupled acoustic emission and weld inputs. *J Intell Manuf*, 1–15
17. Baghbanpourasl A, Kirchberger D, Eitzinger C (2021) In *Proc. IEEE Int. Workshop on Metrology for Industry 4.0 and IoT*, vol 2021. <https://doi.org/10.1109/MetroInd4.0IoT51437.2021.9488550>
18. Lovrić M, Meister R, Steck T, Fadljević L, Gerdenitsch J, Schuster S, Schieffmüller L, Lindstaedt S, Kern R (2020) Parasitic resistance as a predictor of faulty anodes in electro galvanizing: a comparison of machine learning, physical and hybrid models. *Advanced Modeling and Simulation in Engineering Sciences* 7(46). <https://doi.org/10.1186/s40323-020-00184-z>
19. Liu S, Kappes BB, Amin-ahmadi B, Benafan O, Zhang X, Stebner AP (2021) Physics-informed machine learning for composition–process–property design: shape memory alloy demonstration. *Applied Materials Today*, 22. <https://doi.org/10.1016/j.apmt.2020.100898>
20. Kats D, Wang Z, Gan Z, Liu WK, Wagner GJ, Lian Y (2022) A physics-informed machine learning method for predicting grain structure characteristics in directed energy deposition. *Comput Mater Sci* 202(110):958. <https://doi.org/10.1016/j.commatsci.2021.110958>
21. Liu R, Liu S, Zhang X (2021) A physics-informed machine learning model for porosity analysis in laser powder bed fusion additive manufacturing. *Int J Adv Manuf Technol* 113(7):1943–1958
22. Du Y, Mukherjee T, DebRoy T (2021) Physics-informed machine learning and mechanistic modeling of additive manufacturing to reduce defects. *Appl Mater Today* 24(101):123. <https://doi.org/10.1016/j.apmt.2021.101123>
23. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1(4):541–551
24. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
25. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) In: *Proc advances in neural information processing systems (NeurIPS)*, vol 30
26. Katzir L, Elidan G, El-Yaniv R (2021) In: *Proc. int. conf. on learning representations (ICLR)* (virtual)
27. Peng W, Zhou W, Zhang J, Yao W (2020) Accelerating physics-informed neural network training with prior dictionaries. arXiv:2004.08151
28. Ofner AB, Kefalas A, Posch S, Geiger BC (2022) Knock detection in combustion engine time series using a theory-guided 1D convolutional neural network approach. Accepted for publication in *IEEE/ASME Trans. Mechatronics*; arXiv:2201.06990 [cs.LG]
29. Sadoughi M, Hu C (2018) In: *Proc annual conf of the IEEE industrial electronics society (IECON)*, pp 5919–5923. <https://doi.org/10.1109/IECON.2018.8591073>
30. Lu Y, Rajora M, Zou P, Liang SY (2017) Physics-embedded machine learning: case study with electrochemical micro-machining. *Machines* 5(1):4
31. Pawar S, San O, Nair A, Rasheed A, Kvamsdal T (2021) Model fusion with physics-guided machine learning: projection-based reduced-order modeling. *Phys Fluids* 33(6):067,123
32. Jiang H, Hu Q, Zhi Z, Gao J, Gao Z, Wang R, He S, Li H (2021) Convolution neural network model with improved pooling strategy and feature selection for weld defect recognition. *Weld World* 65(4):731–744
33. Mayr A, Lutz B, Weigelt M, Gläsel T, Kißkalt D, Masuch M, Riedel A, Franke J (2018) In: *Proc 8th int electric drives production conf (EDPC)*, pp 1–7
34. Bishop CM (2006) *Pattern recognition and machine learning*, 1st edn. Springer
35. Schweri L, Foucher S, Tang J, Azevedo VC, Günther T., Solenthaler B (2021) A physics-aware neural network approach for flow data reconstruction from satellite observations. *Front Clim* 3:23. <https://doi.org/10.3389/fclim.2021.656505>
36. Karpatne A, Watkins W, Read J, Kumar V (2018) Physics-guided neural networks (PGNN): an application in lake temperature modeling. arXiv:1710.11431 [cs.LG]
37. Sun H, Peng L, Lin J, Wang S, Zhao W, Huang S (2021) Microcrack defect quantification using a focusing high-order SH guided wave EMAT: the physics-informed deep neural network GuwNet. *IEEE Transactions on Industrial Informatics*, 1–1. <https://doi.org/10.1109/TII.2021.3105537>
38. Kronberger G, de Franca FO, Burlacu B, Haider C, Kommenda M (2021) Shape-constrained symbolic regression – improving extrapolation with prior knowledge. *Evol Comput*, 1–24. <https://doi.org/10.1162/evco.a.00294>
39. Stewart R, Ermon S (2017) In: *Proc. AAAI Conf on Artificial Intelligence (AAAI)*, pp 2576–2582
40. Raissi M, Perdikaris P, Karniadakis G (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys*, 378. <https://doi.org/10.1016/j.jcp.2018.10.045>
41. Raissi M, Yazdani A, Karniadakis G (2020) Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* 367(6481):1026–1030
42. Rohrhofer FM, Posch S, Geiger BC (2021) On the Pareto front of physics-informed neural networks. arXiv:2105.00862 [cs.LG]
43. Wang S, Teng Y, Perdikaris P (2021) Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J Sci Comput* 43(5):A3055–A3081
44. Jin X, Cai S, Li H, Karniadakis G (2021) NSFnets (Navier-Stokes flow nets): physics-informed neural networks for the incompressible Navier-Stokes equations. *J Comput Phys* 426(109):951

