Informed Machine Learning: Integrating Prior Knowledge into Data-Driven Learning Systems

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
Laura von Rüden
aus
Marsberg

Bonn, 12.06.2023

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter:	Prof. Dr. Christian Bauckhage
2. Gutachter:	Prof. Dr. Jochen Garcke

Tag der Promotion:	12.10.2023
Erscheinungsjahr:	2023

Abstract

Machine Learning is an important method in Artificial Intelligence (AI). It has shown great success in building models for tasks like prediction or image recognition by learning from patterns in large amounts of data. However, it can have its limits when dealing with insufficient training data. A potential solution is the additional integration of prior knowledge, such as physical laws, logic rules, or knowledge graphs. This leads to the notion of Informed Machine Learning (Informed ML). However, the field is so application-driven that general analyses are rare.

The goal of this PhD thesis is the unification of Informed ML through general, systematic frameworks. In particular, the following research questions are answered: 1) What is the fundamental concept of Informed ML, and how can existing approaches be structurally classified, 2) is it possible to integrate prior knowledge in a universal way, and 3) how can the benefits of Informed ML be quantified, and what are the requirements for the injected knowledge?

First, a concept for Informed ML is proposed, which defines it as learning from a hybrid information source that consists of data and prior knowledge. A taxonomy that serves as a structured classification framework for existing or potential approaches is presented. It considers the knowledge source, its representation type, and the integration stage into the ML pipeline. The concept of Informed ML is further extended to the combination of ML and simulation towards Hybrid AI.

Then, two new methods for a universal knowledge integration are developed. The first method, Informed Pre-Training, allows to initialize neural networks with prototypes from prior knowledge. Experiments show that it improves generalization, especially for small data, and increases robustness. An analysis of the individual neural network layers shows that the improvements come from transferring the deeper layers, which confirms the transfer of semantic knowledge (Informed Transfer Learning). The second method, Geo-Informed Validation, checks models for their conformity with knowledge from street maps. It is developed in the application context of autonomous driving, where it can help to prevent potential predictions errors, e.g., in semantic segmentations of traffic scenes.

Finally, a catalogue of relevant metrics for quantifying the benefits of knowledge injection is defined. Among others, it includes in-distribution accuracy, out-of-distribution robustness, as well as knowledge conformity, and a new metric that combines performance improvement and data reduction is introduced. Furthermore, a theoretical framework that represents prior knowledge in a function space and relates it to data representations is presented. It reveals that the distances between knowledge and data influence potential model improvements, which is confirmed in a systematic experimental study.

All in all, these frameworks support the unification of Informed ML, which makes it more accessible and usable – and helps to achieve trustworthy AI.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Artificial Intelligence (AI) is a rapidly evolving field and is having a transformative impact on many industries and aspects of our daily lives. Some everyday examples are image recognition technologies including automatic photo tagging, conversational AI systems like Siri or ChatGPT, or advanced driver-assistance systems that support humans in driving cars. In the last decades, AI technologies have shown an almost inconceivable exponential development and will have an even more significant role in the future.

One area of AI that has shown particularly great success in recent years is Machine Learning (ML). ML deals with the development of computing algorithms that can automatically build models by learning from data. Once the model has been trained, it can be used to make predictions on new, unseen data. ML is successfully applied for a broad field of tasks, e.g., for image recognition [29], natural language understanding [66], or in recommender systems [47]. These tasks were revolutionized through methods of deep learning, which involves the training of neural network models that are composed of multiple processing layers [30]. In addition to the originally classical AI domains, ML is now also increasingly important in engineering and natural sciences. Application examples span a wide range, including environmental modelling [49], material sciences [12], biomedicine [14], and autonomous driving [7].

However, there are many circumstances where purely data-driven approaches can reach their limits or lead to unsatisfactory results. The most obvious scenario is that not enough data is available to train well-performing and sufficiently generalized models. Another important aspect is that a purely data-driven model might not meet constraints such as dictated by natural laws, or given through regulatory or security guidelines, which are important for trustworthy AI [10]. With ML models becoming more and more complex, there is also a growing need for models to be interpretable and explainable [50].

These issues have led to increased research on how to improve ML models by additionally incorporating prior knowledge into the learning process. Although the integration of implicit knowledge into ML is common, e.g., through labelling or feature engineering, there is a growing interest about the integration of more and explicit knowledge. This additional knowledge is often given by formal knowledge representations, such as logic rules [18, 69], knowledge graphs [3, 34, 27], algebraic equations [28, 62], or simulation models [16, 32, 46]. As an umbrella term for methods that inject such prior knowledge into data-driven learning systems, we henceforth use *Informed Machine Learning (Informed ML)*.

There are many different applications where Informed ML is already successfully used – especially in scientific and engineering domains, where data acquisition can be expensive, and lots of prior knowledge is available. For example: In neural networks for climate prediction, physical laws are injected via knowledge-based loss functions [28]; In robotics, simulations are used as an additional source for training data [48]; and in autonomous driving, the perception of traffic scenes is improved by using knowledge graphs that reflect relations between detected objects [34].

Nevertheless, there are several open research questions about Informed ML. The field is so application-driven that it has led to the development of many different and rather specific approaches. In contrast, general analyses about Informed ML are still missing. This makes it difficult to transfer existing approaches to new applications, or to estimate potential improvements in advance. To improve this situation, the research goal of this PhD thesis is to answer the following central questions:

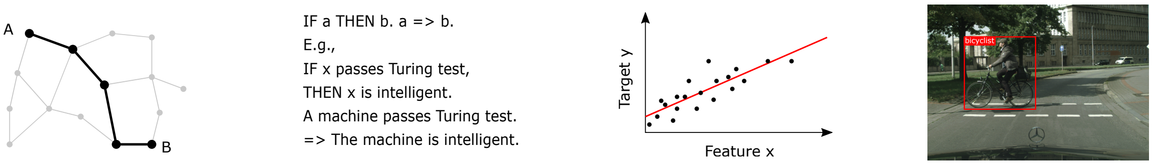
1. What is the fundamental concept of Informed ML, and how can existing approaches for integrating prior knowledge into data-driven learning be structurally classified?

2. Is it possible to integrate prior knowledge into ML in a universal way, and how?

3. How can the benefits of Informed ML be quantified, and what are the requirements for the injected knowledge?

For this, Informed ML is unified in this PhD thesis through the development of general, systematic frameworks. These frameworks comprise different abstraction levels: From concepts, over methods with applications, to theory and systematic analysis. First, a unified concept of Informed ML is proposed, which illustrates its building blocks and serves as the foundation for all further developments. Based on this, a systematic taxonomy is developed and a structured classification of existing approaches is presented. Then, two new methods that permit a universal integration of knowledge into learning systems are proposed. Here, these model-agnostic methods are particularly utilized to improve neural network models. Their relevance is demonstrated in various applications with a primary focus on autonomous driving. Finally, a framework for an Informed Learning Theory is presented, which allows a systematic analysis and quantification of the effectiveness of knowledge integration.

The structure of this dissertation is as follows. This comprehensive introduction chapter consists of three parts: This motivation (Section 1.1), a description of the technical background (Section 1.2), as well as an overview of the PhD thesis contributions (Section 1.3). Chapters 2-5 give further summaries of the published research papers. Finally, Chapter 7 concludes this document with a discussion, and an outlook on future research.



(a) Path searching. (b) Logical reasoning. (c) Prediction. (d) Object detection.

Figure 1.1: **Examples of AI Tasks.** AI can be used for a large variety of tasks that require intelligent actions to solve problems [59], e.g., (a) searching a path (e.g., a route in a street map graph), (b) decision making through inference using first-order logic, (c) prediction through linear regression, or (d) image recognition and object detection (e.g., pedestrian detection (image from Cityscapes dataset [15])). The first two are examples for symbolic AI (knowledge-based AI) as they employ abstract knowledge representations, such as structural graphs and logic rules. The latter two are examples for connectionist AI (data-based AI): These tasks involve learning from examples, which can be well done with machine learning, e.g., by training neural network models.

1.2 Technical Background

Here, the necessary technical background on artificial intelligence, machine learning and neural networks, is shortly summarized. Unless stated differently, the main references that are used for this section are [59, 5, 39, 30].

1.2.1 Artificial Intelligence

Artificial Intelligence (AI) is the ability of machines, especially of computer programs, to perform intelligent actions to solve problems and achieve goals [35]. The required actions can be associated with originally human-like behavior and include perception, understanding, prediction and even some form of manipulation of an environment [59]. The tasks that AI can accomplish are as extensive as the tasks that human thinking can accomplish: From algorithms for searching and planning, over logical reasoning and decision making, to pattern recognition and learning from examples (see Figure 1.1).

Historical Outline

First works of modern AI began shortly after the emergence of digital computers and can be dated back to the 1940s and 1950s [59]. At that time, first versions of neural networks, machine learning algorithms and symbolic reasoning capabilities were already being studied. Also, the famous Turing Test was proposed, which tests if a machine can exhibit intelligent behavior that is equivalent to, or indistinguishable from, that of a human [64].

Since then, AI research roughly considered the two antipodal paradigms of **symbolic AI vs. connectionist AI**. Symbolic AI, also known as rule-based systems or **knowledge-based AI**, can be regarded as a top-down approach that represents information through abstract symbols or logic rules. For example, the AI tasks illustrated on the left side in Figure 1.1 involve such abstract knowledge representations: (a) graph structures are used for searching, and (b) logic rules are used for reasoning. The paradigm of symbolic AI dominated up until the 1980s. A specific type of symbolic AI are expert systems. On the other hand, connectionist AI can be regarded as a bottom-up approach that represents information through neural networks. These networks can be built by learning from data, which is why connectionist AI is a special form of **data-based AI**. For example, the AI tasks illustrated on the right side in Figure 1.1 are typically solved by using data representations: (c) prediction and (d) object detection models can be trained by learning from examples. The connectionist AI paradigm became more popular in the 1990s and especially in the 2010s, deep learning with neural networks has led to impressive performance and became the predominant paradigm in AI [23]. Combining both paradigms, in particular, symbolic reasoning capabilities and neural networks towards a **hybrid AI** approach is a longstanding goal in the area of artificial intelligence [61, 37, 52].

1.2.2 Machine Learning and Neural Networks

The goal of machine learning is to find a **model** that fits **observed data** and that can be used to draw conclusions about **new data**. The modelling process consists mainly of two phases: **Training** and **testing**. First, the model is trained on some given training data, then the model is evaluated or applied on some other test data. A good model shows a good performance not only on the training data, but also on unseen test data – this is called **generalization**.

There are roughly three machine learning scenarios that are different in the form of feedback that is given by the data: Supervised, unsupervised, and reinforcement learning. The focus of this is mainly on **supervised** learning, where the data consists of input-output pairs and the learned model is supposed to be a mapping from input to output. The individual input values are called **features** and the output values are called targets or **labels**. Typical supervised learning tasks are regression or classification (See, e.g., Figure 1.1(c) and 1.1(d)).

Mathematically speaking, machine learning is similar to **function approximation**. It involves finding a function $f \in \mathcal{F}$, also called **hypothesis**, that best approximates an unknown relationship $g : \mathcal{X} \rightarrow \mathcal{Y}$ between features $x \in \mathcal{X}$ and labels $y \in \mathcal{Y}$ in a given dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1...n}$ with sample size n. The optimal function $\hat{f}$ (i.e., the final model[1]) can be found by minimizing the training error, also called empirical risk $R(f)$, with a given **loss** function l:

$$\hat{f} := \arg\min_{f \in \mathcal{F}} R_{\mathcal{D}}(f), \quad R_{\mathcal{D}}(f) = \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} l(f(x), y) \tag{1.1}$$

Empirical risk minimization is a key concept of statistical learning theory [65], for which further details can be found in the overviews in [33, 39].

The choice of the function class $\mathcal{F}$ depends on the problem to be solved and the given data. Some prominent function classes are linear models, decision trees, support vector machines, or neural networks. For example, linear models are simple and easy to interpret and have the form

$$f(\boldsymbol{x}) = \sum_j w_j * x_j + b, \tag{1.2}$$

where $\boldsymbol{x}$ is a feature vector, x_j is a single feature, and w_j and b are the model parameters, also called weights and biases. Neural networks are more complex, but are well-suited to also model non-linear relationships.

In general, there is a tradeoff between selecting a complex function that fits the training data well and a simpler function that may generalize better [39, 59]. This is called the **bias-variance trade-off**:

[1] Here, the words *model* and *function* are used interchangeably for the same concept of a machine learning model. With *model* the inference capabilities are emphasized, and with *function* its mathematical properties are emphasized.

When the function is too simple for a comparably large sample size, it creates a strong bias, which can lead to **underfitting**. On the other hand, when a function is too complex for a comparably small sample size, it allows a large variance, which can lead to **overfitting**. A good model should neither be underfitted, nor overfitted.

Regularization can help to prevent overfitting. Common techniques are L_1- or L_2-Regularization, which add a penalty term to the loss function that encourage the model weights to be small:

$$\hat{f} := \underset{f \in \mathcal{F}}{\arg\min} \left(R_{\mathcal{D}}(f) + \lambda \Omega(f) \right), \quad \Omega(f) = \sum_j |w_j|^q, \quad q = \{1, 2\} \tag{1.3}$$

Here, λ defines the regularization strength, and the regularizer Ω itself quantifies the model complexity. In particular, L_1 regularization tends to produce a sparse model [59], i.e., some weights are set to zero. Other forms of regularization are early stopping, which involves monitoring the loss on an additional validation dataset, or data augmentation, which synthetically increases the data size.

In practice, learning algorithms use optimization techniques to find a model with a minimal loss. A fundamental method is **gradient descent**, which computes the gradient of the loss function with respect to the model parameters and then updates the model parameters in the opposite direction so that the loss becomes smaller. An extension is stochastic gradient descent (SGD), which evaluates the gradient only on a random sample or subset of the dataset and then directly updates the model parameters. If the dataset is split into subsets, these are also called mini-batches. There are plenty of further learning algorithms, e.g., the Adam optimization method.

Finally, it is worth to mention that the learning algorithm is usually executed for several **epochs**, each being a single pass through the entire dataset. Every epoch consists of the following steps: 1. Forward propagation (Pass input data through current model and compute predicted output), 2. Loss calculation (Predicted output vs. target output), 3. Backward propagation (Compute gradients of loss with respect to model parameters), 4. Parameter update (Optimization technique, e.g., SGD).

Artificial Neural Networks

Neural Networks are a specific function class that can be used in machine learning. As illustrated in Figure 1.2, a neural network is composed of several **layers** – usually one input, several hidden, and one output layer – each containing several units (also called **neurons**) [5, 59, 30]. The input to a single unit is computed as a weighted sum of the outputs from the previous layers. In fact, this is like a linear model as shown in Equation 1.2. This value is then fed into a **non-linear activation function**, such as the rectified linear unit, $h(z) = \texttt{max}(0, z)$, to derive the output of that unit. The functional form of such a multilayer feed-forward neural network model is then the concatenation of the individual layers and their activation functions. For example, a network with 2 hidden layers and 1 output layer has the form

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{h}_3 \left(\boldsymbol{W}_3 \boldsymbol{h}_2 \left(\boldsymbol{W}_2 \boldsymbol{h}_1 \left(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1 \right) + \boldsymbol{b}_2 \right) + \boldsymbol{b}_3 \right), \tag{1.4}$$

where $\boldsymbol{x}$ is the input vector, $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$ are the weight matrices and bias vectors of layer i, $\boldsymbol{h}_i$ are the non-linear activation functions, and $\boldsymbol{f}(\boldsymbol{x})$ is the output vector. The **weights** and **biases** are the parameters that are learned using machine learning algorithms.

The training of neural networks with more than a few hidden layers is called **deep learning**. Every layer constitutes a **representation** of the input data in a **latent space**.

The composition of a neural network (also called **architecture**) can have versatile forms. The most

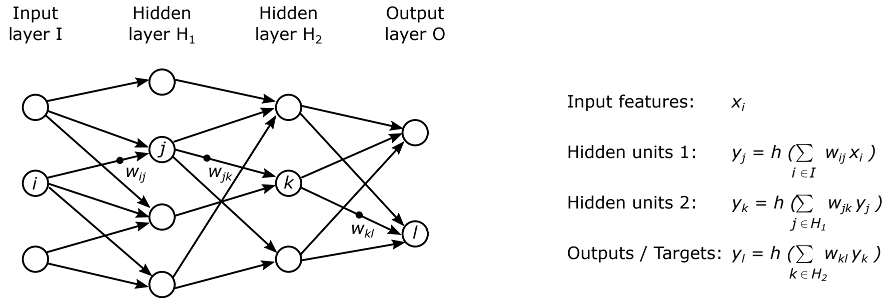

Input layer I
Hidden layer H$_1$
Hidden layer H$_2$
Output layer O
i
j
k
l
w_{ij}
w_{jk}
w_{kl}
Input features: x_i
Hidden units 1: $y_j = h\,(\sum_{i \in I} w_{ij} x_i)$
Hidden units 2: $y_k = h\,(\sum_{j \in H_1} w_{jk} y_j)$
Outputs / Targets: $y_l = h\,(\sum_{k \in H_2} w_{kl} y_k)$

Figure 1.2: **Feed-Forward Neural Network.** This simple neural network model has 1 input, 2 hidden, and 1 output layer. x_i are the input features and y_l are the outputs. The values at each unit are computed in a forward pass as a weighted sum of the previous layer and by applying an activation function h. The weights w are the parameters that are trained using machine learning algorithms. For simplicity, the bias parameters are neglected in this Figure. (Figure recreated from [30]).

basic architecture is the feed-forward neural network, as described above. A more specific architecture type are **convolution neural networks**, which are well suited for processing data with a grid-like structure, such as images. These involve in at least one layer the application of a convolutional kernel, which is a small weight matrix (also known as filter) that is applied to only a small local patch of the input data, but which is slided over the whole input. This type of layers is well-suited to detect local patterns, such as edges in images. A further type are recurrent neural networks, which are well suited for processing sequential data such as text. Another prominent architecture is the Autoencoder, which is used for unsupervised learning, and which is also used together with the so-called Attention mechanism in the more modern transformer architectures [66].

Neural networks usually contain a huge **number of parameters** that need to be learned. The simple example of the 2-hidden layer neural network above (Figure 1.2 and Equation 1.4) has 39 parameters[2]. However, neural networks in real-life applications usually have much more and wider layers. The famous convolutional neural network architecture LeNet-5 [31], which was designed for handwritten digit recognition, contains approximately 60,000 parameters. The groundbreaking AlexNet architecture [29], which was developed in 2012 and demonstrated the superiority of neural networks in the ImageNet object detection and image classification challenge [17], has approximately 60 million parameters. Latest neural networks for natural language processing can have even more than billions of parameters [9].

The large model capacity of neural networks requires **large amounts of training data**. For example, the LeNet-5 architecture can be trained with the MNIST dataset, which comprises handwritten images across 10 digit classes and consists of a total number of 60,000 images. The slightly more complex AlexNet architecture was trained with the ImageNet-1K dataset, which comprises images of 1000 different object classes and consists of over 1 million training images. As explained in Section 1.2.2, a large and diverse dataset is generally required to avoid overfitting and to allow generalization. However, recent studies investigated and advanced the classical bias-variance trade-off for very large neural networks. They observed a double-descent risk curve, which describes the effect that in an over-parameterized regime a decreasing test error can be observed [4, 40].

For the training of neural networks, the **initialization** of its weights and biases plays an important

[2] (3*4 + 4) + (4*3 + 3) + (3*2 + 2) = 39

role ([63]). The weights are usually initialized randomly and follow a uniform or normal distribution ([22, 24]). Random initialization aims at symmetry breaking (i.e., preventing that neurons learn the same features) and avoids the vanishing gradient problem in the beginning of training. The most appropriate distribution depends on the used activation function and the size of the previous network layer.

The neural network parameters can also be initialized by reusing the parameters from a **pre-trained** model, which can then be fine-tuned on given training data for the target task. [19] studied the advantages of unsupervised pre-training and found that it leads to better performing classifiers and is beneficial with small training datasets. They concluded that this is not only an improved optimization procedure but also leads to better generalization. They also found that the largest effect on performance benefits comes from pre-training the early layers, which often represent general features and low-level statistics of the data. In the last years, supervised pre-training on the ImageNet dataset [17] became a common **transfer learning** approach, especially for computer vision tasks [70, 26, 41].

Despite the modelling success of neural networks, their complexity brings certain challenges, such as their massive data requirements and their lack of explainability. In particular, the data need has raised important questions about how to learn from small data, how to generalize to unseen domains, and how to ensure model **robustness** [68, 67, 25]. Moreover, the large number of neural network parameters can make their decision process intransparent and hard to interpret [50]. But especially when they are applied in critical domains, such as healthcare, finance, or autonomous driving, it is important that their decisions are **explainable** and consistent with laws and regulations.

1.3 Thesis Contributions

1.3.1 List of Publications

This cumulative dissertation is comprised of the following peer-reviewed publications, to which I was the primary contributor. The following list is sorted topic-wise and is consistent with the presentation order in this dissertation.

P1) L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauckhage, J. Schuecker. **Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems**. *IEEE Transactions on Knowledge and Data Engineering*. 2021. http://doi.org/10.1109/TKDE.2021.3079836
Preprint:
L. von Rueden, S. Mayer, J. Garcke, C. Bauckhage, J. Schuecker. **Informed Machine Learning - Towards a Taxonomy of Explicit Integration of Knowledge into Machine Learning**. *arXiv preprint arXiv:1903.12394*. 2019.

P2) L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, J. Garcke. **Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions**. *Advances in Intelligent Data Analysis (IDA)*. 2020. http://doi.org/10.1007/978-3-030-44584-3_43

P3) L. von Rueden, S. Houben, K. Cvejoski, J. Garcke, C. Bauckhage, N. Piatkowski. **Informed Pre-Training of Neural Networks Using Prototypes from Prior Knowledge**. *Accepted at: International Joint Conference on Neural Networks (IJCNN)*. 2023.
Preprint:
L. von Rueden, S. Houben, K. Cvejoski, C. Bauckhage, N. Piatkowski. **Informed Pre-Training on Prior Knowledge**. *arXiv preprint arXiv:2205.11433*. 2022.

P4) L. von Rueden, T. Wirtz, F. Hueger, J.D. Schneider, C. Bauckhage, N. Piatkowski. **Street-Map Based Validation of Semantic Segmentation in Autonomous Driving**. *International Conference on Pattern Recognition (ICPR)*. 2020. http://doi.org/10.1109/ICPR48806.2021.9413292
Preprint:
L. von Rueden, T. Wirtz, F. Hueger, J.D. Schneider, C. Bauckhage. **Towards Map-Based Validation of Semantic Segmentation Masks**. *Workshop on AI for Autonomous Driving (AIAD), International Conference on Machine Learning (ICML)*. 2020.

P5) L. von Rueden, J. Garcke, C. Bauckhage. **How Does Knowledge Injection Help in Informed Machine Learning?** *Accepted at: International Joint Conference on Neural Networks (IJCNN)*. 2023.

Moreover, I have also contributed to the following publications. These are not part of this dissertation:

P6) K. Beckh, S. Müller, M. Jakobs, V. Toborek, H. Tan, R. Fischer, P. Welke, S. Houben, L. von Rueden. **Harnessing Prior Knowledge for Explainable Machine Learning: An Overview**. *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*. 2023

P7) M. Günder, N. Piatkowski, L. von Rueden, R. Sifa, C. Bauckhage. **Towards Intelligent Food Waste Prevention: An Approach Using Scalable and Flexible Harvest Schedule Optimization With Evolutionary Algorithms**. *IEEE Access*. 2021.

P8) J. Wörmann, D. Bogdoll, E. Bührle, ..., L. von Rueden, ..., S. Zwicklbauer . **Knowledge Augmented Machine Learning with Applications in Autonomous Driving: A Survey**. *arXiv preprint arXiv:2205.04712*. 2022.

1.3.2 List of Key Contributions

The goal of this PhD thesis is the unification of Informed ML through general, systematic frameworks for all abstraction levels: From concepts, over methods with applications, to theory and systematic analysis. The key contributions of the relevant publications P1-P5 (see list in Section 1.3.1) are:

P1) Informed ML: Integrating Prior Knowledge into Data-Driven Learning Systems – Concept, Taxonomy, and Survey

a) Proposition of a concept for Informed ML and definition as learning from a hybrid information source that consists of data and prior knowledge

b) Development of a taxonomy that classifies approaches according to knowledge source, formal representation type, and integration stage in the ML pipeline

c) Survey and description of available approaches

P2) Combining ML and Simulation to Hybrid AI

a) Proposition of structural frameworks for ML (turning data into models) and simulation (turning models into data)

b) Identification of combination possibilities

P3) Informed Pre-Training of Neural Networks using Knowledge Prototypes

a) Proposition of a novel approach: Informed Pre-Training on knowledge prototypes

b) Improvements in generalization and out-of-distribution robustness

c) Investigation of neural network layers that are affected by the knowledge transfer (Informed Transfer Learning)

P4) Geo-Informed Validation of ML Models for Autonomous Driving

a) Proposition of novel approach: Checking knowledge conformity with street maps

b) Application to semantic segmentation models for traffic scene perception

P5) Quantifying the Benefits: How Knowledge Injection Helps in Informed ML – Metrics, Theory and Systematic Analysis

a) Proposition of a catalogue of performance metrics that are relevant for Informed ML

b) Development of a framework for an Informed Learning Theory: Representation of data and knowledge in function space

c) Systematic Analysis: Dependency of performance improvements on distance between data and knowledge

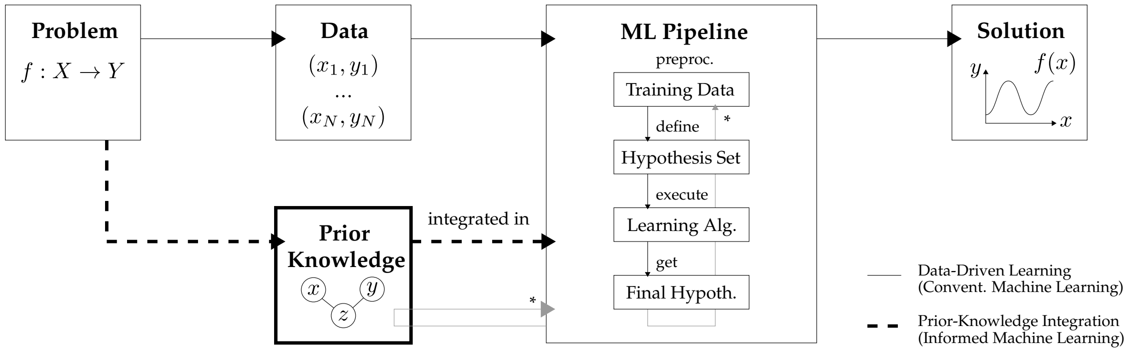


Figure 1.3: **P1) Informed ML: Concept.** The Informed ML pipeline requires a hybrid information source with two components: Data and prior knowledge. In conventional ML, knowledge is used for data preprocessing and feature engineering, but this process is deeply intertwined with the learning pipeline (*). In contrast, in *Informed ML* prior knowledge comes from an independent source, is given by formal representations (e.g., by algebraic equations, logic rules, or knowledge graphs), and is explicitly integrated.

1.3.3 Contributions Summary

In the following, the contents and the contributions of the published research papers P1-P5 (see list in Section 1.3.1) are shortly described for an overview.

P1) Informed ML: Integrating Prior Knowledge into Data-Driven Learning Systems – Concept, Taxonomy, and Survey

Despite its great success, ML can have its limits when dealing with insufficient training data. A potential solution is the additional integration of prior knowledge into the training process, which leads to the notion of *Informed Machine Learning (Informed ML)*. In this paper, we present a structured overview of various approaches in this field. We provide a definition and propose a concept for Informed ML that illustrates its building blocks and distinguishes it from conventional ML (see Figure 1.3). We introduce a taxonomy that serves as a classification framework for Informed ML approaches. It considers the source of knowledge, its representation, and its integration into the ML pipeline. Based on this taxonomy, we survey related research and describe how different knowledge representations such as algebraic equations, logic rules, or simulation results can be used in learning systems. This evaluation of numerous papers on the basis of our taxonomy uncovers key methods in the field of Informed ML.

Further results, such as the developed taxonomy, are presented in the paper summary in Chapter 2.

Central research question: What is the fundamental concept of Informed ML, and how can existing approaches for integrating prior knowledge into data-driven learning be structurally classified? (see Section 1.1, Question 1)

Connection to subsequent papers: The definition and concept of Informed ML is used as the basis for all papers (P1-P5). The taxonomy helps to classify other approaches or identify opportunities

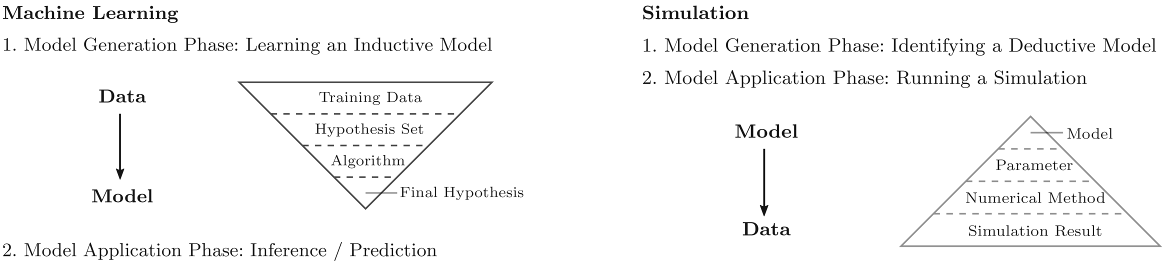

Machine Learning
1. Model Generation Phase: Learning an Inductive Model
Data
Model
Training Data
Hypothesis Set
Algorithm
Final Hypothesis
2. Model Application Phase: Inference / Prediction
Simulation
1. Model Generation Phase: Identifying a Deductive Model
2. Model Application Phase: Running a Simulation
Model
Data
Model
Parameter
Numerical Method
Simulation Result

Figure 1.4: **P2) Combining ML and Simulation: Structural Frameworks.** These frameworks illustrate the components of ML (left) and Simulation (right). The focus of ML is to turn data into models (i.e., discover new knowledge), whereas the focus of simulations is to turn models into data (i.e., transform prior knowledge into data). Both approaches can be combined at different stages to simulation-assisted ML, or ML-assisted simulation.

for novel methods, such as the proposed Informed Pre-Training (P3) and Geo-Informed Validation (P4). Moreover, the survey revealed a heterogeneity of existing approaches, which motivates the development of an Informed Learning Theory and a systematic analysis framework (P5).

P2) Combining ML and Simulation to Hybrid AI

In this paper, we describe the combination of ML and simulation towards a hybrid modelling approach. Such a combination of data-based and knowledge-based modelling is motivated by applications that are partly based on causal relationships, while other effects result from hidden dependencies that are represented in huge amounts of data. Our aim is to bridge the knowledge gap between the two individual communities from ML and simulation to promote the development of hybrid systems. We present conceptual frameworks that describe the stages of ML in terms of transforming data into models, and the stages of simulation in terms of transforming models into data (see Figure 1.4). The frameworks help to identify potential combination approaches and we employ it to give an overview of simulation-assisted ML and ML-assisted simulation.

The results are further presented in the respective paper summary in Chapter 3.

Central research question: What is the fundamental concept of Informed ML (here: Hybrid AI), and how can existing approaches for integrating prior knowledge (here: simulations) into data-driven learning be structurally classified? (see Section 1.1, Question 1)

Connection to preceding and subsequent papers: The conceptual frameworks in this paper are an extension of the Informed ML concept and taxonomy (P1). In particular, simulation-assisted ML is a specific type of Informed ML. Moreover, simulation can transform prior knowledge (e.g., physical formulas, or geospatial prototypes) into data representations, which is relevant for the proposed methods in the subsequent papers (P3-P5).

P3) Informed Pre-Training of Neural Networks using Knowledge Prototypes

We present a novel approach for hybrid AI and propose Informed Pre-Training on prototypes from prior knowledge. Generally, when training data is scarce, the incorporation of additional knowledge

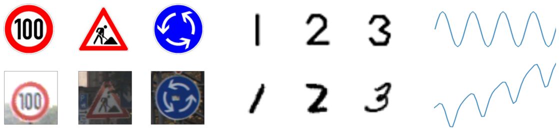


Figure 1.5: **P3) Informed Pre-Training.** Our idea is that data distributions often follow domain invariant relationships that are given by prototypes from prior knowledge. The figure shows examples for such knowledge prototypes (top) and corresponding natural data items (bottom) for three datasets: GTSRB (left), MNIST (middle), CO2 (right). We show that Informed Pre-Training on such prior knowledge is possible and significantly improves generalization and robustness.

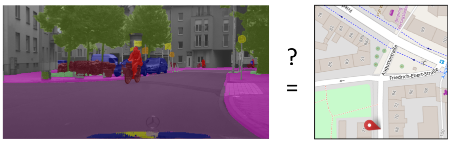


Figure 1.6: **P4) Geo-Informed Validation.** We use Informed ML to validate autonomous driving models with prior knowledge from street maps. For example, the left image shows a segmentation of a traffic scene in the Cityscapes dataset [15] and the right image shows the corresponding map [43]. Here, a road intersection to the right is not detected in the segmentation, but is given by the map. Such potential errors of AI modules can be identified with Informed ML.

can assist the learning process of neural networks. An approach that recently gained a lot of interest is *Informed* ML, which integrates prior knowledge that is explicitly given by formal representations, such as graphs or equations. However, the integration often is application-specific and can be time-consuming. Another more straightforward approach is *pre-training* on other large datasets, which allows to reuse knowledge that is implicitly stored in trained models. This raises the question, if it is also possible to pre-train a neural network on a small set of knowledge representations.

In this paper, we investigate this idea and propose *Informed Pre-Training on knowledge prototypes*. Such prototypes are often available and represent characteristic semantics of the domain (see Figure 1.5). We show that it (i) improves generalization capabilities, (ii) increases out-of-distribution robustness, and (iii) speeds up learning. Moreover, we analyze which parts of a neural network model are affected most by our Informed Pre-Training approach. We discover that (iv) improvements come from deeper layers that typically represent high-level features, which confirms the transfer of semantic knowledge. This is a before unobserved effect and shows that Informed Transfer Learning has additional and complementary strengths to existing approaches.

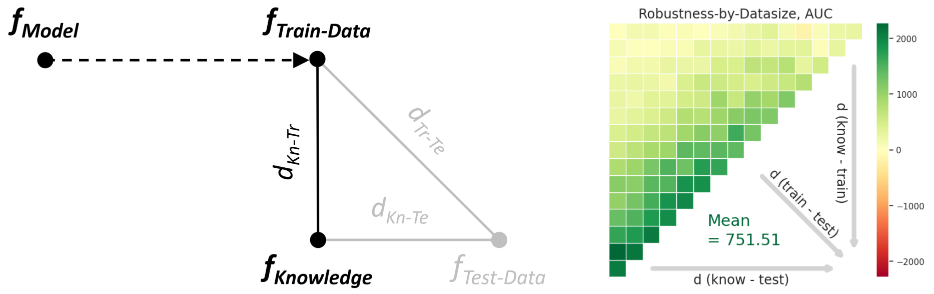
The results are further outlined in the paper summary in Chapter 4.

Central research question: Is it possible to integrate prior knowledge into ML in a universal way, and how? (see Section 1.1, Question 2)

Connection to preceding and subsequent papers: The taxonomy and survey of Informed ML (P1) helped to find out that the novel approach of Informed Pre-Training had not been studied before. The advantage of this novel approach is that it is a unified approach for all the prior knowledge representations of the taxonomy (P1), e.g., algebraic equations, spatial invariances, or knowledge graphs. Simulation (P2) can be used to transform the representations into data format. In the subsequent paper (P5), the novel Informed Pre-Training is further investigated in terms of the distance between data and knowledge prototypes.

P4) Geo-Informed Validation of ML Models for Autonomous Driving

AI for autonomous driving must meet strict requirements on safety and robustness, which motivates the thorough validation of ML models. However, current validation approaches mostly require ground



(a) Theoretical Framework:
Representation in Function Space.

(b) Systematic Analysis:
Benefit Dependency on Distances.

Figure 1.7: **P5) Theoretical Framework and Systematic Analysis.** In Informed ML, prior knowledge is integrated into data-based learning [53]. To better understand its effect, we propose a framework that represents data and knowledge in a function space (See (a)). We analyze how the distances between knowledge, train and test data influence the potential model improvements. E.g., we find that Informed ML greatly improves model robustness, especially when the knowledge is close to out-of-distribution data (See (b)).

truth data and are thus both cost-intensive and limited in their applicability. We propose to overcome these limitations by a model agnostic, Geo-Informed Validation using a-priori knowledge from street maps. In particular, we show how to validate semantic segmentation masks and demonstrate the potential of our approach using OpenStreetMap. We introduce validation metrics that indicate false positive or negative road segments. Besides the validation approach, we present a method to correct the vehicle's GPS position so that a more accurate localization can be used for the street-map based validation. Lastly, we present quantitative results on the Cityscapes dataset indicating that our validation approach can indeed uncover errors of semantic segmentation models (See Figure 1.6).

The results are outlined in the paper summary in Chapter 5.

Central research question: Is it possible to integrate prior knowledge into ML in a universal way, and how? (see Section 1.1, Question 2)

Connection to preceding and subsequent papers: The taxonomy and survey of Informed ML (P1) helped to find out that the novel approach of Geo-Informed Validation had not been studied before.

P5) Quantifying the Benefits: How Knowledge Injection Helps in Informed ML – Metrics, Theory and Systematic Analysis

Informed ML describes the injection of prior knowledge into learning systems. It can help to improve generalization, especially when training data is scarce. However, the field is so application-driven that general analyses about the effect of knowledge injection are rare. This makes it difficult to transfer existing approaches to new applications, or to estimate potential improvements. Therefore, in this paper, we present a framework for quantifying the value of prior knowledge in Informed ML. Our main contributions are three-fold. Firstly, we propose a set of relevant metrics for quantifying the benefits of knowledge injection, comprising in-distribution accuracy, out-of-distribution robustness, and knowledge conformity. We also introduce a metric that combines performance improvement and data reduction. Secondly, we present a theoretical framework that represents prior knowledge in a function space and relates it to data representations and a trained model (See Figure 1.7(a)). This suggests that the distances between knowledge and data influence potential model improvements. Thirdly, we perform a systematic experimental study with controllable toy problems (See Figure 1.7(b)). All in all, this helps to find general answers to the question how knowledge injection helps in Informed ML.

The results are outlined in the paper summary in Chapter 6.

Central research question: How can the benefits of Informed ML be quantified, and what are the requirements for the injected knowledge? (see Section 1.1, Question 3)

Connection to preceding and subsequent papers: The concept of Informed ML (P1) points out the hybrid information source consisting of training data and prior knowledge. These are used as two counterparts in the theoretical framework and systematic analysis of this paper. The surveys (P1, P2) identified the heterogeneity of approaches, which motivated the systematization in this paper. Furthermore, the proposed approach of Informed Pre-Training (P3) is further investigated in this paper to find out what the requirements for the injected knowledge are.

CHAPTER 2

P1) Informed ML: Integrating Prior Knowledge into Data-Driven Learning Systems – Concept, Taxonomy, and Survey

The research summarized in this chapter has been published in the following paper [53]:

P1) L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauckhage, J. Schuecker. **Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems**. *IEEE Transactions on Knowledge and Data Engineering*. 2021. http://doi.org/10.1109/TKDE.2021.3079836

An earlier preprint version has been published here [54]:

L. von Rueden, S. Mayer, J. Garcke, C. Bauckhage, J. Schuecker. **Informed Machine Learning - Towards a Taxonomy of Explicit Integration of Knowledge into Machine Learning**. *arXiv preprint arXiv:1903.12394*. 2019.

2.1 Research Question

In this paper, we investigate the fundamental concept of Informed ML and how existing approaches can be structurally classified (see Section 1.1, central question No.1). In particular, we answer the following open research questions:

- What is the definition of Informed ML?
- What kind of prior knowledge can be integrated into ML, and how?
- Are there similarities between the various available approaches and how can they be unified?

2.2 Results Summary

Our contributions in this paper are threefold: We propose an abstract **concept** for Informed ML that clarifies its building blocks and relation to conventional ML (See Figure 1.3). It states that

Informed ML uses a hybrid information source that consists of data and prior knowledge, which comes from an independent source and is given by formal representations. Our main contribution is the introduction of a **taxonomy** that classifies Informed ML approaches, which is novel and the first of its kind (See Figure 2.1). It contains the dimensions of the knowledge source, its representation, and its integration into the ML pipeline. We put a special emphasis on categorizing various knowledge representations, since this may enable practitioners to incorporate their domain knowledge into ML processes. Moreover, we present a literature **survey** and a description of available approaches and explain how different knowledge representations, e.g., algebraic equations, logic rules, or simulation results, can be used in Informed ML.

In the following, we describe our developed taxonomy. Further results can be found in the paper itself (see Appendix A).

Taxonomy

Our guiding question is how prior knowledge can be integrated into the ML pipeline and our answers particularly focus on three aspects:

1. **Source**:
   Which source of knowledge is integrated?
2. **Representation**:
   How is the knowledge represented?
3. **Integration**:
   Where in the learning pipeline is it integrated?

Based on a comparative and iterative literature survey, we identified a taxonomy with dimensions for these three aspects (See Figure 2.1). Each dimension contains a set of elements that represent the spectrum of different approaches found in the literature.

With respect to knowledge sources, we found three broad categories: Rather specialized and formalized scientific knowledge, everyday life's world knowledge, and more intuitive expert knowledge. For scientific knowledge, we found the most Informed ML papers. With respect to knowledge representations, we found versatile and fine-grained approaches and distilled eight categories (Algebraic equations, differential equations, simulation results, spatial invariances, logic rules, knowledge graphs, probabilistic relations, and human feedback). Regarding knowledge integration, we found approaches for all stages of the ML pipeline, from the training data and the hypothesis set, over the learning algorithm, to the final hypothesis. However, most Informed ML papers consider the two central stages.

Depending on the perspective, the taxonomy can be regarded from either one of two sides: An application-oriented user might prefer to read the taxonomy from left to right, starting with some given knowledge source and then selecting representation and integration. Vice versa, a method-oriented developer or researcher might prefer to read the taxonomy from right to left, starting with some given integration method. For both perspectives, knowledge representations are important building blocks and constitute an abstract interface that connects the application- and the method-oriented side.

