Doctoral Dissertation
Doctoral Program in Computer Science (38thcycle)

Knowledge-Informed Machine Learning for Industrial Application

By

Stefano Zampini

******

Supervisors:
Prof. Davide Anguita
Prof. Luca Oneto

Doctoral Examination Committee:
-

Politecnico di Torino
2025

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Stefano Zampini
2025

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Ai professori Luca Oneto e Davide Anguita per avermi seguito sempre con grande disponibilità. Il loro supporto e la loro guida sono stati fondamentali in questo percorso.

Alla mia compagna di vita, Clelia, alla quale ho chiesto tantissimo. Grazie per aver resitito, non ti sarò mai grato abbastanza.

Alla Ró, ai miei genitori, ai miei nonni, ai miei zii, a tutti i miei cugini Andrea, Erika, Sofia e, l'ultimo arrivato, Jacopo. A tutta la mia famiglia. Con le spalle coperte qualsiasi traguardio è più semplice.

A Nico, che non è solo mio cugino, ma il mio migliore amico da sempre. Ogni passo l'abbiamo fatto insieme, così diversi così uguali.

Agli amici di Bolano, si torna sempre dove si sta bene. E non c'è posto dove si stia meglio.

Ai miei amici nel mondo, da Charlottesville a Tokyo. Siete riusciti a farmi sentire a casa anche lontano da casa.

Ai professori Nando Fioretto, Andrea Coraddu e Daniele Fiscaletti, che mi hanno accolto nei loro gruppi di ricerca, offrendomi grandi opportunità di crescita umana e professionale.

Ai coordinatori del Dottorato del Politecnico di Torino, dell'Università di Genova, dell'Università della Virginia e dell'Università Tecnica di Delft, ai colleghi, ai coinquilini e a tutte le persone con cui abbiamo intrecciato il percorso.

Grazie.

Stefano

Acknowledgements

This PhD was carried out as part of the National PhD in Artificial Intelligence program, managed by the Politecnico di Torino and the University of Genoa. The program is co-financed by the Italian Ministry of University and Research and participating universities and research institutions.

Part of this research was conducted at the RAISE Lab, University of Virginia, in collaboration with Prof. F. Fioretto and PhD student J. Christopher. Another part was carried out at Delft University of Technology, in collaboration with Prof. A. Coraddu, Prof. D. Fiscaletti, and PhD student M. Costa.

Abstract

Artificial Intelligence models are proving extremely useful and are rapidly spreading across the industrial landscape; however, their robustness remains an open challenge that requires further investigation. The use of Machine Learning models—mostly data-driven—poses issues in high-risk domains where model reliability is essential, where understanding the reasoning behind a model's decision is crucial, or in situations where data is scarce or noisy, making purely data-driven approaches unreliable. These concerns have given rise to various research directions, all aiming to improve the trustworthiness and reliability of Machine Learning models. One of the most promising approaches focuses on enhancing models' robustness by incorporating domain knowledge of various kinds into predominantly data-driven Machine Learning models. These techniques are referred as *Knowledge-Informed* Machine Learning.

This Ph.D. project has been dedicated to advancing research in this area by organizing existing techniques and pushing the boundaries of the field through new applications, with the ultimate goal of accelerating technology transfer to industry. This final thesis presents the key outcomes of this work. Firstly, we report a comprehensive study and classification of existing Knowledge-Informed Machine Learning techniques. To date, in industrial contexts, these techniques have been applied almost exclusively to predictive modeling. For the first time, a formal analysis of these techniques is provided, comparing them with traditional Machine Learning approaches and offering an extensive overview of those already adopted in industry. Having laid these foundations, the thesis moves on to its core focus: the development of new practical applications and the expansion of research into emerging directions. Two representative case studies are presented. The first is a practical implementation of a Knowledge-Informed Predictive Model, specifically a power flow analysis using physics to inform a Machine Learning model. In this case study, existing iterative models

for solving the underlying physical equations were highly inefficient, making them unsuitable for real-time control applications. To address this, a surrogate model was developed by integrating multiple knowledge-informing techniques, enabling accurate predictions while significantly outperforming traditional models in terms of both efficiency and reliability. The second application explores a Knowledge-Informed Generative Model, a class of models still in its early stages but already showing potential for industrial use. In this case, the generative modeling framework, specifically diffusion models, was employed to tackle an inverse design problem applied to mechanical metamaterials. A completely novel technique was introduced, which we demonstrated to be highly effective in scenarios where strict adherence to design constraints is crucial. Both applications demonstrate the advantages of using Knowledge-Informed Machine Learning to enhance model robustness and validate the research carried out throughout this doctoral project. This manuscript concludes by outlining the ongoing developments and what we believe to be the future prospects of this line of research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, Artificial Intelligence (AI) has been rapidly spreading across the globe. Yet, interest in AI has existed for quite some time, beginning in the mid 20th century closely following the birth of computer science. At that time, the idea emerged of building machines capable of performing tasks typical of human intelligence, and several research projects began to pursue this goal [1, 2]. However, its early development proceeded at a relatively slow pace. During the 1960s, 1970s, and 1980s, AI research was primarily based on a symbolic approach, where knowledge about the world was represented through symbols and logical rules, aiming to mimic human reasoning through deductive inference [3]. Despite these efforts, this approach never truly spread in the global market. The real breakthrough came with the availability of large datasets and computational power, which led to the Machine Learning (ML) revolution in the 1990s and early 2000s, shifting the focus to statistical data-driven models [3]. The main advantage of these models lies in their domain-agnostic nature, which allows them to learn behavior directly from data, thereby obviating the need for specific domain expertise.