45. Maddu SM, Sturm D, Müller CL, Sbalzarini IF (2021) Inverse-Dirichlet weighting enables reliable training of physics informed neural networks. *Machine Learning, Science and Technology*
46. Jagtap AD, Karniadakis G (2020) Extended physics-informed neural networks (XPINNs): a generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Commun Comput Phys* 28(5):2002–2041
47. Almajid MM, Abu-Al-Saud MO (2022) Prediction of porous media fluid flow using physics informed neural networks. *J Pet Sci Eng* 208(109):205
48. Mao Z, Jagtap AD, Karniadakis G (2020) Physics-informed neural networks for high-speed flows. *Comput Methods Appl Mech Eng* 360(112):789
49. Hu L, Zhang J, Xiang Y, Wang W (2020) Neural networks-based aerodynamic data modeling: a comprehensive review. *IEEE Access* 8:90,805–90,823
50. Chen Y, Lu L, Karniadakis G, Dal Negro L (2020) Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Opt Express* 28(8):11,618–11,633
51. Mishra S, Molinaro R (2021) Physics informed neural networks for simulating radiative transfer. *J Quant Spectros Radiat Transfer* 270(107):705
52. Sahli Costabal F, Yang Y, Perdikaris P, Hurtado DE, Kuhl E (2020) Physics-informed neural networks for cardiac activation mapping. *Front Phys* 8:42
53. Zhu Q, Liu Z, Yan J (2021) Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Comput Mech* 67:619–635. <https://doi.org/10.1007/s00466-020-01952-9>
54. Rao C, Sun H, Liu Y (2021) Physics-informed deep learning for computational elastodynamics without labeled data. *J Eng Mech* 147(8):04021,043
55. Haghighat E, Raissi M, Moure A, Gomez H, Juanes R (2021) A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput Methods Appl Mech Eng* 379(113):741
56. Ghaderi A, Morovati V, Dargazany R (2020) A physics-informed assembly of feed-forward neural network engines to predict inelasticity in cross-linked polymers. *Polymers* 12(11):2628
57. Cai S, Wang Z, Wang S, Perdikaris P, Karniadakis G (2021) Physics-informed neural networks for heat transfer problems. *J Heat Transfer* 143(6):060,801
58. Zhu Q, Liu Z, Yan J (2021) Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Comput Mech* 67(2):619–635
59. Hess M, Alla A, Quaini A, Rozza G, Gunzburger M (2019) A localized reduced-order modeling approach for PDEs with bifurcating solutions. *Comput Methods Appl Mech Eng* 351:379–403. <https://doi.org/10.1016/j.cma.2019.03.050>
60. Pfaff T, Fortunato M, Sanchez-Gonzalez A, Battaglia PW (2021) In: *Proc. Int. Conf. on Learning Representations (ICLR)*
61. Martínez-Martínez F, Rupérez-Moreno MJ, Martínez-Sober M, Solves-Llorens JA, Lorente D, Serrano-López A, Martínez-Sanchis S, Monserrat C, Martín-Guerrero JD (2017) A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time, vol 90
62. Qi Z, Zhang N, Liu Y, Chen W (2019) Prediction of mechanical properties of carbon fiber based on cross-scale FEM and machine learning. *Compos Struct* 212:199–206
63. Pellicer-Valero OJ, Rupérez MJ, Martínez-Sanchis S, Martín-Guerrero JD (2020) Real-time biomechanical modeling of the liver using machine learning models trained on finite element method simulations. *Expert Syst Appl* 143(113):083
64. Önder A (2019) A forming load analysis for extrusion process of AZ31 magnesium. *Trans Nonferrous Metals Soc China* 29(4):741–753
65. Gudur P, Dixit U (2008) A neural network-assisted finite element analysis of cold flat rolling. *Eng Appl Artif Intel* 21(1):43–52
66. Roberts S, Kusiak J, Liu Y, Forcellese A, Withers P (1998) Prediction of damage evolution in forged aluminium metal matrix composites using a neural network approach. *J Mater Process Technol* 80:507–512
67. Hanna BN, Dinh NT, Youngblood RW, Bolotnov IA (2020) Machine-learning based error prediction approach for coarse-grid computational fluid dynamics (CG-CFD). *Prog Nucl Energy* 118(103):140
68. Raj KH, Sharma RS, Srivastava S, Patvardhan C (2000) Modeling of manufacturing processes with ANNs for intelligent manufacturing. *Int J Mach Tools Manuf* 40(6):851–868
69. Bingöl S, Kılıçgedik HY (2018) Application of gene expression programming in hot metal forming for intelligent manufacturing. *Neural Comput Applic* 30(3):937–945
70. Hoffer JG, Geiger BC, Ofner P, Kern R (2021) Mesh-free surrogate models for structural mechanic FEM simulation: a comparative study of approaches. *Appl Sci* 11(20):9411
71. Dupuis R, Jouhaud JC, Sagaut P (2018) In: *Proc. AIAA/ASCE/AHS/ASC structures, structural dynamics, and materials conf.*, p 1905
72. Masood Z, Khan S, Qian L (2021) Machine learning-based surrogate model for accelerating simulation-driven optimisation of hydropower Kaplan turbine. *Renew Energy* 173:827–848
73. D'Addona DM, Antonelli D (2018) Neural network multiobjective optimization of hot forging. *Procedia CIRP* 67:498–503
74. Fahlman SE et al (1988) An empirical study of learning speed in back-propagation networks. *Carnegie Mellon University, Computer Science Department Pittsburgh, PA USA*
75. Pearl J, Mackenzie D (2018) *The book of why: the new science of cause and effect*. Basic books
76. Schölkopf B (2019) Causality for machine learning. *arXiv:1911.10500*
77. Bollen KA (1989) *Structural equations with latent variables*, vol 210. Wiley
78. Mooij J, Janzing D, Schölkopf B (2013) In: *Proc. conf on uncertainty in artificial intelligence (UAI)*, pp 440–448
79. Suzuki E, Shinozaki T, Yamamoto E (2020) Causal diagrams: pitfalls and tips. *J Epidemiol* 30:153–162. <https://doi.org/10.2188/jea.JE20190192>
80. Schölkopf B, Hogg DW, Wang D, Foreman-Mackey D, Janzing D, Simon-Gabriel CJ, Peters J (2016) Modeling confounding by half-sibling regression. *Proc Nat Acad Sci* 113(27):7391–7398
81. Nikolaou N, Waldmann IP, Tsiaras A, Morvan M, Edwards B, Yip KH, Tinetti G, Sarkar S, Dawson JM, Borisov V et al (2020) Lessons learned from the 1st ARIEL, machine learning challenge: correcting transiting exoplanet light curves for stellar spots. *arXiv:2010.15996*
82. Schölkopf B, Janzing D, Peters J, Sgouritsa E, Zhang K, Mooij J (2012) In: *Proc. int. conf. on machine learning (ICML)*. Edinburgh, pp 459–466
83. Castro DC, Walker I, Glocker B (2020) Causality matters in medical imaging. *Nat Commun* 11(1):1–10
84. Peters J, Bühlmann P, Meinshausen N (2016) Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B (Statistical Methodology)*, 947–1012