We observe that, while various paths through the taxonomy are possible, specific ones occur more frequently and we will call them main paths. For example, we often observed the approach that scientific knowledge is represented in algebraic equations, which are then integrated into the learning

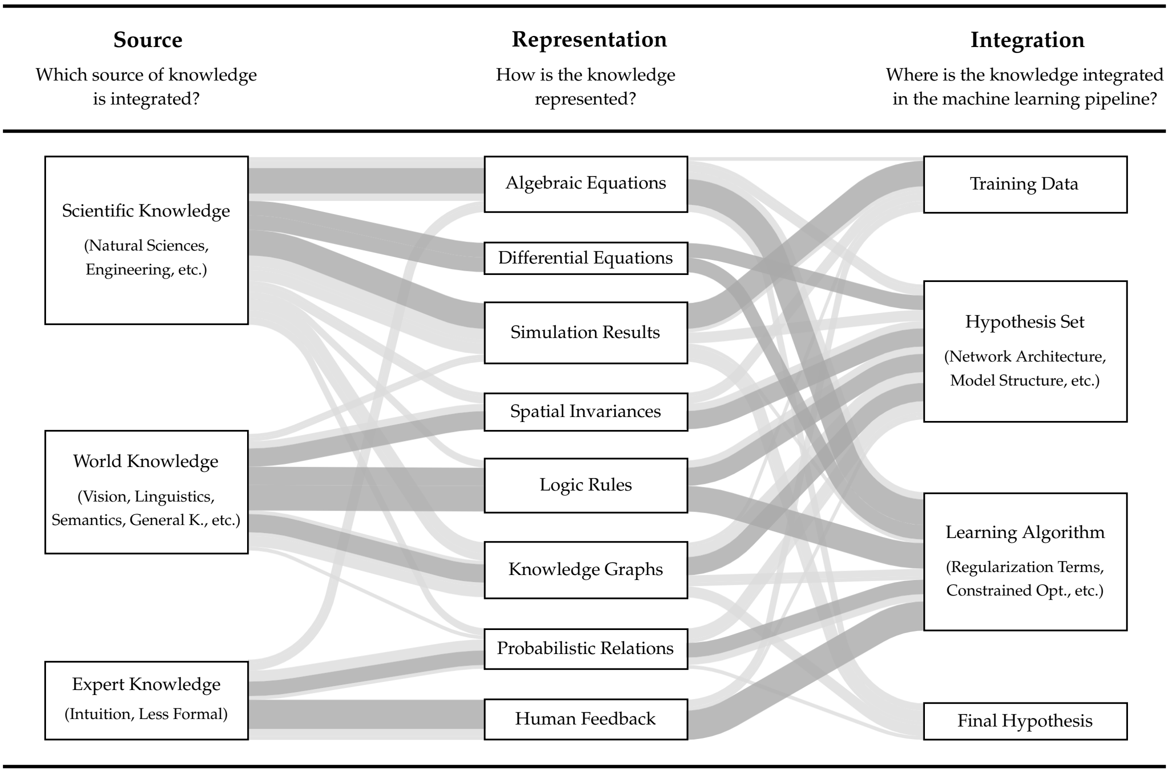

Source
Which source of knowledge is integrated?
Representation
How is the knowledge represented?
Integration
Where is the knowledge integrated in the machine learning pipeline?
Scientific Knowledge
(Natural Sciences, Engineering, etc.)
World Knowledge
(Vision, Linguistics, Semantics, General K., etc.)
Expert Knowledge
(Intuition, Less Formal)
Algebraic Equations
Differential Equations
Simulation Results
Spatial Invariances
Logic Rules
Knowledge Graphs
Probabilistic Relations
Human Feedback
Training Data
Hypothesis Set
(Network Architecture, Model Structure, etc.)
Learning Algorithm
(Regularization Terms, Constrained Opt., etc.)
Final Hypothesis

Figure 2.1: **P1) Informed ML: Taxonomy.** This taxonomy serves as a classification framework for *Informed ML* and structures approaches according to the three above analysis questions about the *knowledge source*, *knowledge representation* and *knowledge integration*. Based on a comparative and iterative literature survey, we identified for each dimension a set of elements that represent a spectrum of different approaches. The size of the elements reflects the relative count of papers. We combine the taxonomy with a Sankey diagram in which the paths connect the elements across the three dimensions and illustrate the approaches that we found in the analyzed papers. The broader the path, the more papers we found for that approach. Main paths (at least four or more papers with the same approach across all dimensions) are highlighted in darker grey and represent central approaches of Informed ML.

algorithm, e.g., the loss function. As another example, we often found that world knowledge, such as linguistics, is represented by logic rules, which are then integrated into the hypothesis set, e.g., the network architecture.

Further details on the concept, taxonomy, or the survey can be found in the paper itself (see Appendix A).

2.3 Author's Contribution

The idea to do a survey about knowledge integration in ML and to derive a concept of Informed ML was due to C. Bauckhage, J. Schuecker, and me. The concept itself was developed by me, J. Schuecker, and S. Mayer. The idea to develop a taxonomy for a structured classification of available approaches was due to me. The main parts of the paper (Sections 1-4, 6-8) were written by me, with support from J. Schuecker, and S. Mayer.

The literature about individual Informed ML approaches was surveyed and described in the paper (Section 5) by individual expert groups of the following co-authors (ordered by paper's author list): 1) Algebraic Equations: L. von Rueden, R. Heese, M. Walczak; 2) Differential Equations: S. Mayer, R. Heese, J. Schuecker; 3) Simulation Results: L. von Rueden, S. Mayer, R. Heese, M. Walczak, J. Garcke; 4) Spatial Invariances: L. von Rueden, B. Georgiev, M. Walczak; 5) Logic Rules: L. von Rueden, B. Kirsch; 6) Knowledge Graphs: S. Giesselbach, A. Pick, J. Schuecker; 7) Probabilistic Relations: A. Pick, J. Schuecker; 8) Human Feedback: K. Beckh, R. Ramamurthy.

The whole research project was managed by me.

CHAPTER 3

P2) Combining ML and Simulation to Hybrid AI

The research summarized in this chapter has been published in the following paper [52]:

P2) L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, J. Garcke. **Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions**. *Advances in Intelligent Data Analysis (IDA)*. 2020. http://doi.org/10.1007/978-3-030-44584-3_43

3.1 Research Question

In this paper, we further investigate the concept of Informed ML and a structured classification of existing approaches (see Section 1.1, central question No.1), but here, we focus on prior knowledge in the form of simulations. In particular, we answer the following open research questions:

- What is the difference between ML and simulation, especially in terms of data- vs. knowledge-based modelling?
- How can they be combined to hybrid AI?
- Can simulation be used to transform prior knowledge into data representation?

3.2 Results Summary

ML and *simulation* have a similar goal: To predict the behavior of a system with data analysis and mathematical modelling. On the one side, ML has shown great successes in fields like image classification [29], language processing [36], or socio-economic analysis [8], where causal relationships are often only sparsely given but huge amounts of data are available. On the other side, simulation is traditionally rooted in natural sciences and engineering, e.g. in computational fluid dynamics [60], where the derivation of causal relationships plays an important role, or in structural mechanics for the performance evaluation of structures regarding reactions, stresses, and displacements [6].

However, some applications can benefit from combining ML and simulation. Such a hybrid approach can be useful when the processing capabilities of classical simulation computations can not handle the available dimensionality of the data, for example in earth system sciences [49], or when the behavior of a system that is supposed to be predicted is based on both known, causal relationships and unknown, hidden dependencies, for example in risk management [38].

However, such challenges are in practice often still approached distinctly with either ML or simulation, apparently because they historically originate from distinct fields. This raises the question how these two modelling approaches can be combined into a hybrid approach to foster intelligent data analysis. Here, a key challenge in developing a hybrid modelling approach is to bridge the knowledge gap between the two individual communities, which are mostly either experts for ML or experts for simulation.

Our goal is to make the key components of the two modelling approaches *ML* and *simulation* transparent and to show the versatile, potential combination possibilities to inspire and foster future developments of hybrid systems.

The contributions of this paper are: 1. **Conceptual frameworks** of ML and simulation that serve as an orientation aid for comparing and combining both methodologies, 2. a **structured overview of combinations** of both modelling approaches, and 3. our **vision of a hybrid approach** with a stronger interplay of data- and simulation based analysis.

Conceptual Frameworks

We developed structural frameworks that point out the individual components of both approaches, ML and simulation (see Figure 1.4).

The main goal of ML is that a machine automatically learns a model that describes patterns in given data. ML consists of two phases 1. model generation, and 2. model application, where the focus is usually made on the first phase, in which an inductive model is learned from data. The components of this phase are the training data, a hypothesis set, a learning algorithm, and a final hypothesis [1, 53]. It describes the finding of patterns in an initially large data space, which are finally represented in a condensed form by the final hypothesis. This can be described as a bottom-up approach.

The goal of a simulation is to predict the behavior of a system or process for a particular situation. Simulation comprises the two phases 1. model generation, and 2. model application, where the focus often is on the second phase, in which an earlier identified deductive model is used to create simulation results. The components of this phase are the simulation model, input parameters, a numerical method, and the simulation result. It describes the unfolding of local interactions from a compactly represented initial model into an expanded data space. This and can be described as a top-down approach.

Combinations: Simulation-Assisted ML and ML-Assisted Simulation

There are several types of Simulation-Assisted ML. Simulations, in particular the simulation results, can be generally integrated into the four different components of ML. The simulation results can be used to a) augment the training data, b) define parts of the hypothesis set in the form of empirical functions, c) steer the training algorithm in generative adversarial networks, or d) verify the final hypothesis against scientific consistency.

Also, there are several types of ML-Assisted Simulation. ML techniques, in particular the final hypothesis, can be used in different simulation components. Exemplary use cases for ML models in simulation are a) model order reduction and the development of surrogate models that offer approximate but simpler solutions, b) the automated inference of an intelligent choice of input parameters for a next simulation run, c) a partly trainable solver for differential equations, or d) the identification of patterns in simulation results for scientific discovery.

Further details can be found in the paper itself (see Appendix B).

3.3 Author's Contribution

The idea for the paper about the combination of ML and simulation was due to me. I developed the structural frameworks, in particular for ML, ML-based simulation, and Simulation-Based ML. S. Mayer supported the development of the simulation framework. I wrote the main part of the paper (Section 1-4, and 6). S. Mayer wrote the section about Industry 4.0 (Section 5).

CHAPTER 4

P3) Informed Pre-Training of Neural Networks Using Knowledge Prototypes

The research summarized in this chapter has been published in the following paper [55]:

P3) L. von Rueden, S. Houben, K. Cvejoski, J. Garcke, C. Bauckhage, N. Piatkowski. **Informed Pre-Training of Neural Networks Using Prototypes from Prior Knowledge**. *Accepted at: International Joint Conference on Neural Networks (IJCNN)*. 2023.

An earlier preprint version has been published here [56]:

L. von Rueden, S. Houben, K. Cvejoski, C. Bauckhage, N. Piatkowski. **Informed Pre-Training on Prior Knowledge**. *arXiv preprint arXiv:2205.11433*. 2022.

4.1 Research Question

In this paper, we investigate how prior knowledge can be integrated with Informed ML in an application-independent, universal way (see Section 1.1, central question No.2). In particular, we answer the following open research questions:

- How can a neural network be initialized with prior knowledge?
- Is it possible to pre-train on a few knowledge representations?
- If yes, how does it improve the neural network model?

4.2 Results Summary

One approach to alleviate problems due to insufficient training data for a specific learning task is to build upon models that have been pre-trained on other large datasets. However, this relies on reusing the implicit information from large datasets, which is not necessarily controllable. Thus, relevant task-specific concepts still need to be learned. Moreover, appropriate large datasets or pre-trained models are not always available. Another promising approach is to inject additional prior knowledge via Informed ML methods, as proposed by [53]. While this ensures an alignment with semantic concepts, the integration of formally represented knowledge into learning algorithms or

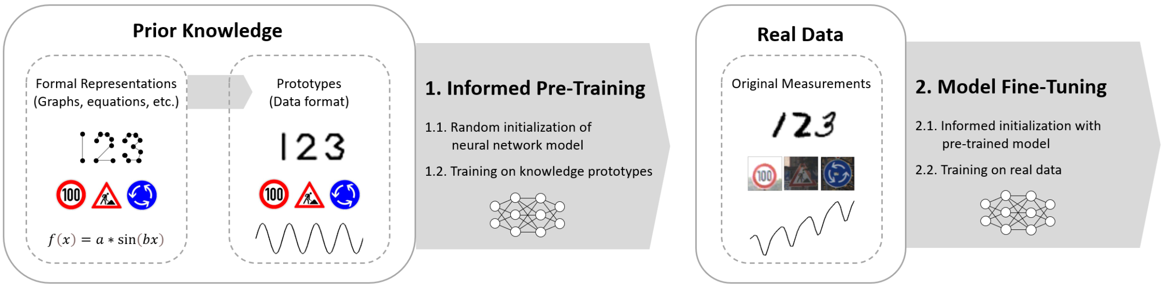

Prior Knowledge
Formal Representations (Graphs, equations, etc.)
Prototypes (Data format)
$f(x) = a * \sin(bx)$
1. Informed Pre-Training
1.1. Random initialization of neural network model
1.2. Training on knowledge prototypes
Real Data
Original Measurements
2. Model Fine-Tuning
2.1. Informed initialization with pre-trained model
2.2. Training on real data

Figure 4.1: **P3) Informed Pre-Training on Knowledge Prototypes.** We propose Informed Pre-Training on prototypes from prior knowledge in order to improve neural network training, especially when real data is scarce. Prior knowledge is often given by formal representations, e.g., by graph structures, image templates, or scientific equations. They can be represented in data space - we then call them "knowledge prototypes". This allows for Informed Pre-Training, i.e., to train a model on prior knowledge. Afterwards, the pre-trained model is fine-tuned on real training data. The Informed Pre-Training leads to significantly increased generalization capabilities and improved robustness.

model architectures can be application-specific and time-consuming, which in turn raises the need for an improved method. This leads us to the questions, how we can we transfer prior knowledge into a neural network? And is it possible to pre-train a neural network on a few knowledge representations?

Our main contributions in this paper are 1) the proposition of the **novel approach Informed Pre-Training** using prototypes from prior knowledge representations, and 2) an investigation of neural network layers that are affected by the knowledge transfer (**Informed Transfer Learning**).

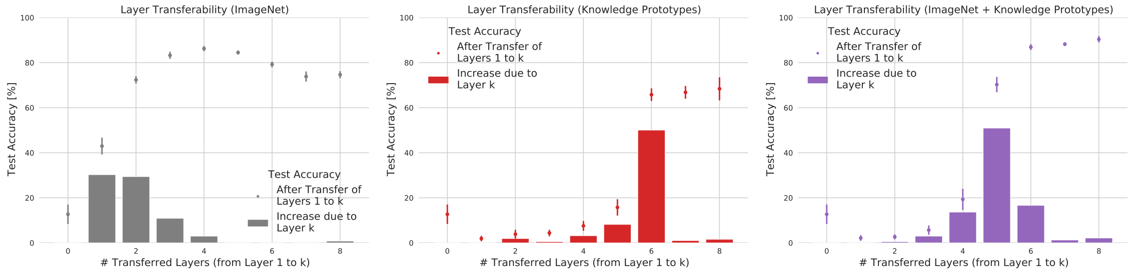
Novel Approach: Informed Pre-Training

We propose the novel approach of Informed Pre-Training on prototypes from prior knowledge. Given prior knowledge is often represented by image templates, graph structures or physical equations. We utilize the fact that these representations can also be given in data or image space, we call them *knowledge prototypes* (see Figure 1.5). These prototypes already reflect major concepts of a target domain, which suggests that pre-training on them can lead to significant improvements.

From a practical point of view our approach consists of the following two main phases, as illustrated in Figure 4.1:

1. Informed Pre-Training
   a) Initialization of neural network model
   b) Training on knowledge prototypes
2. Model fine-tuning
   a) Informed initialization with pre-trained model
   b) Training on real data

In contrast to existing pre-training methods, our method utilizes concise and controllable prior knowledge that is given by a small set of semantic prototypes. In contrast to existing Informed ML methods, our method can be universally applied to various knowledge formalization types and domains.



(a) Pre-Training: ImageNet (Conventional Transfer Learning)

(b) Pre-Training: Knowledge (Informed Transfer Learning)

(c) Pre-Training: ImageNet + Know. (Convent. + Informed Transfer Learning)

Figure 4.2: **P3) Layer Transferability Analysis: Informed Transfer Learning** Importance of individual network layers for different pre-training types. ((a)) Pre-Training on the ImageNet dataset. Improvements come from early layers, which typically represent low-level features. ((b)) Pre-Training on knowledge prototypes (augmented). Improvements come from late layers, which typically represent high-level, semantic concepts. ((c)) Combining both types by using a model pre-trained on ImageNet for the initialization of a subsequent pre-training on knowledge prototypes. After the respective pre-training schemes only the first k layers are transferred. The bar charts highlight the performance gain after fine-tuning due to the transfer of the layers preceding layer k. The experiment shows that knowledge-based, Informed Transfer Learning has an additional and complementary effect to conventional, data-based transfer learning.

It is neither restricted to the types that we use in this paper (i.e., image templates, graph structures, and scientific equations), nor domain specific. Instead, Informed Pre-Training can be used for any kind of prior knowledge that can be represented in a data space, e.g., by rendering or simulation.

Our results show an improvement in test accuracy for small training data by up to 11%. Furthermore, we obtain an increase of 15% on out-of-distribution robustness. Our approach also leads to faster training convergence. Detailed experimental results can be found in the paper itself (see Appendix C).

Layer Transferability Analysis: Informed Transfer Learning

To provide an in-depth analysis we investigate the transfer learning contribution of individual model layers (see Figure 4.2). For traditional data-based pre-training, benefits arise from transferring early layers. In contrast, for our knowledge-based Informed Pre-Training, improvements stem from deeper layers which tend to represent semantic concepts. This is a before unobserved effect, which shows that pre-training on semantic features is viable and significantly different to existing approaches. We refer to this effect as *Informed Transfer Learning*.

We compare our approach to ImageNet pre-training and find that the latter can be further improved by a subsequent Informed Pre-Training on knowledge prototypes. We find an additional increase of 13% in test accuracy. This further confirms the complementary advantages of data-based and knowledge-based pre-training. Further details can be found in the paper itself (see Appendix C).

4.3 Author's Contribution

The idea for Informed Pre-Training was developed by me and N. Piatkowski. I did the implementation, conducted the experiments on MNIST and GTSRB, and wrote the paper. N. Piatkowski derived the prototype initialization bound, and K. Cvejoski conducted additional experiments on the CO2 times series dataset (both are part of the paper's appendix). The idea for the layer transferability analysis and the comparison of data-based vs. knowledge-based transfer learning was due to me. S. Houben supported with discussions especially about model training on the GTSRB traffic sign dataset. All co-authors supported with feedback on the paper manuscript.

CHAPTER 5

P4) Geo-Informed Validation of ML Models for Autonomous Driving

The research summarized in this chapter has been published in the following paper [57]:

P4) L. von Rueden, T. Wirtz, F. Hueger, J.D. Schneider, C. Bauckhage, N. Piatkowski. **Street-Map Based Validation of Semantic Segmentation in Autonomous Driving**. *International Conference on Pattern Recognition (ICPR)*. 2020. http://doi.org/10.1109/ICPR48806.2021.9413292

An earlier preprint version has been published here [58]:

L. von Rueden, T. Wirtz, F. Hueger, J.D. Schneider, C. Bauckhage. **Towards Map-Based Validation of Semantic Segmentation Masks**. *Workshop on AI for Autonomous Driving (AIAD), International Conference on Machine Learning (ICML)*. 2020.

5.1 Research Question

In this paper, we further investigate how prior knowledge can be integrated with Informed ML in a universal way (see Section 1.1, central question No.2). In particular, we answer the following open research questions:

- How can AI models for autonomous driving be validated using prior knowledge?
- How can geospatial knowledge, such as street maps, be used for Informed ML?

5.2 Results Summary

Environmental perception is important for autonomous vehicles in order to assess the surrounding traffic scene and understand its context [13, 45]. A key component is semantic segmentation, which assigns pixel-wise pre-defined class labels to the input images from vehicle's cameras (see e.g. Figure 1.6, left). Current algorithms use machine and deep learning techniques to build models that predict semantic segments and surpass classic computer vision techniques in terms of performance [21, 20].

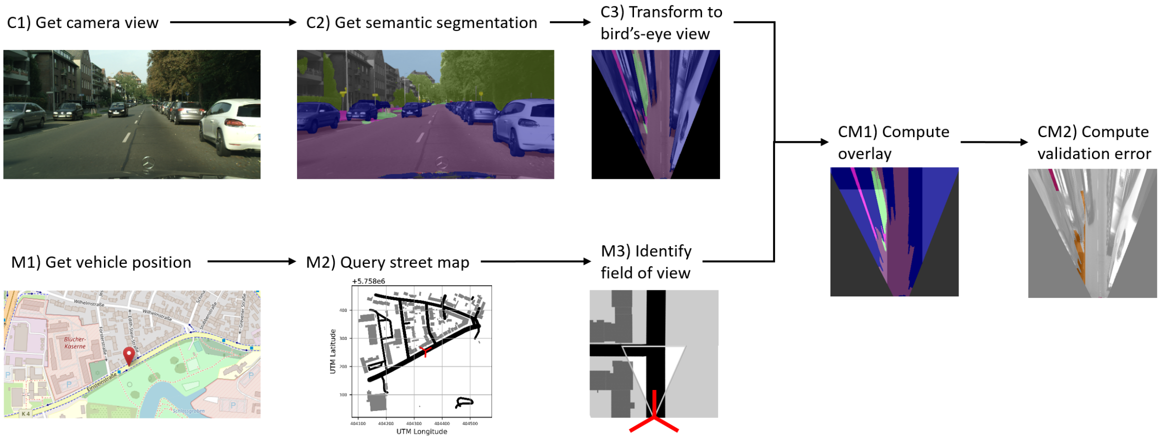

C1) Get camera view
C2) Get semantic segmentation
C3) Transform to bird's-eye view
CM1) Compute overlay
CM2) Compute validation error
M1) Get vehicle position
M2) Query street map
M3) Identify field of view
+5.758e6
UTM Latitude
UTM Longitude

Figure 5.1: **P4) Street-Map Based Validation of Semantic Segmentation Models.** We validate the drivable area in semantic segmentation masks with a-priori geospatial knowledge. For this, we combine the segmentation mask of a given camera view with the street map that corresponds the given vehicle position. We compute an overlay of the segmentation in bird's-eye view and the street-map's field of view. We omit potentially occluding, dynamic objects such as other vehicles or vegetation. The overlay is used to identify validation error regions in the drivable area, which we classify into detected false positives (visualized with orange red pixels) and false negatives (visualized with pink red pixels). The steps with *C* describe tasks involving a *c*amera view and the steps with *M* describe tasks involving a *m*ap view.

Although state-of-the-art neural networks for semantic segmentation achieve promising results, it can still be observed that certain objects of the drivable area are not detected correctly. Moreover, smaller networks being used for embedded purposes are often comparatively less accurate than state-of-the-art networks with arbitrary size. As an example, roads and pedestrian walks could be mixed up in difficult lighting conditions or unusual terrain, such as in Figure 1.6, which could result in false negative or false positive road segments in the prediction.

We propose to support the goal of safe AI in autonomous driving by applying the idea of Informed ML [53] and validate learned models with a-priori geospatial knowledge. We suggest to compare semantic segmentation masks to the structured semantic information in street maps, as illustrated in Figure 1.6, and present a novel method that computes the overlap of drivable area between the segmentation output and the map. Our approach is inspired by how human drivers would perceive environments: When they find themselves in a new environment, they often consult external knowledge sources such as street maps and compare what they see in their vicinity to what they see on the map.

Our main contributions are the following. First, we introduce the **novel approach of a Geo-Informed Validation using street maps** for semantic segmentation models in order to identify potential prediction errors. Second, we define new **Informed validation metrics** that can be used for comparing semantic segmentations of traffic scenes to street maps. Third, we present an algorithm for **localization correction** that can be used to calibrate the street-map position according to the ground truth segmentation. We present experimental results from applying our methods to the Cityscapes traffic scene image dataset, which demonstrate that our approach can identify similar prediction errors as a validation by ground truth data.

Approach for Informed, Street-Map Based Validation

The approach is illustrated in Figure 5.1.

Informed Validation Metrics

The consistency of the semantic segmentation with the street map can be quantified by the two validation metrics that we introduced. a) The Intersection over Segmentation (IoS) quantifies the overlap of the segmentation with the map. A low IoS is an indicator for false positive road segments. b) The Intersection over Map (IoM) quantifies, vice versa, the overlap of the map with the segmentation. Dynamic segments are ignored. A low IoM is an indicator for false negative road segments.

Figure 5.1 shows an example for the detection of a false positive road. The predicted segmentation shows a road straight forward and below the cars parked at the left side of the street. According to the ground truth there is a parking space below that parking cars. Our map-based validation approach identifies this deviation: The street map suggests a less broad road than in the prediction, resulting in the detection of a false positive region (see orange red color at the left side of the validation error image). For this image the validation metrics are $IoS = 88.03\%$, and $IoM = 97.22\%$, also reflecting the false positive road prediction.

Further details on the approach, but also on the developed method for localization correction, as well as more experimental results, can be found in the paper itself (see Appendix D).

5.3 Author's Contribution

The basis for the Geo-Informed Validation using street maps was developed in a project together with Volkswagen that was conducted by T. Wirtz and me. I refined and implemented the approach (Informed validation and localization correction), and conducted the experiments. The paper was mainly written by me, with helpful feedback from all co-authors.

CHAPTER 6

P5) Quantifying the Benefits: How Knowledge Injection Helps in Informed ML – Metrics, Theory and Systematic Analysis

The research summarized in this chapter has been published in the following paper [51]:

P5) L. von Rueden, J. Garcke, C. Bauckhage. **How Does Knowledge Injection Help in Informed Machine Learning?** *Accepted at: International Joint Conference on Neural Networks (IJCNN)*. 2023.

6.1 Research Question

In this paper, we investigate the quantification of Informed ML and its benefits (see Section 1.1, central question No.3) and answer particularly the following open research questions:

- How can knowledge injection improve ML and how can the benefits be quantified?
- How can knowledge Informed ML be formulated theoretically?
- What are the requirements for the injected knowledge so that it helps?

6.2 Results Summary

There are many different applications where Informed ML is successfully used – especially in scientific and engineering domains, where data acquisition can be expensive, but lots of prior knowledge is available. Just to give a few examples: In neural networks for climate prediction, physical laws are injected via knowledge-based loss functions [28]. In robotics, simulations are used as an additional source for training data [48]. Or in autonomous driving, spatial prototypes are employed to improve object detection [56].

However, the field is so application-driven that it has led to the development of many different and rather specific approaches. In contrast, general analyses about Informed ML are still missing [53]. This makes it difficult to transfer existing approaches to new applications, or to estimate potential improvements in advance.

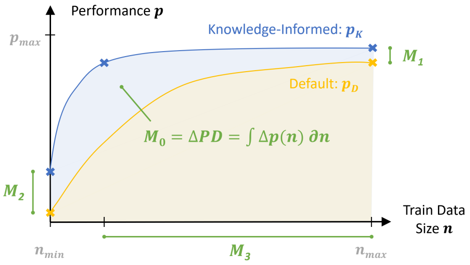


Figure 6.1: **P5) Metric Flavors.** Models that are trained with Informed ML usually achieve a higher performance, e.g., accuracy or robustness, for smaller training data sizes [28, 62, 56]. We propose a new metric that quantifies performance and data need in a single metric in terms of the area under the curve: *Performance-by-Data AUC* (M_0). Other metric flavors are the increase in performance at max. and min. data size (M_1 and M_2), and data reduction for a specific performance (M_3).

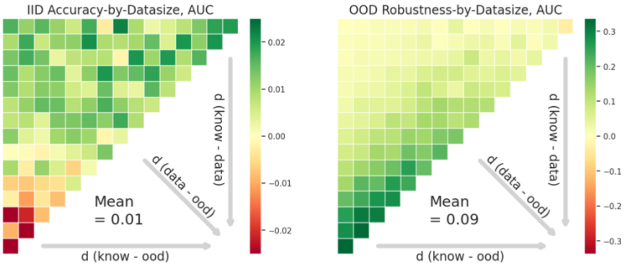


Figure 6.2: **P5) Experimental Results.** Every pixel in in the triangular matrices represents the results from a single experiment by showing the improvement of training a neural network with Informed ML over default training in terms of Performance-by-Data AUC. The left plot shows improvements in In-Distribution (IID) Test Accuracy, and the right plot in Out-of-Distribution (OOD) Robustness. The individual experiments are varied in three dimensions: Distance between prior knowledge and training data (vertical), distance between prior knowledge and OOD data (horizontal), as well as distance between training data and OOD data (diagonal). The left plot confirms our Conjecture 3, and the right our Conjecture 4.

Therefore, our objective is to find general answers to the research question of how knowledge injection via Informed ML does help. Our approach is to develop a framework for quantifying the value of prior knowledge in Informed ML.

In summary, the main contributions in this paper are:

1. We propose a **metrics catalogue** for quantifying the benefits of Informed ML. We also propose a **new metric** that combines performance improvements and data reduction.
2. We present a first **theoretical framework** for Informed ML.
3. We perform a **systematic experimental study** with controllable toy problems.

Metrics

The main goals of Informed ML are to train with less data, to achieve a better model performance, to increase knowledge conformity, or to increase interpretability. However, most works about Informed ML present individual metrics to quantify the benefits of their method and measure it for individual dataset sizes.

We propose to measure performance (e.g., test accuracy) for various train data sizes and summarize the results in a single metric that we call *Performance-by-Data AUC*. As illustrated in Figure 6.1, the metric quantifies the area under the curve of performance p vs. training data size n.

Furthermore, we suggest a metrics catalogue where we focus on relevance for model generalization: In-Distribution Test Accuracy, Out-of-Distribution Robustness, and Knowledge Conformity. The generic performance p from above can be any of these 3 metric types.

Theoretical Framework

We propose a theoretical framework that helps to formalize prior knowledge injection. It is a first step towards an Informed Learning Theory. Our main idea is to regard prior knowledge as a function that can be represented in the same space as the model or the training data (See Figure 1.7(a)). We conjecture that the distance between data and knowledge determines the potential benefits of Informed ML.

Lemma 1 (Prior Knowledge). Prior knowledge describes relations between concepts and can be represented as a function.

Lemma 2 (Knowledge Representation in Function Space). Prior knowledge can be represented in the same function space as given data representations.

Conjecture 3 (Informed Generalization Improvement). The smaller the distance between knowledge and data, the larger the improvement through Informed ML on in-distribution generalization.

Conjecture 4 (Informed Robustness Improvement). The smaller the distance between knowledge and out-of-distribution (OOD) data, and the larger the distance between in-distribution and OOD data, the larger the potential improvement through Informed ML on the OOD robustness.

Systematic Analysis

To illustrate the framework, we perform a systematic experimental study with toy problems. As the toy problem we propose a classification task, which allows to vary the knowledge and the injection method in a controllable manner. We vary relevant parameters, such as the distance between data and knowledge, and measure the potential improvements through Informed ML.

Figure 6.2 shows an excerpt of the results. For in-distribution test accuracy, we can see that the improvements through Informed ML are largest when the distance between prior knowledge and training data is small (upper pixel rows). This confirms Conjecture 3 from above. For the OOD robustness, we can see that the improvement is largest when the distance between knowledge and training data is large (lower pixel rows) and the distance between knowledge and OOD test data is small (closer to diagonal). This confirms our Conjecture 4. We also nicely see that our introduced metric of Performance-by-Data AUC is a good summary of the other metrics.

Further details, such as the metrics catalogue, more details on the theoretical framework, as well as the experimental results, can be found in the paper itself (see Appendix E).

6.3 Author's Contribution

I contributed all parts of the paper (theoretical framework, metrics catalogues, and systematic analysis) – from idea, over implementation and experimentation, to writing the paper. J. Garcke and C. Bauckhage supported with helpful discussions.

CHAPTER 7

Conclusion

Informed ML describes the idea to inject additional prior knowledge into data-driven learning systems. It is a very relevant topic for AI, because it can help to train well-performing ML models even when not enough data is available, or help to increase the models' knowledge conformity, such as the obedience to natural laws or regulatory guidelines, which is important for trustworthy AI.

In this PhD thesis, Informed ML was unified through the development of general, systematic frameworks. This is significant, because the field has so far been very application-driven, which led to the development of various specific approaches. That heterogeneity made it difficult to transfer existing approaches to new applications, or to estimate potential improvements in advance. In contrast, a unification of Informed ML makes it more accessible, practical, and valuable.

7.1 Discussion

The following central research questions were answered in this PhD thesis: 1) What is the fundamental concept of Informed ML, and how can existing approaches can be classified? 2) Is it possible to integrate prior knowledge in a universal way, and how? 3) How can the benefits of Informed ML be quantified, and what are the requirements for the injected knowledge? In this section, the achieved results are discussed regarding these guiding questions.

What is the fundamental concept of Informed ML, and how can existing approaches for integrating prior knowledge into data-driven learning be structurally classified?

This question has been answered thoroughly in this PhD thesis (especially in papers P1 and P2).

A concept for Informed ML was proposed and it was defined as learning from a hybrid information source that consists of data and prior knowledge (P1). Furthermore, a taxonomy that serves as a structured classification framework for Informed ML approaches was developed. It considers the knowledge source, its representation, and the integration into the ML pipeline. Based on this, available approaches were surveyed, and it was described how knowledge representations, such as algebraic equations, logic rules, or simulation results, can be injected into data-driven learning systems.

The proposed Informed ML concept and the developed taxonomy have already demonstrated to be established, useful tools. Since their publication, these structural frameworks are used regularly by researchers and developers to get an overview of the field, and to identify potential Informed ML approaches for their application. The frameworks even helped to structure whole research projects, such as the “KI Wissen” project, which investigates the integration of prior knowledge in AI models for autonomous driving. Thus, the main goal of the PhD thesis – a unification of Informed ML to make it more accessible and usable – has been clearly achieved.

The concept of Informed ML was further extended to the combination of ML and simulation towards Hybrid AI (P2). For this, structural frameworks were proposed that explain the general steps of both approaches. In summary, they depict ML as a bottom-up approach that generates an inductive, data-based model (i.e., transforming data into knowledge) – whereas simulation is described as a top-down approach that applies a deductive, knowledge-based model (i.e., transforming prior knowledge into data). Versatile combination possibilities to hybrid AI systems were identified, including simulation-assisted ML and ML-assisted simulation.

These structural frameworks have also turned out to be a helpful tool for other researchers. For this PhD thesis itself, they are particularly relevant because they describe simulation as transforming prior knowledge into data. The last aspect turned out to be a helpful view for the next central question.

Is it possible to integrate prior knowledge into ML in a universal way, and how?

Yes, it is possible to inject knowledge into data-driven learning in a universal way. For this, two new methods were developed, which are Informed Pre-Training (paper P3) and Geo-Informed Validation (paper P4).

The first method, Informed Pre-Training (P3), applies the idea that data distributions often follow domain invariant relationships that are given by prototypes from prior knowledge. Such prototypes can be given by various knowledge representation, e.g., geospatial templates, graphs, or equations. It was shown that pre-training on such knowledge prototypes improves generalization capabilities, especially for small data, and also increases out-of-distribution robustness. Furthermore, an analysis of the neural network layers that are affected most by the Informed Pre-Training, has shown that the improvements come from transferring the deeper layers. These typically represent high-level features, which confirms the transfer of semantic knowledge through Informed Pre-Training (i.e., it induces Informed Transfer Learning). This demonstrated that conventional, data-based pre-training and the novel, knowledge-based pre-training have complementary strengths, so that the latter can be used to achieve further improvements.

The second method, Geo-Informed Validation (P4), can be used to check ML models for their conformity with geospatial knowledge. This method was developed in the application context of AI for autonomous driving, which must meet strict requirements on safety and robustness. In particular, it was proposed to validate models that have been trained for semantic segmentation of traffic scenes by using prior knowledge about spatial perspectives and from street maps. It was shown, that Informed Validation can be used to identify potential errors of ML models. For this, specific validation metrics were defined that quantify the conformity of semantic segmentation predictions with geospatial knowledge. This brings the advantage, that AI models can also be tested, even when no ground truth data is available.

Both methods (P3, and P4) utilize the idea to transform formalized knowledge representations, such as equations, graphs, or geospatial objects, into data representations (e.g., by using simulation), so that they can be easier integrated into the ML pipeline. Although this idea was not clear from the beginning, it turned out to be very practical for developing universal Informed ML methods. Other ideas could have been to improve existing approaches like knowledge-based neural network architectures or knowledge-based loss functions, but this would still require time-consuming engineering (e.g., developing a modular architecture according to given knowledge rules) for every new application. Therefore, knowledge integration into the data-based steps of the ML pipeline through Pre-Training and Validation are preferable, because this has the advantage of simple, rather application-independent integration methods. A further advantage is that the knowledge is not hard coded into the model, so that there is still the flexibility to learn from new patterns in real data.

The idea to transform knowledge into data (and vice versa) is also connected to the idea of continual lifelong learning [44]. This interplay between knowledge and data shows that a transformation between them is also a natural approach for Informed ML.

How can the benefits of Informed ML be quantified, and what are the requirements for the injected knowledge?

This question has been answered by several investigations in this PhD thesis (especially in paper P5, and also in P3 and P4). In particular, a metrics catalogue for quantifying the benefits through Informed ML was proposed, a framework for an Informed Learning Theory was developed, and a systematic analysis of the dependency on the distance between data and knowledge was conducted.

The systematic metrics catalogue comprises three main benefits of Informed ML: Improving in-distribution accuracy, out-of-distribution robustness, and knowledge conformity (P5). With respect to the concrete quantification, a list of specific metrics was suggested, including a new metric that combines performance improvement and data efficiency (called performance-by-data AUC). The catalogue allows a transparent, and standardized comparison of various methods. Thus, it provides the basis for future benchmarks of Informed ML. Several experiments have shown that all those metrics can be improved particularly when the original training data is scarce. This confirms one of the main motivations for using Informed ML: Being able to train good ML models with less data.

An interesting insight from this PhD thesis is that Informed ML is especially beneficial for improving out-of-distribution robustness, i.e., for situations when a model is applied to test data from an environment that is different than the training environment. This has been shown for the developed method of Informed Pre-Training (P3) and in the systematic analysis (P5). Furthermore, the main benefit of the developed Geo-Informed Validation (P4) is to increase knowledge conformity. The insight that the injection of prior knowledge into ML can especially improve out-of-distribution robustness contributes a more differentiated view on improvements through Informed ML. Simply said, it does not only hold the assumption

"If you inject prior knowledge, then you can train a model with less data",

but rather:

"If you inject prior knowledge that is valid across all environments, then you can train a model in one environment and foster its performance in other environments".

This idea also aligns with other ML research, e.g., with invariant risk minimization [2]. However, the difference is that with this approach, it is still needed to gather data from several environments and then learn the invariants. In contrast, in Informed ML, the prior knowledge already brings the invariants and these can explicitly be integrated into the learning process.

The second part of the above central question, i.e., regarding the requirements on the knowledge, has been answered through the development of a new theoretical framework: It describes prior knowledge and data in a joint representation space, leading to the conjectures that the distances between them influence the performance improvement (P5). Evaluating and specifying the requirements on the injected knowledge has been non-trivial, because knowledge can be represented in versatile forms. As depicted in the Informed ML taxonomy, typical representations of prior knowledge are algebraic equations, logic rules, knowledge graphs, simulation results, or human feedback. An investigation on the requirements for each type would have been exhaustive. Therefore, an abstract view was taken and knowledge was formulated as relations between concepts that can be represented as a function, which brings the advantage that it can be related to other data representations. It was conjectured that a) the smaller the distance between knowledge and data, the larger the improvement through Informed ML on in-distribution generalization, and b) the smaller the distance between knowledge and out-of-distribution test data, and the larger the distance between training data and that out-of-distribution test data, the larger the potential improvement through Informed ML on the robustness. Finally, a systematic experimental study with controllable toy problems was performed. These confirmed the developed theories about the influence of the distances between knowledge and data on potential model improvements.

7.2 Outlook

All in all, the main research questions of this PhD thesis have been thoroughly investigated and could be well answered, so that the goal of a unification of Informed ML has been achieved. Of course, every finding leads to new research ideas. In the following, potential future work is shortly described.

Informed ML Benchmark. What is currently missing in the field of Informed ML is a benchmark that allows a standardized comparison of different approaches. This PhD thesis already provided the fundamentals for such a benchmark, which are the unified metrics for quantification and the systematic analysis approach with standardized variations of the distance between data an knowledge. Furthermore, the developed method of Informed Pre-Training using knowledge prototypes brings the opportunity to investigate Informed ML using the MNIST dataset, which is a well-established benchmark for common ML tasks. Using this, the next step would be to create an extended suite of benchmark datasets and then conduct a large-scale comparison of available Informed ML approaches.

Informed ML Theory. The framework for an Informed Learning Theory, which was proposed in this PhD thesis, can also be investigated more deeply. The derivation of the conjectures about the effect of distances between knowledge and data on model improvements can be further elaborated. It would be interesting to conduct further theoretical investigations about the transformation of prior knowledge into data representation and the ensuing relation between knowledge and training data in function space, so that the current theoretical basics become more expounded.

Further Applications. There are plenty of applications, where Informed ML can be used more extensively in the future. The frameworks that were developed in this PhD thesis can help to do this. The Informed ML taxonomy can be used to identify further knowledge integration strategies for individual applications. In particular, in science and engineering, where data acquisition is expensive and lots of knowledge is available, there is a lot of potential for Informed ML. For example, the proposed Geo-Informed ML can be extended and further applied in autonomous driving or in any other spatial applications. Another example that would also be interesting to investigate is the application of Informed ML in language modelling, which is closely related to the next idea for future work.

Informed ML for Conversational and General AI. Current AI developments include conversational systems, such as ChatGPT [42], which can be used for tasks like question answering, creative writing, or problem solving. These systems go in the direction of an artificial general intelligence (AGI) [11] and suggest that a new era has begun. However, the information that is provided by these systems is not necessarily guaranteed to be accurate. Therefore, the verification of such AI systems, e.g., to check the correctness of the provided answers or their conformity with regulatory or security guidelines, is an important task. For this, Informed ML can provide potential solutions, because it can incorporate factual knowledge where training data is scarce or validate and verify the AI systems' predictions. Thus, the investigation of how prior knowledge can be incorporated into conversational and general AI systems constitutes very important future work – because Informed ML is helpful to achieve trustworthy AI.

Bibliography

[1] Yaser S. Abu-Mostafa, Malik Magdon-Ismail and Hsuan-Tien Lin. *Learning from data*. AMLBook, 2012.

[2] Martin Arjovsky et al. “Invariant risk minimization”. In: *arXiv preprint arXiv:1907.02893* (2019).

[3] Peter Battaglia et al. “Interaction networks for learning about objects, relations and physics”. In: *Advances in Neural Information Processing Systems (NIPS)* (2016).

[4] Mikhail Belkin et al. “Reconciling modern machine-learning practice and the classical bias–variance trade-off”. In: *Proceedings of the National Academy of Sciences (PNAS)* (2019).

[5] Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006.

[6] Bastian Bohn et al. “Analysis of car crash simulation data with nonlinear machine learning methods”. In: *Procedia Computer Science* (2013).

[7] Mariusz Bojarski et al. “End to end learning for self-driving cars”. In: *arXiv preprint arXiv:1604.07316* (2016).

[8] Johan Bollen, Huina Mao and Alberto Pepe. “Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena”. In: *International AAAI Conference on Web and Social Media (ICWSM)* (2011).

[9] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in Neural Information Processing Systems (NIPS)* (2020).

[10] Miles Brundage et al. “Toward trustworthy AI development: mechanisms for supporting verifiable claims”. In: *arXiv preprint arXiv:2004.07213* (2020).

[11] Sébastien Bubeck et al. “Sparks of artificial general intelligence: Early experiments with gpt-4”. In: *arXiv preprint arXiv:2303.12712* (2023).

[12] Keith T. Butler et al. “Machine learning for molecular and materials science”. In: *Nature* (2018).

[13] Mark Campbell et al. “Autonomous driving in urban environments: approaches, lessons and challenges”. In: *Philosophical Transactions of the Royal Society A* (2010).

[14] Travers Ching et al. “Opportunities and obstacles for deep learning in biology and medicine”. In: *Journal of The Royal Society Interface* (2018).