Since then, ML has seen exponential growth. The number of AI-related publications has risen exponentially, as shown in Figure 1.1, while private investments have increased by 1300% since 2014 [4]. In the last fifteen years, we have witnessed the rapid emergence of groundbreaking innovations such as Deep Learning, Generative AI, Transformers, Large Language Models (LLMs), and AI Agents, marking a new era in technological advancement. This revolution

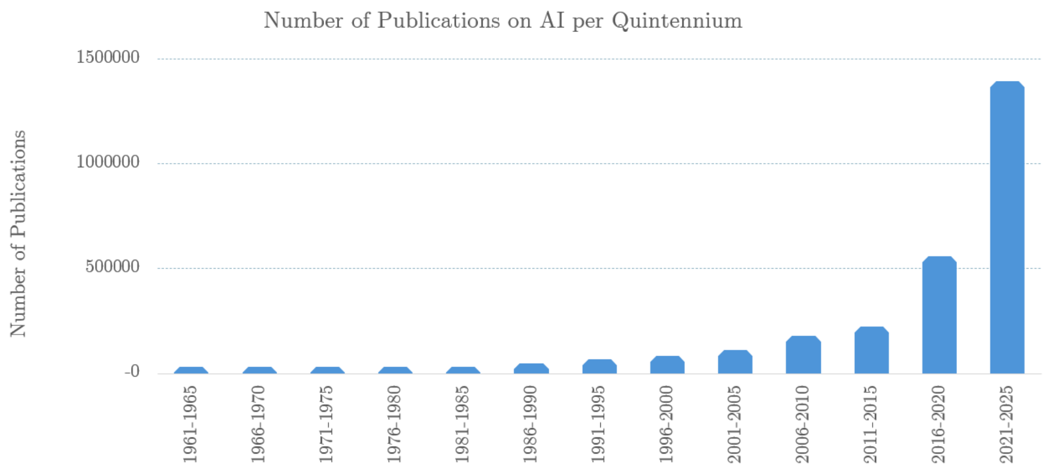

Fig. 1.1 Annual number of publications on Scopus[7] related to AI, obtained through a keyword-based search using terms associated with AI.

has not been an end in itself, but it is extending to virtually every aspect of life. An example of the wide range of domains that AI has infiltrated is the awarding of the 2024 Nobel Prizes in Physics (Geoffrey Hinton and John Hopfield) and Chemistry (David Baker, Demis Hassabis, and John M. Jumper), which included AI experts whose contributions enabled major breakthroughs in their respective fields. However, this is only a glaring example, and it is not just research that has benefited from AI. What we are experiencing is AI models moving from research labs to the market at an astonishing pace. We have become accustomed to near-daily advancements, and people's optimism towards AI is growing worldwide [4]. In just a few years, end-users have embraced concepts that were previously unthinkable, such as intelligent chatbots, personalized recommendation systems, autonomous driving, and commercially-ready robots. Not only end-users, but also the labor market has benefited from these innovations: new AI tools open up new possibilities, like smart industries, increasingly efficient automation, enhanced data analysis, and countless other innovations. According to a 2024 global survey by McKinsey & Company [5], 78% of companies are using AI for their processes with an overall clearly positive impact, although opinions on the extent of the economic benefits vary. It is undeniable, however, that AI has significantly boosted productivity and industrial efficiency [4].

Alongside these benefits, the first concerns have begun to emerge regarding the challenges posed by this profound change [6]. In accordance with the AI Incident Database[1] (AIID), AI-related incidents have steadily increased in recent years, reaching 233 in 2024, a rise of 56.4% compared to 2023. In addition, other critical issues must be addressed, such as unemployment, lack of trustworthiness, environmental impact, responsibility, bias, and privacy, all of which require a rational and systematic approach. Still, these problems seem to be frequently overlooked; in fact, concerns about AI safety are often sidelined to avoid stalling research, as major tech companies continue to push the boundaries of AI [4]. This is highlighted by the McKinsey & Company [5] survey, which shows that less than 64% of the business leaders interviewed are taking active steps to mitigate risks.

Thankfully, there are still hopeful signs. Worldwide, there is a growing commitment to a more responsible approach to AI. Dedicated research groups are working to ensure the sustainable development of AI, while institutions such as the European Union, the Organisation for Economic Co-operation and Development, the United Nations, and the African Union are taking steps to identify risks and monitor the development of AI [4]. Research publications in the field of reliable AI have increased by 28.8% in the last year [4]. Companies such as Safe Superintelligence[2], comprising some of the most highly regarded researchers in the field, explicitly aim to promote safer and more controlled AI development. They have also received huge funding to support their efforts[2]. The European Union has made significant strides with the release of the AI Act[3], the first legislative framework aiming to regulate AI and control its fast development. The practical implementation of the AI Act has only just begun, yet it must keep pace with a field evolving at extraordinary speed.