85. Pamfil R, Sriwattanaworachai N, Desai S, Pilgerstorfer P, Georgatzis K, Beaumont P, Aragam B (2020) In: Proc. int. conf. on artificial intelligence and statistics (AISTATS). PMLR, pp 1595–1605
86. Koutroulis G, Botler L, Mutlu B, Diwold K, Römer K, Kern R (2021) KOMPOS: connecting causal knots in large nonlinear time series with non-parametric regression splines. *ACM Trans Intell Syst Technol (TIST)* 12(5):1–27
87. Achille A, Soatto S (2018) Information dropout: learning optimal representations through noisy computation. *IEEE Trans Pattern Anal Mach Intell* 40(12):2897–2905

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Johannes G. Hoffer¹ · Andreas B. Ofner¹ · Franz M. Rohrhofer¹ · Mario Lovrić^{1,2} · Roman Kern^{1,3} · Stefanie Lindstaedt^{1,3} · Bernhard C. Geiger^{1,4} 

Johannes G. Hoffer
jhoffer@know-center.at

Andreas B. Ofner
aofner@know-center.at

Franz M. Rohrhofer
frohrhofer@know-center.at

Mario Lovrić
mlovric@know-center.at

Roman Kern
rkern@know-center.at; rkern@tugraz.at

Stefanie Lindstaedt
slind@know-center.at

¹ Know-Center GmbH, Inffeldgasse 13, Graz, 8010, Austria

² Institute for Anthropological Research, Gajeva ul. 32, Zagreb, 10000, Croatia

³ Institute of Interactive Systems and Data Science, Graz University of Technology, Inffeldgasse 16c, Graz, 8010, Austria

⁴ Signal Processing and Speech Communication Laboratory, Graz University of Technology, Inffeldgasse 16c, Graz 8010, Austria