[15] Marius Cordts et al. “The cityscapes dataset for semantic urban scene understanding”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016).

[16] Antoine Cully et al. “Robots that can adapt like animals”. In: *Nature* (2015).

[17] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).

[18] Michelangelo Diligenti, Soumali Roychowdhury and Marco Gori. “Integrating prior knowledge into deep learning”. In: *International Conference on Machine Learning and Applications (ICMLA)* (2017).

[19] Dumitru Erhan et al. “The difficulty of training deep architectures and the effect of unsupervised pre-training”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)* (2009).

[20] Di Feng et al. “Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges”. In: *IEEE Transactions on Intelligent Transportation Systems* (2020).

[21] Alberto Garcia-Garcia et al. “A survey on deep learning techniques for image and video semantic segmentation”. In: *Applied Soft Computing* (2018).

[22] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)* (2010).

[23] Ian Goodfellow, Yoshua Bengio and Aaron Courville. *Deep learning*. The MIT Press, 2016.

[24] Kaiming He et al. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015).

[25] Dan Hendrycks and Thomas Dietterich. “Benchmarking neural network robustness to common corruptions and perturbations”. In: *International Conference on Learning Representations (ICLR)* (2019).

[26] Minyoung Huh, Pulkit Agrawal and Alexei A Efros. “What makes ImageNet good for transfer learning?” In: *arXiv preprint arXiv:1608.08614* (2016).

[27] Chenhan Jiang et al. “Hybrid knowledge routed modules for large-scale object detection”. In: *Advances in Neural Information Processing Systems (NIPS)* (2018).

[28] Anuj Karpatne et al. “Physics-guided neural networks (pgnn): An application in lake temperature modeling”. In: *arXiv preprint arXiv:1710.11431* (2017).

[29] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Neural Information Processing Systems (NIPS)* (2012).

[30] Yann LeCun, Yoshua Bengio and Geoffrey Hinton. “Deep learning”. In: *Nature* (2015).

[31] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* (1998).

[32] Kuan-Hui Lee et al. “Spigan: Privileged adversarial learning from simulation”. In: *International Conference on Learning Representations (ICLR)* (2019).

[33] Ulrike von Luxburg and Bernhard Schölkopf. “Statistical learning theory: Models, concepts, and results”. In: *Handbook of the History of Logic* (2011).

[34] Kenneth Marino, Ruslan Salakhutdinov and Abhinav Gupta. "The more you know: Using knowledge graphs for image classification". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[35] John McCarthy. *What is artificial intelligence*. 2007.

[36] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems (NIPS)* (2013).

[37] Marvin L. Minsky. "Logical versus analogical or symbolic versus connectionist or neat versus scruffy". In: *AI magazine* (1991).

[38] Kirsten Mitchell-Wallace et al. *Natural catastrophe risk management and modelling: A practitioner's guide*. John Wiley & Sons, 2017.

[39] Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[40] Preetum Nakkiran et al. "Deep double descent: Where bigger models and more data hurt". In: *Journal of Statistical Mechanics: Theory and Experiment* (2021).

[41] Behnam Neyshabur, Hanie Sedghi and Chiyuan Zhang. "What is being transferred in transfer learning?" In: *Advances in Neural Information Processing Systems (NIPS)* (2020).

[42] OpenAI. "GPT-4 Technical Report". In: *arXiv preprint arXiv:2303.08774* (2023).

[43] OpenStreetMap. https://www.openstreetmap.org.

[44] German I. Parisi et al. "Continual lifelong learning with neural networks: A review". In: *Neural Networks* (2019).

[45] Scott Drew Pendleton et al. "Perception, planning, control, and coordination for autonomous vehicles". In: *Machines* (2017).

[46] Julius Pfrommer et al. "Optimisation of manufacturing process parameters using deep neural networks as surrogate models". In: *Procedia CiRP* (2018).

[47] Ivens Portugal, Paulo Alencar and Donald Cowan. "The use of machine learning algorithms in recommender systems: A systematic review". In: *Expert Systems with Applications* (2018).

[48] Akshara Rai et al. "Using simulation to improve sample-efficiency of Bayesian optimization for bipedal robots". In: *Journal of Machine Learning Research* (2019).

[49] Markus Reichstein et al. "Deep learning and process understanding for data-driven Earth system science". In: *Nature* (2019).

[50] Ribana Roscher et al. "Explainable machine learning for scientific insights and discoveries". In: *IEEE Access* (2020).

[51] Laura von Rueden, Jochen Garcke and Christian Bauckhage. "How does knowledge injection help in informed machine learning?" In: *International Joint Conference on Neural Networks (IJCNN)* (2023).

[52] Laura von Rueden et al. "Combining machine learning and simulation to a hybrid modelling approach: Current and future directions". In: *Advances in Intelligent Data Analysis (IDA)* (2020).

[53] Laura von Rueden et al. "Informed machine learning – A taxonomy and survey of integrating prior knowledge into learning systems". In: *IEEE Transactions on Knowledge and Data Engineering* (2021).

[54] Laura von Rueden et al. "Informed machine learning – Towards a taxonomy of explicit integration of knowledge into machine learning". In: *arXiv preprint arXiv:1903.12394* (2019).

[55] Laura von Rueden et al. "Informed pre-training of neural networks using prototypes from prior knowledge". In: *International Joint Conference on Neural Networks (IJCNN)* (2023).

[56] Laura von Rueden et al. "Informed pre-training on prior knowledge". In: *arXiv preprint arXiv:2205.11433* (2022).

[57] Laura von Rueden et al. "Street-map based validation of semantic segmentation in autonomous driving". In: *International Conference on Pattern Recognition (ICPR)* (2021).

[58] Laura von Rueden et al. "Towards map-based validation of semantic segmentation masks". In: *International Conference on Machine Learning (ICML), Workshop on AI for Autonomous Driving (AIAD)* (2020).

[59] Stuart Jonathan Russell and Peter Norvig. *Artificial intelligence a modern approach*. Pearson Education, Inc., 2020.

[60] Chris T. Shaw. *Using computational fluid dynamics*. Prentice Hall, 1992.

[61] Paul Smolensky. "Connectionist AI, symbolic AI, and the brain". In: *Artificial Intelligence Review* (1987).

[62] Russell Stewart and Stefano Ermon. "Label-free supervision of neural networks with physics and domain knowledge". In: *AAAI Conference on Artificial Intelligence* (2017).

[63] Ilya Sutskever et al. "On the importance of initialization and momentum in deep learning". In: *International Conference on Machine Learning (ICML)* (2013).

[64] Alan M. Turing. "Computing Machinery and Intelligence". In: *Mind* (1950).

[65] Vladimir Vapnik. "Principles of risk minimization for learning theory". In: *Advances in Neural Information Processing Systems (NIPS)* (1991).

[66] Ashish Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems (NIPS)* (2017).

[67] Riccardo Volpi et al. "Generalizing to unseen domains via adversarial data augmentation". In: *Advances in Neural Information Processing Systems (NIPS)* (2018).

[68] Yaqing Wang et al. "Generalizing from a few examples: A survey on few-shot learning". In: *ACM Computing Surveys* (2020).

[69] Jingyi Xu et al. "A semantic loss function for deep learning with symbolic knowledge". In: *International Conference on Machine Learning (ICML)* (2018).

[70] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems (NIPS)* (2014).

List of Figures

APPENDIX A

Paper P1) Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3079836, IEEE Transactions on Knowledge and Data Engineering

Informed Machine Learning – A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems

Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Michal Walczak, Jochen Garcke, Christian Bauckhage and Jannis Schuecker

Abstract—Despite its great success, machine learning can have its limits when dealing with insufficient training data. A potential solution is the additional integration of prior knowledge into the training process which leads to the notion of *informed machine learning*. In this paper, we present a structured overview of various approaches in this field. We provide a definition and propose a concept for informed machine learning which illustrates its building blocks and distinguishes it from conventional machine learning. We introduce a taxonomy that serves as a classification framework for informed machine learning approaches. It considers the source of knowledge, its representation, and its integration into the machine learning pipeline. Based on this taxonomy, we survey related research and describe how different knowledge representations such as algebraic equations, logic rules, or simulation results can be used in learning systems. This evaluation of numerous papers on the basis of our taxonomy uncovers key methods in the field of informed machine learning.

Index Terms—Machine Learning, Prior Knowledge, Expert Knowledge, Informed, Hybrid, Neuro-Symbolic, Survey, Taxonomy.

1 Introduction

Machine learning has shown great success in building models for pattern recognition in domains ranging from computer vision [1] over speech recognition [2] and text understanding [3] to Game AI [4]. In addition to these classical domains, machine learning and in particular deep learning are increasingly important and successful in engineering and the sciences [5], [6], [7]. These success stories are grounded in the data-based nature of the approach of learning from a tremendous number of examples.

However, there are many circumstances where purely data-driven approaches can reach their limits or lead to unsatisfactory results. The most obvious scenario is that not enough data is available to train well-performing and sufficiently generalized models. Another important aspect is that a purely data-driven model might not meet constraints such as dictated by natural laws, or given through regulatory or security guidelines, which are important for trustworthy AI [8]. With machine learning models becoming more and more complex, there is also a growing need for models to be interpretable and explainable [9].

These issues have led to increased research on how to improve machine learning models by additionally incorporating prior knowledge into the learning process. Although integrating knowledge into machine learning is common, e.g. through labelling or feature engineering, we observe a growing interest in the integration of more knowledge, and especially of further formal knowledge representations. For example, logic rules [10], [11] or algebraic equations [12], [13] have been added as constraints to loss functions. Knowledge graphs can enhance neural networks with information about relations between instances [14], which is of interest in image classification [15], [16]. Furthermore, physical simulations have been used to enrich training data [17], [18], [19]. This heterogeneity in approaches leads to some redundancy in nomenclature; for instance, we find terms such as physics-informed deep learning [20], physics-guided neural networks [12], or semantic-based regularization [21]. The recent growth of research activities shows that the combination of data- and knowledge-driven approaches becomes relevant in more and more areas. However, the growing number and increasing variety of research papers in this field motivates a systematic survey.

A recent survey synthesizes this into a new paradigm of theory-guided data science and points out the importance of enforcing scientific consistency in machine learning [22]. Even for support vector machines there exists a survey about the incorporation of knowledge into this formalism [23]. The fusion of symbolic and connectionist AI seems more and more approachable. In this regard, we refer to recent a survey on graph neural networks and a research direction framed as relational inductive bias [24]. Our work

- *All authors are with the Fraunhofer Center for Machine Learning.*
- *Laura von Rueden, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Birgit Kirsch, Annika Pick, Rajkumar Ramamurthy, Christian Bauckhage and Jannis Schuecker are with the Fraunhofer IAIS, Institute for Intelligent Analysis and Information Systems, 53757 Sankt Augustin, Germany.*
- *Sebastian Mayer and Jochen Garcke are with the Fraunhofer SCAI, Institute for Algorithms and Scientific Computing, 53757 Sankt Augustin, Germany.*
- *Raoul Heese and Michał Walczak are with the Fraunhofer ITWM, Institute for Industrial Mathematics, 67663 Kaiserslautern, Germany.*
- *Julius Pfrommer is with the Fraunhofer IOSB, Institute for Optronics, System Technologies and Image Exploitation, 76131 Karlsruhe, Germany.*
- *Corresponding author: laura.von.rueden@iais.fraunhofer.de*

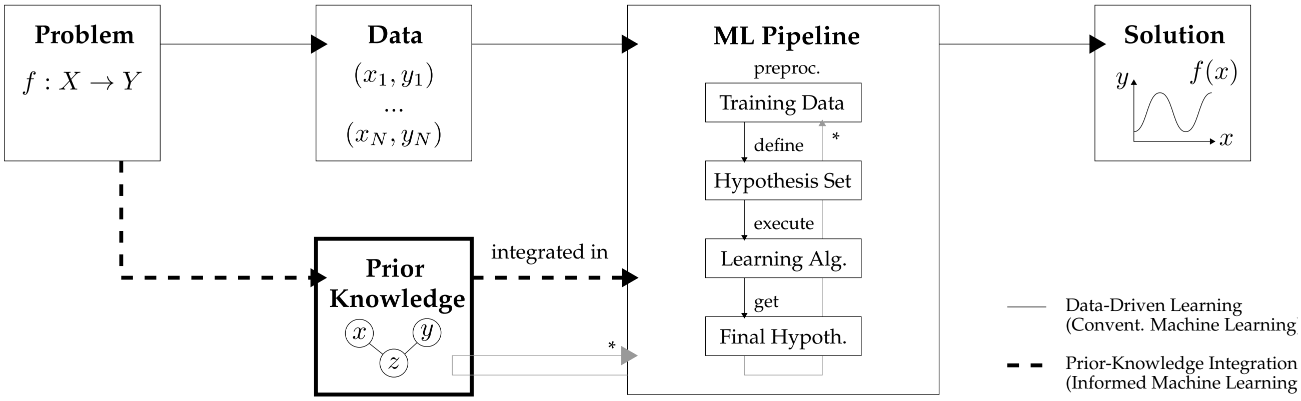


Figure 1: **Information Flow in Informed Machine Learning.** The informed machine learning pipeline requires a hybrid information source with two components: Data and prior knowledge. In conventional machine learning knowledge is used for data preprocessing and feature engineering, but this process is deeply intertwined with the learning pipeline (*). In contrast, in *informed machine learning* prior knowledge comes from an independent source, is given by formal representations (e.g., by knowledge graphs, simulation results, or logic rules), and is explicitly integrated.

complements the aforementioned surveys by providing a systematic categorization of knowledge representations that are integrated into machine learning. We provide a structured overview based on a survey of a large number of research papers on how to integrate additional, prior knowledge into the machine learning pipeline. As an umbrella term for such methods, we henceforth use *informed machine learning*.

Our contributions are threefold: We propose an abstract concept for informed machine learning that clarifies its building blocks and relation to conventional machine learning. It states that informed learning uses a hybrid information source that consists of data and prior knowledge, which comes from an independent source and is given by formal representations. Our main contribution is the introduction of a taxonomy that classifies informed machine learning approaches, which is novel and the first of its kind. It contains the dimensions of the knowledge source, its representation, and its integration into the machine learning pipeline. We put a special emphasis on categorizing various knowledge representations, since this may enable practitioners to incorporate their domain knowledge into machine learning processes. Moreover, we present a description of available approaches and explain how different knowledge representations, e.g., algebraic equations, logic rules, or simulation results, can be used in informed machine learning.

Our goal is to equip potential new users of informed machine learning with established and successful methods. As we intend to survey a broad spectrum of methods in this field, we cannot describe all methodical details and we do not claim to have covered all available research papers. We rather aim to analyze and describe common grounds as well as the diversity of approaches in order to identify the main research directions in informed machine learning.

In Section 2, we begin with a formulation of our concept for *informed machine learning*. In Section 3, we describe how we classified the approaches in terms of our applied surveying methodology and our obtained key insights. Section 4 presents the taxonomy and its elements that we distilled from surveying a large number of research papers. In Section 5, we describe the approaches for the integration of knowledge into machine learning classified according to the taxonomy in more detail. After brief historical account in Section 6, we finally discuss future directions in Section 7 and conclude in Section 8.

2 Concept of Informed Machine Learning

In this section, we present our concept of *informed machine learning*. We first state our notion of knowledge and then present our descriptive definition of its integration into machine learning.

2.1 Knowledge

The meaning of knowledge is difficult to define in general and is an ongoing debate in philosophy [25], [26], [27]. During the generation of knowledge, it first appears as useful information [28], which is subsequently validated. People validate information about the world using the brain's inner statistical processing capabilities [29], [30] or by consulting trusted authorities. Explicit forms of validation are given by empirical studies or scientific experiments [27], [31].

Here, we assume a computer-scientific perspective and understand knowledge as validated information about relations between entities in certain contexts. Regarding its use in machine learning, an important aspect of knowledge is its formalization. The degree of formalization depends on whether knowledge has been put into writing, how structured the writing is, and how formal and strict the language is that was used (e.g., natural language vs. mathematical formula). The more formally knowledge is represented, the more easily it can be integrated into machine learning.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3079836, IEEE Transactions on Knowledge and Data Engineering

2.2 Integrating Prior Knowledge into Machine Learning

Apart from the usual information source in a machine learning pipeline, the training data, one can additionally integrate knowledge. If this knowledge is pre-existent and independent of learning algorithms, it can be called prior knowledge. Moreover, such prior knowledge can be given by formal representations, which exist in an external, separated way from the learning problem and the usual training data. Machine learning that explicitly integrates such knowledge representations will henceforth be called *informed machine learning*.

Definition. *Informed machine learning* describes learning from a hybrid information source that consists of data and prior knowledge. The prior knowledge comes from an independent source, is given by formal representations, and is explicitly integrated into the machine learning pipeline.

This notion of informed machine learning thus describes the flow of information in Figure 1 and is distinct from conventional machine learning.

2.2.1 Conventional Machine Learning

Conventional machine learning starts with a specific problem for which there is training data. These are fed into the machine learning pipeline, which delivers a solution. Problems can typically be formulated as regression tasks where inputs X have to be mapped to outputs Y. Training data is generated or collected and then processed by algorithms, which try to approximate the unknown mapping. This pipeline comprises four main components, namely the training data, the hypothesis set, the learning algorithm, and the final hypothesis [32].

In traditional approaches, knowledge is generally used in the learning pipeline, however, mainly for training data preprocessing (e.g. labelling) or feature engineering. This kind of integration is involved and deeply intertwined with the whole learning pipeline, such as the choice of the hypothesis set or the learning algorithm, as depicted in Figure 1. Hence, this knowledge is not really used as an independent source or through separated representations, but is rather used with adaption and as required.

2.2.2 Informed Machine Learning

The information flow of informed machine learning comprises an additional prior-knowledge integration and thus consists of two lines originating from the problem, as shown in Figure 1. These involve the usual training data and additional prior knowledge. The latter exists independently of the learning task and can be provided in form of logic rules, simulation results, knowledge graphs, etc.

The essence of *informed machine learning* is that this prior knowledge is explicitly integrated into the machine learning pipeline, ideally via clear interfaces defined by the knowledge representations. Theoretically, this applies to each of the four components of the machine learning pipeline.

3 Classification of Approaches

To comprehend how the concept of informed machine learning is implemented, we performed a systematic classification of existing approaches based on an extensive literature survey. Our goals are to uncover different methods, identify their similarities or differences, and to offer guidelines for users and researchers. In this section, we describe our classification methodology and summarize our key insights.

3.1 Methodology

The methodology of our classification is determined by specific analysis questions which we investigated in a systematic literature survey.

3.1.1 Analysis Questions

Our guiding question is how prior knowledge can be integrated into the machine learning pipeline. Our answers will particularly focus on three aspects: Since prior knowledge in informed machine learning consists of an independent source and requires some form of explicit representations, we consider knowledge sources and representations. Since it also is essential at which component of the machine learning pipeline what kind of knowledge is integrated, we also consider integration methods. In short, our literature survey addresses the following three questions:

1) **Source**:
Which source of knowledge is integrated?
2) **Representation**:
How is the knowledge represented?
3) **Integration**:
Where in the learning pipeline is it integrated?

3.1.2 Literature Surveying Procedure

To systematically answer the above analysis questions, we surveyed a large number of publications describing informed machine learning approaches. We used a comparative and iterative surveying procedure that consisted of different cycles. In the first cycle, we inspected an initial set of papers and took notes as to how each paper answers our questions. Here, we observed that specific answers occur frequently, which then led to the idea of devising a classification framework in the form of a taxonomy. In the second cycle, we inspected an extended set of papers and classified them according to a first draft of the taxonomy. We then further refined the taxonomy to match the observations from the literature. In the third cycle, we re-inspected and resorted papers and, furthermore, expanded our set of papers. This resulted in an extensive literature basis in which all papers are classified according to the distilled taxonomy.

3.2 Key Insights

Next, we present an overview over key insights from our systematic classification. As a preview, we refer to Figure 2, which visually summarizes our findings. A more detailed description of our findings will be given in Sections 4 and 5.

3.2.1 Taxonomy

Based on a comparative and iterative literature survey, we identified a taxonomy that we propose as a classification framework for informed machine learning approaches. Guided by the above analysis questions, the taxonomy consists of the three dimensions *knowledge source*, *knowledge*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3079836, IEEE Transactions on Knowledge and Data Engineering

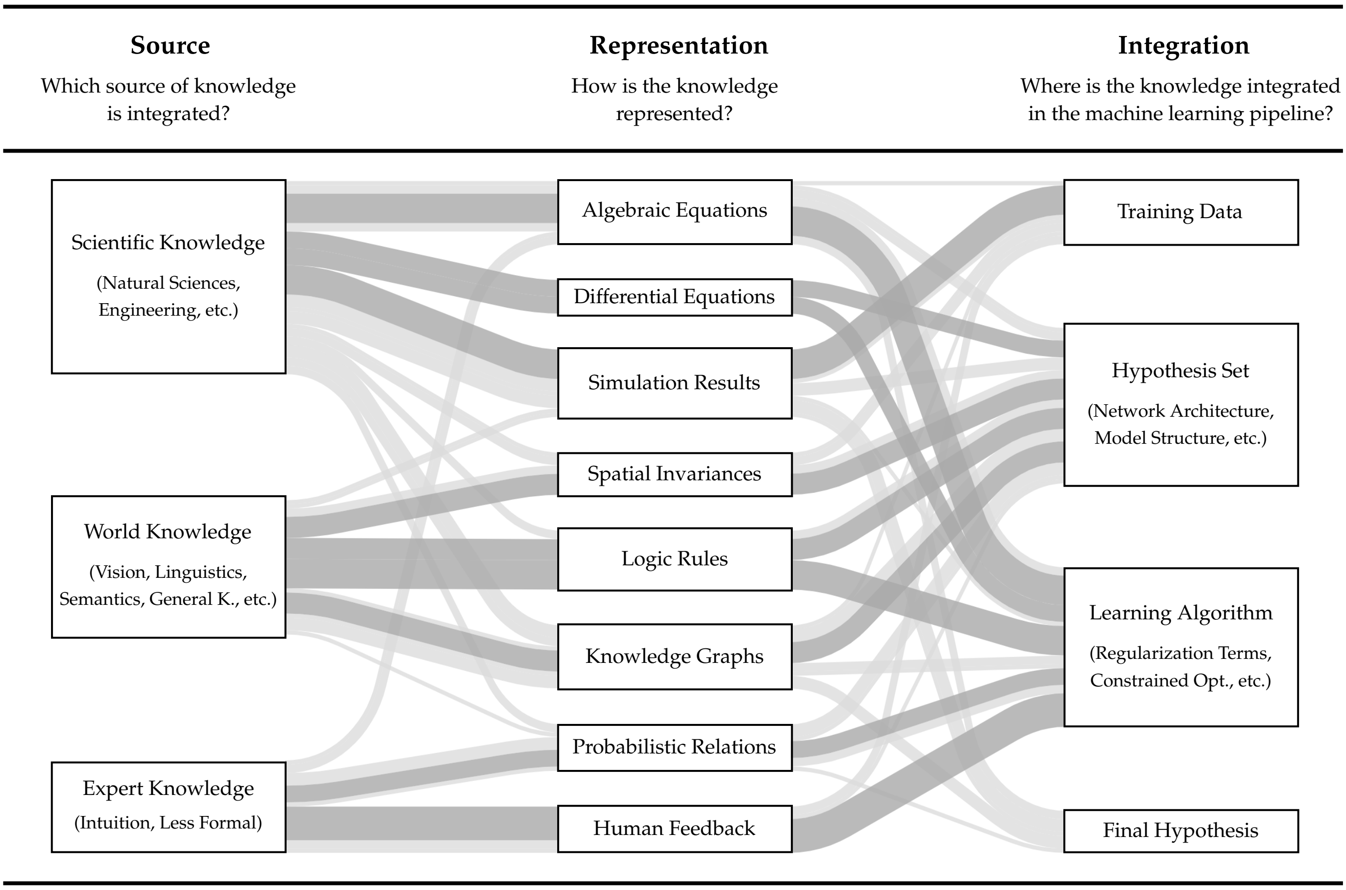

Source
Which source of knowledge is integrated?
Representation
How is the knowledge represented?
Integration
Where is the knowledge integrated in the machine learning pipeline?
Scientific Knowledge
(Natural Sciences, Engineering, etc.)
World Knowledge
(Vision, Linguistics, Semantics, General K., etc.)
Expert Knowledge
(Intuition, Less Formal)
Algebraic Equations
Differential Equations
Simulation Results
Spatial Invariances
Logic Rules
Knowledge Graphs
Probabilistic Relations
Human Feedback
Training Data
Hypothesis Set
(Network Architecture, Model Structure, etc.)
Learning Algorithm
(Regularization Terms, Constrained Opt., etc.)
Final Hypothesis

Figure 2: **Taxonomy of Informed Machine Learning.** This taxonomy serves as a classification framework for *informed machine learning* and structures approaches according to the three above analysis questions about the *knowledge source*, *knowledge representation* and *knowledge integration*. Based on a comparative and iterative literature survey, we identified for each dimension a set of elements that represent a spectrum of different approaches. The size of the elements reflects the relative count of papers. We combine the taxonomy with a Sankey diagram in which the paths connect the elements across the three dimensions and illustrate the approaches that we found in the analyzed papers. The broader the path, the more papers we found for that approach. Main paths (at least four or more papers with the same approach across all dimensions) are highlighted in darker grey and represent central approaches of informed machine learning.

representation and *knowledge integration*. Each dimension contains a set of elements that represent the spectrum of different approaches found in the literature. This is illustrated in the taxonomy in Figure 2.

With respect to knowledge sources, we found three broad categories: Rather specialized and formalized scientific knowledge, everyday life's world knowledge, and more intuitive expert knowledge. For scientific knowledge we found the most informed machine learning papers. With respect to knowledge representations, we found versatile and fine-grained approaches and distilled eight categories (Algebraic equations, differential equations, simulation results, spatial invariances, logic rules, knowledge graphs, probabilistic relations and human feedback). Regarding knowledge integration, we found approaches for all stages of the machine learning pipeline, from the training data and the hypothesis set, over the learning algorithm, to the final hypothesis. However, most informed machine learning papers consider the two central stages.

Depending on the perspective, the taxonomy can be regarded from either one of two sides: An application-oriented user might prefer to read the taxonomy from left to right, starting with some given knowledge source and then selecting representation and integration. Vice versa, a method-oriented developer or researcher might prefer to read the taxonomy from right to left, starting with some given integration method. For both perspectives, knowledge representations are important building blocks and constitute an abstract interface that connects the application- and the method-oriented side.

3.2.2 *Frequent Approaches*

The taxonomy serves as a classification framework and allows us to identify frequent approaches of informed machine learning. In our literature survey, we categorized each research paper with respect to each of the three taxonomy dimensions.

Paths through the Taxonomy. When visually highlighting and connecting them, a specific combination of entries across the taxonomy dimensions figuratively results in a path through the taxonomy. Such paths represent specific approaches towards informed learning and we illustrate

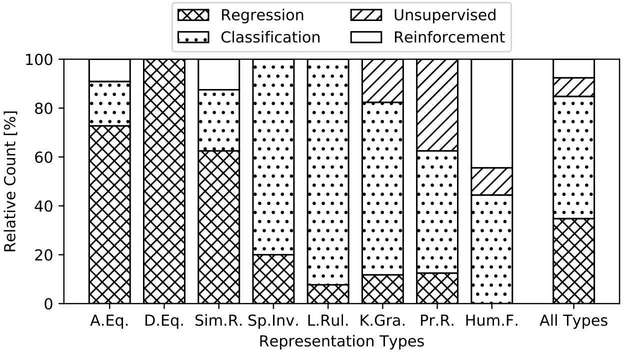


Figure 3: **Knowledge Representations and Learning Tasks.**

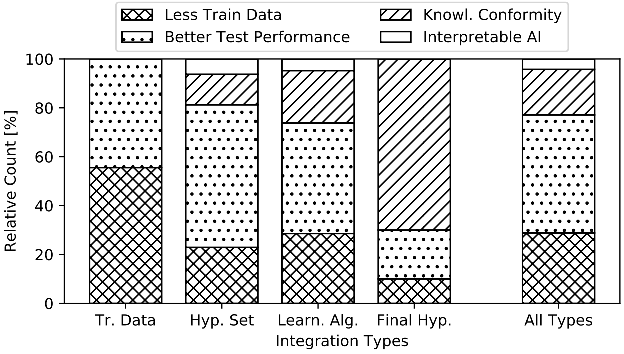


Figure 4: **Knowledge Integration and its Goals.**

this by combining the taxonomy with a Sankey diagram, as shown in Figure 2. We observe that, while various paths through the taxonomy are possible, specific ones occur more frequently and we will call them main paths. For example, we often observed the approach that scientific knowledge is represented in algebraic equations, which are then integrated into the learning algorithm, e.g. the loss function. As another example, we often found that world knowledge such as linguistics is represented by logic rules, which are then integrated into the hypothesis set, e.g. the network architecture. These paths, especially the main paths, can be used as a guideline for users new to the field or provide a set of baseline methods for researchers.

Paths from Source to Representation. We found that the paths from source to representation form groups. That is, for every knowledge source there appear prevalent representation types. Scientific knowledge is mainly represented in terms of algebraic or differential equations or exist in form of simulation results. While other forms of representation are possible, too, there is a clear preference for equations or simulations, likely because most sciences aim at finding natural laws encoded in formulas. For world knowledge, the representation forms of logic rules, knowledge graphs, or spatial invariances are the primary ones. These can be understood as a group of symbolic representations. Expert knowledge is mainly represented by probabilistic relations or human feedback. This is appears reasonable because such representations allow for informality as well as for a degree of uncertainty, both of which might be useful for representing intuition. We also performed an additional analysis on the dependency of the learning task and found a confirmation of the above described representation groups as shown in Figure 3.

From a theoretical point of view, transformations between representations are possible and indeed often apparent within the aforementioned groups. For example, equations can be transformed to simulation results, or logic rules can be represented as knowledge graphs and vice versa. Nevertheless, from a practical point of view, differentiating between forms of representations appears useful as specific representations might already be available in a given set up.

Paths from Representation to Integration. For most of the representation types we found at least one main path to an integration type. The following mappings can be observed. Simulation results are very often integrated into the training data. Knowledge graphs, spatial invariances, and logic rules are frequently incorporated into the hypothesis set. The learning algorithm is mainly enhanced by algebraic or differential equations, logic rules, probabilistic relations, or human feedback. Lastly, the final hypothesis is often checked by knowledge graphs or also by simulation results. However, since we observed various possible types of integration for all representation types, the integration still appears to be problem specific.

Hence, we additionally analyzed the literature for the goal of the prior knowledge integration and found four main goals: Data efficiency, accuracy, interpretability, or knowledge conformity. Although these goals are interrelated or even partially equivalent according to statistical learning theory, it is interesting to examine them as different motivations for the chosen approach. The distribution of goals for the distinct integration types is shown in Figure 4. We observe that the main goal always is to achieve better performance. The integration of prior knowledge into the training data stands out, because its main goal is to train with less data. The integration into the final hypothesis is also special, because it is mainly used to ensure knowledge conformity for secure and trustworthy AI. All in all, this distribution suggests suitable integration approaches depending on the goal.

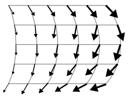
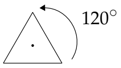
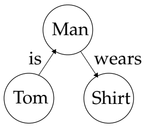
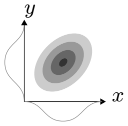
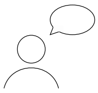
4 Taxonomy

In this section, we describe the *informed machine learning* taxonomy that we distilled as a classification framework in our literature survey. For each of the three taxonomy dimensions *knowledge source*, *knowledge representation* and *knowledge integration* we describe the found elements, as shown in Figure 2. While an extensive approach categorization according to this taxonomy with further concrete examples will be presented in the next section (Section 5), we here describe the taxonomy on a more conceptual level.

4.1 Knowledge Source

The category *knowledge source* refers to the origin of prior knowledge to be integrated in machine learning. We observe that the source of prior knowledge can be an established

Table 1: **Illustrative Overview of Knowledge Representations in the Informed Machine Learning Taxonomy.** Each representation type is illustrated by a simple or prominent example in order to give a first intuitive understanding.

Algebraic Equations	Differential Equations	Simulation Results	Spatial Invariances	Logic Rules	Knowledge Graphs	Probabilistic Relations	Human Feedback
$E = m \cdot c^2$ $v \leqslant c$	$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$ $F(x) = m \frac{d^2 x}{dt^2}$			$A \wedge B \Rightarrow C$			

knowledge domain but also knowledge from an individual group of people with respective experience.

We find that prior knowledge often stems from the sciences or is a form of world or expert knowledge, as illustrated on the left in Figure 2. This list is neither complete nor disjoint but intended show a spectrum from more formal to less formal, or explicitly to implicitly validated knowledge. Although particular knowledge can be assigned to more than one of these sources, the goal of this categorization is to identify paths in our taxonomy that describe frequent approaches of knowledge integration into machine learning. In the following we shortly describe each of the knowledge sources.

Scientific Knowledge. We subsume the subjects of science, technology, engineering, and mathematics under *scientific knowledge*. Such knowledge is typically formalized and validated explicitly through scientific experiments. Examples are the universal laws of physics, bio-molecular descriptions of genetic sequences, or material-forming production processes.

World Knowledge. By *world knowledge* we refer to facts from everyday life that are known to almost everyone and can thus also be called general knowledge. It can be more or less formal. Generally, it can be intuitive and validated implicitly by humans reasoning in the world surrounding them. Therefore, world knowledge often describes relations of objects or concepts appearing in the world perceived by humans, for instance, the fact that a bird has feathers and can fly. Moreover, by world knowledge we also subsume linguistics. Such knowledge can also be explicitly validated through empirical studies. Examples are the syntax and semantics of language.

Expert Knowledge. We consider *expert knowledge* to be knowledge that is held by a particular group of experts. Within the expert's community it can also be called common knowledge. Such knowledge is rather informal and needs to be formalized, e.g., with human-machine interfaces. It is also validated implicitly through a group of experienced specialists. In the context of cognitive science, this expert knowledge can also become intuitive [29]. For example, an engineer or a physician acquires knowledge over several years of experience working in a specific field.

4.2 Knowledge Representation

The category *knowledge representation* describes how knowledge is formally represented. With respect to the flow of information in informed machine learning in Figure 1, it directly corresponds to our key element of prior knowledge. This category constitutes the central building block of our taxonomy, because it determines the potential interface to the machine learning pipeline.

In our literature survey, we frequently encountered certain representation types, as listed in the taxonomy in Figure 2 and illustrated more concretely in Table 1. Our goal is to provide a classification framework of informed machine learning approaches including the used knowledge representation types. Although some types can be mathematically transformed into each other, we keep the representation that are closest to those in the reviewed literature. Here we give a first conceptual overview over these types.

Algebraic Equations. Algebraic equations represent knowledge as equality or inequality relations between mathematical expressions consisting of variables or constants. Equations can be used to describe general functions or to constrain variables to a feasible set and are thus sometimes also called algebraic constraints. Prominent examples in Table 1 are the equation for the mass-energy equivalence and the inequality stating that nothing can travel faster than the speed of light in vacuum.

Differential Equations. Differential equations are a subset of algebraic equations, which describe relations between functions and their spatial or temporal derivatives. Two famous examples in Table 1 are the heat equation, which is a partial differential equation (PDE), and Newton's second law, which is an ordinary differential equation (ODE). In both cases, there exists a (possibly empty) set of functions that solve the differential equation for given initial or boundary conditions. Differential equations are often the basis of a numerical computer simulation. We distinguish the taxonomy categories of differential equations and simulation results in the sense that the former represents a compact mathematical model while the latter represents unfolded, data-based computation results.

Simulation Results. Simulation results describe the numerical outcome of a computer simulation, which is an approximate imitation of the behavior of a real-world process. A simulation engine typically solves a mathematical model using numerical methods and produces results for situation-specific parameters. Its numerical outcome is the simulation result that we describe here as the final knowledge representation. Examples are the flow field of a simulated fluid or pictures of simulated traffic scenes.

Spatial Invariances. Spatial invariances describe properties that do not change under mathematical transformations such as translations and rotations. If a geometric object is invariant under such transformations, it has a symmetry (for example, a rotationally symmetric triangle). A function can be called invariant, if it has the same result for a symmetric transformation of its argument. Connected to invariance is the property of equivariance.

Logic Rules. Logic provides a way of formalizing knowledge about facts and dependencies and allows for translating ordinary language statements (e.g., `IF` A `THEN` B) into formal logic rules ($A \Rightarrow B$). Generally, a logic rule consists of a set of Boolean expressions (A, B) combined with logical connectives ($\wedge$, $\vee$, $\Rightarrow$, $\ldots$). Logic rules can be also called logic constraints or logic sentences.

Knowledge Graphs. A graph is a pair (V, E), where V are its vertices and E denotes edges. In a knowledge graph, vertices (or nodes) usually describe concepts whereas edges represent (abstract) relations between them (as in the example "Man wears shirt" in Table 1). In an ordinary weighted graph, edges quantify the strength and the sign of a relationship between nodes.

Probabilistic Relations. The core concept of probabilistic relations is a random variable X from which samples x can be drawn according to an underlying probability distribution $P(X)$. Two or more random variables X, Y can be interdependent with joint distribution $(x, y) \sim P(X, Y)$. Prior knowledge could be assumptions on the conditional independence or the correlation structure of random variables or even a full description of the joint probability distributions.

Human Feedback. Human feedback refers to technologies that transform knowledge via direct interfaces between users and machines. The choice of input modalities determines the way information is transmitted. Typical modalities include keyboard, mouse, and touchscreen, followed by speech and computer vision, e.g., tracking devices for motion capturing. In theory, knowledge can also be transferred directly via brain signals using brain-computer interfaces.

4.3 Knowledge Integration

The category *knowledge integration* describes where the knowledge is integrated into the machine learning pipeline.

Our literature survey revealed that integration approaches can be structured according to the four components of training data, hypothesis set, learning algorithm, and final hypothesis. Though we present these approaches more thoroughly in Section 5, the following gives a first conceptual overview.

Training Data. A standard way of incorporating knowledge into machine learning is to embody it in the underlying training data. Whereas a classic approach in traditional machine learning is feature engineering where appropriate features are created from expertise, an informed approach according to our definition is the use of hybrid information in terms of the original data set and an additional, separate source of prior knowledge. This separate source of prior knowledge allows to accumulate information and therefore can create a second data set, which can then be used together with, or in addition to, the original training data. A prominent approach is simulation-assisted machine learning where the training data is augmented through simulation results.

Hypothesis Set. Integrating knowledge into the hypothesis set is common, say, through the definition of a neural network's architecture and hyper-parameters. For example, a convolutional neural network applies knowledge as to location and translation invariance of objects in images. More generally, knowledge can be integrated by choosing model structure. A notable example is the design of a network architecture considering a mapping of knowledge elements, such as symbols of a logic rule, to particular neurons.

Learning Algorithm. Learning algorithms typically involve a loss function that can be modified according to additional knowledge, e.g. by designing an appropriate regularizer. A typical approach of informed machine learning is that prior knowledge in form of algebraic equations, for example laws of physics, is integrated by means of additional loss terms.

Final Hypothesis. The output of a learning pipeline, i.e. the final hypothesis, can be benchmarked or validated against existing knowledge. For example, predictions that do not agree with known constraints can be discarded or marked as suspicious so that results are consistent with prior knowledge.

5 Description of Integration Approaches

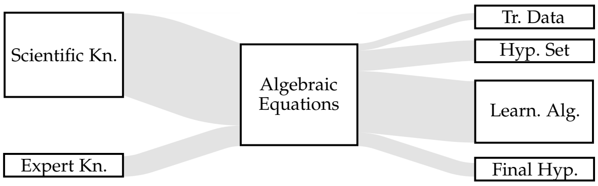
In this section, we give a detailed account of the informed machine learning approaches we found in our literature survey. We will focus on methods and therefore structure our presentation according to knowledge representations. This is motivated by the assumption that similar representations are integrated into machine learning in similar ways as they form the mathematical basis for the integration. Moreover the representations combine both the application- and the method-oriented perspective as described in Section 3.2.1.

For each knowledge representation, we describe the informed machine learning approaches in a separate subsection and present the observed (paths from) knowledge source and the observed (paths to) knowledge integration. We describe each dimension along its entities starting with the main path entity, i.e. the one we found in most papers.

This whole section refers to Table 2 and 3, which lists the paper references sorted according to our taxonomy.

5.1 Algebraic Equations

The main path for algebraic equations that we found in our literature survey comes from scientific knowledge and goes into the learning algorithm, but also other integration types are possible, as illustrated in the following figure.



5.1.1 (Paths from) Knowledge Source

Algebraic equations are mainly used to represent formalized scientific knowledge, but may also be used to express more intuitive expert knowledge.

Insert 1: Knowledge-Based Loss Term

When learning a function f^* from data (x_i, y_i) where the x_i are input features and the y_i are labels, a knowledge-based loss term L_k can be built into the objective function [10], [12]:

$$f^* = \arg\min_f \Big(\overbrace{\lambda_l \sum_i L(f(x_i), y_i)}^{\texttt{Label-based}} + \overbrace{\lambda_r R(f)}^{\texttt{Regul.}} + \underbrace{\lambda_k L_k(f(x_i), x_i)}_{\texttt{Knowledge-based}} \Big) \tag{1}$$

Whereas L is the usual label-based loss and R is a regularization function, L_k quantifies the violation of given prior-knowledge equations. Parameters λ_l, λ_r and λ_k determine the weight of the terms.
Note that L_k only depends on the input features x_i and the learned function f and thus offers the possibility of label-free supervision [13].

Scientific Knowledge. We observed that algebraic equations are used in machine learning in various domains of natural sciences and engineering, particularly in physics [12], [13], [33], [34], [35], but also in biology [36], [37], robotics [38], or manufacturing and production processes [34], [39].

Three representative examples are the following: The trajectory of objects can be described with kinematic laws, e.g., that the position y of a falling object can be described as a function of time t, namely $y(t) = y_0 + v_0 t + at^2$. Such knowledge from Newtonian mechanics can be used to improve object detection and tracking in videos [13]. Or, the proportionality of two variables can be expressed via inequality constraints, for example, that the water density ρ at two different depths $d_1 < d_2$ in a lake must obey $\rho(d_1) \leqslant \rho(d_2)$, which can be used in water temperature prediction [12]. Furthermore, for the prediction of key performance indicators in production processes, relations between control parameters (e.g. voltage, pulse duration) and intermediate observables (e.g. current density) are known to influence outcomes and can be expressed as linear equations derived from principles of physical chemistry [34].

Expert Knowledge. An example for the representation of expert knowledge is to define valid ranges of variables according to experts' intuition as approximation constraints [33] or monotonicity constraints [39].

5.1.2 *(Paths to) Knowledge Integration*

We observe that a frequent way of integrating equation-based knowledge into machine learning is via the learning algorithm. The integration into the other stages is possible, too, and we describe the approaches here ordered by their occurence.

Learning Algorithm. Algebraic equations and inequations can be integrated into learning algorithms via additional loss terms [12], [13], [33], [35] or, more generally, via constrained problem formulation [36], [37], [39].

The integration of algebraic equations as knowledge-based loss terms into the learning objective function is detailed in Insert 1. These knowledge-based terms measure potential inconsistencies w.r.t., say, physical laws [12], [13]. Such an extended loss is usually called physics-based or hybrid loss and fosters the learning from data as well as from prior knowledge. Beyond the measuring inconsistencies with exact formulas, inconsistencies with approximation ranges or general monotonicity constraints, too, can be quantified via rectified linear units [33].

As a further approach, support vector machines can incorporate knowledge by relaxing the optimization problem into a linear minimization problem to which constraints are added in form of linear inequalities [36]. Similarly, it is possible to relax the optimization problem behind certain kernel-based approximation methods to constrain the behavior of a regressor or classifier in a possibly nonlinear region of the input domain [37].

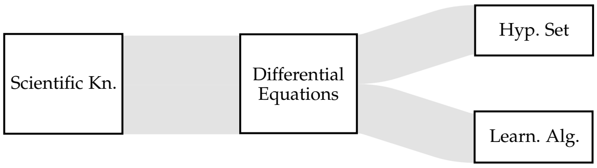
Hypothesis Set. An alternative approach is the integration into the hypothesis set. In particular, algebraic equations can be translated into the architecture of neural networks [34], [38], [40]. One idea is to sequence predefined operations leading to a functional decomposition [40]. More specifically, relations between input parameters, intermediate observables, or output variables reflecting physical constraints can be encoded as linear connections between the layers of a network model [34], [38].

Final Hypothesis. Another integration path applies algebraic equations to the final hypothesis, mainly serving as a consistency check with given constraints from a knowledge domain. This can be implemented as an inconsistency measure that quantifies the deviation of the predicted results from given knowledge similar to the above knowledge-based loss terms. It can then be used as an additional performance metric for model comparison [12]. Such a physical consistency check can also comprise an entire diagnostics set of functions describing particular characteristics [41].

Training Data. Another natural way of integrating algebraic equations into machine learning is to use them for training data generation. While there are many papers in this category, we want to highlight one that integrates prior knowledge as an independent, second source of information by constructing a specific feature vector that directly models physical properties and constraints [42].

5.2 Differential Equations

Next, we describe informed machine learning approaches based on differential equations, which frequently represent scientific knowledge and are integrated into the hypothesis set or the learning algorithm.



5.2.1 *(Paths from) Knowledge Source*

Differential equations model the behavior of dynamical systems by relating state variables to their rate of change. In the literature discussed here, differential equations represent knowledge from the natural sciences.

Scientific Knowledge. Here we give three prominent examples: The work in [20], [43] considers the Burger's equation, which is used in fluid dynamics to model simple one-dimensional currents and in traffic engineering to describe traffic density behavior. Advection-diffusion equations [44] are used in oceanography to model the evolution of sea surface temperatures. The Schrödinger equation studied in [20] describes quantum mechanical phenomena such as wave propagation in optical fibres or the behavior of Bose-Einstein condensates.

5.2.2 *(Paths to) Knowledge Integration*

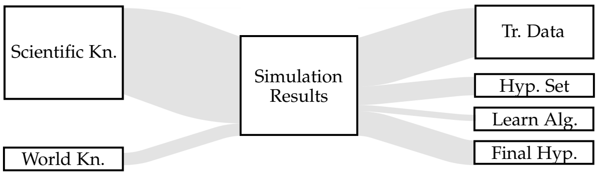
Regarding the integration of differential equations, our survey particularly focuses on the integration into neural network models.

Learning Algorithm. A neural network can be trained to approximate the solution of a differential equation. To this end, the governing differential equation is integrated into the loss function similar to Equation 1 [45]. This requires evaluating derivatives of the network with respect to its inputs, for example, via automatic differentiation, an approach that was recently adapted to deep learning [20]. This ensures the physical plausibility of the neural network output. An extension to generative models is possible, too [43]. Finally, probabilistic models can also be trained by minimizing the distance between the model conditional density and the Boltzmann distribution dictated by a differential equation and boundary conditions [46].

Hypothesis Set. In many applications, differential equations contain unknown time- and space-dependent parameters. Neural networks can model the behavior of such parameters, which then leads to hybrid architectures where the functional form of certain components is analytically derived from (partially) solving differential equations [44], [47], [48]. In other applications, one faces the problem of unknown mappings from input data to quantities whose dynamics are governed by known differential equations, usually called system states. Here, neural networks can learn a mapping from observed data to system states [49]. This also leads to hybrid architectures with knowledge-based modules, e.g. in form of a physics engine.

5.3 Simulation Results

Simulation results are also a prominent knowledge representation in informed machine learning. They mainly come from scientific knowledge and are used to extend the training data.



5.3.1 *(Paths from) Knowledge Source*

Computer simulations have a long tradition in many areas of the sciences. While they are also gaining popularity in other domains, most works on integrating simulation results into machine learning deal with natural sciences and engineering.

Insert 2: Simulation Results as Synthetic Tr. Data

The results from a simulation can be used as synthetic training data and can thus augment the original, real training data. Some papers that follow this approach are [12], [18], [19], [59], [64], [65], [67].

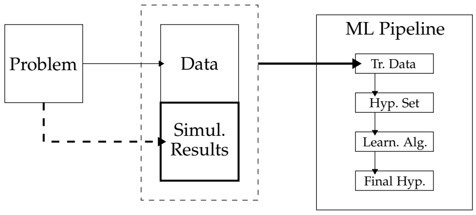


Figure 5: Information flow for synthetic training data from simulations.

Scientific Knowledge. Simulation results informing machine learning can be found in fluid- and thermodynamics [12], material sciences [19], [60], [61], life sciences [59], mechanics and robotics [64], [65], [66], or autonomous driving [18]. To make it more concrete, we give three examples: In material sciences, a density functional theory ab-initio simulation can be used to model the energy and stability of potential new material compounds and their crystal structure [61]. Even complex material forming processes can be simulated, for example a composite textile draping process can be simulated based on a finite-element model [19]. As an example for autonomous driving, urban traffic scenes under specific weather and illumination conditions, which might be useful for the training of visual perception components, can be simulated with dedicated physics engines [18].

5.3.2 *(Paths to) Knowledge Integration*

We find that the integration of simulation results into machine learning is most often happens via the augmentation of training data. Other approaches that occur frequently are the integration into the hypothesis set or the final hypothesis.

Training Data. The integration of simulation results into training data [12], [18], [19], [59], [64], [65], [67] depends on how the simulated, i.e. synthetic, data is combined with the real-world measurements:

Firstly, additional input features are simulated and, together with real data, form input features. For example, original features can be transformed by multiple approximate simulations and the similarity of the simulation results can be used to build a kernel [59].

Secondly, additional target variables are simulated and added to the real data as another feature. This way the model does not necessarily learn to predict targets, e.g. an underlying physical process, but rather the systematic discrepancy between simulated and the true target data [12].

Thirdly, additional target variables are simulated and used as synthetic labels, which is of particular use when the original experiments are very expensive [19]. This approach

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3079836, IEEE Transactions on Knowledge and Data Engineering

Table 2: **References Classified by Knowledge Representation and (Path from) Knowledge Source.**

SOURCE	REPRESENTATION							
	Algebraic Equations	Differential Equations	Simulation Results	Spatial Invariances	Logic Rules	Knowledge Graphs	Probabilistic Relations	Human Feedback
Scientific Knowledge	[12], [13], [33] [34], [35], [36] [37], [38], [41] [39], [42]	[20], [43], [44] [45], [46], [47] [48], [49]	[12], [18], [19] [59], [60], [61] [64], [65], [66] [67], [68], [69]	[50], [51], [52]	[53], [54]	[14], [55], [56] [62], [63]	[57], [58]	
World Knowledge			[67], [70]	[71], [72], [73] [75], [76], [77] [83]	[10], [11], [13] [21], [78], [79] [84], [85], [86] [90], [91], [92]	[15], [16], [56] [80], [81], [82] [87], [88], [89] [93], [94], [95]	[74]	
Expert Knowledge	[33], [39], [40]						[74], [96], [97] [101], [102], [103] [107]	[98], [99], [100] [104], [105], [106] [108], [109], [110]

Table 3: **References Classified by Knowledge Representation and (Path to) Knowledge Integration.**

INTEGRAT.	REPRESENTATION							
	Algebraic Equations	Differential Equations	Simulation Results	Spatial Invariances	Logic Rules	Knowledge Graphs	Probabilistic Relations	Human Feedback
Training Data	[42]		[12], [18], [19] [59], [64], [65] [67]	[52], [77], [83]		[81]		[100], [105]
Hypothesis Set	[34], [38], [40]	[44], [47], [48] [49]	[60], [68], [69]	[50], [51], [73] [71], [72], [75] [76]	[53], [78], [79] [54], [91], [92] [90]	[14], [15], [16] [62], [63], [82] [80], [87], [95]	[74], [97], [101]	[105]
Learning Algorithm	[12], [13], [33] [35], [36], [37] [39]	[20], [43], [45] [46]	[66]		[10], [11], [13] [21], [85], [86] [84]	[55], [56], [89]	[57], [96], [101] [58], [102], [103]	[98], [99], [100] [104], [106], [110] [108], [109]
Final Hypothesis	[12], [41]		[19], [61], [66] [70]			[88], [93], [94]	[107]	

can also be realized with physics engines, for example, pre-trained neural networks can be tailored towards an application through additional training on simulated data [64]. Synthetic training data generated from simulations can also be used to pre-train components of Bayesian optimization frameworks [65].

In informed machine learning, training data thus stems from a hybrid information source and contains both simulated and real data points (see Insert 2). The gap between the synthetic and the real domain can be narrowed via adversarial networks such as SimGAN. These improve the realism of, say, synthetic images and can generate large annotated data sets by simulation [67]. The SPIGAN framework goes one step further and uses additional, privileged information from internal data structures of the simulation in order to foster unsupervised domain adaption of deep networks [18].

Hypothesis Set. Another approach we observed integrates simulation results into the hypothesis set [60], [68], [69], which is of particular interest when dealing with low-fidelity simulations. These are simplified simulations that approximate the overall behaviour of a system but ignore intricate details for the sake of computing speed.

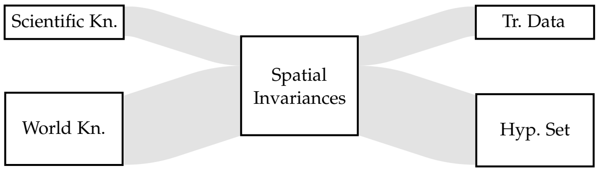
When building a machine learning model that reflects the actual, detailed behaviour of a system, low-fidelity simulation results or a response surface (a data-driven model of the simulation results) can be build into the architecture of a knowledge-based neural network (KBANN [53], see Insert 3), e.g. by replacing one or more neurons. This way, parts of the network can be used to learn a mapping from low-fidelity simulation results to a few real-world observations or high-fidelity simulations [60], [69].

Learning Algorithm. Furthermore, a simulation can directly be integrated into iterations of a a learning algorithm. For example, a realistic positioning of objects in a 3D scene can be improved by incorporating feedback from a solid-body simulation into learning [66]. By means of reinforcement learning, this is even feasible if there are no gradients available from the simulation.

Final Hypothesis. A last but important approach that we found in our survey integrates simulation results into the final hypothesis set of a machine learning model. Specifically, simulations can validate results of a trained model [19], [61], [66], [70].

5.4 Spatial Invariances

Next, we describe informed machine learning approaches involving the representation type of spatial invariances. Their main path comes from world knowledge and goes to the hypothesis set.



5.4.1 (Paths from) Knowledge Source

We mainly found references using spatial invariances in the context of world knowledge or scientific knowledge.

World Knowledge. Knowledge about invariances may fall into the category of world knowledge, for example when modeling facts about local or global pixel correlations in images [73]. Indeed, invariants are often used in image recognition where many characteristics are invariant under metric-preserving transformations. For example, in object recognition, an object should be classified correctly independent of its rotation in an image.

Scientific Knowledge. In physics, Noether's theorem states that certain symmetries (invariants) lead to conserved quantities (first integrals) and thus integrate Hamiltonian systems or equations of motion [52], [50]. For example, in equations modeling planetary motion, the angular momentum serves as such an invariant.

5.4.2 (Paths to) Knowledge Integration

In most references we found spatial invariances informing the hypothesis set.

Hypothesis Set. Invariances from physical laws can be integrated into the architecture of a neural network. For example, invariant tensor bases can be used to embed Galilean invariance for the prediction of fluid anisotropy tensors [50], or the physical Minkowski metric that reflects mass invariance can be integrated via a Lorentz layer into a neural network [51].

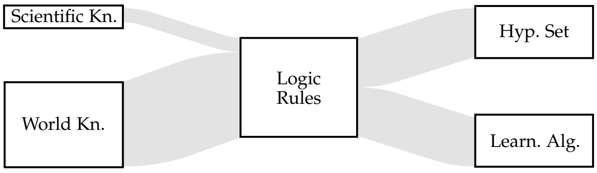
A recent trend is to integrate knowledge as spatial invariances into the architecture or layout of convolutional neural networks, which leads to so called geometric deep learning in [111]. A natural generalization of CNNs are group equivariant CNNs (G-CNNs) [71], [72], [75]. G-convolutions provide a higher degree of weight sharing and expressiveness. Simply put, the idea is to define filters based on a more general group-theoretic convolution. Another approach towards rotation invariance in image recognition considers harmonic network architecture where a certain response entanglement (arising from features that rotate at different frequencies) is resolved [76]. The goal is to design CNNs that exhibits equivariance to patch-wise translation and rotation by replacing conventional CNN filters with circular harmonics.

In support vector machines, invariances under group transformations and prior knowledge about locality can be incorporated by the construction of appropriate kernel functions [73]. In this context, local invariance is defined in terms of a regularizer that penalizes the norm of the derivative of the decision function [23].

Training Data. An early example of integrating knowledge as invariances into machine learning is the creation of virtual examples [77] and it has been shown that data augmentation through virtual examples is mathematically equivalent to incorporating prior knowledge via a regularizer. A similar approach is the creation of meta-features [83]. For instance, in turbulence modelling using the Reynolds stress tensor, a feature can be createad that is rotational, reflectional and Galilean invariant [52]. This is achieved by selecting features fulfilling rotational and Gallilean symmetries and augmenting the training data to ensure reflectional invariance.

5.5 Logic Rules

Logic Rules play an important role for the integration of prior knowledge into machine learning. In our literature survey, we mainly found the the source of world knowledge and the two integration paths into the hypothesis set and the learning algorithm.



5.5.1 (Path from) Knowledge Source

Logic rules can formalize knowledge from various sources, but the most frequent is world knowledge. Here we give some illustrative examples.

World Knowledge. Logic rules often describe knowledge about real-world objects [10], [11], [13], [78], [79] such as seen in images. This can focus on object properties, such as for animals x that $(\text{FLY}(x) \wedge \text{LAYEGGS}(x) \Rightarrow \text{BIRD}(x))$ [10]. It can also focus on relations between objects such as the co-occurrence of characters in game scenes, e.g. $(\text{PEACH} \Rightarrow \text{MARIO})$ [13].

Another knowledge domain that can be well represented by logic rules is linguistics [84], [85], [86], [91], [92], [112], [113]. Linguistic rules can consider the sentiment of a sentence (e.g., if a sentence consists of two sub-clauses connected with a 'but', then the sentiment of the clause after the 'but' dominates [86]); or the order of tags in a given word sequence (e.g., if a given text element is a citation, then it can only start with an author or editor field [84]).

Rules can also describe dependencies in social networks. For example, on a scientific research platform, it can be observed that authors citing each other tend to work in the same field $(\text{Cite}(x, y) \wedge \text{hasFieldA}(x) \Rightarrow \text{hasFieldA}(y))$ [21].

5.5.2 (Path to) Knowledge Integration

We observe that logic rules are integrated into learning mainly in the hypothesis set or, alternatively, in the learning algorithm.

Hypothesis Set. Integration into the hypothesis set comprises both deterministic and probabilistic approaches. The former include neural-symbolic systems, which use rules

Insert 3: Knowledge-Based Artificial Neural Networks (KBANNs)

Rules can be integrated into neural architectures by mapping the rule's components to the neurons and weights with these steps [53]:

1) Get rules. If needed, rewrite them to have a hierarchical structure.
2) Map rules to a network architecture. Construct (positively/negatively) weighted links for (existing/negated) dependencies.
3) Add nodes. These are not given through the initial rule set and represent hidden units.
4) Perturb the complete set of weights.

After the KBANN's architecture is built, the network is refined with learning algorithms.

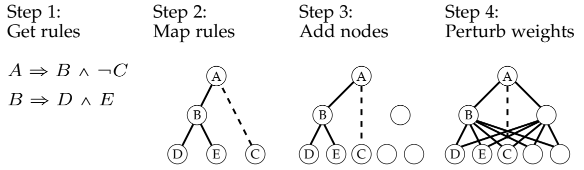


Figure 6: Steps of Rules-to-Network Translation [53]. Simple example for integrating rules into a KBANN.

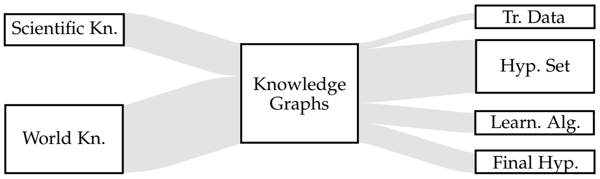
as the basis for the model structure [53], [54], [90]. In Knowledge-Based Artificial Neural Networks (KBANNs), the architecture is constructed from symbolic rules by mapping the components of propositional rules to network components [53] as further explained in Insert 3. Extensions are available that also output a revised rule set [54] or also consider first-order logic [90]. A recent survey about neural-symbolic computing [114] summarizes further methods.

Integrating logic rules into the hypothesis set in a probabilistic manner is yet another approach [78], [79], [91], [92]. These belong to the research direction of statistical relational learning [115]. Corresponding frameworks provide a logic templating language to define a probability distribution over a set of random variables. Two prominent frameworks are markov logic networks [78], [91] and probabilistic soft logic [79], [92], which translate a set of first-order logic rules to a markov random field. Each rule specifies dependencies between random variables and serves as a template for so called potential functions, which assign probability mass to joint variable configurations.

Learning Algorithm. The integration of logic rules into the learning algorithm is often accomplished via additional, semantic loss terms [10], [11], [13], [21], [84], [85], [86]. These augment the objective function similar to the knowledge-based loss terms explained above. However, for logic rules, the additional loss terms evaluate a functional that transforms rules into continuous and differentiable constraints, for example via the t-norm [10]. Semantic loss functions can also be derived from first principles using a set of axioms [11]. As a specific approach for student-teacher architectures, the rules can be first integrated in a teacher network and can then be used by a student network that is trained by minimizing a semantic loss term that measures the imitation of the teacher network [85], [86].

5.6 Knowledge Graphs

The taxonomy paths we observed in our literature survey that are related to knowledge representation are illustrated in the following graphic.



5.6.1 (Paths from) Knowledge Source

Since graphs are very versatile modeling tools, they can represent various kinds of structured knowledge. Typically, they are constructed from databases, however, the most frequent source we found in informed machine learning papers is world knowledge.

World Knowledge. Since humans perceive the world as composed of entities, graphs are often used to represent relations between visual entities. For example, the Visual Genome knowledge graph is build from human annotations of object attributes and relations between objects in natural images [15], [16]. Similarly, the MIT ConceptNet [116] encompasses concepts of everyday life and their relations automatically built from text data. In natural language processing, knowledge graphs often represent knowledge about relations among concepts, which can be referred to by words. For example, WordNet [117] represents semantic and lexical relations of words such as synonymy. Such knowledge graphs are often used for information extraction in natural language processing, but information extraction can also be used to build new knowledge graphs [118].

Scientific Knowledge. In physics, graphs can immediately describe physical systems such as spring-coupled masses [14]. In medicine, networks of gene-protein interactions describe biological pathway information [55] and the hierarchical nature of medical diagnoses is captured by classification systems such as the International Classification of Diseases (ICD) [56], [63].

5.6.2 (Paths to) Knowledge Integration

In our survey, we observed the integration of knowledge graphs in all four components of the machine learning pipeline but most prominently in the hypothesis set.

Hypothesis Set. The fact that the world consists of interrelated objects can be integrated by altering the hypothesis set. Graph neural networks operate on graphs and thus feature an object- and relation-centric bias in their architecture [24]. A recent survey [24] gives an overview over this field and explicitly names this knowledge integration relational inductive bias. This bias is of benefit, e.g. for learning physical dynamics [14], [62] or object detection [16].

In addition, graph neural networks allow for the explicit integration of a given knowledge graph as a second source of information. This allows for multi-label classification in natural images where inference about a particular object is facilitated by using relations to other objects in an image [15] (see Insert 4). More generally, a graph reasoning layer can be inserted into any neural network [82]. The main idea is to enhance representations in a given layer by propagating through a given knowledge graph.

Insert 4: Integrating Knowledge Graphs in CNNs for Image Classification

Image classification through convolutional neural networks can be improved by using knowledge graphs that reflect relations between detected objects. Technically, such relations form adjacency matrices in gated graph neural networks [15]. During the detection, the network graph is propagated, starting with detected nodes and then expanding to neighbors [24].

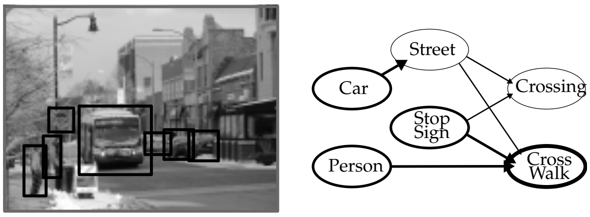


Figure 7: Illustrative application example of using neural networks and knowledge graphs for image classification, similar as in [15]. The image (from the COCO dataset) shows a pedestrian cross walk.

Another approach is to use attention mechanisms on a knowledge graph in order to enhance features. In natural language analysis, this facilitates the understanding as well as the generation of conversational text [80]. Similarly, graph-based attention mechanism are used to counteract too few data points by using more general categories [63]. Also, attention on related knowledge graph embedding can support the training of word embeddings like ERNIE [87], which are fed into language models like BERT [95], [119].

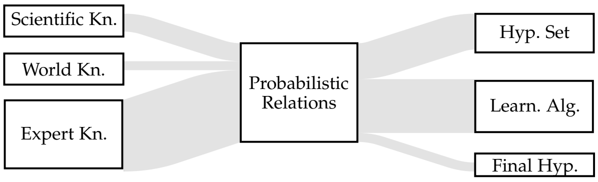
Training Data. Another prominent approach is distant supervision where information in a graph is used to automatically annotate texts to train natural language processing systems. This was originally done naïvely by considering each sentence that matches related entities in a graph as a training sample [81]; however, recently attention-based networks have been used to reduce the influence of noisy training samples [120].

Learning Algorithm. Various works discuss the integration of graph knowledge into the learning algorithm. For instance, a regularization term based on the graph Laplacian matrix can enforce strongly connected variables to behave similarly in the model, while unconnected variables are free to contribute differently. This is commonly used in bioinformatics to integrate genetic pathway information [55], [56]. Some natural language models, too, include information from a knowledge graph into the learning algorithm, e.g. when computing word embeddings. Known relations among words can be utilized as augmented contexts [89] in word2vec training [121].

Final Hypothesis. Finally, graph can also be used to improve or validate final hypotheses or trained models. For instance, a recent development is to post-process word embeddings based on information from knowledge graphs [88], [93]. In object detection, predicted probabilities of a learning system can be refined using semantic consistency measures [94] derived form knowledge graphs. In both cases, the knowledge graphs are used to indicate whether the prediction is consistent with available knowledge.

5.7 Probabilistic Relations

The most frequent paths probabilistic relations found in our literature survey comes from expert knowledge and goes to the hypothesis set or the learning algorithm.



5.7.1 *(Paths from) Knowledge Source*

Knowledge in form of probabilistic relations originates most prominently from domain experts, but can also come from other sources such as natural sciences.

Expert Knowledge. A human expert has intuitive knowledge over a domain, for example, which entities are related to each other and which are independent. Such relational knowledge, however, is often not quantified and validated and differs from, say, knowledge in natural sciences. Rather, it involves degrees of belief or uncertainty.

Human expertise exists in all domains. In the car insurance, driver features like age relate to risk aversion [96]. Another examples is computer expertise for troubleshooting, i.e relating a device status to observations [91].

Scientific Knowledge. Correlation structures can also be obtained from natural sciences knowledge. For example, correlations between genes can be obtained from gene interaction networks [122] or from a gene ontology [57].

5.7.2 *(Paths to) Knowledge Integration*

We generally observe the integration of probabilistic relations into the hypothesis set as well as into the learning algorithm and the final hypothesis.

Hypothesis Set. Expert knowledge is the basis for probabilistic graphical models. For example, Bayesian network structures are typically designed by human experts and thus fall into the category of informing the hypothesis set. Here, we focus on contributions where knowledge and Bayesian inference are combined in more intricate ways, for instance, by learning network structures from knowledge and from data. A recent overview [123] categorizes the type of prior knowledge about network structures into the presence or absence of edges, edge probabilities, and knowledge about node orders.

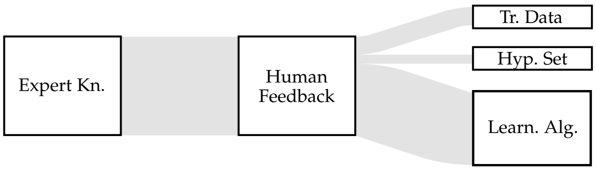
Probabilistic knowledge can be used directly in the hypothesis set. For example, extra nodes can be added to a Bayesian network thus altering the hypothesis set [97], or the structure of a probabilistic model can be chosen in accordance to given spatio-temporal structures [124]. In other hybrid approaches, the parameters of the conditional distribution of the Bayesian network are either learned from data or obtained from knowledge [74], [101].

Learning Algorithm. Human knowledge can also be used to define an informative prior [101], [125], which affects the learning algorithm as is has a regularizing effect. Structural constraints can alter score functions or the selection policies of conditional independence test, informing the search for the network structure [96]. More qualitative knowledge, e.g. observing one variable increases the probability of another, was integrated using isotonic regression, i.e. parameter estimation with order constraints [103]. Causal network inference can make use of ontologies to select the tested interventions [57]. Furthermore, prior causal knowledge can be used to constrain the direction of links in a Bayesian network [58].

Final Hypothesis. Finally, predictions obtained from a Bayesian network can be judged by probabilistic relational knowledge in order to refine the model [107].

5.8 Human Feedback

Finally, we look at informed machine learning approaches belonging to the representation type of human feedback. The most common path begins with expert knowledge and ends at the learning algorithm.



5.8.1 *(Paths from) Knowledge Source*

Compared to other categories in our taxonomy, knowledge representation via human feedback is less formalized and mainly stems from expert knowledge.

Expert Knowledge. Examples of knowledge that fall into this category include knowledge about topics in text documents [98], agent behaviors [99], [100], [104], [105], and data patterns and hierarchies [98], [106], [110]. Knowledge is often provided in form of relevance or preference feedback and humans in the loop can integrate their intuitive knowledge into the system without providing an explanation for their decision. For example, in object recognition, users can provide their corrective feedback about object boundaries via brush strokes [108]. As another example, in Game AI, an expert user can give spoken instructions for an agent in an Atari game [100].

5.8.2 *(Paths to) Knowledge Integration*

Human feedback for machine learning is usually assumed to be limited to feature engineering and data annotation. However, it can also be integrated into the learning algorithm itself. This often occurs in areas of reinforcement learning, or interactive learning combined with visual analytics.

Learning Algorithm. In reinforcement learning, an agent observes an unknown environment and learns to act based on reward signals. The TAMER framework [99] provides the agent with human feedback rather than (predefined) rewards. This way, the agent learns from observations and human knowledge alike. While these approaches can quickly learn optimal policies, it is cumbersome to obtain the human feedback for every action. Human preference w.r.t. whole action sequences, i.e. agent behaviors, can circumvent this [104]. This enables the learning of reward functions. Expert knowledge can also be incorporated through natural language interfaces [100]. Here, a human provides instructions and agents receive rewards upon completing these instructions.

Active learning offers a way to include the "human in the loop" to efficiently learn with minimal human intervention. This is based on iterative strategies where a learning algorithm queries an annotator for labels [126]. We do not consider this standard active learning as an informed learning method because the human knowledge is essentially used for label generation only. However, recent efforts integrate further knowledge into the active learning process.

Visual analytics combines analysis techniques and interactive visual interfaces to enable exploration of –and inference from– data [127]. Machine learning is increasingly combined with visual analytics. For example, visual analytics systems allow users to drag similar data points closer in order to learn distance functions [106], provide corrective feedback in object recognition [108], or even to alter correctly identified instances where the interpretation is not in line with human explanations [109], [110].

Lastly, various tools exist for text analysis, in particular for topic modeling [98] where users can create, merge and refine topics or change keyword weights. They thus impart knowledge by generating new reference matrices (term-by-topic and topic-by-document matrices) that are integrated in a regularization term that penalizes the difference between the new and the old reference matrices. This is similar to the semantic loss term described above.

Training Data and Hypothesis Set. Another approach towards incorporating expert knowledge in reinforcement learning considers human demonstration of problem solving. Expert demonstrations can be used to pre-train a deep Q-network, which accelerates learning [105]. Here, prior knowledge is integrated into the hypothesis set and the training data since the demonstrations inform the training of the Q-network and, at the same time, allow for interactive learning via simulations.

6 Historical Background

The idea of integrating knowledge into learning has a long history. Historically, AI research roughly considered the two antipodal paradigms of symbolism and connectionism. The former dominated up until the 1980s and refers to reasoning based on symbolic knowledge; the latter became more popular in the 1990s and considers data-driven decision making using neural networks. Especially Minsky [128] pointed out limitations of symbolic AI and promoted a stronger focus on data-driven methods to allow for causal and fuzzy reasoning. Already in the 1990s were knowledge data bases used together with training data to obtain knowledge-based artificial neural networks [53]. In the 2000s, when support vector machines (SVMs) were the de-facto paradigm in classification, there was interest in incorporating knowledge into this formalism [23]. Moreover, in the geosciences, and most prominently in weather forecasting, knowledge integration dates back to the 1950s. Especially the discipline of data assimilation deals with techniques that combine statistical and mechanistic models to improve prediction accuracy [129], [130].

7 Discussion of Challenges and Directions

Our findings about the main approaches of informed machine learning are summarized in Table 4. It gives for each approach the taxonomy path, its main motivation, the central approach idea, remarks to potential challenges, and our viewpoint on current or future directions. For further details on the methods themselves and the corresponding papers, we refer to Section 5. In the following, we discuss the challenges and directions for these main approaches, sorted by the integrated knowledge representations.

Prior knowledge in the form of algebraic equations can be integrated as constraints via knowledge-based loss terms (e.g., [12], [13], [35]). Here, we see a potential challenge in finding the right weights for supervision from knowledge vs. data labels. Currently, this is solved by setting the hyperparameters for the individual loss terms [12]. However, we think that strategies from more recently developed learning algorithms, such as self-supervised [131] or few-shot learning [132], could also advance the supervision from prior knowledge. Moreover, we suggest further research on theoretical concepts based on the existing generalization bounds from statistical learning theory [133], [134] and the connection between regularization and effective hypothesis space [135].

Differential equations can be integrated similarly, but with a specific focus on physics-informed neural networks that constrain the model derivatives by the underlying differential equation (e.g., [20], [45], [46]). A potential challenge is the robustness of the solution, which is the subject of current research. One approach is to investigate the the model quality by a suitable quanitification of its uncertainty [43], [46]. We think, a more in-depth comparison with existing numerical solvers [136] would also be helpful. Another challenge of physical systems is the generation and integration of sensor data in real-time. This is currently tackled by online learning methods [48]. Furthermore, we think that techniques from data assimilation [130] could also be helpful to combine modelling from knowledge and data.

Simulation results can be used for synthetic data generation or augmentation (e.g., [18], [19], [59]), but this can bring up the challenge of a mismatch between real and simulated data. A promising direction to close the gap is domain adaptation, especially adversarial training [67], [137], or domain randomization [138]. Moreover, for future work we see further potential in the development of new hybrid systems that combine machine learning and simulation in more sophisticated ways [139].

The utilization of spatial invariances through model architectures with invariant characteristics, such as group equivariant or convolutional networks, diminish the model search space (e.g., [71], [72], [76]). Here, a potential challenge is the proper invariance specification and implementation [76] or expensive evaluations on more complex geometries [111]. Therefore, we think that the efficient adaptation of invariant-based models to further scenarios can further improve geometric-based representation learning [111].

Logic rules can be encoded in the architecture of knowledge-based neural networks (KBANNs), (e.g., [53], [54], [90]). Since this idea was already developed when neural networks had only a few layers, a question is, if it is still feasible for deep neural networks. In order to improve the practicality, we suggest to develop automated interfaces for knowledge integration. A future direction could be the development of new neuro-symbolic systems. Although the combination of connectionist and symbolic systems into hybrid systems is a longtime idea [140], [141], it is currently getting more attention [142], [143]. Another challenge, especially in statistical relational learning (SRL), such as Markov logic networks or probabilistic soft logic (e.g., [79], [92], [144]). is the aquisition of rules when they are not yet given. An ongoing research topic to this end is the learning of rules from data, which is called structure learning [145].

Knowledge graphs can be integrated into learning systems either explicitly via graph propagation and attention mechanisms, or implicitly via graph neural networks with relational inductive bias (e.g., [14], [15], [16]). A challenge is the comparability between different methods, because authors often use template like ConceptNet [80] or VisualGenome [15], [16] and customize the graphs in to improve running time and performance. Since the choice of graph can have high influence [82], we suggest a pool of standardized graphs in order to improve comparability, or even to establish benchmarks. Another interesting direction is to combine graph using and graph learning. A requirement here is the need for good entity linking models in approaches such as KnowBERT [95] and ERNIE [87] and the continuous embedding of new facts in the graph.

Probabilistic Relations can be integrated as prior knowledge in terms of a-priori probability distributions that are refined with additional observations (e.g., [74], [97], [101]). The main challenges are the large computational effort and the formalization of knowledge in terms of inductive priors. Directions responding to this are variational methods with origins in optimization theory and functional analysis [146] and variational neural networks [147]. Besides scaling issues, an explicit treatment of causality is becoming more important in machine learning and closely related to graphical probabilistic models [148].

Human feedback can be integrated into the learning algorithm by human-in-the-loop (HITL) reinforcement learning (e.g., [99], [104]), or by explanation alignment through interactive learning combined with visual analytics (e.g., [109], [110]). However, the exploration of human feedback can be very expensive due to its latency in real systems. Exploratory actions could hamper user experience [149], [150], so that online reinforcement learning is generally avoided. A promising approach is learning a reward estimator [151], [152] from collected logs, which then provides unlimited feedback for unseen instances that do not have any human judgments. Another challenge is that human feedback is often intuitive and not formalized and thus difficult to incorporate into machine learning systems. Also

Table 4: **Main Approaches of Informed Machine Learning.** The approaches are sorted by taxonomy path and knowledge representation. Methodical details can be found in Section 5. Challenges and directions are discussed in Section 7.

Taxonomy Path			**Main Motivation**	**Central Approach Idea**	**Potential Challenge**	**Current / Future Directions**
Source	Represent.	Integration				
Scientific Knowl.	Algebraic Equations (See Sec. 5.1)	Learning Algor.	Less data, Knowl. conform.	Knowledge-based loss terms from constraints (see Insert 1)	Weighting supervision from data labels vs. knowledge	Hyperparameter setting, Novel learning algorithms, Extension of learning theory
	Differential Equations (See Sec. 5.2)	Learning Algor.	Knowl. conform., Less data	Physics-informed neural networks with derivatives in loss function	Solution robustness, Real-time data generation and integration	Uncertainty quantification, Numerical solver comparison, Online learning, data assimilation
	Simulation Results (See Sec. 5.3)	Training Data	Less data	Synthetic data generation or data augmentation (see Insert 2)	Sim-to-real gap, i.e. mismatch between real and simulated data	Adversarial domain adaptation, Domain randomization; Hybrid systems
World Knowl.	Spatial Invariances (See Sec. 5.4)	Hypoth. Set	Performance (Small models)	Models with invariant characteristics, e.g. group equivariant DNNs/CNNs	Invariance specification, expensive geometric evaluations	Geometric-based representation learning, Adaptaion to complex scenarios
	Logic Rules (See Sec. 5.5)	Hypoth. Set	Performance	KBANNs (see Insert 3); SRL (e.g., Markov logic networks, prob. soft logic)	Feasibility for deep neural networks; Acquisition of rules	Automated integration interface, Neuro-symbolic systems; Structure learning
	Knowl. Graphs (See Sec. 5.6)	Hypoth. Set	Performance, Less data	Gr. propagation (see Insert 4), attention; Gr. neural networks (relational inductive bias)	Comparability with custom graphs, Getting the graph, Entity linking	Standardized graph data pool, Combine graph using and learning, Neuro-symbolic systems
Expert Knowl.	Probabilistic Relations (See Sec. 5.7)	Hypoth. Set	Less data	Informed structure of prob. graphical models, informative priors	High computational effort, Formalization of knowledge	Variational methods combining prob. models with numerical opt., Probabilistic neural networks
	Human Feedback (See Sec. 5.8)	Learning Algor.	Less data, Performance, Interpretability	HITL Reinforcement learning; Explanation alignment via Visual anal./interactive ml	Feedback latency; Formalization of intuition, Evaluation methods	Reward estimation from logs; Representation transformation, Utilization for interpretability

human-gorunded evaluation is very costly, especially compared to functionally-grounded evaluation [153]. Therefore we suggest to further study representation transformations to formalize intuitive knowledge, e.g. from human feedback to logical rules. Furthermore, we found that improved interpretability still only is a minor goal for knowledge integration (see Figure 4). This, too, suggests opportunities for future work.

Even if these directions are motivated by specific approaches, we think that they are generally relevant and can advance the whole field of informed machine learning.

8 Conclusion

In this paper, we presented a unified classification framework for the explicit integration of additional prior knowledge into machine learning, which we described using the umbrella term of *informed machine learning*. Our main contribution is the development of a taxonomy that allows a structured categorization of approaches and the uncovering of main paths. Moreover, we presented a conceptual clarification of informed machine learning, as well as a systematic and comprehensive research survey. This helps current and future users of informed machine learning to identify the right methods to use their prior knowledge, for example, to deal with insufficient training data or to make their models more robust.

Acknowledgments

This work is a joint effort of the Fraunhofer Research Center for Machine Learning (RCML) within the Fraunhofer Cluster of Excellence Cognitive Internet Technologies (CCIT) and the Competence Center for Machine Learning Rhine Ruhr (ML2R) which is funded by the Federal Ministry of Education and Research of Germany (grant no. 01|S18038B). The team of authors gratefully acknowledges this support.

Moreover the authors would like to thank Dorina Weichert, Daniel Paurat, Lars Hillebrand, Theresa Bick and Nico Piatkowski for helpful discussions.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems (NIPS)*, 2012.

[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *Signal Processing Magazine*, vol. 29, 2012.

[3] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," *arxiv:1606.01781*, 2016.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, 2016.

[5] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, "Machine learning for molecular and materials science," *Nature*, vol. 559, no. 7715, 2018.

[6] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman *et al.*, "Opportunities and obstacles for deep learning in biology and medicine," *J. Royal Society Interface*, vol. 15, no. 141, 2018.

[7] J. N. Kutz, "Deep learning in fluid dynamics," *J. Fluid Mechanics*, vol. 814, 2017.

[8] M. Brundage, S. Avin, J. Wang, H. Belfield, G. Krueger, G. Hadfield, H. Khlaaf, J. Yang, H. Toner, R. Fong *et al.*, "Toward trustworthy ai development: mechanisms for supporting verifiable claims," *arXiv:2004.07213*, 2020.

[9] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," *arxiv:1905.08883*, 2019.

[10] M. Diligenti, S. Roychowdhury, and M. Gori, "Integrating prior knowledge into deep learning," in *Int. Conf. on Machine Learning and Applications (ICMLA)*. IEEE, 2017.

[11] J. Xu, Z. Zhang, T. Friedman, Y. Liang, and G. V. d. Broeck, "A semantic loss function for deep learning with symbolic knowledge," *arxiv:1711.11157*, 2017.

[12] A. Karpatne, W. Watkins, J. Read, and V. Kumar, "Physics-guided neural networks (pgnn): An application in lake temperature modeling," *arxiv:1710.11431*, 2017.

[13] R. Stewart and S. Ermon, "Label-free supervision of neural networks with physics and domain knowledge," in *Conf. Artificial Intelligence*. AAAI, 2017.

[14] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende *et al.*, "Interaction networks for learning about objects, relations and physics," in *Neural Information Processing Systems (NIPS)*, 2016.

[15] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," in *Conf. Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[16] C. Jiang, H. Xu, X. Liang, and L. Lin, "Hybrid knowledge routed modules for large-scale object detection," in *Neural Information Processing Systems (NIPS)*, 2018.

[17] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, 2015.

[18] K.-H. Lee, J. Li, A. Gaidon, and G. Ros, "Spigan: Privileged adversarial learning from simulation," in *Int. Conf. Learning Representations (ICLR)*, 2019.

[19] J. Pfrommer, C. Zimmerling, J. Liu, L. Kärger, F. Henning, and J. Beyerer, "Optimisation of manufacturing process parameters using deep neural networks as surrogate models," *Procedia CIRP*, vol. 72, no. 1, 2018.

[20] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations," *arxiv:1711.10561*, 2017.

[21] M. Diligenti, M. Gori, and C. Sacca, "Semantic-based regularization for learning and inference," *Artificial Intelligence*, vol. 244, 2017.

[22] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar, "Theory-guided data science: A new paradigm for scientific discovery from data," *Trans. Knowledge and Data Engineering*, vol. 29, no. 10, 2017.

[23] F. Lauer and G. Bloch, "Incorporating prior knowledge in support vector machines for classification: A review," *Neurocomputing*, vol. 71, no. 7-9, 2008.

[24] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, "Relational inductive biases, deep learning, and graph networks," *arxiv:1806.01261*, 2018.

[25] M. Steup, "Epistemologyp," in *The Stanford Encyclopedia of Philosophy (Winter 2018 Edition), Edward N. Zalta (ed.)*, 2018.

[26] L. Zagzebski, *What is Knowledge?* John Wiley & Sons, 2017.

[27] P. Machamer and M. Silberstein, *The Blackwell guide to the philosophy of science*. John Wiley & Sons, 2008, vol. 19.

[28] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI magazine*, vol. 17, no. 3, 1996.

[29] D. Kahneman, *Thinking, Fast and Slow*. Macmillan, 2011.

[30] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and brain sciences*, vol. 40, 2017.

[31] H. G. Gauch, *Scientific method in practice*. Cambridge University Press, 2003.

[32] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data*. AMLBook, 2012, vol. 4.

[33] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, and N. Ramakrishnan, "Incorporating prior domain knowledge into deep neural networks," in *Int. Conf. Big Data*. IEEE, 2018.

[34] Y. Lu, M. Rajora, P. Zou, and S. Liang, "Physics-embedded machine learning: Case study with electrochemical micro-machining," *Machines*, vol. 5, no. 1, 2017.

[35] R. Heese, M. Walczak, L. Morand, D. Helm, and M. Bortz, "The good, the bad and the ugly: Augmenting a black-box model with expert knowledge," in *Int. Conf. Artificial Neural Networks (ICANN)*. Springer, 2019.

[36] G. M. Fung, O. L. Mangasarian, and J. W. Shavlik, "Knowledge-based support vector machine classifiers," in *Neural Information Processing Systems (NIPS)*, 2003.

[37] O. L. Mangasarian and E. W. Wild, "Nonlinear knowledge-based classification," *Trans. Neural Networks*, vol. 19, no. 10, 2008.

[38] R. Ramamurthy, C. Bauckhage, R. Sifa, J. Schücker, and S. Wrobel, "Leveraging domain knowledge for reinforcement learning using mmc architectures," in *Int. Conf. Artificial Neural Networks (ICANN)*. Springer, 2019.

[39] M. von Kurnatowski, J. Schmid, P. Link, R. Zache, L. Morand, T. Kraft, I. Schmidt, and A. Stoll, "Compensating data shortages in manufacturing with monotonicity knowledge," *arXiv:2010.15955*, 2020.

[40] C. Bauckhage, C. Ojeda, J. Schücker, R. Sifa, and S. Wrobel, "Informed machine learning through functional composition."

[41] R. King, O. Hennigh, A. Mohan, and M. Chertkov, "From deep to physics-informed learning of turbulence: Diagnostics," *arxiv:1810.07785*, 2018.

[42] S. Jeong, B. Solenthaler, M. Pollefeys, M. Gross *et al.*, "Data-driven fluid simulations using regression forests," *ACM Trans. Graphics*, vol. 34, no. 6, 2015.

[43] Y. Yang and P. Perdikaris, "Physics-informed deep generative models," *arxiv:1812.03511*, 2018.

[44] E. de Bezenac, A. Pajot, and P. Gallinari, "Deep learning for physical processes: Incorporating prior scientific knowledge," *arxiv:1711.07970*, 2017.

[45] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *Trans. Neural Networks*, vol. 9, no. 5, 1998.

[46] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris, "Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data," *J. Computational Physics*, vol. 394, 2019.

[47] D. C. Psichogios and L. H. Ungar, "A hybrid neural network-first principles approach to process modeling," *AIChE Journal*, vol. 38, no. 10, pp. 1499–1511, 1992.

[48] M. Lutter, C. Ritter, and J. Peters, "Deep lagrangian networks: Using physics as model prior for deep learning," *arxiv:1907.04490*, 2019.

[49] F. D. A. Belbute-peres, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "End-to-end differentiable physics for learning and control," in *Neural Information Processing Systems (NIPS)*, 2018.

[50] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *J. Fluid Mechanics*, vol. 807, 2016.

[51] A. Butter, G. Kasieczka, T. Plehn, and M. Russell, "Deep-learned top tagging with a lorentz layer," *SciPost Phys*, vol. 5, no. 28, 2018.

[52] J.-L. Wu, H. Xiao, and E. Paterson, "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," *Physical Review Fluids*, vol. 3, no. 7, 2018.

[53] G. G. Towell and J. W. Shavlik, "Knowledge-based artificial neural networks," *Artificial Intelligence*, vol. 70, no. 1-2, 1994.

[54] A. S. d. Garcez and G. Zaverucha, "The connectionist inductive learning and logic programming system," *Applied Intelligence*, vol. 11, no. 1, 1999.

[55] T. Ma and A. Zhang, "Multi-view factorization autoencoder with network constraints for multi-omic integrative analysis," in *Int. Conf. Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018.

[56] Z. Che, D. Kale, W. Li, M. T. Bahadori, and Y. Liu, "Deep computational phenotyping," in *Int. Conf. Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 2015.

[57] M. B. Messaoud, P. Leray, and N. B. Amor, "Integrating ontological knowledge for iterative causal discovery and visualization," in *European Conf. Symbolic and Quantitative Approaches to Reasoning and Uncertainty*. Springer, 2009.

[58] G. Borboudakis and I. Tsamardinos, "Incorporating causal prior knowledge as path-constraints in bayesian networks and maximal ancestral graphs," *arxiv:1206.6390*, 2012.

[59] T. Deist, A. Patti, Z. Wang, D. Krane, T. Sorenson, and D. Craft, "Simulation assisted machine learning." *Bioinformatics (Oxford, England)*, 2019.

[60] H. S. Kim, M. Koc, and J. Ni, "A hybrid multi-fidelity approach to the optimal design of warm forming processes using a knowledge-based artificial neural network," *Int. J. Machine Tools and Manufacture*, vol. 47, no. 2, 2007.

[61] G. Hautier, C. C. Fischer, A. Jain, T. Mueller, and G. Ceder, "Finding nature's missing ternary oxide compounds using machine learning and density functional theory," *Chemistry of Materials*, vol. 22, no. 12, 2010.

[62] M. B. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum, "A compositional object-based approach to learning physical dynamics," *arxiv:1612.00341*, 2016.

[63] E. Choi, M. T. Bahadori, L. Song, W. F. Stewart, and J. Sun, "Gram: Graph-based attention model for healthcare representation learning," in *Int. Conf. Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 2017.

[64] A. Lerer, S. Gross, and R. Fergus, "Learning physical intuition of block towers by example," *arxiv:1603.01312*, 2016.

[65] A. Rai, R. Antonova, F. Meier, and C. G. Atkeson, "Using simulation to improve sample-efficiency of bayesian optimization for bipedal robots." *J. Machine Learning Research*, vol. 20, no. 49, 2019.

[66] Y. Du, Z. Liu, H. Basevi, A. Leonardis, B. Freeman, J. Tenenbaum, and J. Wu, "Learning to exploit stability for 3d scene parsing," in *Neural Information Processing Systems (NIPS)*, 2018.

[67] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Conf. Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[68] F. Wang and Q.-J. Zhang, "Knowledge-based neural models for microwave design," *Trans. Microwave Theory and Techniques*, vol. 45, no. 12, 1997.

[69] S. J. Leary, A. Bhaskar, and A. J. Keane, "A knowledge-based approach to response surface modelling in multifidelity optimization," *J. Global Optimization*, vol. 26, no. 3, 2003.

[70] L. von Rueden, T. Wirtz, F. Hueger, J. D. Schneider, N. Piatkowski, and C. Bauckhage, "Street-map based validation of semantic segmentation in autonomous driving," in *Int. Conf. Pattern Recognition (ICPR)*. IEEE, 2020.

[71] T. S. Cohen and M. Welling, "Group equivariant convolutional networks," in *Int. Conf. Machine Learning (ICML)*, 2016.

[72] S. Dieleman, J. De Fauw, and K. Kavukcuoglu, "Exploiting cyclic symmetry in convolutional neural networks," *arxiv:1602.02660*, 2016.

[73] B. Schölkopf, P. Simard, A. J. Smola, and V. Vapnik, "Prior knowledge in support vector kernels," in *Neural Information Processing Systems (NIPS)*, 1998.

[74] B. Yet, Z. B. Perkins, T. E. Rasmussen, N. R. Tai, and D. W. R. Marsh, "Combining data and meta-analysis to build bayesian networks for clinical decision support," *J. Biomedical Informatics*, vol. 52, 2014.

[75] J. Li, Z. Yang, H. Liu, and D. Cai, "Deep rotation equivariant network," *Neurocomputing*, vol. 290, 2018.

[76] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: Deep translation and rotation equivariance," in *Conf. Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[77] P. Niyogi, F. Girosi, and T. Poggio, "Incorporating prior information in machine learning by creating virtual examples," *Proc. of the IEEE*, vol. 86, no. 11, 1998.

[78] M. Schiegg, M. Neumann, and K. Kersting, "Markov logic mixtures of gaussian processes: Towards machines reading regression data," in *Artificial Intelligence and Statistics*, 2012.

[79] M. Sachan, K. A. Dubey, T. M. Mitchell, D. Roth, and E. P. Xing, "Learning pipelines with limited data and domain knowledge: A study in parsing physics problems," in *Neural Information Processing Systems (NIPS)*, 2018.

[80] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu, "Commonsense knowledge aware conversation generation with graph attention." in *Int. Joint Conf. Artificial Intelligence (IJCAI)*, 2018.

[81] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Association for Computational Linguistics (ACL)*, 2009.

[82] X. Liang, Z. Hu, H. Zhang, L. Lin, and E. P. Xing, "Symbolic graph reasoning meets convolutions," in *Neural Information Processing Systems (NIPS)*, 2018.

[83] D. L. Bergman, "Symmetry constrained machine learning," in *SAI Intelligent Systems Conf.* Springer, 2019.

[84] M.-W. Chang, L. Ratinov, and D. Roth, "Guiding semi-supervision with constraint-driven learning," in *Association for Computational Linguistics (ACL)*, 2007.

[85] Z. Hu, Z. Yang, R. Salakhutdinov, and E. Xing, "Deep neural networks with massive learned knowledge," in *Conf. Empirical Methods in Natural Language Processing*, 2016.

[86] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing, "Harnessing deep neural networks with logic rules," *arxiv:1603.06318*, 2016.

[87] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "Ernie: Enhanced language representation with informative entities," *arxiv:1905.07129*, 2019.

[88] N. Mrkšić, D. O. Séaghdha, B. Thomson, M. Gašić, L. Rojas-Barahona, P.-H. Su, D. Vandyke, T.-H. Wen, and S. Young, "Counter-fitting word vectors to linguistic constraints," *arxiv:1603.00892*, 2016.

[89] J. Bian, B. Gao, and T.-Y. Liu, "Knowledge-powered deep learning for word embedding," in *Joint European Conf. machine learning and knowledge discovery in databases*. Springer, 2014.

[90] M. V. França, G. Zaverucha, and A. S. d. Garcez, "Fast relational learning using bottom clause propositionalization with artificial neural networks," *Machine Learning*, vol. 94, no. 1, 2014.

[91] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, 2006.

[92] A. Kimmig, S. Bach, M. Broecheler, B. Huang, and L. Getoor, "A short introduction to probabilistic soft logic," in *NIPS Workshop on Probabilistic Programming: Foundations and Applications*, 2012.

[93] G. Glavaš and I. Vulić, "Explicit retrofitting of distributional word vectors," in *Association for Computational Linguistics (ACL)*, 2018.

[94] Y. Fang, K. Kuan, J. Lin, C. Tan, and V. Chandrasekhar, "Object detection meets knowledge graphs," 2017.

[95] M. E. Peters, M. Neumann, R. Logan, R. Schwartz, V. Joshi, S. Singh, and N. A. Smith, "Knowledge enhanced contextual word representations," in *Conf. Empirical Methods in Natural Language Processing (EMNLP), Int. Joint Conf. Natural Language Processing (IJCNLP)*, 2019.

[96] L. M. de Campos and J. G. Castellano, "Bayesian network learning algorithms using structural restrictions," *Int. J. Approximate Reasoning*, vol. 45, no. 2, 2007.

[97] A. C. Constantinou, N. Fenton, and M. Neil, "Integrating expert knowledge with data in bayesian networks: Preserving data-driven expectations when the expert variables remain unobserved," *Expert Systems with Applications*, vol. 56, 2016.

[98] J. Choo, C. Lee, C. K. Reddy, and H. Park, "Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization," *Trans. Visualization and Computer Graphics*, vol. 19, no. 12, 2013.

[99] W. B. Knox and P. Stone, "Interactively shaping agents via human reinforcement: The tamer framework," in *Int. Conf. Knowledge Cature (K-CAP)*. ACM, 2009.

[100] R. Kaplan, C. Sauer, and A. Sosa, "Beating atari with natural language guided reinforcement learning," *arxiv:1704.05539*, 2017.

[101] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, 1995.

[102] M. Richardson and P. Domingos, "Learning with knowledge from multiple experts," in *Int. Conf. Machine Learning (ICML)*, 2003.

[103] A. Feelders and L. C. Van der Gaag, "Learning bayesian network parameters under order constraints," *Int. J. Approximate Reasoning*, vol. 42, no. 1-2, 2006.

[104] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Neural Information Processing Systems (NIPS)*, 2017.

[105] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband *et al.*, "Deep q-learning from demonstrations," in *Conf. Artificial Intelligence*. AAAI, 2018.

[106] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang, "Dis-function: Learning distance functions interactively," in *Conf. Visual Analytics Science and Technology (VAST)*. IEEE, 2012.

[107] B. Yet, Z. Perkins, N. Fenton, N. Tai, and W. Marsh, "Not just data: A method for improving prediction with knowledge," *J. Biomedical Informatics*, vol. 48, 2014.

[108] J. A. Fails and D. R. Olsen Jr, "Interactive machine learning," in *Int. Conf. Intelligent User Interfaces*. ACM, 2003.

[109] L. Rieger, C. Singh, W. J. Murdoch, and B. Yu, "Interpretations are useful: penalizing explanations to align neural networks with prior knowledge," *arXiv:1909.13584*, 2019.

[110] P. Schramowski, W. Stammer, S. Teso, A. Brugger, H.-G. Luigs, A.-K. Mahlein, and K. Kersting, "Right for the wrong scientific reasons: Revising deep networks by interacting with their explanations," *arXiv:2001.05371*, 2020.

[111] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *Signal Processing*, vol. 34, no. 4, 2017.

[112] M.-W. Chang, L. Ratinov, and D. Roth, "Structured learning with constrained conditional models," *Machine Learning*, vol. 88, no. 3, 2012.

[113] D. Sridhar, J. Foulds, B. Huang, L. Getoor, and M. Walker, "Joint models of disagreement and stance in online debate," in *Association for Computational Linguistics (ACL)*, 2015.

[114] A. S. d. Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran, "Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning," *arxiv:1905.06088*, 2019.

[115] L. D. Raedt, K. Kersting, and S. Natarajan, *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Morgan & Claypool Publishers, 2016.

[116] R. Speer and C. Havasi, "Conceptnet 5: A large semantic network for relational knowledge," in *The People's Web Meets NLP*. Springer, 2013, pp. 161–176.

[117] G. A. Miller, "Wordnet: A lexical database for english," *Communications ACM*, vol. 38, no. 11, 1995.

[118] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel *et al.*, "Never-ending learning," *Communications ACM*, vol. 61, no. 5, 2018.

[119] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arxiv:1810.04805*, 2018.

[120] Z.-X. Ye and Z.-H. Ling, "Distant supervision relation extraction with intra-bag and inter-bag attentions," *arxiv:1904.00143*, 2019.

[121] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arxiv:1301.3781*, 2013.

[122] M. S. Massa, M. Chiogna, and C. Romualdi, "Gene set analysis exploiting the topology of a pathway," *BMC systems biology*, vol. 4, no. 1, 2010.

[123] N. Angelopoulos and J. Cussens, "Bayesian learning of bayesian networks with informative priors," *Annals of Mathematics and Artificial Intelligence*, vol. 54, no. 1-3, 2008.

[124] N. Piatkowski, S. Lee, and K. Morik, "Spatio-temporal random fields: Compressible representation and distributed estimation," *Machine Learning*, vol. 93, no. 1, 2013.

[125] R. Fischer, N. Piatkowski, C. Pelletier, G. I. Webb, F. Petitjean, and K. Morik, "No cloud on the horizon: Probabilistic gap filling in satellite image series," in *Int. Conf. Data Science and Advanced Analytics (DSAA)*, 2020.

[126] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

[127] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, "Visual analytics: Definition, process, and challenges," in *Information visualization*. Springer, 2008.

[128] M. L. Minsky, "Logical versus analogical or symbolic versus connectionist or neat versus scruffy," *AI magazine*, vol. 12, no. 2, 1991.

[129] E. Kalnay, *Atmospheric modeling, data assimilation and predictability*. Cambridge university press, 2003.

[130] S. Reich and C. Cotter, *Probabilistic forecasting and Bayesian data assimilation*. Cambridge University Press, 2015.

[131] M. Janner, J. Wu, T. D. Kulkarni, I. Yildirim, and J. B. Tenenbaum, "Self-supervised intrinsic image decomposition," in *Neural Information Processing Systems (NIPS)*, 2017.

[132] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, 2020.

[133] F. Cucker and D. X. Zhou, *Learning theory: an approximation theory viewpoint*. Cambridge University Press, 2007, vol. 24.

[134] I. Steinwart and A. Christmann, *Support vector machines*. Springer Science & Business Media, 2008.

[135] F. Cucker and S. Smale, "Best choices for regularization parameters in learning theory: on the bias-variance problem," *Foundations of computational Mathematics*, vol. 2, no. 4, 2002.

[136] L. Lapidus and G. F. Pinder, *Numerical solution of partial differential equations in science and engineering*. John Wiley & Sons, 2011.

[137] M. Wulfmeier, A. Bewley, and I. Posner, "Addressing appearance change in outdoor robotics with adversarial domain adaptation," in *Int. Conf. Intelligent Robots and Systems (IROS)*. IEEE, 2017.

[138] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *Int. Conf. Robotics and Automation (ICRA)*. IEEE, 2018.

[139] L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, and J. Garcke, "Combining machine learning and simulation to a hybrid modelling approach: Current and future directions," in *Int. Symp. Intelligent Data Analysis (IDA)*. Springer, 2020.

[140] K. McGarry, S. Wermter, and J. MacIntyre, "Hybrid neural systems: From simple coupling to fully integrated neural networks," *Neural Computing Surveys*, vol. 2, no. 1, 1999.

[141] R. Sun, "Connectionist implementationalism and hybrid systems," *Encyclopedia of Cognitive Science*, 2006.

[142] A. S. d. Garcez and L. C. Lamb, "Neurosymbolic ai: The 3rd wave," *arXiv:2012.05876*, 2020.

[143] T. Dong, C. Bauckhage, H. Jin, J. Li, O. Cremers, D. Speicher, A. B. Cremers, and J. Zimmermann, "Imposing category trees onto word-embeddings using a geometric construction," in *Int. Conf. Learning Representations (ICLR)*, 2018.

[144] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, "Hinge-loss markov random fields and probabilistic soft logic," *arxiv:1505.04406*, 2015.

[145] V. Embar, D. Sridhar, G. Farnadi, and L. Getoor, "Scalable structure learning for probabilistic soft logic," *arXiv:1807.00973*.

[146] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. American Statistical Association*, vol. 112, no. 518, 2017.

[147] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *arXiv:1906.02691*, 2019.

[148] J. Pearl, *Causality*. Cambridge university press, 2009.

[149] J. Kreutzer, S. Riezler, and C. Lawrence, "Learning from human feedback: Challenges for real-world reinforcement learning in nlp," *arXiv:2011.02511*, 2020.

[150] G. Dulac-Arnold, D. Mankowitz, and T. Hester, "Challenges of real-world reinforcement learning," *arXiv:1904.12901*, 2019.

[151] J. Kreutzer, J. Uyheng, and S. Riezler, "Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning," in *Association for Computational Linguistics (ACL)*, 2018.

[152] Y. Gao, C. M. Meyer, and I. Gurevych, "April: Interactively learning to summarise by combining active preference learning and reinforcement learning," in *Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[153] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv:1702.08608*, 2017.

Laura von Rueden received a B.Sc. degree in physics and a M.Sc. degree in simulation sciences in 2015 from RWTH Aachen University. Afterwards, she was a data scientist at Capgemini. Since 2018, she is a research scientist at Fraunhofer IAIS and a PhD candidate in computer science at the Universität Bonn. Her research interests include machine learning and especially the combination of data- and knowledge-based modelling.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2021.3079836, IEEE Transactions on Knowledge and Data Engineering

Sebastian Mayer received his diploma degree in Mathematics in 2011 from TU Darmstadt and his PhD in Mathematics in 2018 from University Bonn. Since 2017, he is a research scientist at Fraunhofer SCAI. His research interests are machine learning and biologically-inspired algorithms in the context of cyber-physical systems.

Katharina Beckh received her M.Sc. degree in Human-Computer Interaction in 2019 from Julius Maximilian University of Wuerzburg. Since 2019, she is a research scientist at Fraunhofer IAIS. Her research interests include interactive machine learning, human oriented modeling and text mining with a primary focus in the medical domain.

Bogdan Georgiev received his Ph.D. degree in Mathematics in 2018 from Max-Planck-Institute and Bonn University. Since 2018, he is a research scientist at Fraunhofer IAIS. His present research interests include aspects of learning theory such as generalization/compression bounds, geometric learning and quantum computing.

Sven Giesselbach received his M.Sc. degree in computer science in 2012 from university of Bonn. Since 2015, he is a data scientist at Fraunhofer IAIS and is also lead of the team natural language understanding at the department knowledge discovery. His research interests include the use of external knowledge in natural language processing.

Raoul Heese received the Diploma and PhD from the Institute of Quantum Physics, Ulm University, Germany in 2012 and 2016. He is currently working as a research scientist at Fraunhofer ITWM, Kaiserslautern, Germany. His research interests include informed learning, supervised learning and their application to real-world problems.

Birgit Kirsch received her M.Sc. degree in business informatics in 2017 from Hochschule Trier. Since 2017, she is a research scientist at Fraunhofer IAIS. Her research interests include Natural Language Processing and Statistical Relational Learning.

Michal Walczak received his PhD degree in physics from the Georg-August University of Goettingen, Germany, in 2014. Since 2016 he is a research scientist at Fraunhofer ITWM in Kaiserslautern, Germany. His research interests include machine learning, decision support, multi-criteria optimization, and their application to radiotherapy planning and process engineering.

Julius Pfrommer received his PhD degree in computer science in 2019 from Karlsruhe Institute of Technology (KIT). Since 2018, he is the head of a research group at Fraunhofer IOSB. His research interests include distributed systems, planning under uncertainty, and optimization theory with its many applications for machine learning and optimal control.

Annika Pick received her M.Sc. degree in Computer Science in 2018 from University of Bonn. Since 2019, she is a data scientist at Fraunhofer IAIS. Her research interests include learning from healthcare data and pattern mining.

Rajkumar Ramamurthy received his M.Sc. in Media Informatics in 2016 from RWTH Aachen University. Since 2018, he is a Data Scientist at Fraunhofer IAIS and a Doctoral candidate at the University of Bonn. His research interests include reinforcement learning and natural language processing.

Jochen Garcke received his diploma degree in mathematics in 1999, and the Ph.D. degree in mathematics in 2004, both from the Universität Bonn. He was a Postdoctoral Fellow at the Australian National University from 2004 to 2006, a Postdoctoral Researcher from 2006 to 2008 and a Junior Research Group Leader from 2008 to 2011, both at the Technical University Berlin. Since 2011 he is Professor of numerics at the University of Bonn and department head at Fraunhofer SCAI, Sankt Augustin. His research interests include machine learning, scientific computing, reinforcement learning, and high-dimensional approximation. Prof. Garcke is a member of DMV, GAMM, and SIAM. He is a reviewer for IEEE Transactions on Industrial Informatics, IEEE Transactions on Neural Networks, and IEEE Transactions on Pattern Analysis and Machine Intelligence.

Christian Bauckhage (M'02) received M.Sc. and Ph.D. degrees in computer science from Bielefeld University in 1998 and 2002, respectively. Since 2008, he is a professor of computer science at the University of Bonn and lead scientist for machine learning at Fraunhofer IAIS. Previously, he was with the Centre for Vision Research in Toronto, Canada and a Senior Research Scientist at Deutsche Telekom Laboratories, Berlin. His research focuses on theory and practice of learning systems and next generation computing. He regularly reviews for the IEEE Transactions on Neural Networks and Learning Systems, the IEEE Transactions on Pattern Analysis and Machine Intelligence, and the IEEE Transactions on Games for which he is also an associate editor.

Jannis Schuecker received his doctoral degree in physics from the RWTH Aachen University. Until 2019, he was a research scientist at Fraunhofer IAIS. His research interests include machine learning, in particular, time series modeling using neural networks and interpretable machine learning.

APPENDIX B

Paper P2) Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions

Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions

Laura von Rueden[1,2](✉), Sebastian Mayer[1,3], Rafet Sifa[1,2], Christian Bauckhage[1,2], and Jochen Garcke[1,3,4]

[1] Fraunhofer Center for Machine Learning, Sankt Augustin, Germany
[2] Fraunhofer IAIS, Sankt Augustin, Germany
[3] Fraunhofer SCAI, Sankt Augustin, Germany
[4] Institute for Numerical Simulation, University of Bonn, Bonn, Germany
laura.von.rueden@iais.fraunhofer.de

Abstract. In this paper, we describe the combination of machine learning and simulation towards a hybrid modelling approach. Such a combination of data-based and knowledge-based modelling is motivated by applications that are partly based on causal relationships, while other effects result from hidden dependencies that are represented in huge amounts of data. Our aim is to bridge the knowledge gap between the two individual communities from machine learning and simulation to promote the development of hybrid systems. We present a conceptual framework that helps to identify potential combined approaches and employ it to give a structured overview of different types of combinations using exemplary approaches of simulation-assisted machine learning and machine-learning assisted simulation. We also discuss an advanced pairing in the context of Industry 4.0 where we see particular further potential for hybrid systems.

Keywords: Machine learning · Simulation · Hybrid approaches

1 Introduction

Machine learning and *simulation* have a similar goal: To predict the behaviour of a system with data analysis and mathematical modelling. On the one side, machine learning has shown great successes in fields like image classification [21], language processing [24], or socio-economic analysis [7], where causal relationships are often only sparsely given but huge amounts of data are available. On the other side, simulation is traditionally rooted in natural sciences and engineering, e.g. in computational fluid dynamics [35], where the derivation of causal relationships plays an important role, or in structural mechanics for the performance evaluation of structures regarding reactions, stresses, and displacements [6].

However, some applications can benefit from combining machine learning and simulation. Such an hybrid approach can be useful when the processing

M. R. Berthold et al. (Eds.): IDA 2020, LNCS 12080, pp. 548–560, 2020.
https://doi.org/10.1007/978-3-030-44584-3_43

capabilities of classical simulation computations can not handle the available dimensionality of the data, for example in earth system sciences [30], or when the behaviour of a system that is supposed to be predicted is based on both known, causal relationships and unknown, hidden dependencies, for example in risk management [25]. However, such challenges are in practice often still approached distinctly with either machine learning or simulation, apparently because they historically originate from distinct fields. This raises the question how these two modelling approaches can be combined into a hybrid approach in order to foster intelligent data analysis. Here, a key challenge in developing a hybrid modelling approach is to bridge the knowledge gap between the two individual communities, which are mostly either experts for machine learning or experts for simulation. Both groups have extremely deep knowledge about the methods used in their particular fields. However, the respectively used terminologies are different, so that an exchange of ideas between both communities can be impeded.

Related work that describes a combination of machine learning with simulation can roughly be divided in two groups, not surprisingly, either from a machine learning or a simulation point of view. The first group frequently describes the integration of simulation into machine learning as an additional source for training data, for example in autonomous driving [23], thermodynamics [19], or biomedicine [13]. A typical motivation is the augmentation of data for scenarios that are not sufficiently represented in the available data. The second group of related works describes the integration of machine learning techniques in simulation, often for a specific application, such as car crash simulation [6], fluid simulation [38], or molecular simulation [26]. A typical motivation is to identify surrogate models [16], which offer an approximate but cheaper to evaluate model to replace the full simulation. Another technique that is used to adapt a dynamical simulation model to new measurements is data assimilation, which is traditionally used in weather forecasting [22]. Related work that considers an equal combination of machine learning and simulation is quite rare. A work that is closest to describing such a hybrid, symbiotic modelling approach is [4].

More general, the integration of prior knowledge into machine learning can be described as *informed machine learning* [34] or *theory-guided data science* [18]. The paper [34] presents a survey with a taxonomy that structures approaches according to the knowledge type, representation, and integration stage. We reuse those categories in this paper. However, that survey considers a much broader spectrum of knowledge representations, from logic rules over simulation results to human interaction, while this paper puts an explicit focus on simulations.

Our goal is to make the key components of the two modelling approaches *machine learning* and *simulation* transparent and to show the versatile, potential combination possibilities in order to inspire and foster future developments of hybrid systems. We do not intend to go into technical details but rather give a high-level methodological overview. With our paper we want to outline a vision of a stronger, more automated interplay between data- and simulation-based analysis methods. We mainly aim our findings at the data analysis and machine

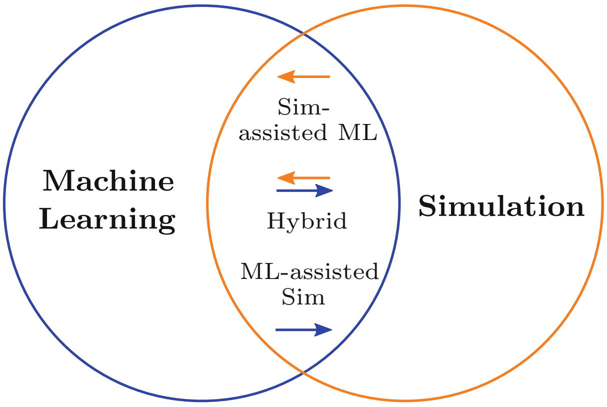


Fig. 1. Subfields of Combining Machine Learning and Simulation. The fields of machine learning and simulation have an intersecting area, which we partition into three subfields: 1. Simulation-assisted machine learning describes the integration of simulations into machine learning. 2. Machine-learning assisted simulation describes the integration of machine learning into simulation. 3. A hybrid combination describes a combination of machine learning and simulation with a strong mutual interplay.

learning community, but also those from the simulation community are welcome to read on. Generally, our target audience are researchers and users of one of the two modelling approaches who want to learn how they can use the other one.

The contributions of this paper are: 1. A conceptual framework serving as an orientation aid for comparing and combining machine learning and simulation, 2. a structured overview of combinations of both modelling approaches, 3. our vision of a hybrid approach with a stronger interplay of data- and simulation based analysis.

The paper is structured as follows: In Sect. 2 we give a brief overview of the subfields that result from combining machine learning and simulation. In Sect. 3 we present these two separate modelling approaches along our conceptual framework. In Sect. 4 we describe the versatile combinations by giving exemplary references and applications. In Sect. 5 we further discuss our observations in Industry 4.0 projects that lead us to a vision for the advanced pairing of machine learning and simulation. Finally we conclude in Sect. 6.

2 Overview

In this section, we give a short overview about the subfields that result from a combination of machine learning with simulation. We view the combination with equal focus on both fields, driving our vision of a hybrid modelling approach with a stronger and automated interplay. Figure 1 illustrates our view on the fields' overlap, which can be partitioned into the three subfields simulation-assisted machine learning, machine-learning assisted simulation, and a hybrid combination. Even though the first two can be regarded as one-sided approaches because they describe the integration with a point of view from one approach, the last one can be regarded as a two-sided approach. Although the term *hybrid*

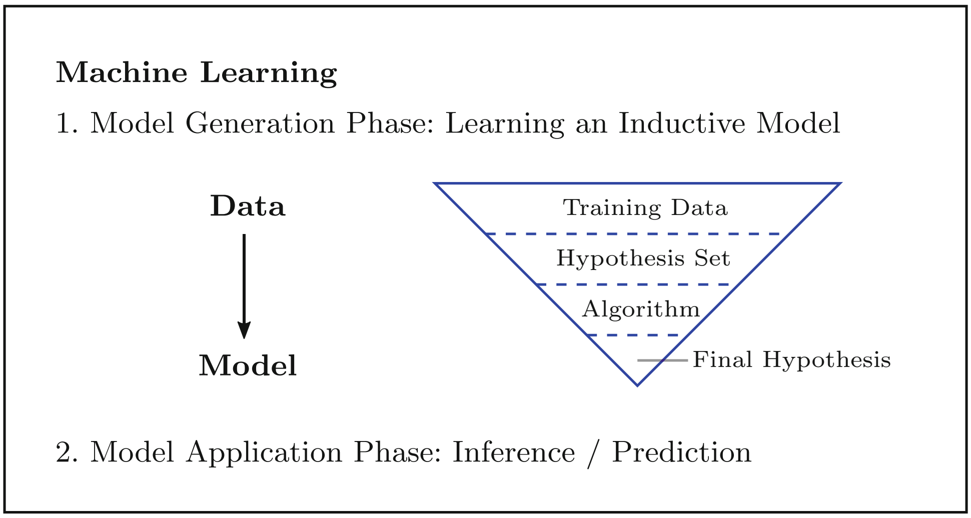

Machine Learning
1. Model Generation Phase: Learning an Inductive Model
Data
Model
Training Data
Hypothesis Set
Algorithm
Final Hypothesis
2. Model Application Phase: Inference / Prediction

Fig. 2. Components of Machine Learning. Machine Learning consists of two phases 1. model generation, and 2. model application, where the focus is usually made on the first phase, in which an inductive model is learned from data. The components of this phase are the training data, a hypothesis set, a learning algorithm, and a final hypothesis [1,34]. It describes the finding of patterns in an initially large data space, which are finally represented in a condensed form by the final hypothesis. This is illustrated by the reversed triangle and can be described as a "bottom-up approach".

is in the literature often used for the above one-sided approaches, we prefer to use it only for the two-sided approach where machine learning and simulation have a strong mutual, symbiotic-like interplay.

3 Modelling Approaches

In this section, we describe the two modelling approaches by means of a conceptual framework that aims to make them and their components transparent and comparable.

3.1 Machine Learning

The main goal of machine learning is that a machine automatically learns a model that describes patterns in given data. The typical components of machine learning are illustrated in Fig. 2. In the first, main phase an inductive model is learned. Inductive means that the model is built by drawing conclusions from samples and is thus not guaranteed to depict causal relationships, but can instead identify hidden, previously unknown patterns, meaning that the model is usually not knowledge-based but rather data-based. This inductive model can finally be applied to new data in order to predict or infer a desired target variable.

The model generation phase can be roughly split into four sub-phases or respective components [1,34]. Firstly, training data is prepared that depicts historical records of the investigated process or system. Secondly, a hypothesis set

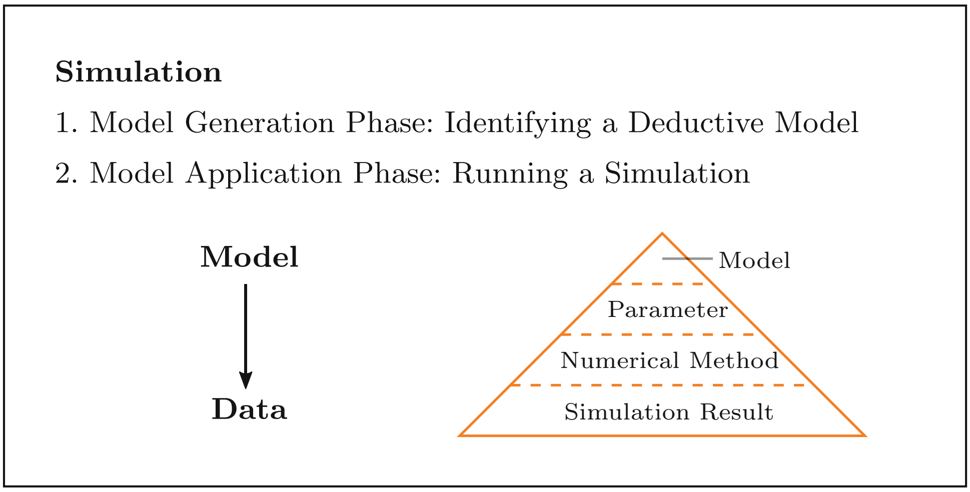

Simulation
1. Model Generation Phase: Identifying a Deductive Model
2. Model Application Phase: Running a Simulation
Model
Data
Model
Parameter
Numerical Method
Simulation Result

Fig. 3. Components of Simulation. Simulation comprises the two phases 1. model generation, and 2. model application, where the focus often is on the second phase, in which an earlier identified deductive model is used in order to create simulation results. The components of this phase are the simulation model, input parameters, a numerical method, and the simulation result. It describes the unfolding of local interactions from a compactly represented initial model into an expanded data space. This is supposed be illustrated by the triangle and can be described as a "top-down approach".

is defined in the form of a function class or network architecture that is assumed to map input features to the target variables. Thirdly, a learning algorithm tunes the parameters of the hypothesis set so that the performance of the mapping is maximized by using optimization algorithms like gradient descent and results in, fourthly, the final hypothesis, which is the desired inductive model. This model generation phase is often repeated in a loop-like manner by tuning hyper-parameters until a sufficient model performance is achieved.

3.2 Simulation

The goal of a simulation is to predict the behaviour of a system or process for a particular situation. There are different types of simulations, ranging from cellular automata, over agent-based simulations, to equation-based simulations [9,15,36]. In the following we concentrate on the last type, which is based on mathematical models and is especially used in science and engineering. The first, required stage preceding the actual simulation is the identification of a deductive model, often in the form of differential equations. Deductive in this context means that the model describes causal relationships and can thus be called knowledge-based. Such models are often developed through extensive research, starting with a derivation, for example in theoretical physics, and continuing with plentiful experimental validations. Some recent research exists of proof-of-concepts for identifying models directly from data [8,33].

The main phase of a simulation is the application of the identified model for a specific scenario, often called running a simulation. This phase can be described in four typical main components or sub-phases, which are, as illustrated in Fig. 3, the mathematical model, the input parameters, the numerical

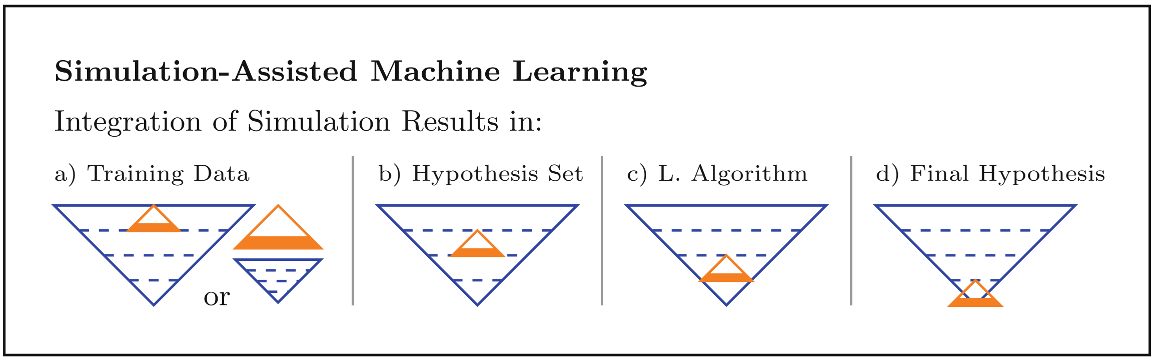

Simulation-Assisted Machine Learning
Integration of Simulation Results in:
a) Training Data
or
b) Hypothesis Set
c) L. Algorithm
d) Final Hypothesis

Fig. 4. Types of Simulation-Assisted Machine Learning. Simulations, in particular the simulation results, can be generally integrated into the four different components of machine learning. The triangles illustrate the machine learning (blue/dark gray) or the simulation (orange/light gray) approach and their components, which are themselves presented in Figs. 2 and 3. The simulation results can be used to (a) augment the training data, (b) define parts of the hypothesis set in the form of empirical functions, (c) steer the training algorithm in generative adversarial networks, or (d) verify the final hypothesis against scientific consistency. (Color figure online)

method, and finally the simulation result [36]. After the selection of a mathematical model, the input parameters that describe the specific scenario are defined in the second sub-phase. They can comprise general parameters such as the spatial domain or time of interest, as well as initial conditions quantifying the systems' or processes' initial status and boundary conditions defining the behaviour at domain borders. In the third sub-phase, a numerical method computes the solution of the given model observing the constraints resulting from the input parameters. Examples for numerical methods are finite differences, finite elements or finite volume methods for spatial discretization [36], or particle methods based on interaction forces [26]. These form the basis for an approximate solution, which is the final simulation result. This model application phase is often repeated in a loop-like manner, e.g., by tuning the discretization to achieve a desired approximation accuracy and stability of the solution.

4 Combining Machine Learning and Simulation

In this section, we describe combinations of machine learning and simulation by using our conceptual framework from Sect. 3. Here, we focus on simulation-assisted machine learning and machine-learning assisted simulation. For each of the methodical combination types, we give exemplary application references.

4.1 Simulation-Assisted Machine Learning

Simulation offers an additional source of information for machine learning that goes beyond typically available data and that is rich of knowledge. This additional information can be integrated into the four components of machine learning as illustrated in Fig. 4. In the following, we will give an overview about these integration types by giving for each an illustrative example and refer for a more detailed discussion to [34].

Simulations are particularly useful for creating additional training data in a controlled environment. This is for example applied in autonomous driving, where simulations such as physics engines are employed to create photo-realistic traffic scenes, which can be used as synthetic training data for learning tasks like semantic segmentation [14], or for adversarial test generation [40]. As another example, in systems biology, simulations can be integrated in the training data of kernelized machine learning methods [13].

Moreover, simulations can be integrated into the hypothesis set, either directly as the solvers or through deduced, empirical functions that compactly describe the simulations results. These functions can be built into the architecture of a neural network, as shown for the application of finding an optimal design strategy for a warm forming process [20].

The integration of simulations into the learning algorithm can for example be realized by generative adversarial networks (GANs), which learn a prediction function that obeys constraints, which might be unknown but are implicitly given through a simulation [31].

Another important integration type is in the validation of the final hypothesis by simulations. An example for this comes from material discovery, where first a machine learning model suggests new compounds based on patterns in a data basis, and second the physical properties are computed and thus checked by a density functional theory simulation [17].

An approach that uses simulations along the whole machine learning pipeline is reinforcement learning (RL), when the model is learned in a simulated environment [2]. Studies under the keyword "sim-to-real" are often concerned with robots learning to grip or move unknown objects in simulations and usually require retraining in reality. An application for controlling the temperature of plasma follows the analogous approach, i.e., a training based on a software-physics model, where the learned RL model is then further adapted for use in reality [41].

4.2 Machine-Learning Assisted Simulation

Machine learning is often used in simulation with the intention to support the solution process or to detect patterns in the simulation data. With respect to our conceptual framework presented in Sect. 3, machine learning techniques can be used for the initial model, the input parameters, the numerical method, and the final simulation results, as illustrated in Fig. 4. In the following we will give an overview about the integration types. Again, we do not intend to cover the full spectrum of machine-learning assisted simulation, we rather want to illustrate its diverse approaches through representative examples.

A prominent integration type of machine learning techniques into simulation is the identification of simpler models, such as surrogate models [11,12,16,26].

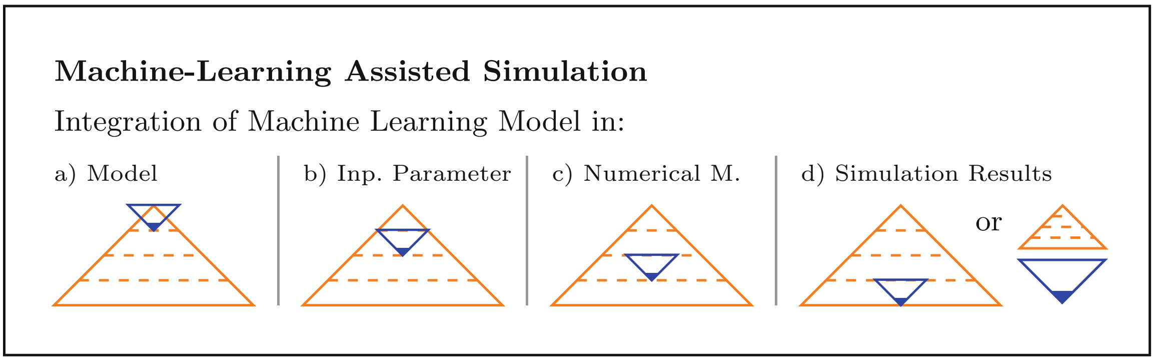

Machine-Learning Assisted Simulation
Integration of Machine Learning Model in:
a) Model
b) Inp. Parameter
c) Numerical M.
d) Simulation Results
or

Fig. 5. Types of Machine-Learning Assisted Simulation. Machine learning techniques, in particular the final hypothesis, can be used in different simulation components. The triangles illustrate the machine learning (blue/dark gray) or the simulation (orange/light gray) approach and their components, which are themselves explained in Figs. 2 and 3. Exemplary use cases for machine learning models in simulation are (a) model order reduction and the development of surrogate models that offer approximate but simpler solutions, (b) the automated inference of an intelligent choice of input parameters for a next simulation run, (c) a partly trainable solver for differential equations, or d) the identification of patterns in simulation results for scientific discovery. (Color figure online)

These are approximate and cheap to evaluate models that are particularly of interest when the solution of the original, more precise model is very time- or resource-consuming. The surrogate model can then be used to analyse the overall behaviour of the system in order to reveal scenarios that should be further investigated with the detailed original simulation model. Such surrogate models can be developed with machine-learning techniques either with data from real-world experiments, or with data from high-fidelity simulations. One application example is the optimization of process parameters using deep neural networks as surrogate models [27]. Kernel-based approaches are also commonly used as surrogate models for simulations, an example to improve the energetic efficiency of a gas transport network is shown in [10]. A well-established approach for surrogate modelling is model order reduction, for example with proper orthogonal decomposition, which is closely related to principal component analysis [5,37].

Data assimilation, which includes the calibration of constitutive models and the estimation of system states, is another area where machine learning techniques enhance simulations. Data assimilation problems can be modelled using dynamic Bayesian networks with continuous physically interpretable state spaces where the evaluation of transition kernels and observation operators requires forward-simulation runs [29].

Machine learning techniques can also be used to study the parameter dependence of simulation results. For example, after an engineer executes a sequence of simulations, a machine learning model can detect different behavioral modes in the results and thus reduce the analysis effort during the engineering process [6]. This supports the selection of the parameter setting for the next simulation, for which active learning techniques can also be employed. For example, [39] studied it for selecting the molecules for which the internal energy shall be determined by computationally expensive quantum-mechanical calculations, as well as for determining a surrogate model for the fluid flow in a well-bore while drilling.

The integration of machine learning techniques into the numerical method can support to obtain the numerical solution. One approach is to exchange parts of the model that are resource-consuming to solve, with learned models that can be computed faster, for example with machine learning generated force fields in molecular dynamics simulations [26]. Another approach that is recently investigated are trainable solvers for partial differential equations that determine the complete solution through a neural network [28].

A further, very important integration type is the application of machine learning techniques on the simulation results in order to detect patterns, often motivated by the goal of scientific discovery. While there are plenty of application domains, two exemplary representatives are particle physics [3] and earth-sciences, for example with the use of convolutional neural networks for the detection of weather patterns on climate simulation data [30]. For further examples we refer to a survey about explainable machine learning for scientific discovery [32].

5 Advanced Pairing of Machine Learning and Simulation

Section 4 gave a brief overview of the versatile existing approaches that integrate aspects of machine learning into simulation and vice versa, or that combine simulation and machine learning sequentially. Yet, we think that the integration of these two established worlds is only at the beginning, both in terms of modelling approaches and in terms of available software solutions.

In the following, we describe a number of observations from our project experience in the development of cyber-physical systems for Industry 4.0 applications that support this assessment. Note that the key technical goal of Industry 4.0 is the flexibilization of production processes. In addition to the broad integration of digital equipment in the production machinery, a key provider of flexibilization is a decrease of process design and dimensioning times and ideally, a merging of planning and production phase that are today still strictly separated. This requires a new generation of computer-aided engineering (CAE) software systems that allow for very fast process optimization cycles with real time feedback loops to the production machinery. An advanced pairing of machine learning and simulation will be key to realize such systems by addressing the following issues:

- **Simulation results are not fully exploited:** Especially in the industrial practice, simulations are run with a very specific analysis goal based on expert-designed quantities of interest. This ignores that the simulation result might reveal more patterns and regularities, which might be irrelevant for the current analysis goal but useful in other contexts.
- **Selective surrogate modelling**: Even if modern machine learning approaches are used, surrogate models are built for very specific purposes and the decision when and where to use a surrogate model is left to domain

experts. In this way, it is exploited too little that similar underlying systems might lead to similar surrogate models and in consequence, too many costly high-fidelity simulations are run to generate the data basis, although parts of the learned surrogate models could be transferred.
- **Parameter studies and simulation engines:** Parameter and design studies are well-established tools in many fields of engineering. Surprisingly, the frameworks to conduct these studies and to build the surrogate models are third-party solutions that are separated from the core simulation engines. For the parameter study framework, the simulation engine is a black box, which does not know that it is currently used for a parameter study. In turn, the standard rules to generate sampling points in the parameter space are not aware about the internals of the simulation engine. This raises the question how much more efficient parameter studies could be conducted so that both software systems were stronger connected to each other.

These observations lead us to a research concept that we propose in this paper and call it **learning simulation engines**. A learning simulation engine is a hybrid system that combines machine learning and simulation in an optimal way. Such an engine can automatically decide when and where to apply learned surrogate models or high-fidelity simulations. Surrogate models are efficiently organized and re-used through the use of transfer learning. Parameter and design optimization is an integral component of the learning simulation engine and active learning methods allow the efficient re-use of costly high-fidelity computations.

Of course, the vision of a learning simulation engine raises numerous research questions. We describe some of them in view of Fig. 1. First of all, the question is how learning and simulation can be technically combined to such an advanced hybrid approach, especially, if they can only be integrated into each other by using the final simulation results and the final hypothesis (as shown in Figs. 4 and 5), or if they can also be combined at an earlier sub-phase. Moreover, the counterparts of the learning's model generation phase and the simulation's model application phase (see Figs. 2 and 3) should be investigated further in order to better understand the similarities and differences to the simulation's model generation phase and a learning's model application phase.

6 Conclusion

In this paper, we described the combination of machine learning and simulation motivated by fostering intelligent analysis of applications that can benefit from a combination of data- and knowledge-based solution approaches.

We categorized the overlap between the two fields into three sub-fields, namely, simulation-assisted machine learning, machine-learning assisted simulation, and a hybrid approach with a strong and mutual interplay. We presented a conceptual framework for the two separate approaches, in order to make them and their components transparent for the development of a potential combined approach. In summary, it describes machine learning as a bottom-up approach that generates an inductive, data-based model and simulation as a top-down approach that applies a deductive, knowledge-based model. Using this conceptual framework as an orientation aid for their integration into each other, we gave a structured overview about the combination of machine learning and simulation. We showed the versatility of the approaches through exemplary methods and use cases, ranging from simulation-based data augmentation and scientific consistency checking of machine learning models, to surrogate modelling and pattern detection in simulations for scientific discovery. Finally, we described the scenario of an advanced pairing of machine learning and simulation in the context of Industry 4.0 where we see particular further potential for hybrid systems.

References

1. Abu-Mostafa, Y.S., Magdon-Ismail, M., Lin, H.T.: Learning From Data (2012)
2. Akkaya, I., et al.: Solving rubik's cube with a robot hand (2019). arXiv:1910.07113
3. Albertsson, K., Altoe, P., Anderson, D., Andrews, M., Espinosa, J.P.A., Aurisano, A., Basara, L., Bevan, A., Bhimji, W., et al.: Machine learning in high energy physics community white paper. J. of Phys.: Conf. Ser. **1085**, 022008 (2018)
4. Baker, R.E., Pena, J.M., Jayamohan, J., Jérusalem, A.: Mechanistic models versus machine learning, a fight worth fighting for the biological community? Biol. Lett. **14**(5), 20170660 (2018)
5. Benner, P., Gugercin, S., Willcox, K.: A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev. **57**(4), 483–531 (2015)
6. Bohn, B., Garcke, J., Iza-Teran, R., Paprotny, A., Peherstorfer, B., Schepsmeier, U., Thole, C.A.: Analysis of car crash simulation data with nonlinear machine learning methods. Proc. Comput. Sci. **18**, 621–630 (2013)
7. Bollen, J., Mao, H., Pepe, A.: Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In: AAAI Conference Weblogs and Social Media (2011)
8. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Nat. Acad. Sci. **113**(15), 3932–3937 (2016)
9. Bungartz, H.J., Zimmer, S., Buchholz, M., Pflger, D.: Modeling and Simulation (2014)
10. Clees, T., Hornung, N., Nikitin, I., Nikitina, L., Steffes-lai, D.: RBF-metamodel driven multi-objective optimization and its applications. Int. J. Adv. Intell. Syst. **9**(1), 19–24 (2016)
11. Cozad, A., Sahinidis, N.V., Miller, D.C.: Learning surrogate models for simulation-based optimization. AIChE J. **60**(6), 2211–2227 (2014)
12. Cranmer, K., Brehmer, J., Louppe, G.: The frontier of simulation-based inference (2019). arXiv:1911.01429
13. Deist, T.M., Patti, A., Wang, Z., Krane, D., Sorenson, T., Craft, D.: Simulation-assisted machine learning. Bioinformatics **35**(20), 4072–4080 (2019)
14. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator (2017). arXiv:1711.03938
15. Durán, J.M.: Computer Simulations in Science and Engineering. TFC. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-90882-3

16. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modelling: A Practical Guide. John Wiley, Hoboken (2008)
17. Hautier, G., Fischer, C.C., Jain, A., Mueller, T., Ceder, G.: Finding natures missing ternary oxide compounds using machine learning and density functional theory. Chem. Mater. **22**(12), 3762–3767 (2010)
18. Karpatne, A., Atluri, G., Faghmous, J.H., Steinbach, M., Banerjee, A., Ganguly, A., Shekhar, S., Samatova, N., Kumar, V.: Theory-guided data science: a new paradigm for scientific discovery from data. IEEE Trans. Knowl. Data Eng. **29**(10), 2318–2331 (2017)
19. Karpatne, A., Watkins, W., Read, J., Kumar, V.: Physics-guided neural networks (pgnn): an application in lake temperature modeling (2017). arXiv:1710.11431
20. Kim, H.S., Koc, M., Ni, J.: A hybrid multi-fidelity approach to the optimal design of warm forming processes using a knowledge-based artificial neural network. Int. J. Mach. Tools Manuf. **47**(2), 211–222 (2007)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
22. Lahoz, W., Khattatov, B., Menard, R. (eds.): Data Assimilation. Making Sense of Observations. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-540-74703-1
23. Lee, K.H., Li, J., Gaidon, A., Ros, G.: Spigan: Privileged adversarial learning from simulation. In: ICLR (2019)
24. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS (2013)
25. Mitchell-Wallace, K., Foote, M., Hillier, J., Jones, M.: Natural Catastrophe Risk Management and Modelling: A practitioner's Guide. John Wiley, Hoboken (2017)
26. Noé, F., Tkatchenko, A., Müller, K.R., Clementi, C.: Machine learning for molecular simulation (2019). arXiv:1911.02792
27. Pfrommer, J., Zimmerling, C., Liu, J., Kärger, L., Henning, F., Beyerer, J.: Optimisation of manufacturing process parameters using deep neural networks as surrogate models. Proc. CIRP **72**(1), 426–431 (2018)
28. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations (2017). arXiv:1711.10561
29. Reich, S., Cotter, C.: Probabilistic Forecasting and Bayesian Data Assimilation. Cambridge University Press, Cambridge (2015)
30. Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al.: Deep learning and process understanding for data-driven earth system science. Nature **566**(7743), 195–204 (2019)
31. Ren, H., Stewart, R., Song, J., Kuleshov, V., Ermon, S.: Adversarial constraint learning for structured prediction. In: IJCAI (2018)
32. Roscher, R., Bohn, B., Duarte, M.F., Garcke, J.: Explainable machine learning for scientific insights and discoveries (2020). IEEE Access
33. Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Data-driven discovery of partial differential equations. Sci. Adv. **3**(4), e1602614 (2017)
34. von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., Walczak, M., Garcke, J., Bauckhage, C., Schuecker, J.: Informed machine learning - a taxonomy and survey of integrating knowledge into learning systems (2020). arXiv:1903.12394v2
35. Shaw, C.T.: Using Computational Fluid Dynamics (1992)
36. Strang, G.: Computational Science and Engineering, vol. 791 (2007)

37. Swischuk, R., Mainini, L., Peherstorfer, B., Willcox, K.: Projection-based model reduction: formulations for physics-based machine learning. Comput. Fluids **179**, 704–717 (2019)
38. Tompson, J., Schlachter, K., Sprechmann, P., Perlin, K.: Accelerating Eulerian fluid simulation with convolutional networks. In: ICML (2017)
39. Tsymbalov, E., Makarychev, S., Shapeev, A., Panov, M.: Deeper connections between neural networks and gaussian processes speed-up active learning (2019). arXiv:1902.10350
40. Tuncali, C.E., Fainekos, G., Ito, H., Kapinski, J.: Simulation-based adversarial test generation for autonomous vehicles with machine learning components. In: IEEE Intelligent Vehicles Symposium (2018)
41. Witman, M., Gidon, D., Graves, D.B., Smit, B., Mesbah, A.: Sim-to-real transfer reinforcement learning for control of thermal effects of an atmospheric pressure plasma jet plasma sources. Sci. Technol. **28**(9), 095019 (2019)

APPENDIX C

Paper P3) Informed Pre-Training of Neural Networks Using Prototypes from Prior Knowledge

Informed Pre-Training of Neural Networks Using Prototypes from Prior Knowledge

Laura von Rueden[1,2], Sebastian Houben[2], Kostadin Cvejoski[2], Jochen Garcke[1,3], Christian Bauckhage[1,2], Nico Piatkowski[2]
[1]*University of Bonn,* [2]*Fraunhofer IAIS,* [3]*Fraunhofer SCAI,* Sankt Augustin, Germany

***Abstract*—We present a novel approach for hybrid AI and propose informed pre-training on prototypes from prior knowledge.**

Generally, when training data is scarce, the incorporation of additional knowledge can assist the learning process of neural networks. An approach that recently gained a lot of interest is *informed* machine learning, which integrates prior knowledge that is explicitly given by formal representations, such as graphs or equations. However, the integration often is application-specific and can be time-consuming. Another more straightforward approach is *pre-training* on other large data sets, which allows to reuse knowledge that is implicitly stored in trained models. This raises the question, if it is also possible to pre-train a neural network on a small set of knowledge representations.

In this paper, we investigate this idea and propose *informed pre-training on knowledge prototypes*. Such prototypes are often available and represent characteristic semantics of the domain. We show that it (i) improves generalization capabilities, (ii) increases out-of-distribution robustness, and (iii) speeds up learning. Moreover, we analyze which parts of a neural network model are affected most by our informed pre-training approach. We discover that (iv) improvements come from deeper layers that typically represent high-level features, which confirms the transfer of semantic knowledge. This is a before unobserved effect and shows that informed transfer learning has additional and complementary strengths to existing approaches.

***Index Terms*—Hybrid AI, Informed Machine Learning, Pre-Training, Transfer Learning, Prototypes, Prior Knowledge**

I. Introduction

Combining neural and symbolic reasoning capabilities towards a hybrid modelling approach is a longstanding goal in the area of artificial intelligence [1]–[3]. Over the last decade, deep learning with neural networks has led to impressive performance and became the predominant paradigm in AI [4]. However, their massive data requirements have raised important questions about how to learn from small data, how to generalize to unseen domains, and how to ensure model robustness [5]–[7].

One approach to alleviate problems due to insufficient training data for a specific learning task is to build upon models that have been pre-trained on other large datasets. However, this relies on reusing the implicit information from large datasets, which is not necessarily controllable. Thus relevant task-specific concepts still need to be learned. Moreover, appropriate large datasets or pre-trained models are not always available. Another promising approach is to inject additional prior knowledge via informed machine learning methods, as proposed by [8]. While this ensures an alignment with semantic concepts, the integration of formally represented knowledge into learning algorithms or model architectures can be application-specific and time-consuming, which in turn raises the need for an improved method.

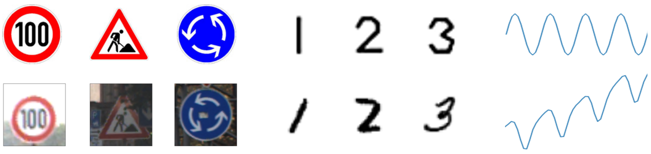


Fig. 1: Data distributions often follow domain invariant relationships that are given by prototypes from prior knowledge. The figure shows examples for knowledge prototypes (top) and corresponding natural data items (bottom) for three datasets: GTSRB (left), MNIST (middle), CO2 (right). Our paper shows that informed pre-training on such prior knowledge is possible and significantly improves generalization and robustness.

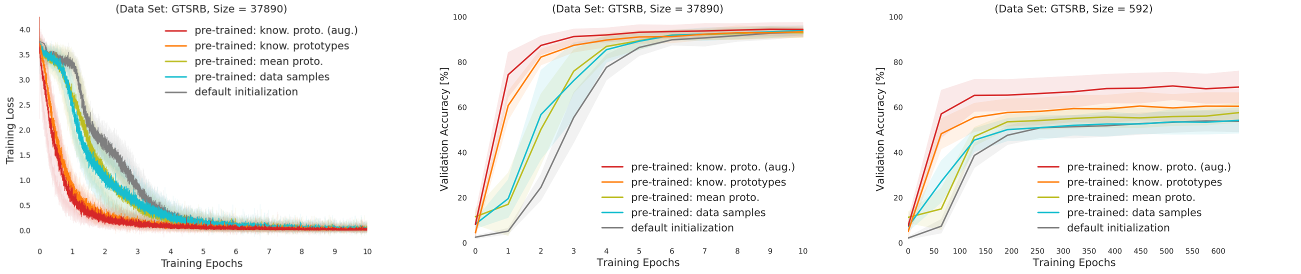
We propose to merge pre-training and knowledge-informed learning. Our idea is inspired by human learning: When students learn to read digits, teachers show them a prototypical image for each category (see Figure 1). This lets students initially focus on the relevant information and avoids distraction by any semantically unimportant feature. After having internalized the main concept, the learner can simply refine it based on new observations. We transfer this idea to machine learning and answer the following research questions:

How can we transfer prior knowledge into a neural network? Can we pre-train a neural network on a few knowledge representations?

In this paper, we propose the novel approach of informed pre-training on prototypes from prior knowledge. Given prior knowledge is often represented by image templates, graph structures or physical equations. We utilize the fact that these representations can also be given in data or image space, we call them ”knowledge prototypes”. These prototypes already reflect major concepts of a target domain, which suggests that pre-training on them can lead to significant improvements.

We investigate our approach on three different image classification datasets, namely GTSRB [9], MNIST [10] and USPS [11]. Our approach is highly relevant for computer vision tasks, but can also be applied to versatile other AI tasks. For example, additional experiments with the NOAA CO2 dataset [12] also show the applicability to regression problems.

Our study led to several new and remarkable findings. The main contributions can be summarized as follows:



(a) Training Loss

(b) Validation Accuracy

(c) Val. Accuracy (Small Data)

Fig. 2: Informed pre-training on knowledge prototypes leads to a learning speed-up and improves generalization, especially for small training data. This figure shows the learning curves for our method (pre-trained: know. prototypes / know. proto. (aug.)) in comparison to default initialization and other pre-training methods. As shown in (a), the training loss after pre-training on knowledge prototypes does not encounter local minima or saddle points but instead shows a continuous descent. This suggests that informed pre-training can initialize the learning algorithm in favourable regions of the loss landscape. Moreover, for small training data the accuracy converges to a higher and better value (c). This effect can be further intensified when augmenting the knowledge prototypes.

1) We present the novel approach of informed pre-training using prototypes from prior knowledge. In contrast to existing pre-training methods, our method utilizes concise and controllable prior knowledge that is given by a small set of semantic prototypes. In contrast to existing informed learning methods, our method can be universally applied. To the best of our knowledge our approach has not been considered or studied before and constitutes a novel avenue for both pre-training and informed machine learning.
2) Our results show an improvement in test accuracy for small training data by up to 11%. Furthermore, we obtain an increase of 15% on out-of-distribution robustness. Our approach also leads to faster training convergence. We compare our approach to several baselines and find that our approach yields the best results.
3) To provide an in-depth analysis we investigate the transfer learning contribution of individual model layers. For traditional data-based pre-training, benefits arise from transferring early layers. In contrast, for our informed knowledge-based pre-training, improvements stem from deeper layers which tend to represent semantic concepts. This is a before unobserved effect, which shows that pre-training on semantic features is viable and significantly different to existing approaches. We refer to this effect as “informed transfer learning”.
4) We compare our approach to ImageNet pre-training and find that the latter can be further improved by a subsequent informed pre-training on knowledge prototypes. We find an additional increase of 13% in test accuracy. This further confirms the complementary advantages of data-based and knowledge-based pre-training.
5) Finally, we compare our method to other informed learning approaches, which shows that a concurrent training on data and prototypes is beneficial, too. However, pre-training still prevails as it initializes parameters of learning algorithms in a favourable region of the loss landscape while still providing the flexibility of fine-tuning to natural data features afterwards.

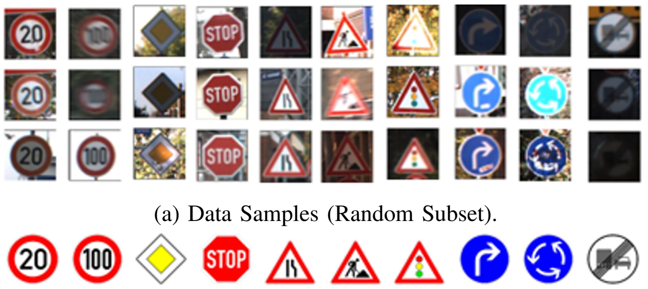
Please note, that one advantage of our method is that it can be applied to various knowledge formalization types and domains. It is neither restricted to the types that we use in this paper (i.e. image templates, graph structures, and scientific equations), nor domain specific. Instead informed pre-training can be used for any kind of prior knowledge that can be represented in a data space, e.g., by rendering or simulation. This makes our approach especially beneficial for all domains where real data acquisition is hard.

II. Related Work

Our paper is located at the intersection of informed machine learning, pre-training and transfer learning, and also relates to the use of prototypes in artificial intelligence.

a) Informed Machine Learning: Informed learning belongs to the field of hybrid AI and describes the idea to improve data-based learning systems through the integration of additional prior knowledge [8]. The utilization of prior information for regularization has already been discussed many years ago in the context of statistical learning theory [13], [14]. Recently, knowledge injection into neural networks became a popular approach to alleviate the problem of small training data or to ensure knowledge conformity. Given knowledge representations can be integrated into the machine learning pipeline using different approaches [8]. For example, logic rules about object properties or physical equations describing object dynamics can be integrated into the loss function as constraints [15], [16], or they can be incorporated into the model architecture as inductive biases [17]–[19]. Informed machine learning is also closely related to causally-aware machine learning [20], [21], as well as to neuro-symbolic AI [22].

b) Pre-Training and Transfer Learning: The aim of transfer learning is to improve a target task by reusing knowledge from another source domain or task [23]. The weights of a neural network are usually initialized randomly [24], [25] but



(a) Data Samples (Random Subset).

(b) Knowledge Prototypes.

Fig. 3: GTSRB data and prototypes from prior knowledge. A subset of 10 from the total 43 classes are shown.

can also be initialized by reusing the parameters from a pre-trained model. It can then be fine-tuned on given training data for the target task. In the last years, supervised pre-training on the ImageNet dataset [26] became a common transfer learning approach, especially for computer vision tasks [27]–[29]. [30] put the advantages of pre-training with ImageNet into question and claimed that it does not necessarily result in better test performance but only in a speed-up of the learning process. Nevertheless, [31] showed that pre-training can improve robustness. Several works also studied layer transferability [27], [29], [32] and found that the largest benefits in performance stem from pre-training the early layers. These often represent general features and low-level statistics of the data.

c) Prototypes in AI: In the context of machine learning prototypes are representatives of a data distribution. Such prototypes can be available from prior knowledge. This is natural in computer vision, where prototypical images or structural prototypes are traditionally used as templates for object classification. For example, [33] used deformable prototypes for handwritten digit recognition, or [34] used spatial prototypes for traffic sign recognition. Prototypical images are also used for creating synthetic training data [35], or can be employed with autoencoders for one-shot learning [36]. When prototypes are not yet available, they can be learned from the dataset [37]–[39]. It is also possible to identify such prototypes in the latent space and use them implicitly within so called prototypical networks [40]. Moreover, there are also parallels between our work and curriculum or continual learning [41], [42]. The main difference is that curriculum learning usually uses a subset of training examples, not knowledge prototypes.

III. Approach

We present the novel approach of informed pre-training of neural networks using prototypes stemming from prior knowledge. We first give a brief formalization that motivates the approach. We then describe the utilized knowledge prototypes and the practical approach.

A. Formalization

We consider a supervised learning scenario. Here, we especially regard the task of image classification. Let a data sample $\mathcal{D} = \{(x_i, y_i)\}_{i=1\ldots n}$, with images $x \in \mathcal{X}$ and labels $y \in \mathcal{Y}$, be given. The learning task is to find a model $f : \mathcal{X} \to \mathcal{Y}$ with $f \in \mathcal{F}$.

We assume that additional prior knowledge is given in the form of prototypes $x_p \in \mathcal{X_P}$ that represent the underlying concepts of the actual data. In short we call these ”knowledge prototypes”. Each of the prototypes is assigned to one class so that we have a sample $\mathcal{P} = \{(x_{p,j}, y_j)\}_{j=1\ldots m}$.

Intuitively, each data element x is constructed from a prototype x_p via

$$x = t(x_p)$$

with some, possibly non-linear, transformation $t : \mathcal{X}_p \to \mathcal{X}$ for $t \in \mathcal{T}$.

Using this, it is straightforward to show that a model that is pre-trained on knowledge prototypes is a good initialization for the main learning task. In particular, the framework of minimizing the empirical risk $R(f)$ with a given loss function l can be used to formulate the pre-training on the knowledge prototypes $\mathcal{P}$ as follows:

$$f^* := \arg\min_{f \in \mathcal{F}} R_{\mathcal{P}}(f), \quad R_{\mathcal{P}}(f) = \frac{1}{|\mathcal{P}|} \sum_{(x,y)\in\mathcal{P}} l(f(x), y)$$

The main learning task on the real data $\mathcal{D}$ is:

$$\hat{f} := \arg\min_{f \in \mathcal{F}} R_{\mathcal{D}}(f), \quad R_{\mathcal{D}}(f) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} l(f(x), y)$$

Clearly, $|\mathbb{E}\left[R_{\mathcal{D}}(f) - R_{\mathcal{P}}(f)\right]|$ should be as small as possible. To show this, we have derived a prototype initialization bound. Intuitively, it says that any model that is pre-trained on the prototypes will be a good initialization, whenever the prototypes are likely to be generated by the data distribution. The result can also be refined to address the distance between prototypes and data points. However, the final insight is the same: prototypes shall represent concepts which appear in the actual data.

B. Knowledge Prototypes

The knowledge prototypes are semantic representatives for the structure of the underlying distribution of the data. Such prototypes are available for a wide range of applications. They can be based on different types of formal knowledge representations, e.g., on image templates, but also on more sophisticated forms such as structural graphs, or physical equations.

In this paper, we consider the traffic sign recognition with the GTSRB dataset, where templates of traffic sign symbols are publicly available as the knowledge prototypes. In fact, we simply recycled the image templates from the official GTSRB result analysis application [9]. A subset of the knowledge prototypes are shown in 3b. Moreover, we consider handwritten digit recognition with MNIST and USPS, where we employ the deformable graph prototypes from [33]. For this, we simply made a screenshot and increased the line width so that edges and nodes are smooth and transform them into images. An excerpt of resulting knowledge prototypes is shown in Figure 1.

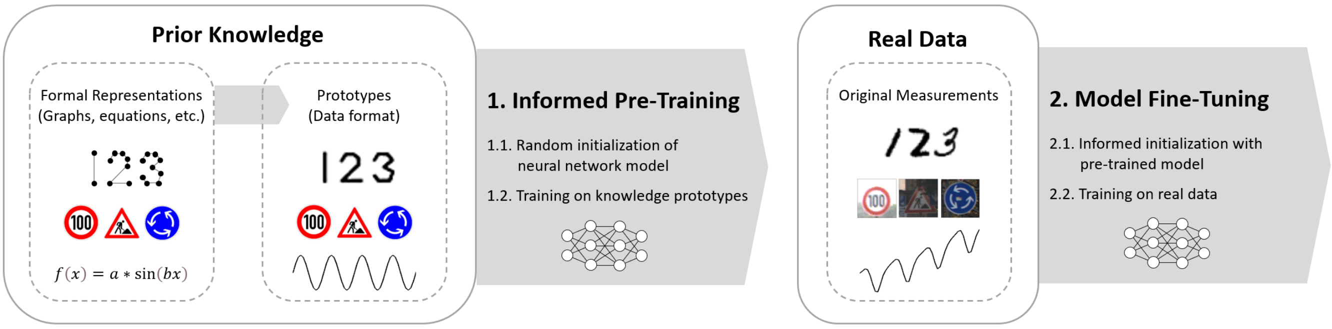

Prior Knowledge
Formal Representations (Graphs, equations, etc.)
$f(x) = a * \sin(bx)$
Prototypes (Data format)
123
1. Informed Pre-Training
1.1. Random initialization of neural network model
1.2. Training on knowledge prototypes
Real Data
Original Measurements
123
2. Model Fine-Tuning
2.1. Informed initialization with pre-trained model
2.2. Training on real data

Fig. 4: Approach overview. We propose informed pre-training on prototypes from prior knowledge in order to improve neural network training, especially when real data is scarce. Prior knowledge is often given by formal representations, e.g. by graph structures, image templates, or scientific equations. They can also be represented in data space - we then call them ”knowledge prototypes”. This allows for informed pre-training, i.e. to train a model on prior knowledge. Afterwards, the pre-trained model is fine-tuned on real training data. The informed pre-training leads to significantly increased generalization capabilities and improved robustness.

C. Informed Pre-Training on Knowledge Prototypes

From a practical point of view our approach consists of the following two main phases, as illustrated in Figure 4:

1) Informed pre-training
   a) Initialization of neural network model
   b) Training on knowledge prototypes $\mathcal{P}$
2) Model fine-tuning
   a) Informed initialization with pre-trained model
   b) Training on real data $\mathcal{D}$

In the first phase, the model is trained only on the knowledge prototypes. Taking into account that the number of model parameters is much bigger then the number of prototype samples, we consciously let the model memorize the prototypes and run the learning algorithm until the training loss approaches zero. In this situation this is favourable because the prototypes contain no noise and we want to fully exploit the prior knowledge. We further support this design choice with the recent findings about the double-descent risk curve, which describes the effect that in an over-parameterized regime a decreasing test error can be observed [43], [44].

In the second phase, the model is initialized with the pre-trained model and then fine-tuned to the real training data. For both phases, we use the same model architecture and the same hyperparameters.

a) Augmentation of Knowledge Prototypes: We found that geometric augmentation of knowledge prototypes can improve the benefits from informed pre-training. This can be seen as a variation of the method. The first variation is pre-training on only 1 prototype per class, the second variation is pre-training on more than 1 prototype per class. In particular, we experimented with up until 100 prototypes per class. For this, GTSRB prototypes were augmented with plausible random affine transformations: Rescaling, small translations, rotation up to 5 degree. A random subset of 10 out of the maximum 100 augmented knowledge prototypes are shown in Figure 5. MNIST prototypes were potentially augmented with random 2D perspective transformations (e.g. rescaling, rotation, shearing).

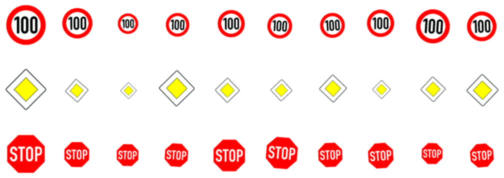


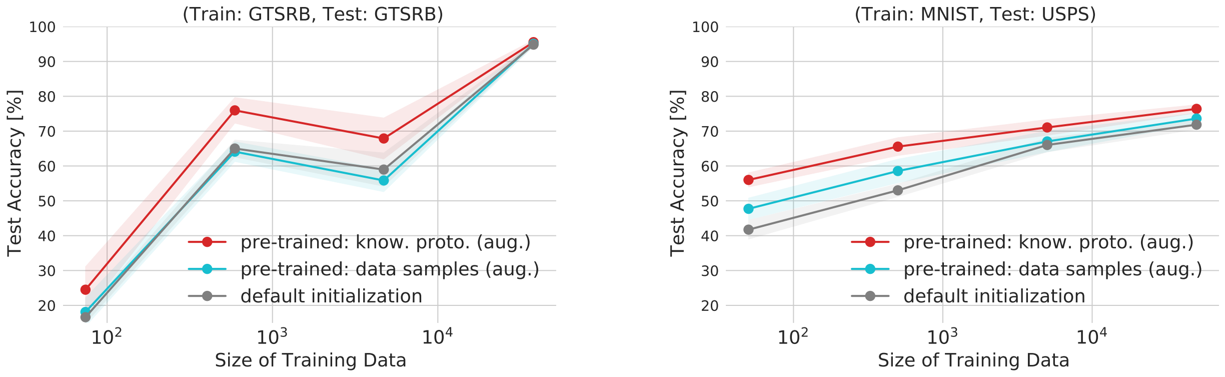
Fig. 5: Augmentation Examples of Knowledge Prototypes (GTSRB).

IV. Experimental Setup

We demonstrate our approach with experiments on the task of image classification. Here, we shortly report the most important information on the experimental setup.

a) Datasets: We employ three different image datasets, which are the German Traffic Sign Recognition Benchmark (GTSRB) [9], the widespread handwritten digit database from MNIST [10], and the related but different dataset USPS [11]. We employ the usual splits into train / val / test subsets. For GTSRB these amount to 37,890 / 1,290 / 12,630 and for MNIST to 50,000 / 10,000 / 10,000 images. The USPS dataset is used for testing for out-of-distribution robustness. It consists of handwritten digit images as well, but they stem from a different underlying data distribution then MNIST. Potential domain shifts between training and target data are a challenge in machine learning and it is desirable that a trained model is robust to out-of-distribution data. The test set contains 2,007 images.

We investigate learning with small training data sets. Therefore, we run our experiments for four different training subsets: 100%, 10%, 1%, and 0.1%. For MNIST this means: 50,000 / 5,000 / 500 / 50 images. For GTSRB we create the subsets through dividing by 8: 37,890 / 4710 / 592 / 74 images. The exact numbers result from maintaining the organization into image tracks (30 images each) for the larger two subsets, and permitting the use of individual images for the smaller two data subsets.



(a) GTSRB (In-Distribution Generalization)

(b) USPS (Out-of-Distribution Robustness)

Fig. 6: Test accuracies for different training data sizes (0.1%, 1%, 10%, and 100%) for **informed pre-training on knowledge prototypes (our approach)**, pre-training on data samples, and default initialization. a) In-distribution generalization is improved by informed pre-training especially for smaller training data ($\leq$ 10%). The zick-zack is due to the data set organization in image tracks. b) Out-of-distribution robustness is improved by informed pre-training for all sizes.

b) Models: We show the benefits of our method on three benchmark datasets using well-established standard architectures and hyperparameters. For GTSRB, we used the AlexNet [45] and for MNIST the LeNet-5 convolutional neural network architecture [46]. We have also tested our method on several additional architectures to confirm that the observed effects generalize to other setups.

c) Training Parameters: We repeat every experiment run 10 times, and for every run we redo both the pre-training and the fine-tuning. The repetitions differ in the random initialization of the pre-training phase. We further shuffle the split into train and val subset for the fine-tuning phase. We run the fine-tuning for a fixed budget of 10 training epochs. The 10 epochs are enough to reach a sufficient convergence, but we have also run extended experiments for 100 epochs, which show that the improvements from informed pre-training persist.

d) Baselines: We compare our method of informed pre-training on knowledge prototypes to several baselines: 1) default initialization, 2) pre-training on data samples, 3) pre-training on mean prototypes. Finally, for GTSRB we also compare our method to another baseline 4) pre-training on ImageNet. For baseline 2, the same amount of data samples and training iterations as for our method is used. It shows that the benefits of informed pre-training are not due to the advances from the learning algorithm, but really come from the knowledge prototypes themselves. Baseline 3 provides a comparison to pre-training on alternative prototypes, which we compute as the class-wise mean in pixel space. Baseline 4 show interesting differences between our knowledge-based, informed pre-training and the conventional, data-based pre-training. For our method and all baselines the same number of epochs and other hyperparameters are used.

V. Results

With our experiments we now show that informed pre-training on knowledge prototypes leads to improved generalization for small training data, improved model robustness with respect to out-of-distribution data, and also to a learning speed-up. Moreover, we observe that these benefits result from transfer learning of late layers. Finally, we compare informed pre-training to other informed learning strategies. We present the results on each of these findings.

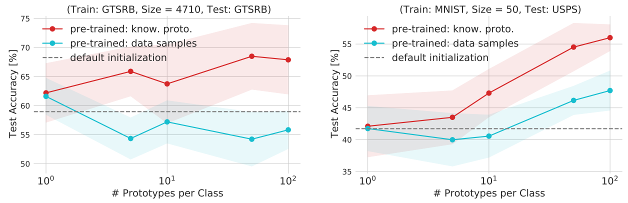
A. Improved Generalization and Robustness

One motivation for the integration of prior knowledge into machine learning approaches is to alleviate the problem of little training data. In fact, our experimental results show that informed pre-training improves in-distribution generalization for small training data, as well as out-of-distribution robustness for all data set sizes (See Figure 6 and Table I). In particular, for GTSRB and small training data the improvement in test accuracy with our approach (pre-trained: know. proto. (aug.)) is 8-11% compared to default initialization (Figure 6a). However, for applications out-of-distribution robustness is also very important. To investigate this, we have trained a model on MNIST and tested on USPS. Here, we observe a significant improvement by nearly 15% for small data, and even a remarkable improvement by about 5% for large training data (Figure 6b). This shows that our proposed approach can substantially improve out-of-distribution generalization.

We have also investigated augmentation of knowledge prototypes. Pre-training on only one prototype per class (know. prototypes) already leads to improvements, but these can still be significantly increased through geometric augmentations (know. proto. (aug.)). Whereas in the baseline augmentation of data samples can even lead to deterioration, the augmentation of our knowledge prototypes leads to improvements. Results are shown in Figure 7. .

Train Data	Test Data	Pre-Training	**Test Accuracies [%], for different Train Data Sizes** **≈ 0.1%**	 **≈ 1%**	 **≈ 10%**	 **100%**
GTSRB	GTSRB	(default init.)	16.61 ± 1.74	65.01 ± 2.46	58.97 ± 4.79	94.82 ± 0.48
		data samples	18.18 ± 2.96	65.38 ± 2.89	61.60 ± 3.13	94.89 ± 0.57
		data samples (aug.)	17.69 ± 4.15	64.16 ± 3.01	55.39 ± 3.07	95.07 ± 0.72
		mean prototypes	17.45 ± 3.62	65.69 ± 4.57	58.47 ± 6.54	94.82 ± 0.79
		know. prototypes - (ours)	**19.87** ± 3.21	**69.81** ± 3.89	**62.20** ± 5.07	**95.07** ± 0.73
		know. proto. (aug.) - (ours)	**24.50** ± 6.61	**75.96** ± 3.69	**67.89** ± 5.91	**95.53** ± 0.62
MNIST	MNIST	(default init.)	73.00 ± 3.04	89.55 ± 1.12	97.25 ± 0.16	98.53 ± 0.11
		data samples	73.38 ± 3.11	89.93 ± 0.62	97.16 ± 0.15	98.46 ± 0.18
		data samples (aug.)	75.43 ± 4.09	90.96 ± 0.77	97.08 ± 0.26	98.54 ± 0.14
		mean prototypes	69.81 ± 3.47	89.53 ± 1.19	97.08 ± 0.26	98.42 ± 0.22
		know. prototypes - (ours)	**72.51** ± 2.70	**89.90** ± 1.12	**97.26** ± 0.15	**98.46** ± 0.18
		know. proto. (aug.) - (ours)	**77.57** ± 4.26	**91.77** ± 0.81	**97.16** ± 0.23	**98.69** ± 0.13
MNIST	USPS	(default init.)	41.73 ± 2.73	53.01 ± 1.83	66.03 ± 1.99	71.85 ± 1.38
		data samples	41.75 ± 3.50	52.45 ± 2.83	64.98 ± 1.58	72.54 ± 1.83
		data samples (aug.)	47.71 ± 3.09	58.56 ± 3.40	67.04 ± 2.99	73.61 ± 1.56
		mean prototypes	39.70 ± 3.24	52.10 ± 2.54	64.76 ± 2.66	71.78 ± 1.55
		know. prototypes - (ours)	**42.11** ± 4.82	**55.06** ± 2.31	**66.93** ± 1.81	**72.68** ± 1.33
		know. proto. (aug.) - (ours)	**56.00** ± 2.05	**65.56** ± 2.58	**71.06** ± 2.27	**76.39** ± 1.18

TABLE I: Test accuracies in % (higher is better) for different data sets and training data sizes. We report mean values and standard deviations from 10 repetitions. The bold rows highlight the results based on **informed pre-training on knowledge prototypes (our approach)**. Underlined results highlight the best results for each test case and training data size. We report results for pre-training on only 1 prototype per class (know. prototypes), as well as for augmentation with 100 prototypes per class (know. proto. (aug.)).



(a) GTSRB

(b) USPS

Fig. 7: Test accuracy after pre-training with respect to augmentation strength, i.e., number of prototypes (From 1 to 100 per class).

B. *Informed Transfer Learning of Semantic Layers*

The goal of this experiment is to better understand where the improvements of informed pre-training are coming from. For conventional, data-based pre-training, e.g., on ImageNet, the improvement results from the transfer of early neural network layers, representing low-level, statistical features [27]. This raises the question, which network layers are responsible for performance gains after pre-training on knowledge prototypes.

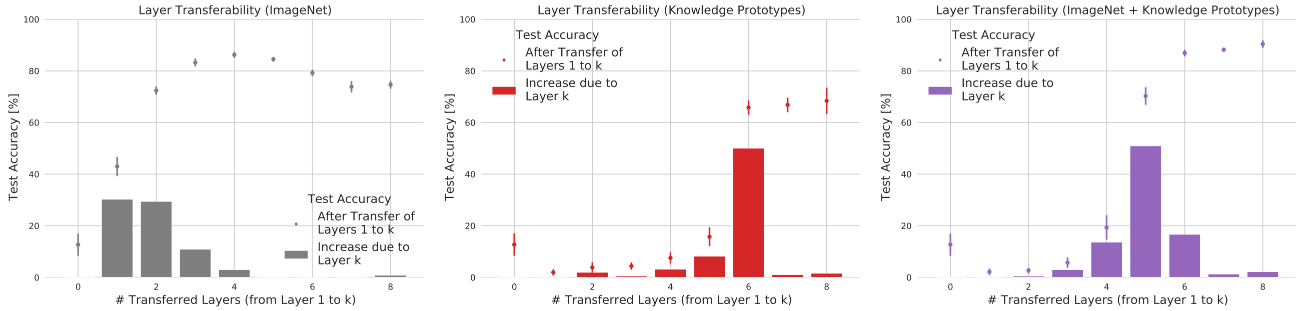
We therefore designed an experiment to analyze the importance of individual network layers for transferring learning performance. As before, we pre-train a neural network on knowledge prototypes, but now we reuse only the first k layers of the pre-trained model, i.e., the pre-trained model is truncated after layer k. The remaining layers are re-initialized with the default random procedure. We use the AlexNet architecture, which has a total of eight layers resulting in nine partitioning cases ($k = 0$ corresponds to a random initialization of all layers and $k = 8$ to deploying the full pre-trained model). For each k, we perform a separate fine-tuning and then evaluate the test accuracy. Figure 8 shows the results for the small GTSRB data subset (1%) and training the AlexNet architecture, which has 8 layers. As the effect of pre-training is especially apparent in the early training phase, we deliberately evaluate the test accuracy after the first epoch for the full training set, which correspond to 64 Epochs for the small data set.

Figure 8a shows that for pre-training on ImageNet the performance gain results, as expected, from transferring of the early network layers. These account for general low-level image features. In contrast, Figure 8b shows that for pre-training on knowledge prototypes the performance gain results from transferring the deep layers. These typically represent high-level semantic features. This is a remarkable and before unseen effect (we call this ”informed transfer learning”). Our results suggest that the two pre-training types have complementary strengths.

We furthermore investigate a third pre-training method that consists of two pre-training phases: initializing the first five layers with a regular ImageNet-pretraining and subsequently pre-training on knowledge prototypes (see Figure 8c.) As it depicts the contribution of the layers from knowledge-based pre-training, we again measure the largest gains truncating the late layers, albeit with a slight shift to the middle. Even more remarkable, such a subsequent knowledge-based pre-training can further improve pre-training on ImageNet as reported in Table II.

C. *Learning Speed-Up*

Figure 2 depicts the learning curves for our experiments on the GTSRB dataset. They show that pre-training on knowledge prototypes leads to a learning speed-up, which indicates that



(a) Pre-Training: ImageNet
(Conventional Transfer Learning)

(b) Pre-Training: Knowledge Prototypes
(Informed Transfer Learning)

(c) Pre-Training: ImageNet + Know. Proto.
(Convent. + Informed Transfer Learning)

Fig. 8: Importance of individual network layers for different pre-training types. (a) Pre-Training on the ImageNet dataset. Improvements come from early layers, which typically represent low-level features. (b) Pre-Training on knowledge prototypes (augmented). Improvements come from late layers, which typically represent high-level, semantic concepts. (c) Combining both types by using a model pre-trained on ImageNet for the initialization of a subsequent pre-training on knowledge prototypes. After the respective pre-training schemes only the first k layers are transferred. The bar charts highlight the performance gain after fine-tuning due to the transfer of the layers preceding layer k. The experiment shows that knowledge-based, informed transfer learning has an additional and complementary effect to conventional, data-based transfer learning.

Test Data	**Pre-Training**	**Test Accuracies [%], for different Train Data Sizes** $\approx$ **0.1%**	 $\approx$ **1%**	 $\approx$ **10%**	 **100%**
GTSRB	(default init.)	16.61 ± 1.74	65.01 ± 2.46	58.97 ± 4.79	94.82 ± 0.48
	know. proto. (aug.) - (ours)	**24.50** ± 6.61	**75.96** ± 3.69	**67.89** ± 5.91	**95.53** ± 0.62
	ImageNet	28.50 ± 1.12	76.31 ± 1.71	62.25 ± 3.75	96.92 ± 0.34
	ImageNet + know. proto. (aug.) - (ours)	<u>**41.72**</u> ± 2.72	<u>**85.48**</u> ± 1.85	<u>**74.37**</u> ± 3.44	<u>**97.27**</u> ± 0.21

TABLE II: Test accuracies for 1) informed transfer learning (pre-training: know. proto. (aug.)), 2) conventional transfer learning (pre-training: ImageNet), and 3) conventional + informed transfer learning (pre-training: first ImageNet, then know. proto. (aug.)). The bold rows highlight results based on **informed pre-training (our approach)**. Underlined results highlight the best results.

the model is initialized in a more favourable region of the loss landscape. We conjecture that pre-training on knowledge prototypes reflects the overall structure of the main learning task. This interpretation can be likened to continuation methods, where a non-convex optimization problem is transformed into a convex optimization problem and then gradually converted back while following the path of the minimizer [41], [42].

D. Concurrent Informed Learning

Finally, we have also tested other forms of informed learning. We injected the knowledge prototypes in the training data itself and trained models concurrently on both. We also combined informed pre-training and that concurrent learning. The results show that informed transfer learning is not only possible via pre-training, but also via concurrent learning. However, for small training data and especially the in-distribution case the pure pre-training on knowledge prototypes leads to the better results. An advantage of pre-training is the following: Knowledge can also change over time. If we first pre-train on knowledge and then refine to real data observations, the inductive bias is less strong and still allows the flexibility to adapt. This again makes our proposed method more robust.

VI. Conclusion

We proposed informed pre-training on knowledge prototypes and found that it (i) improves generalization capabilities (especially for small data), (ii) increases out-of-distribution robustness, and (iii) speeds up the learning process. Moreover, we showed that (iv) the improvements come from transferring the deeper network layers that typically represent high-level, semantic features (”informed transfer learning”). This is in contrast to traditional data-based pre-training and shows that both have complementary strengths. This is an interesting insight, which also seems to agree with cognitive sciences [2], [47]. A huge advantage of our method is that it is domain agnostic and can be applied to various knowledge representations types and domains. However, a potential challenge is the transformation of formal knowledge into prototypes. Therefore, as future research we see the application to more sophisticated knowledge types. For scientific and visual domains technique like rendering and simulation offer a helpful resource. For example, one could use graph structures or high-dimensional geometrical models that are used for synthetic data generation (such as by [48]) and employ them as knowledge prototypes for informed pre-training.

Acknowledgment

This research has been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence.

References

[1] P. Smolensky, "Connectionist ai, symbolic ai, and the brain," *Artificial Intelligence Review*, vol. 1, no. 2, 1987.

[2] M. L. Minsky, "Logical versus analogical or symbolic versus connectionist or neat versus scruffy," *AI Magazine*, vol. 12, no. 2, 1991.

[3] L. Von Rueden, S. Mayer, R. Sifa, C. Bauckhage, and J. Garcke, "Combining machine learning and simulation to a hybrid modelling approach: Current and future directions," in *International Symposium on Intelligent Data Analysis*. Springer, 2020.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[5] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Computing Surveys*, vol. 53, no. 3, 2020.

[6] R. Volpi, H. Namkoong, O. Sener, J. Duchi, V. Murino, and S. Savarese, "Generalizing to unseen domains via adversarial data augmentation," *NeurIPS*, 2018.

[7] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *ICLR*, 2019. [Online]. Available: https://openreview.net/forum?id=HJz6tiCqYm

[8] L. Von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, and J. Schuecker, "Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems," *IEEE TKDE*, 2021.

[9] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German Traffic Sign Recognition Benchmark: A multi-class classification competition," in *IJCNN*, 2011.

[10] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]*, vol. 2, 2010.

[11] J. J. Hull, "A database for handwritten text recognition research," *IEEE PAMI*, vol. 16, no. 5, 1994.

[12] E. Dlugokencky and P. Tans, "Trends in atmospheric carbon dioxide, global monthly mean co2, gml.noaa.gov/ccgg/trends/," 2021. [Online]. Available: gml.noaa.gov/ccgg/trends/

[13] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 1995.

[14] P. Niyogi, F. Girosi, and T. Poggio, "Incorporating prior information in machine learning by creating virtual examples," *Proc. of the IEEE*, vol. 86, no. 11, 1998.

[15] M. Diligenti, M. Gori, and C. Sacca, "Semantic-Based Regularization for Learning and Inference," *Artificial Intelligence*, vol. 244, 2017.

[16] R. Stewart and S. Ermon, "Label-free supervision of neural networks with physics and domain knowledge," in *AAAI*, vol. 31, no. 1, 2017.

[17] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, "Relational Inductive Biases, Deep Learning, and Graph Networks," *arXiv:1806.01261*, 2018.

[18] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, 2021.

[19] N. M. Gürel, X. Qi, L. Rimanic, C. Zhang, and B. Li, "Knowledge enhanced machine learning pipeline against diverse adversarial attacks," in *ICML*, 2021.

[20] T. Kyono and M. van der Schaar, "Improving model robustness using causal knowledge," *arXiv:1911.12441*, 2019.

[21] T. Kyono, "Towards causally-aware machine learning," *PhD Thesis, University of California*, 2021.

[22] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision," in *ICLR*, 2018.

[23] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, 2009.

[24] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *AISTATS*, 2010.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. of the IEEE ICCV*, 2015.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[27] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *NeurIPS*, 2014.

[28] M. Huh, P. Agrawal, and A. A. Efros, "What makes imagenet good for transfer learning?" *arXiv:1608.08614*, 2016.

[29] B. Neyshabur, H. Sedghi, and C. Zhang, "What is being transferred in transfer learning?" *arXiv:2008.11687*, 2020.

[30] K. He, R. Girshick, and P. Dollár, "Rethinking imagenet pre-training," in *ICCV*, 2019.

[31] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *ICML*, 2019.

[32] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training," in *AISTATS*, 2009.

[33] T. Hastie and R. Tibshirani, "Handwritten digit recognition via deformable prototypes," in *Statistics and Data Analysis Research Department, AT&T Bell Laboratories*, 1994.

[34] F. Larsson and M. Felsberg, "Using fourier descriptors and spatial models for traffic sign recognition," in *Scandinavian Conference on Image Analysis*, 2011.

[35] D. Spata, D. Horn, and S. Houben, "Generation of natural traffic sign images using domain translation with cycle-consistent generative adversarial networks," in *Intelligent Vehicles Symposium*, 2019.

[36] J. Kim, T.-H. Oh, S. Lee, F. Pan, and I. S. Kweon, "Variational prototyping-encoder: One-shot learning with prototypical images," in *CVPR*, 2019.

[37] B. Kim, O. Koyejo, and R. Khanna, "Examples are not enough, learn to criticize! criticism for interpretability," *NeurIPS*, 2016.

[38] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," in *AAAI*, vol. 32, no. 1, 2018.

[39] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, "This looks like that: Deep learning for interpretable image recognition," *NeurIPS*, 2019.

[40] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *NeurIPS*, 2017.

[41] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum Learning," in *ICML*, 2009.

[42] H. Mobahi and J. Fisher III, "A theoretical analysis of optimization by gaussian continuation," in *AAAI*, 2015.

[43] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling Modern Machine-Learning Practice and the Classical Bias–Variance Trade-Off," *Proc. of the National Academy of Sciences*, vol. 116, no. 32, 2019.

[44] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: Where bigger models and more data hurt," in *ICLR*. OpenReview.net, 2020.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NeurIPS*, 2012.

[46] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, 1998.

[47] S. Chipman and A. L. Meyrowitz, *Foundations of knowledge acquisition: Cognitive models of complex learning*. Springer Science & Business Media, 2012, vol. 194.

[48] M. Schwarz and S. Behnke, "Stillleben: Realistic scene synthesis for deep learning in robotics," in *ICRA*. IEEE, 2020.

APPENDIX D

Paper P4) Street-Map Based Validation of Semantic Segmentation in Autonomous Driving

Street-Map Based Validation of Semantic Segmentation in Autonomous Driving

(Preprint - Final version will be published at IEEE)

Laura von Rueden*‡, Tim Wirtz*, Fabian Hueger†, Jan David Schneider†, Nico Piatkowski*, Christian Bauckhage*
* Fraunhofer Center for Machine Learning, Fraunhofer IAIS, Sankt Augustin, Germany
† Volkswagen Group Automation, Wolfsburg, Germany
‡ laura.von.rueden@iais.fraunhofer.de

***Abstract*—Artificial intelligence for autonomous driving must meet strict requirements on safety and robustness, which motivates the thorough validation of learned models. However, current validation approaches mostly require ground truth data and are thus both cost-intensive and limited in their applicability. We propose to overcome these limitations by a model agnostic validation using a-priori knowledge from street maps. In particular, we show how to validate semantic segmentation masks and demonstrate the potential of our approach using OpenStreetMap. We introduce validation metrics that indicate false positive or negative road segments. Besides the validation approach, we present a method to correct the vehicle's GPS position so that a more accurate localization can be used for the street-map based validation. Lastly, we present quantitative results on the Cityscapes dataset indicating that our validation approach can indeed uncover errors in semantic segmentation masks.**

I. Introduction

Environmental perception is important for autonomous vehicles in order to assess the surrounding traffic scene and understand its context [1], [2]. A key component is semantic segmentation, which assigns pixel-wise pre-defined class labels to the input images from vehicle's cameras. Current algorithms use machine and deep learning techniques to build models that predict semantic segments and surpass classic computer vision techniques in terms of performance [3], [4].

The development of artificial intelligence systems brings certain challenges, especially when they are applied in safety-critical areas. Building deep neural networks that generalize well and are robust often comes with the need for large amounts of ground truth data, which is typically acquired in expensive manual labelling processes. To ensure the safety of AI-based systems, mechanisms that support a trustworthy development like interpretability, auditing and risk assessment are discussed with growing interest [5].

The validation of machine learning models is particularly important in the area of highly automated driving, for example the identification and mitigation of risks of potential functional insufficiencies in neural networks used for perception [6]. Since the perception component is responsible for the first assessment of the vehicle's surroundings, the detection and reduction of errors in this component can increase the reliability of the resulting environment model. Proposed approaches for mitigation are the detection of prediction uncertainties [7] and the estimation of an according error propagation [8].

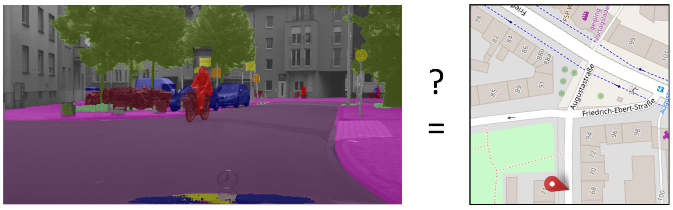


Fig. 1: **Research question.** Can predicted semantic segmentation masks be validated with a-priori knowledge from street maps? The left image shows a segmentation of a traffic scene in the Cityscapes dataset [9] and the right image shows the corresponding map [10]. Here, an intersection to the right, which is shown in the map, is not reflected in the segmentation.

Although state-of-the-art neural networks for semantic segmentation achieve promising results, it can still be observed that certain objects of the drivable area are not detected correctly. Moreover, smaller networks being used for embedded purposes are often comparatively less accurate than state-of-the-art networks with arbitrary size. As an example, roads and pedestrian walks could be mixed up in difficult lighting conditions or unusual terrain like in the segmentation in Figure 1.

We propose to support the goal of safe artificial intelligence in autonomous driving by applying the idea of informed machine learning [11] and validate learned models with a-priori knowledge. In this paper, we suggest to compare semantic segmentation masks to the structured semantic information in street maps, as illustrated in Figure 1, and present a novel method that computes the overlap of drivable area between the segmentation output and the map. Our approach is inspired by how human drivers would perceive environments: When they find themselves in a new environment, they often consult external knowledge sources such as street maps and compare what they see in their vicinity to what they see on the map.

Related work comprises approaches for multi-modal perception for autonomous driving, combining the inputs from various driving data [4]. The combination of camera inputs and street maps for semantic segmentation has already been used to assign geographical addresses to detect buildings [12], or to build conditional random fields for scene understanding [13],

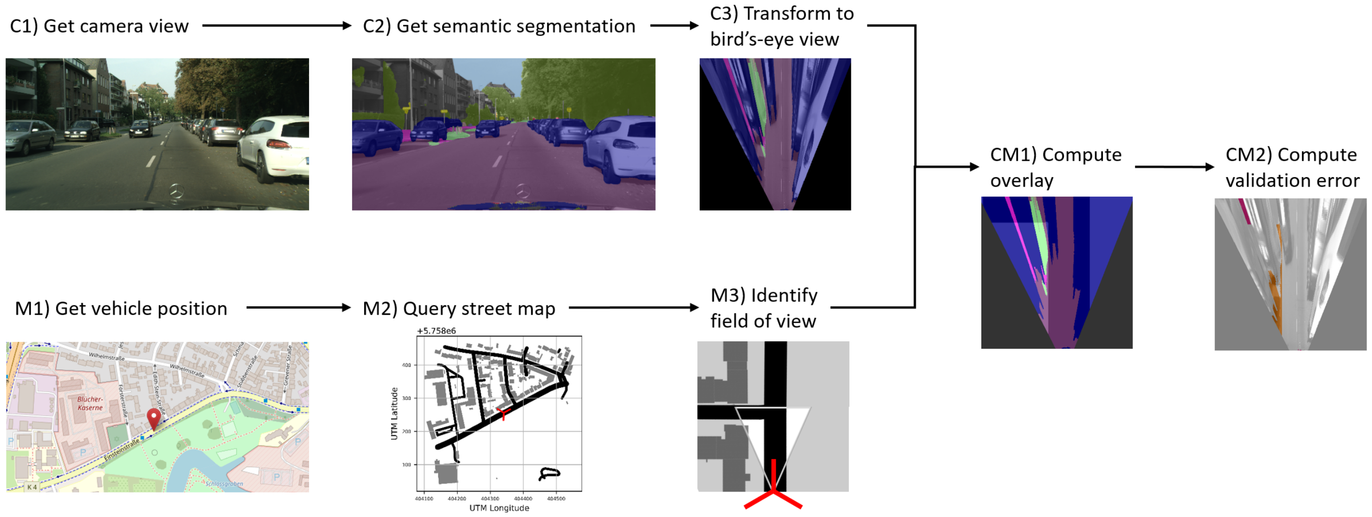


Fig. 2: **Approach overview**: We validate the drivable area in semantic segmentation masks with a-priori knowledge from street maps. For this, we combine the segmentation mask of a given camera view with the street map that corresponds the given vehicle position. We compute an overlay of the segmentation in bird's-eye view and the street map's field of view, omitting occluding, dynamic objects such as other vehicles or vegetation. The overlay is used to identify validation error regions, which we classify into detected false positives (indicated by a low IoS; visualized with orange red pixels) and false negatives (indicated by a low IoM; visualized with pink red pixels). The steps with C describe tasks involving a *c*amera view and the steps with M describe tasks involving a *m*ap view.

[14]. The integration of general geographic or geometric a-priori knowledge in perception tasks has been investigated in versatile forms, for example as shape priors for object localization [15], temporal priors for revisited locations [16], or spatial relation graphs for object detection [17]. However, to the best of our knowledge, there is not yet an approach that uses street maps for a validation of semantic segmentation masks.

An advantage of our proposed method is that it facilitates a model-agnostic validation of segmentation masks. It can be applied to predictions from deep learning approaches, but also to traffic scene segmentations from any other approach including the ground truth data generation process itself.

Furthermore, our validation method can be applied either offline in the testing phase of learned models, or even online to assess potentials errors within the current prediction, since the approach does not require ground truth to be available. Another advantage is that it can be used to test models even in geographical regions that had not been represented in the training data. This is relevant because the characteristics of semantic concepts such as roads, cars, or vegetation can be very diverse across different regions, but training datasets do not always reflect this domain variety [18], [19]. For autonomous driving, external data sources such as street maps thus provide a valuable alternative information source for static objects present in ground truth data.

Our paper presents four contributions. First, we introduce the approach to validate the drivable area in semantic segmentation masks using a-priori knowledge obtained from street maps, which can be used to identify prediction errors. Second, we define new validation metrics that can be used for comparing semantic segmentations of traffic scenes to street maps. Third, we present an algorithm for localization correction that can be used to calibrate the street map position according to the ground truth segmentation. Fourth, we present experimental results on the Cityscapes dataset, which demonstrate that our approach can identify similar prediction errors as a validation by ground truth data. The paper is structured accordingly.

II. Approach: Segmentation Validation

Our approach validates the drivable area in a given semantic segmentation mask using the corresponding geometric structures in a given street map. In this section we present how we combine the segmentation and the street map in an overlay. We define the validation metrics that we use for the identification of potential validation errors. Finally, we demonstrate our approach with two examples.

A. Overlay of segmentation and street map

An overview of our approach is illustrated in Figure 2. In the following we shortly describe each step within the approach.

Step C1) Get camera view: We retrieve an image from the vehicle's front view of a traffic scene. Here we use the Cityscapes [9] dataset.

Step C2) Get semantic segmentation: Using a neural network, we obtain a segmentation mask that maps each image pixel to a set of pre-defined class labels. In the example image in Figure 2, the labels are visualized in a chosen color coding: *road* is violet, *car* is blue, *pedestrian walk* is pink, etc. Here, we used a model trained by the ERFNet encoder-decoder architecture [20] to create the predictions.

Step C3) Transform to bird's-eye view: To prepare the validation of the drivable area segments using the street map, we transform the segmentation image into a bird's-eye view, which corresponds to the view space of the street map.

Step M1) Get vehicle position: We get the position by reading the GPS coordinates of the vehicle and thus retrieve latitude, longitude and the heading. These values are given in the Cityscapes dataset for each camera image. Since the accuracy of the GPS position can be a challenge, we develop a localization correction algorithm, which is further described in Section III, and apply it to alleviate inaccuracies in the position.

Step M2) Query street map graph: For the given latitude and longitude we get the street map graph for the surrounding area. For our analysis we use data from OpenStreetMap [10], because this source offers a freely available option with sufficient data coverage for an experimental demonstration of our approach. For future application in production other map providers that offer more detailed and accurate information, for example in high-definition maps, might be preferable.

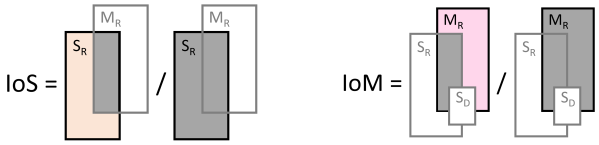
Step M3) Identify field of view: We transform the street map graph to an image and rotate it in the direction of the vehicle's heading, which is obtained from the metadata of the camera image. We zoom in so that it corresponds to the potential field of view from the camera mounted on the car.

Step CM1) Compute overlay: We combine the prepared images in an overlay of the semantic segmentation in bird's eye view with the road from the map image. As shown in Figure 2, the road is illustrated as a transparent black area. This allows us to recognize the overlap between the predicted road segments from the semantic segmentation mask and the street map.

Step CM2) Compute validation error: Finally we compute the regions where the predictions of the trained model deviate from the a-priori knowledge contained in the map. Two types of validation errors can be derived: False positive regions, i.e., where the segmentation shows a road, but the map does *not* (here visualized by orange red), and false negative regions, i.e., where the segmentation does *not* show a road, but the map does (visualized by pink red). For computing reliable error regions, we omit pixels that are assigned to labels that could be occluding the drivable area like, e.g., vegetation or cars.

B. Validation metrics

To quantify the overlap of semantic road segments with the street map, we introduce two new validation metrics that we call *Intersection over Segmentation* (IoS) and *Intersection over Map* (IoM). In semantic segmentation, the most commonly used evaluation metric for quantifying the model performance is the *Intersection over Union (IoU)*, which is defined by the area of overlap between predicted and ground truth segments divided by the area of union of both segments. However, in our approach we are especially interested in identifying either false positive or false negative road prediction errors. Our introduced metrics evaluate these errors for the road segments, so that they should precisely be called IoS_R and IoM_R. For simplicity we omit this subscript and just call them IoS and IoM in this paper.



(a) Intersection over Segment.

(b) Intersection over Map

Fig. 3: **Validation metrics.** The consistency of the semantic segmentation with the street map can be quantified by the two validation metrics that we introduced. a) The Intersection over Segmentation (IoS) quantifies the overlap of the segmentation with the map. A low IoS is an indicator for false positive road segments (see orange red region). b) The Intersection over Map (IoM) quantifies, vice versa, the overlap of the map with the segmentation. A low IoM is an indicator for false negative road segments (see pink red region).

The IoS metric quantifies the share of the semantic road segments that are covered by the roads in the map and thus helps to identify false positive errors. Here, we define false positives as semantic road segments, where the street map does not show a road segment. As illustrated in Figure 3a, we compute the IoS as the overlap area from the semantic road segment S_R and the road in the map M_R, divided by the area of the semantic road segment S_R itself. If the IoS is high, the segmentation and street map are mostly consistent, but the lower the IoS, the more false positive (FP) pixels are in the segmentation. Thus, the IoS can also be described as the share of true positive road segments (TP) of all road segments:

$$IoS = \frac{S_R \cap M_R}{S_R} = \frac{TP}{TP + FP}$$

Vice versa, the IoM metric quantifies the share of the road segments in the map that are covered by the road segments in the segments and thus helps to identify false negative errors. We define false negative errors as road segments that are not represented in the segmentation although they are present in the map. For its calculation, any dynamic semantic segments, such as vehicles or pedestrians, and vegetation that might occlude the road segment, are omitted. As illustrated in Figure 3b, we compute the IoM as the intersection of the semantic road segment S_R and the road in the map M_R, divided by the area of the road in the map M_R minus intersections with dynamic semantic segments S_D. The lower the IoM, the more false negative (FN) pixels are in the segmentation. The IoM can also be described as the share of true positive road segments from all road segments in the

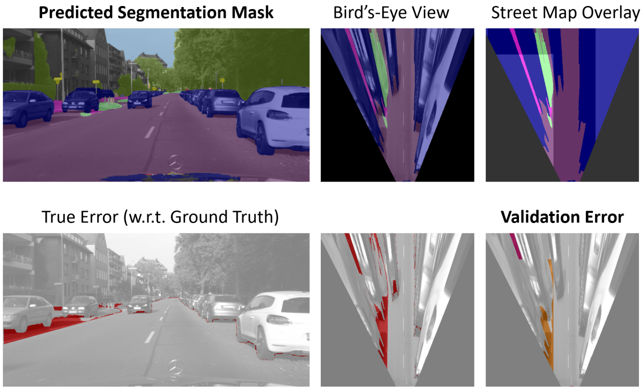


Fig. 4: **Example for the detection of a false positive road.** The predicted segmentation shows a road straight forward and below the cars parked at the left side of the street. According to the ground truth there is a parking space below that parking cars. Our map-based validation approach identifies this deviation: The street map suggests a less broad road than in the prediction, resulting in the detection of a false positive region (see orange red color at the left side of the validation error image). For this image the validation metrics are $IoS = 88.03\%$, and $IoM = 97.22\%$, also reflecting the false positive road prediction.

map:

$$IoM = \frac{S_R \cap M_R}{M_R - (M_R \cap S_D)} = \frac{TP}{TP + FN}$$

Another metric that combines the IoS and IoM is the dice coefficient. It can be used to quantify the general overlap between the road in the map and in the semantic segmentation. We use it in our localization correction method, as further described in Section III, and in our extended experiments as an initial estimation if a segmentation mask contains potential errors, as further explained in Section IV.

$$dice = \frac{2 \cdot (S_R \cap M_R)}{S_R + M_R - (M_R \cap S_D)} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

C. Example results

We demonstrate our approach with examples for two different traffic scenes. The first example, given in Figures 4, shows a traffic scene where the ground truth semantic segmentation shows a parking space at the left side, but the prediction shows a road. Our approach identifies this false positive road segment, also indicated by a lower IoS metric. The second example, given in Figure 5, shows a scene where the ground truth semantic segmentation shows a road intersection to the right, but the prediction does not. Again, our approach identifies this false negative, also supported by a lower IoM metric.

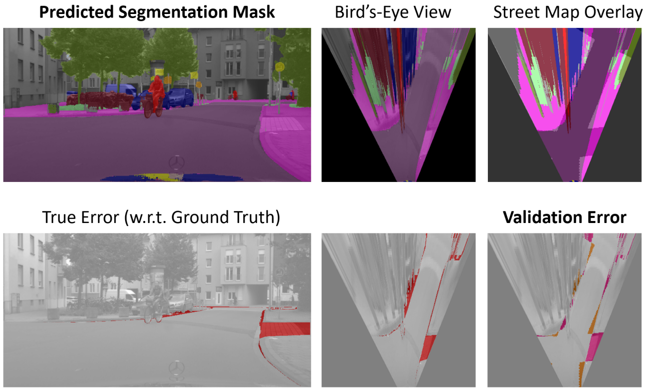


Fig. 5: **Example for the detection of a false negative road.** The predicted segmentation shows a road running straight forward, although there is an intersection to the right according to the ground truth. Our approach identifies this deviation, too: The street map shows an intersection to the right. This results in a detected false negative region (see pink red color at the right side of the error image). For this image the validation metrics are $IoS = 94.27\%$, and $IoM = 90.33\%$, also reflecting the false negative road prediction.

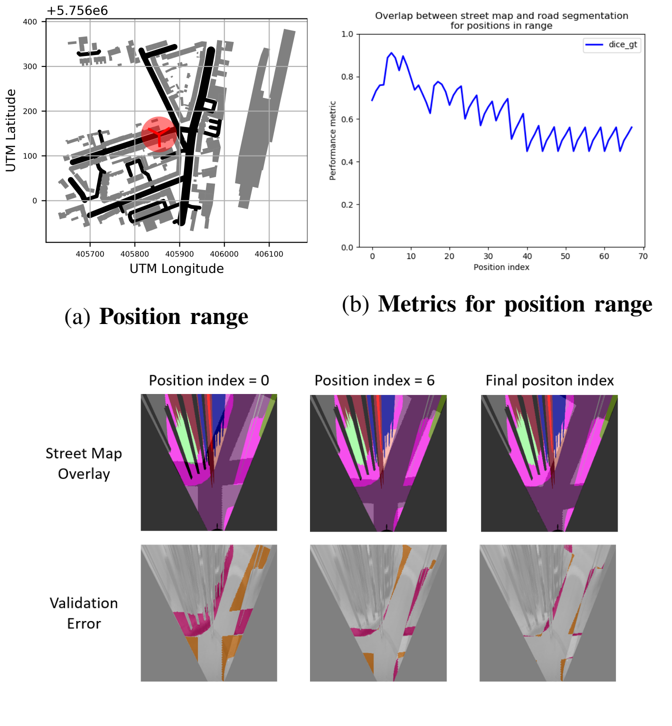
III. Approach: Localization correction

The validation of segmentation masks using street map data poses a challenge with respect to the localization precision. Inaccuracies in the vehicle localization, e.g. through a GPS position, would lead to inaccuracies in the correspondingly selected street map area. In our experiments we employed the widely used Cityscapes [9] dataset, for which we observed such inaccuracies. Apart from this dataset, the currently most used semantic segmentation datasets in relevant publications do not contain all the required vehicle localization data in terms of latitude, longitude and heading.

We thus used the Cityscapes dataset and avoid GPS localization errors via an automatic correction algorithm applied to the position, in order to demonstrate our approach. Nevertheless, modern and future technologies like landmark detection and priors can provide a localization within a few centimeters [21] and in real-world applications of our street map based validation approach, data with such precise localization information should be used.

We developed an algorithm that corrects the GPS position based on an optimal fit with the ground truth segmentation. While in the predicted segmentation larger errors with respect to the street map can be expected (and it is our goal to identify them with our validation approach), the ground truth segmentation should have no or only small errors with respect to the street map. Using this assumption, the position can be calibrated according to the ground truth segmentation and can then be re-used for the street map based validation of the predicted segmentation.

The goal of our algorithm is to identify the most accurate position within a GPS error range of a few metres for which



(a) **Position range**

(b) **Metrics for position range**

(c) **Overlay and error for selected iterations**

Fig. 6: **Localization correction.** This figure illustrates our algorithm for a localization correction based on ground-truth segmentation masks. The algorithm takes a range around the original GPS position into account (a) and computes the overlap for potential positions that lie on roads within this range (b). The saw tooth pattern in the metric diagram is the result from scanning the road not only along its length, but also along its width. Figure (c) shows the overlap for specific iterations of positions and illustrates how the search algorithm finds the optimal position.

the overlap between the street map and the ground truth segmentation is maximal. Our algorithm is consists of three steps. First, the position range according to the GPS error around the original position is determined. The road elements that lie in this range according to the street map are identified (see Figure 6a). The second step executes an optimization algorithm over a position search space: The identified road elements are rasterized into a grid of positions that lie along the length and the width of the road. The position headings are set to the angle of the corresponding road element. For each position in this search space, the dice coefficient, as defined in Section II-B, is computed (see Figure 6b). The higher the dice coefficient, the better the fit of the ground truth segmentation to the street map and thus it is more probable that the position is the true position (see Figure 6c). The position with the maximum dice coefficient is saved as the new position. The third step executes an additional optimization algorithm over a *fine* position search space that covers the positions between the new position and the closest other positions from the previous search space. This step refines the found new position by applying a similar search routine as before, but now for the finer grid of positions.

IV. Experimental Results

In this section we show comprehensive, statistical results from applying our approach to the Cityscapes segmentation dataset. We present results on three aspects: First, we apply our localization correction algorithm to the ground truth segmentations in order to find the most accurate GPS positions. Using the corrected positions, we then apply our street map based validation approach on the predicted segmentations in order to identify potential prediction errors. Finally, we compare our approach to a validation of the predictions with ground truth data.

For our experiments, we use the Cityscapes train and validation subsets, for which ground truth segmentations are available and which comprise a total number of 3475 traffic scenes. We apply an additional data cleaning step and remove traffic scenes that contain segments with the label *ground*, which describes areas that cars and pedestrians share equally. This label can not be assigned to either *road* or no-*road*, which is strictly relevant for our approach.

A. Localization correction

The localization correction significantly improves the accuracy of the GPS positions, as shown in the distribution of the *dice* coefficients before and after the correction in Figure 7. For the Cityscapes dataset we achieve an improvement from initially $dice = 0.50 \pm 0.28$, to $dice = 0.88 \pm 0.11$. The found GPS positions are saved for re-use in the validation of the predicted segmentations.

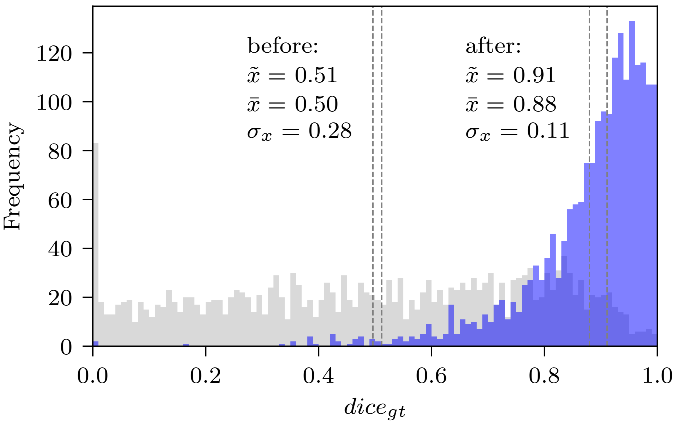


Fig. 7: **Improvement by localization correction.** Our algorithm optimizes the overlap between street map and ground truth segmentation, which is quantified by the *dice* coefficient. While it originally follows an equal distribution, which indicates a bad fit, after the correction it has a clear maximum at high values, which indicates a strongly improved and good fit.

Although our algorithm significantly improves the fit between street map and ground truth segmentation, some segmentations cannot be sufficiently aligned. A reason for this is

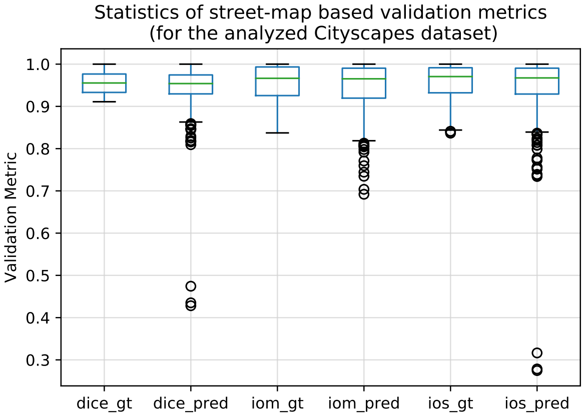


Fig. 8: **Statistical summary of validation metrics.** The box-whisker plots show the metrics $dice$, IoM, and IoS, each for the ground truth and the predicted segmentation masks, for the analyzed Cityscapes dataset. The metrics for the predicted masks show clear outliers below the lower whiskers. These are indicators for validation errors, which we want to identify.

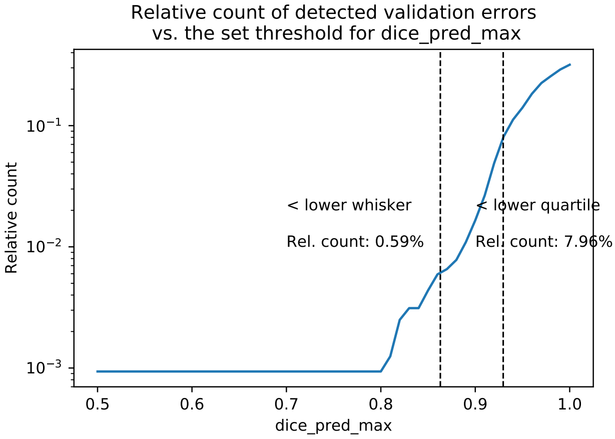


Fig. 9: **Relative count of identified validation errors.** The higher the threshold $dice_{pred,max}$, the larger the identification quantity. Predicted segmentations that have a $dice$ coefficient below are counted as an validation error. The highlighted values for the lower whisker and lower quartile refer to the thresholds deduced from the statistical summary in Figure 8.

the partially insufficient precision and information density of the used map itself. Although open map providers like OpenStreetMap integrate content from various contributors, which can easily be updated, such sources often lack details. These could include, for example, the precise geometric curvatures or the exact width of a road. To alleviate such deficiencies, we apply a further data cleaning step and filter out the traffic scenes with a mediocre fit between ground truth segmentation and street map. We continue our analysis with the data subset above the median, i.e. for which $dice > 0.91$.

B. Street map based segmentation validation

We apply our street map based validation approach in order to identify those segmentations that contain potential prediction errors. Figure 8 shows the statistical summary of our validation metrics. All in all the metrics have high values of around 95%, reflecting a general large overlap between the road segmentation and the street map. However, for the predicted segmentation there are some outliers that indicate validation errors. The existence of these outliers is expected and shows the functionality of our validation metrics.

We identify the validation errors with the following procedure. First, we query all predicted segmentations that have a $dice$ coefficient below a specified threshold, which we call $dice_{\text{pred, max}}$. The set threshold defines the relative count of returned segmentations, as shown in Figure 9. For each returned instance, we further determine if it indicates a false positive or false negative error, depending on its IoS and IoM validation metrics. Exemplary results for identified prediction errors are shown in the left columns of Figure 11.

C. Comparison to ground-truth based validation

In order to analyze the quality of the identified validation errors in the predicted segmentations, we compare them to the true prediction errors, which are the errors resulting from a validation using the ground truth segmentation. We use two pixel-based measures to quantify the performance of our method. We compute the recall, which is the probability of detection, i.e., the probability that our method indicates a validation error, given a true error. Moreover, we compute the precision, which describes the probability of correctness, i.e., the probability that there is a true error, given that our method indicates a validation error.

Figure 11 lists the recall and precision for exemplary results. It also visualizes the pixel regions from the identified validation error and the true error. As the regions mainly overlap, this shows that our method can uncover similar errors as a validation using ground truth.

The recall and precision for the total analyzed dataset are shown in Figure 10. The smaller the selected threshold for filtering the outliers, the higher the precision and recall. However, a trade-off between quantity and quality of the found errors implies a medium threshold value, such as the lower quartile or whisker (based on the statistical summary in Figure 8). Further analysis shows that precision and recall increase further with the size of the error region.

V. Conclusion

We proposed to validate machine learning models with a-priori domain knowledge and presented an approach that validates semantic segmentation masks with given street maps.

In particular, in our approach we combine the segmentation in bird's eye view with the field of view in the street map and use this overlay to compute validation errors. We introduced two new validation metrics called Intersection over Segmentation (IoS) and Intersection over Map (IoM) that we used to identify segmentation masks with false positive and false negative road segments. Furthermore, we developed

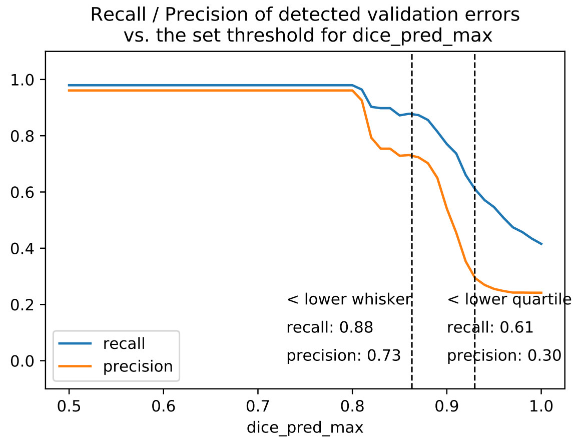


Fig. 10: **Recall and precision of identified validation errors.** The lower the threshold $dice_{pred,max}$, the higher the identification quality in terms of recall and precision. The vertical lines highlight specific thresholds deduced from the statistical summary in Figure 8.

an algorithm that can correct inaccurate vehicle positions by finding the best overlap with ground truth segmentation. We performed an experimental study with the Cityscapes dataset and OpenStreetMap, and showed that road segmentation errors can indeed be detected by our proposed validation procedure.

A general challenge is the precision of the street map itself. The validation procedure can only be as good as the information density in the map. Although our experiments with OpenStreetMap showed good results, even better results can achieved with high definition maps. For future work we therefore intend to perform the experiments with more precise and comprehensive maps, which would then also allow to investigate other classes than roads with our approach.

All in all, our proposed approach of a street map based validation of semantic segmentations offers a new and valuable way to support the goal of safe artificial intelligence in autonomous driving.

Acknowledgements

Initial work has been funded by Volkswagen Group Automation (See Section II and [22]). The remaining, extended elaboration of the work is funded by the Federal Ministry of Education and Research of Germany as part of the competence center for machine learning ML2R (01—S18038B).

References

[1] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, “Autonomous driving in urban environments: approaches, lessons and challenges,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, 2010.

[2] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus, and M. H. Ang, “Perception, planning, control, and coordination for autonomous vehicles,” *Machines*, vol. 5, no. 1, 2017.

[3] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, “A survey on deep learning techniques for image and video semantic segmentation,” *Applied Soft Computing*, vol. 70, 2018.

[4] D. Feng, C. Haase-Schuetz, L. Rosenbaum, H. Hertlein, F. Duffhauss, C. Glaeser, W. Wiesbeck, and K. Dietmayer, “Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges,” *arXiv:1902.07830*, 2019.

[5] M. Brundage, S. Avin, J. Wang, H. Belfield, G. Krueger, G. Hadfield, H. Khlaaf, J. Yang, H. Toner, R. Fong *et al.*, “Toward trustworthy ai development: Mechanisms for supporting verifiable claims,” *arXiv:2004.07213*, 2020.

[6] S. Burton, L. Gauerhof, and C. Heinzemann, “Making the case for safety of machine learning in highly automated driving,” in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2017.

[7] A. Kendall and Y. Gal, “What uncertainties do we need in Bayesian deep learning for computer vision?” in *Advances in Neural Information Processing Systems*, 2017.

[8] R. McAllister, Y. Gal, A. Kendall, M. Van Der Wilk, A. Shah, R. Cipolla, and A. V. Weller, “Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning.” International Joint Conferences on Artificial Intelligence, 2017.

[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[10] OpenStreetMap, https://www.openstreetmap.org.

[11] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauckhage, and J. Schuecker, “Informed machine learning - a taxonomy and survey of integrating knowledge into learning systems,” *arXiv:1903.12394v2*, 2020.

[12] S. Ardeshir, K. Malcolm Collins-Sibley, and M. Shah, “Geo-semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[13] S. Wang, S. Fidler, and R. Urtasun, “Holistic 3d scene understanding from a single geo-tagged image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[14] R. Díaz, M. Lee, J. Schubert, and C. C. Fowlkes, “Lifting gis maps into strong geometric context for scene understanding,” in *IEEE Winter Conference on Applications of Computer Vision*, 2016.

[15] J. K. Murthy, S. Sharma, and K. M. Krishna, “Shape priors for real-time monocular object localization in dynamic environments,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[16] B. Schroeder and A. Alahi, “Using a priori knowledge to improve scene understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[17] H. Xu, C. Jiang, X. Liang, and Z. Li, “Spatial-aware graph relation network for large-scale object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

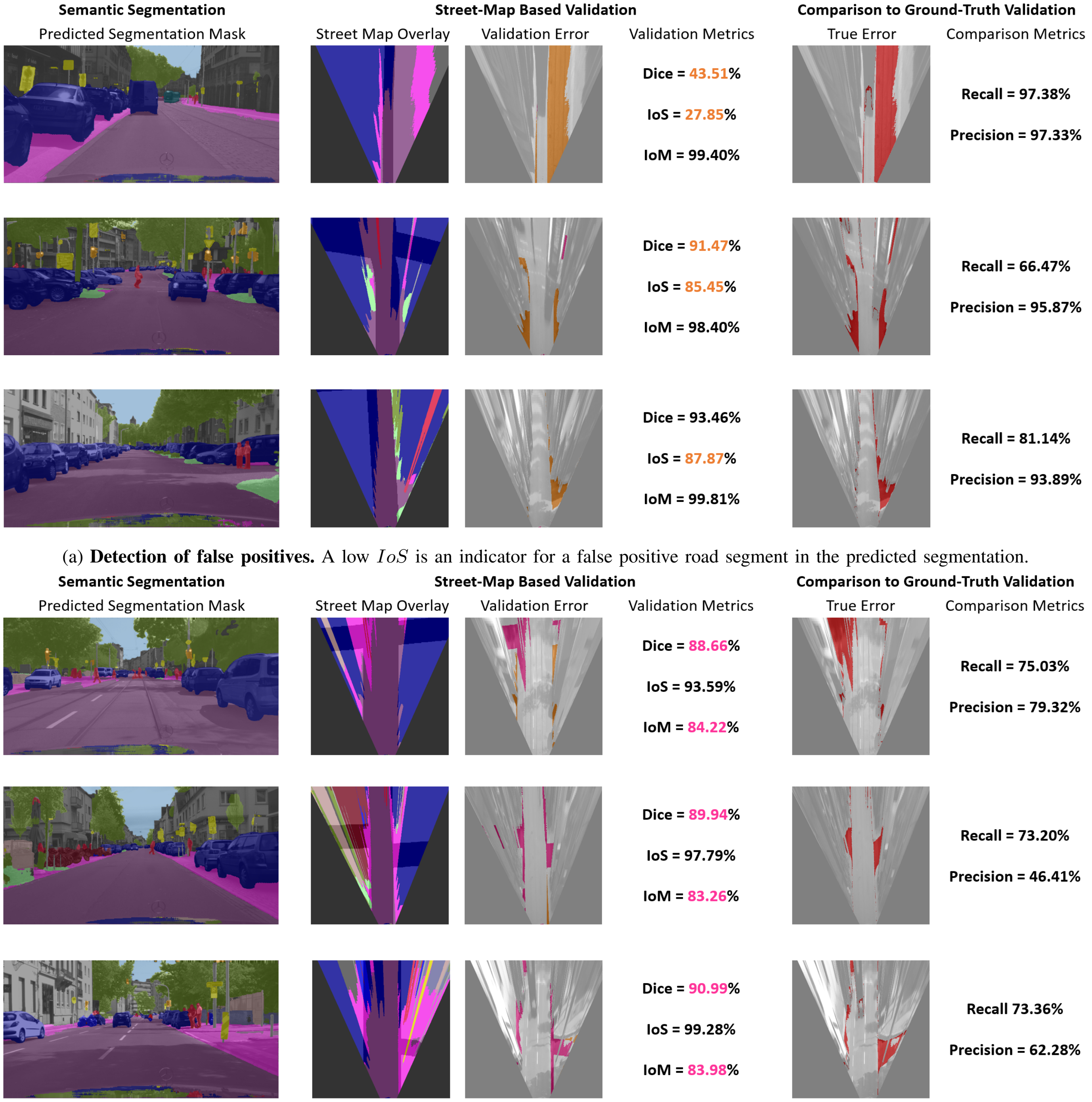
[18] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, “Learning to adapt structured output space for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[19] Y. Wang, X. Chen, Y. You, L. Erran, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao, “Train in germany, test in the usa: Making 3d object detectors generalize,” *arXiv:2005.08139*, 2020.

[20] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, “Erfnet: Efficient residual factorized convnet for real-time semantic segmentation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, 2017.

[21] D. Wilbers, C. Merfels, and C. Stachniss, “Localization with sliding window factor graphs on third-party maps for automated driving,” in *IEEE International Conference on Robotics and Automation*, 2019.

[22] L. von Rueden, T. Wirtz, F. Hueger, J. D. Schneider, and C. Bauckhage, “Towards map-based validation of semantic segmentation masks,” *Workshop on AI for Autonomous Driving (AIAD) on the 37th International Conference on Machine Learning (ICML).*



(a) **Detection of false positives.** A low IoS is an indicator for a false positive road segment in the predicted segmentation.

(b) **Detection of false negatives.** A low IoM is an indicator for a false negative road segment in the predicted segmentation.

Fig. 11: **Examples of experimental results for detected errors using our street map based validation.** This figure shows how our street-map based validation approach can help to identify prediction errors. The rows show results from the validation error list, which is the automated output from our algorithm. The six rows here are a manual selection of that automated output. Figure (a) shows exemplary results for detected false positives, and (b) for detected false negatives. Next to the actual error detection, we also show in the right columns how our method compares to a ground-truth based validation. It shows that the error regions are overlapping and the precision and recall give high values. This means that our validation method using street maps can identify similar error regions as a validation using ground truth data.

APPENDIX E

Paper P5) How Does Knowledge Injection Help in Informed Machine Learning?

How Does Knowledge Injection Help in Informed Machine Learning?

Laura von Rueden[1,2], Jochen Garcke[1,3], Christian Bauckhage[1,2]
[1]*University of Bonn,* [2]*Fraunhofer IAIS,* [3]*Fraunhofer SCAI*
Sankt Augustin, Germany
laura.von.rueden@iais.fraunhofer.de

***Abstract*—Informed machine learning describes the injection of prior knowledge into learning systems. It can help to improve generalization, especially when training data is scarce. However, the field is so application-driven that general analyses about the effect of knowledge injection are rare. This makes it difficult to transfer existing approaches to new applications, or to estimate potential improvements. Therefore, in this paper, we present a framework for quantifying the value of prior knowledge in informed machine learning. Our main contributions are threefold. Firstly, we propose a set of relevant metrics for quantifying the benefits of knowledge injection, comprising in-distribution accuracy, out-of-distribution robustness, and knowledge conformity. We also introduce a metric that combines performance improvement and data reduction. Secondly, we present a theoretical framework that represents prior knowledge in a function space and relates it to data representations and a trained model. This suggests that the distances between knowledge and data influence potential model improvements. Thirdly, we perform a systematic experimental study with controllable toy problems. All in all, this helps to find general answers to the question how knowledge injection helps in informed machine learning.**

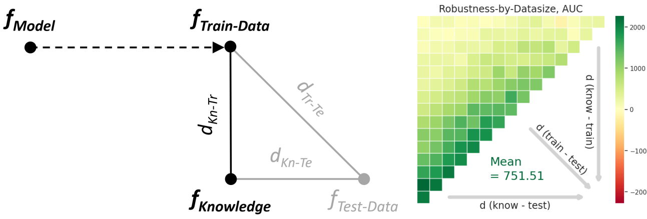
***Index Terms*—Hybrid AI, Informed Machine Learning, Prior Knowledge Injection, Neural Networks**

I. Introduction

Hybrid AI combines data-driven and knowledge-based models [1]–[3]. A particular approach that recently gained a lot of popularity is informed machine learning, which describes the injection of additional prior knowledge into learning systems [4]. This can help to improve model performance, especially when relevant training data is scarce [5]–[8]. Other potential benefits are that it can increase model robustness [9]–[11], help to ensure knowledge conformity [12]–[14], or can even improve explainability [15].

There are many applications where informed machine learning is successfully used – especially in scientific and engineering domains, where data acquisition can be expensive, but lots of prior knowledge is available. Just to give a few examples: In neural networks for climate prediction, physical laws are injected via knowledge-based loss functions [5]. In robotics, simulations are used as an additional source for training data [8]. Or in autonomous driving, spatial prototypes are employed to improve object detection [11].

However, the field is so application-driven that it has led to the development of many and rather specific approaches. In contrast, general analyses about informed machine learning are still missing [4]. This makes it difficult to transfer existing approaches to new applications, or to estimate potential improvements in advance.



(a) Function Space Illustration with Representations of Data and Prior Knowledge.

(b) Model Improvement using Prior Knowledge.

Fig. 1: In Informed Machine Learning, prior knowledge is integrated into data-based learning [4], [16]. To better understand its effect, we propose a framework that represents knowledge in a function space (See a). We analyze how the distances between knowledge, train, and test data influence the potential model improvements. E.g., we find that informed learning greatly improves model robustness, especially when the knowledge is close to out-of-distribution test data (See b).

Therefore, in this paper our objective is to find general answers to the research question of how knowledge injection via informed machine learning does help. We further subdivide this into the following subquestions:

- How can knowledge injection improve machine learning?
- What are the requirements for the injected knowledge?
- How should the knowledge be injected?

Our approach is to develop a framework for quantifying the value of prior knowledge in informed machine learning. For this, we first define a set of metrics that quantify potential benefits. Then we propose a theoretical framework that helps to formalize prior knowledge injection. It is a first step towards an informed learning theory. Our main idea is to regard prior knowledge as a function that can be represented in the same space as the model or the training data (See Figure 1a). We conjecture that the distance between data and knowledge determines the potential benefits of informed machine learning. To illustrate the framework, we perform a systematic experimental study with toy problems. As the toy problems we propose a classification task, which allow to vary the knowledge and the injection method in a controllable manner. We vary relevant parameters, such as the distance between data and knowledge (See Figure 1b), and measure the potential improvements through informed machine learning.

In summary, the main contributions of this paper are:

1) We propose a set of metrics for quantifying the benefits of informed machine learning.
2) We present a first theoretical framework for informed machine learning.
3) We perform a systematic experimental study with controllable toy problems.

Each of these contributions helps to answer the above research questions. The paper is structured accordingly.

II. Related Work

Our work is mainly related to hybrid AI and informed machine learning, but also reuses concepts from learning theory.

A. *Informed Machine Learning*

In informed machine learning, pre-given formalized knowledge is injected into data-driven learning systems [4], [16], [17]. It is sometimes also called theory-guided data science [18], or causally-aware machine learning [19]. The taxonomy of informed learning depicts the diversity of applications and methods in terms of knowledge source, representation type, and integration method [4].

However, related work about the general, application-independent, effect of knowledge injection in informed machine learning is rare. We shortly describe the works that go in this direction, ordered by our contributions in terms of 1) metric quantification, 2) theoretical framework and 3) experimental study.

A first work that presents an approach for the quantification of domain knowledge in informed machine learning is given by Yang et al. [20]. They proposed a method based on the Shapley value to quantify the contribution of injected prior knowledge to the model performance improvement. *The main difference to our work is that they consider a set of knowledge pieces and attribute the contribution of the individual pieces, whereas we consider knowledge as an abstract unit and analyze which properties it needs to have.*

A first theoretical study about physics-informed neural networks was presented by Shin et al. [21]. They provide a convergence theory with respect to the number of data samples. Yang et al. have also presented a theoretical study on informed learning by wide neural networks [22]. They especially investigate the trade-off between knowledge and data labels.

A first experimental comparison of informed learning methods is given by Monaco et al. [23]. They consider three application examples and on each they evaluate two informed learning methods. In particular they measure the performance for variations of the training data size. *The difference to our work is that they investigate pre-given applications, whereas we investigate toy problems, which allow us to adapt the experiments and the knowledge in a controllable manner. Moreover, they only measure the prediction error for various data sizes, whereas we develop and measure a total catalogue of metrics.*

B. *Learning Theory*

The foundations of statistical learning theory have been developed already many years ago by Vapnik et al. [24]. Overviews about learning theory can be found in [25], [26]. At the heart of it is the principle of empirical risk minimization, which we shortly recap. The goal of a learning task is to find a model $f : \mathcal{X} \to \mathcal{Y}$, with $f \in \mathcal{F}$, based on some given training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1\ldots n}$, with features $x \in \mathcal{X}$, labels $y \in \mathcal{Y}$ and sample size n. The model can then be approximated by minimizing the empirical risk $R(f)$ with a given loss function l:

$$\hat{f} := \arg\min_{f \in \mathcal{F}} R_{\mathcal{D}}(f), \quad R_{\mathcal{D}}(f) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} l(f(x), y) \quad (1)$$

Recently, an extension of the statistical learning theory was proposed in terms of also taking into account the preservation of invariants [27], also called invariant risk minimization [28], [29]. This approach can be motivated by the goal of out-of-distribution generalization [30]: Assuming training data is collected in various environments, then statistical invariants across them should also hold in novel testing environments [31]. *This idea is similar to our understanding of informed machine learning: Prior knowledge describes causal relationships that are underlying a given data distribution, i.e. invariants. Integrating these into a learning task can thus improve model performance. The main difference is that in invariant risk minimization the invariants still need to be learned, whereas in informed machine learning they are given by prior knowledge.*

III. Metrics for Informed Learning

As described in [4], the main goals of informed machine learning are to train with less data, to achieve a better model performance, to increase knowledge conformity, or to increase

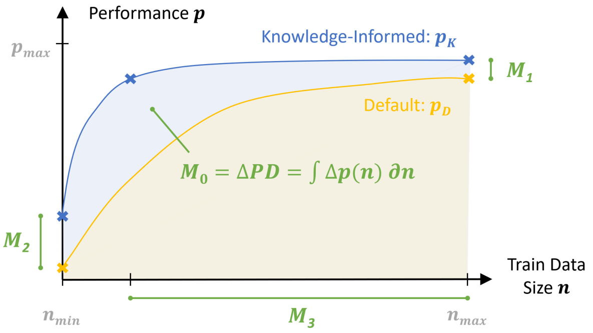


Fig. 2: Illustration of Performance vs. Size of Training Data. Models that are trained with informed machine learning usually achieve a higher performance, e.g. accuracy or robustness, for smaller training data sizes [5], [6], [11]. We propose a new metric that quantifies performance and data need in a single metric in terms of the area under the curve: *Performance-by-Data AUC*, in short PD (see Section III-A). All in all, we suggest to quantify improvements in terms of four metric flavours: Increase in Performance-by-Data AUC (M_0), increase in performance at max. and min. data size (M_1 and M_2), as well as data reduction for a specific performance (M_3).

interpretability. However, most works about informed learning methods present individual metrics to quantify the benefits of their method. For example, [5] reports test error and physical inconsistency for various data sizes, [20] compares test accuracy for full data size, or [11] reports ouf-of-distribution robustness for various data sizes.

Here, we propose a systematic metric catalogue, as well as a new metric that combines performance improvement and data efficiency. These allow a more transparent, and standardized comparison of various methods. Moreover, they provide the basis for future benchmarks of informed learning methods.

A. *Performance-by-Data AUC*

We propose to measure performance (e.g., test accuracy) for various train data sizes and summarize the results in a single metric that we call *Performance-by-Data AUC*. As illustrated in Figure 2, the metric quantifies the area under the curve of performance p vs. training data size n.

Definition 1 (Performance-by-Data AUC)**.**

$$PD = \int_{n_{min}}^{n_{max}} p(n)\,\mathrm{d}n \tag{2}$$

This metric can be normalized through dividing by the maximum possible area, i.e. by $p_{max} * (n_{max} - n_{min})$, where p_{max} is the maximum possible performance (e.g., 100% test accuracy). Then $PD \in [0.0, 1.0]$ and the larger the better.

For comparing two models, e.g., a (knowledge-)informed model with performance p_K and a default, data-based model with performance p_D, the difference between the two area integrals can be computed.

Definition 2 (Improvement of Performance-by-Data AUC)**.**

$$\Delta PD = \int_{n_{min}}^{n_{max}} \Delta p(n)\,\mathrm{d}n \tag{3}$$

$$= \int_{n_{min}}^{n_{max}} (p_{\mathrm{K}}(n) - p_{\mathrm{D}}(n))\,\mathrm{d}n \tag{4}$$

The proposed ΔPD metric has the advantage that it encapsulates the performance for all data set sizes in a single metric. This means, one does not need to choose a specific data set size for which to compare the performance, or vice versa.

B. *Metrics Catalogue*

For evaluating informed learning methods, we focus on metrics that are especially relevant for model generalization: In-Distribution Test Accuracy, Out-of-Distribution Robustness, and Knowledge Conformity. The generic performance p from above can be any of these 3 metric types. As indicated in Figure 2, we specifically evaluate each metric in 4 metric flavours: The above described Performance-by-Data AUC (M_0 in Figure 2), but also the performance at max. and min. data size (M_1 and M_2), as well as the data amount that is required to achieve a specific performance (M_3). Further metric types for evaluating informed methods are training time and model size. Also a measurement of the model interpretability is interesting, however, such a quantification is currently still an open research question [15].

Box 1: Metrics Catalogue: Improvements through Informed Learning

This catalogue represents the various goals of informed learning and depicts how knowledge injection can improve machine learning.

1) **Increase of In-Distribution (IID) Test Accuracy**
   a) Increase: IID Accuracy-by-Datasize
   b) Increase: IID Accuracy for Max. Datasize
   c) Increase: IID Accuracy for Min. Datasize
   d) Reduction: Training Datasize for specific IID Accuracy
2) **Increase of Out-of-Distribution (OOD) Robustness**
   a) Increase: OOD Robustness-by-Datasize
   b) Increase: OOD Robustness for Max. Datasize
   c) Increase: OOD Robustness for Min. Datasize
   d) Reduction: Training Datasize for specific OOD Robustness
3) **Increase of Knowledge Conformity**
   a) Increase: Knowledge Conf.-by-Datasize
   b) Increase: Knowledge Conf. for Max. Datasize
   c) Increase: Knowledge Conf. for Min. Datasize
   d) Reduction: Training Datasize for specific Knowledge Conf.
4) **Reduction of Training Data ***
5) **Reduction of Training Time**
6) **Reduction of Model Size**
7) **Improvement in Interpretability**

* Please note that the important goal of data reduction is represented below each of first three metric types (See 1a+d, 2a+d, 3a+d).

In summary, we suggest the metric catalogue that is shown in Box 1 for evaluating informed learning methods.

IV. A Framework for an Informed Learning Theory

We want to better understand what influences the expected performance gains of informed learning. In particular, it is of great interest what the requirements on the injected knowledge are. To investigate this, we employ and extend concepts from statistical learning theory [25], [26]. This way, we hope to make a first step in the direction of an *informed* learning theory.

A. *Knowledge in Function Space*

The question about the requirements on the injected knowledge is non-trivial, because knowledge can be represented in versatile forms. As depicted in the informed learning taxonomy [4], typical representations of prior knowledge are algebraic equations, logic rules, knowledge graphs, simulation results, or human feedback. An investigation on the requirements for each type could already be exhaustive.

Here, we therefore take an abstract view and conjecture:

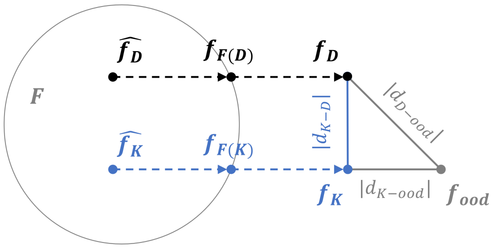


Fig. 3: Function space with representations of prior knowledge f_K, data f_D, and OOD test data f_{ood} (right), and decomposition in generalization error terms (left). The circle illustrates the function space F used by a learning algorithm. Beyond the circle is the space of all possible functions. $\hat{f_D}$ is the empirical best solution of the algorithm (See Equation 1). $f_{F(D)}$ is the best possible solution in F. f_D represents the (*unknown*) data distribution. The blue elements show the respective representations for prior knowledge (our proposed informed learning extension). Particularly, f_K represents the (*known*) prior knowledge.

Axiom 3 (**Prior Knowledge**). **Prior knowledge describes relations between concepts and can be represented as a function.**

We use this to relate it to given data:

Axiom 4 (**Knowledge Representation in Function Space**). **Prior knowledge can be represented in the same function space as given data representations.**

Figure 3 illustrates the knowledge representation in a function space. Here, we also illustrate the distance $|d_{K-D}|$ between the *known* knowledge representation f_K and the *unknown* data representation f_D. In addition to the in-distribution data, we also consider an out-of-distribution data, which is represented by the *unknown* data representation f_{ood}. The illustration in function space depicts how prior knowledge can give hints about the unknown data representations.

B. *Knowledge-to-Data Distance*

Let us consider the case for in-distribution (IID) generalization. This means that a model is tested on data that follows the same underlying distribution as the training data.

We are interested in the expected performance improvement through informed learning by using the prior knowledge f_K. In the statistical learning theory, maximizing model performance is equivalent to minimizing the empirical risk (see Equation 1). We thus regard the risks $R(\hat{f_D})$ and $R(\hat{f_K})$. The generalization error for the default, data-based model can be decomposed as follows (see *black* drawing in Figure 3):

$$\begin{aligned}\underbrace{R(\hat{f_D}) - R(f_D)}_{\text{generalization error}} &= \underbrace{\Big(R(\hat{f_D}) - R(f_{F(D)})\Big)}_{\text{estimation error}} \\ &+ \underbrace{\big(R(f_{F(D)}) - R(f_D)\big)}_{\text{approximation error}} \qquad (5)\end{aligned}$$

We propose to also formalize the error for a purely informed model with respect to generalization to the in-distribution data (see *blue* drawing in Figure 3):

$$\begin{aligned}\underbrace{R(\hat{f_K}) - R(f_D)}_{\text{know. generalization error}} &= \underbrace{\Big(R(\hat{f_K}) - R(f_{F(K)})\Big)}_{\text{know. estimation error}} \\ &+ \underbrace{\big(R(f_{F(K)}) - R(f_K)\big)}_{\text{know. approximation error}} \\ &+ \underbrace{\big(R(f_K) - R(f_D)\big)}_{\text{know.-to-data error}} \qquad (6)\end{aligned}$$

For the model distance in terms of their generalization errors follows then:

$$\begin{aligned}R(\hat{f_K}) - R(\hat{f_D}) &= C + \underbrace{\big(R(f_K) - R(f_D)\big)}_{\text{know.-to-data error}} \\ &\propto C + \underbrace{|d_{K-D}|}_{\text{know.-to-data distance}} \qquad (7)\end{aligned}$$

Conjecture 5 (**Informed IID-Generalization Improvement**). **The smaller the distance between knowledge and data, the larger the improvement through informed learning on in-distribution generalization.**

C. *Knowledge-to-OOD Distance*

Let us consider the case of out-of-distribution generalization. Out-of-distribution generally refers to the evaluation on test data that follows another distribution then the train data [30].

Here, for the model distance in terms of their out-of-distribution generalization errors follows then:

$$R_{ood}(\hat{f_K}) - R_{ood}(\hat{f_D}) \propto C_{ood} + \underbrace{|d_{K-ood}|}_{\text{know.-to-ood dist.}} - \underbrace{|d_{D-ood}|}_{\text{data-to-ood dist.}} \qquad (8)$$

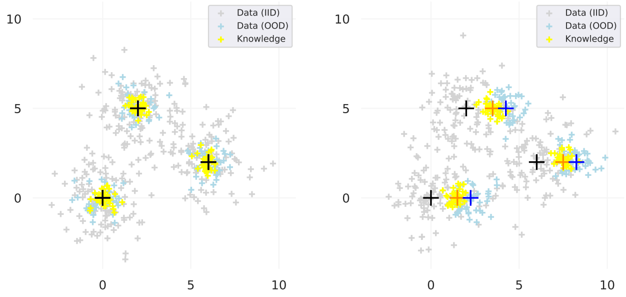
Conjecture 6 (**Informed OOD-Generalization Improvement**). **The smaller the distance between knowledge and the OOD data, and the larger the distance between IID and OOD data, the larger the potential improvement through informed learning on the OOD generalization.**

V. Systematic Experimental Analysis

We performed a systematic experimental study of the effect of knowledge injection in informed machine learning. For this, we defined a controllable toy problem. We measured the performance metrics as defined in Section III and employ the theoretical framework from Section IV.

A. *Experimental Setup*

1) Toy Datasets: Let us consider a toy problem for the task of classification, as illustrated in Figure 4b. We have also investigated a toy problem for regression, which shows similar results. Since the effects of knowledge injection, especially the influence of the distances between knowledge and data, can



(a) No Distances.

(b) With Distances.

Fig. 4: Toy dataset for classification with 3 classes. We use distinct sets for in-distribution data (grey), out-of-distribution data (blue), and prior knowledge (yellow). The distance can be varied between the sets (as motivated by the theoretical framework in Section IV). (a) shows the case when the centers of IID data, OOD data, and knowledge overlap, i.e. for $|d_{K-D}| = 0$, $|d_{K-OOD}| = 0$. (b) shows an example with distances between them (Here: $|d_{K-D}| = 1.5$, $|d_{K-OOD}| = 0.75$). In our experimental study, we measure the effect of informed machine learning for various distance setups.

be more clearly with the classification problem, we consider this in the following.

The toy dataset contains three classes. Each blob in Figure 4b represents another class (i.e. the top blob, lower left blob, and middle right blob). The number of samples is 288, with 96 samples per class.

In addition to the main (IID) data, we also consider a smaller sets of OOD data, and of prior knowledge representations. Here, the original prior knowledge representation can be understood as class prototypes, similar as in [11]. In applications, such prototypes can, e.g., be structural templates (e.g. traffic sign templates for image recognition). Such knowledge can be transformed into a data format by rendering. Since prior knowledge is more concise than data, we consciously chose smaller standard deviations for the knowledge set.

The distances between the main (IID) data, the OOD data, and the prior knowledge can be controlled and varied. An example for a distance setup is shown in Figure 4b.

2) Systematic Analysis: In our systematic study, we vary several parameters: 1) Distances between knowledge and data, 2) Amount of training data, 3) (informed) learning method. For each setup, we measure the metrics from our metrics catalogue. Especially, we focus on IID Test Accuracy, OOD Robustness, and Knowledge Conformity (i.e., accuracy on the IID data set, accuracy on the OOD data set, and accuracy on knowledge samples).

We investigate a range of distance setups, as illustrated in Figure 5. For this, we keep the position of the IID data set fixed and move the OOD data set and/or the knowledge to the side. In particular, we consider a maximum distance of 3.5 with a step size of 0.25, i.e. a total of 15 positions. We combine distances of know-data with distances of know-ood, resulting in the illustrated position triangles. For every position we perform separate trainings.

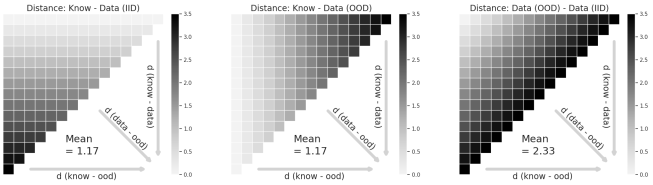


Fig. 5: Illustration of distance variation in systematic experimental study: Every position represents a unique experimental set up in terms of distances between prior knowledge, (IID) data, and OOD data. For every setup, we perform a default neural network training and informed trainings in order to measure the gained performance improvements. In addition, we train every setup for a range of training data sizes.

Furthermore, we vary the size of the training data. We consider 6 unique sizes from 10 to 300 data samples with an exponential growth. By taking into account various training data sizes, we can measure the metric flavours, as described in Section III: Performance-by-Data AUC, performance at max. data, performance at min. data, and data need to reach a specific performance.

As informed machine learning, we apply two methods, similar as in [11]: Combining training data and knowledge samples in terms of 1) Concurrent Training, 2) Informed Pre-Training.

3) Learning Setup: We apply a neural network with 1 hidden layer with 100 neurons. We use stochastic gradient descent and cross entropy loss for the learning algorithm. As the hyperparameters we use: batch size = 18, learning rate = 0.01, momentum = 0.9, early stopping after 3 stagnating epochs, regularization with weight decay = 0.2. Each experiment is repeated 10 times. For every run the data samples are generated randomly.

B. Results

The complete results in terms of improvements of informed learning over the default setup can be found in Figure 7. Results for Informed Pre-Training are shown in Figure 8. Both informed learning methods show that our distance theorems from Section IV are confirmed. We also nicely see, that our introduced metric of Performance-by-Data AUC (Definition 2) is a good summary of the other metrics. In general, we see that informed learning can greatly improve OOD robustness.

A subset of the results is shown for a closer look in Figure 6. The left subfigure shows the improvement in IID generalization. We observe that the smaller the distance between knowledge and training data (upper pixel rows) the larger the improvement. This confirms our Conjecture 5 from above. The right subfigure shows the improvement in OOD robustness. Here, we can see that the the improvement is largest when the distance between knowledge and training data is large (lower pixel rows) and the distance between knowledge and OOD test data is small (closer to diagonal). This confirms our Conjecture 6 from above.

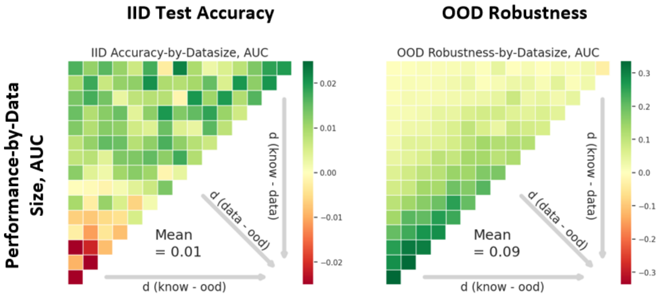


Fig. 6: Experimental Results: Improvements in IID-Generalization and OOD-Robustness through Informed Training. The left plot confirms our Conjecture 5, and the right our Conjecture 6. (Complete Results can be found in Figures 7 and 8.)

VI. Conclusion

In this paper, we presented a framework for quantifying the value of prior knowledge in informed machine learning. We first proposed a set of relevant metrics for quantifying the benefits of knowledge injection, comprising in-distribution accuracy, out-of-distribution robustness, and knowledge conformity. We also introduced a metric that combines performance improvement and data reduction, called performance-by-data AUC. Secondly, we presented a theoretical framework that represents prior knowledge in a function space and relates it to data representations and a trained model. Thirdly, we performed a systematic experimental study with controllable toy problems. These confirmed our theories about the influence of the distances between knowledge and data on potential model improvements. All in all, our contributions hopefully help to find general answers to the question how knowledge injection helps. In particular they form the basis for potential benchmarks of informed machine learning.

Acknowledgment

This research has been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence.

References

[1] G. Marcus, “The next decade in ai: four steps towards robust artificial intelligence,” *arXiv preprint arXiv:2002.06177*, 2020.

[2] L. von Rueden, S. Mayer, R. Sifa, C. Bauckhage, and J. Garcke, “Combining machine learning and simulation to a hybrid modelling approach: Current and future directions,” in *Advances in Intelligent Data Analysis(IDA)*. Springer, 2020.

[3] M. K. Sarker, L. Zhou, A. Eberhart, and P. Hitzler, “Neuro-symbolic artificial intelligence,” *AI Communications*, 2021.

[4] L. Von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauckhage, and J. Schuecker, “Informed machine learning – a taxonomy and survey of integrating prior knowledge into learning systems,” *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[5] A. Karpatne, W. Watkins, J. Read, and V. Kumar, “Physics-guided neural networks (pgnn): An application in lake temperature modeling,” *arXiv preprint arXiv:1710.11431*, 2017.

[6] R. Stewart and S. Ermon, “Label-free supervision of neural networks with physics and domain knowledge,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[7] T. M. Deist, A. Patti, Z. Wang, D. Krane, T. Sorenson, and D. Craft, “Simulation-assisted machine learning,” *Bioinformatics*, 2019.

[8] A. Rai, R. Antonova, F. Meier, and C. G. Atkeson, “Using simulation to improve sample-efficiency of bayesian optimization for bipedal robots,” *The Journal of Machine Learning Research*, 2019.

[9] T. Kyono and M. van der Schaar, “Improving model robustness using causal knowledge,” *arXiv preprint arXiv:1911.12441*, 2019.

[10] N. M. Gürel, X. Qi, L. Rimanic, C. Zhang, and B. Li, “Knowledge enhanced machine learning pipeline against diverse adversarial attacks,” in *International Conference on Machine Learning*. PMLR, 2021.

[11] L. Von Rueden, S. Houben, K. Cvejoski, C. Bauckhage, and N. Piatkowski, “Informed pre-training on prior knowledge,” *arXiv preprint arXiv:2205.11433*, 2022.

[12] M. Bahari, I. Nejjar, and A. Alahi, “Injecting knowledge in data-driven vehicle trajectory predictors,” *Transportation research part C: emerging technologies*, 2021.

[13] L. von Rueden, T. Wirtz, F. Hueger, J. D. Schneider, N. Piatkowski, and C. Bauckhage, “Street-map based validation of semantic segmentation in autonomous driving,” in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021.

[14] J. Wörmann, D. Bogdoll, E. Bührle, H. Chen, E. F. Chuo, K. Cvejoski, L. van Elst, T. Gleißner, P. Gottschall, S. Griesche *et al.*, “Knowledge augmented machine learning with applications in autonomous driving: A survey,” *arXiv preprint arXiv:2205.04712*, 2022.

[15] K. Beckh, S. Müller, M. Jakobs, V. Toborek, H. Tan, R. Fischer, P. Welke, S. Houben, and L. von Rueden, “Explainable machine learning with prior knowledge: An overview,” *arXiv preprint arXiv:2105.10172*, 2021.

[16] L. Von Rueden, S. Mayer, J. Garcke, C. Bauckhage, and J. Schuecker, “Informed machine learning – towards a taxonomy of explicit integration of knowledge into machine learning,” *arXiv preprint arXiv:1903.12394*, 2019.

[17] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, 2021.

[18] A. Karpatne, G. Atluri, J. H. Faghmous, M. Steinbach, A. Banerjee, A. Ganguly, S. Shekhar, N. Samatova, and V. Kumar, “Theory-guided data science: A new paradigm for scientific discovery from data,” *IEEE Transactions on knowledge and data engineering*, 2017.

[19] T. Kyono, “Towards causally-aware machine learning,” *PhD Thesis, University of California*, 2021.

[20] J. Yang and S. Ren, “A quantitative perspective on values of domain knowledge for machine learning,” *arXiv preprint arXiv:2011.08450*, 2020.

[21] Y. Shin, J. Darbon, and G. E. Karniadakis, “On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type pdes,” *arXiv preprint arXiv:2004.01806*, 2020.

[22] J. Yang and S. Ren, “Informed learning by wide neural networks: Convergence, generalization and sampling complexity,” *arXiv preprint arXiv:2207.00751*, 2022.

[23] S. Monaco, D. Apiletti, and G. Malnati, “Theory-guided deep learning algorithms: An experimental evaluation,” *Electronics*, 2022.

[24] V. Vapnik, “Principles of risk minimization for learning theory,” *Advances in neural information processing systems*, 1991.

[25] U. Von Luxburg and B. Schölkopf, “Statistical learning theory: Models, concepts, and results,” in *Handbook of the History of Logic*. Elsevier, 2011.

[26] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[27] V. Vapnik and R. Izmailov, “Rethinking statistical learning theory: learning using statistical invariants,” *Machine Learning*, 2019.

[28] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, “Invariant risk minimization,” *arXiv preprint arXiv:1907.02893*, 2019.

[29] K. Ahuja, J. Wang, A. Dhurandhar, K. Shanmugam, and K. R. Varshney, “Empirical or invariant risk minimization? a sample complexity perspective,” *arXiv preprint arXiv:2010.16412*, 2020.

[30] Z. Shen, J. Liu, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui, “Towards out-of-distribution generalization: A survey,” *arXiv preprint arXiv:2108.13624*, 2021.

[31] M. Arjovsky, “Out of distribution generalization in machine learning,” Ph.D. dissertation, New York University, 2020.

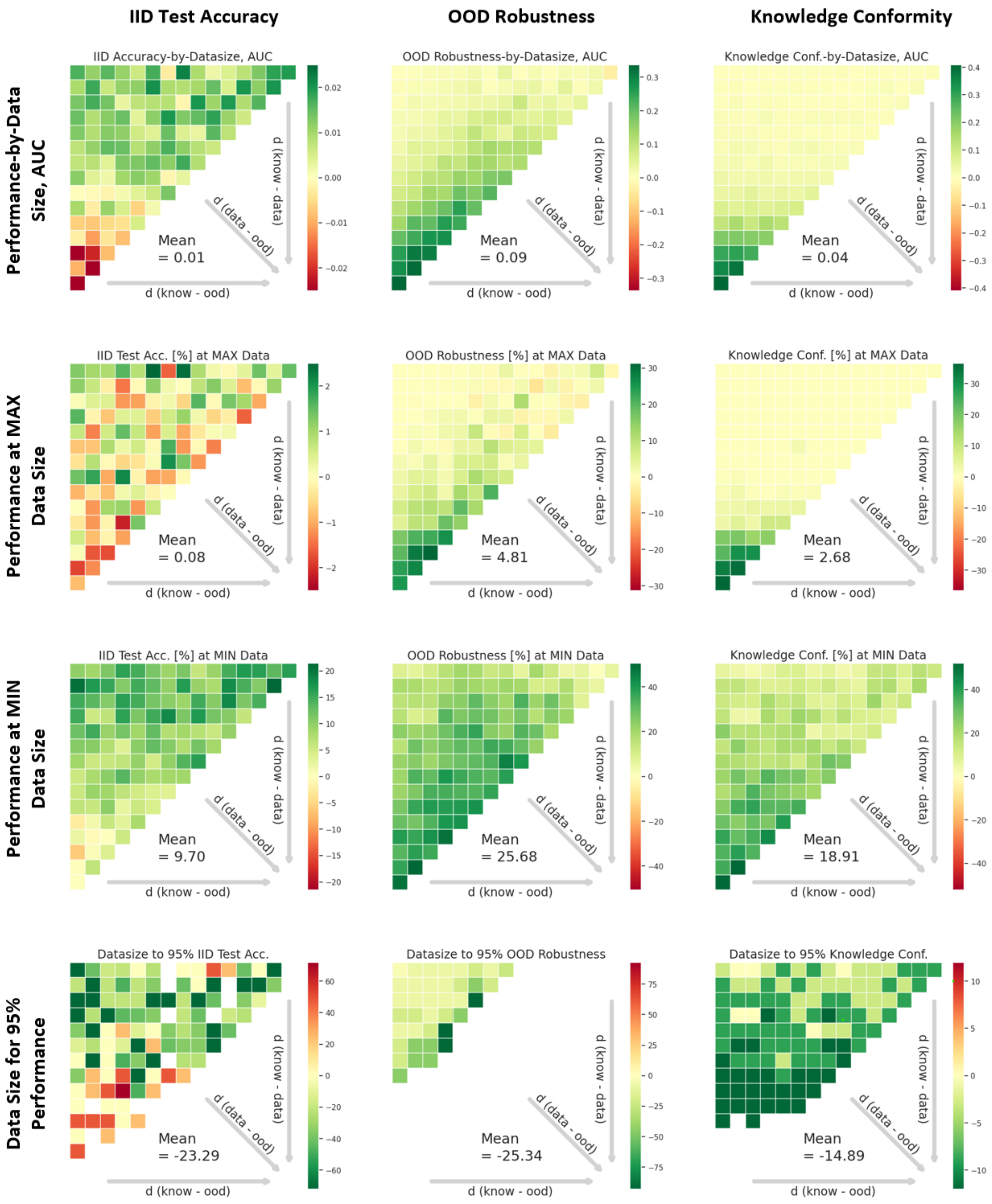


Fig. 7: Experimental Results: Improvements through Informed Training.

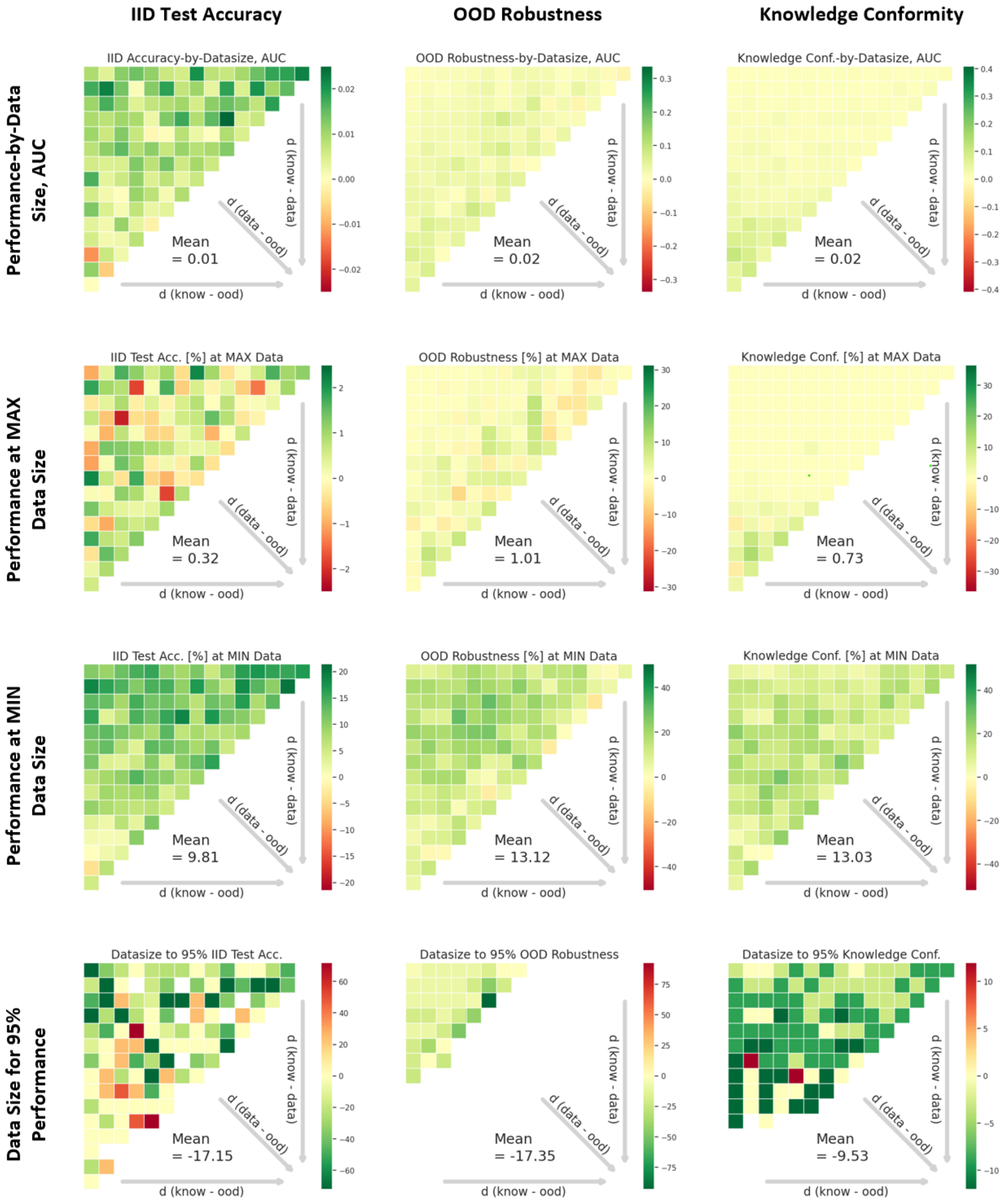


Fig. 8: Experimental Results: Improvements through Informed Pre-Training.