Geoffrey Hinton, Nobel Prize and Turing Award laureate, has for years been raising awareness about AI safety, advocating for a supranational agreement on controlled AI development. Unfortunately, this goal remains far from realization. To date, the analysis of associated risks is often overshadowed by the focus on AI ability to enhance many aspects of life and work, which continues to dominate both the market and research priorities. It is therefore our responsibility, as

[1] https://incidentdatabase.ai/
[2] https://ssi.inc/
[3] https://artificialintelligenceact.eu/the-act/

future AI professionals, to ensure that future AI development follows a safe path for humanity. It is up to us, as those who will lead tomorrow's development, to ensure that AI serves all of society, improving people's lives, rather than taking control, increasing inequalities, or causing conflicts.

As a result, it is essential to focus on creating AI that is both trustworthy and responsible, particularly in high-stakes and sensitive fields. With this goal in mind, the first question to ask is: *Where do the open problems with AI originate?* This question is open to a thousand answers. To us, one of the key challenges lies in the fact that the very strength of AI, namely its sole dependency on data, is perhaps also its most problematic. We explain this statement in more detail. On the one hand, the domain-agnostic nature of ML models, trained solely on data, has enabled their adoption across diverse domains and allowed a few model architectures (e.g., neural networks, support vector machines) to be applied to countless scenarios. In addition, ML has been proven to be highly effective and the statistical approach is sufficient in many fields. To be fully accurate, not only is it sufficient, but it has even enabled applications that were previously considered impractical, thanks to the lack of need for domain-specific insights. However, on the other hand, this approach comes with several intrinsic issues that are worth analyzing in detail.

- First, statistical models are built to perform well *on average*, while the model point-wise predictions can still deviate from actual outcomes, meaning that individual output may not be entirely reliable. This is a well-known issue that is sometimes consciously accepted and, at other times, can cause problems and be mistakenly overlooked. For instance, the work of Raissi et al. [7] highlighted how models can violate fundamental physical principles, such as mass conservation, despite performing reasonably well on average with the training data. Hence, it is important to recognize that every prediction is a statistical estimate inherently marked by uncertainty.

- A second relevant aspect is that the data-driven approach relies on correlations rather than causal relationships. This feature, while powerful, is also a double-edged sword. The positive side is that ML models can extract useful information from data considered unusable in traditional physics-based models. For instance, vibration data from the external casing of a mechanical machine, largely worthless for physical modeling, can be employed by an ML model, given sufficient historical data, to estimate the health of components. However, at the same time, the use of correlations is dangerous and can be misleading. The so-called *spurious correlations* may occur between variables that are completely unrelated, creating the illusion of a relationship where none truly exists. This might lead to models that incorrectly rely on seemingly high correlation values, with no guarantee that future events will continue to follow the same trend. Therefore, trusting solely in correlations should be done with caution, as demonstrated by striking cases, documented for both cautionary and entertainment purposes by websites such as Spurious Correlations[4]. For example, they report the apparent correlation between the popularity of the name Theodore in the US and fossil fuel usage in Burundi, which shows a Pearson correlation of 0.981, even though it is clear that the two are unrelated. Cases like these are manifest, but they also prompt reflection on how relying on correlations without any understanding of causality can lead to gross errors.

- A third problem is that data-driven models often struggle with extrapolation, meaning that when historical data are absent or insufficient, or when conditions change, the model performance deteriorates. A prime example of this is Google Flu Trends, which initially showed promise but underperformed when tested with new data [8]. Such cases demonstrate how even the world's top experts face challenges of this nature. Then, when using a predictive ML model, one must be aware that its validity is limited to the range covered by the training set and that applying it outside this range should be done with caution.

- A fourth intrinsic feature of the statistical approach, is that performance is highly sensitive to data quality. High-quality datasets lead to high-quality models, while poor datasets result in poor models, or as is often said: *garbage in, garbage out*. It is logical that a model trained on an inaccurate dataset will result in an inaccurate model, but it is important to note that when we talk about dataset quality, we are not only referring to accuracy or fidelity. This concept is much broader and includes aspects such as fairness, robustness, diversity, and completeness. Examples of datasets

[4] https://www.tylervigen.com/spurious-correlations

lacking fairness controls lead to biased models in criminal recidivism prediction tools [9] and biased hiring algorithms [10].

- The fifth point raised about the intrinsic nature of ML is the lack of algorithmic transparency and the responsibility for errors. These two concepts appear to be closely related, as machine choices can often seem arbitrary and difficult to justify. When the reasoning behind a model decisions is opaque, its use can generate unease and mistrust. This is the case of SyRI, a model used in the Netherlands for welfare decisions, where the reasoning behind some discriminatory outcomes was opaque, ultimately leading to its abandonment [11]. Similar concerns arise in autonomous driving incidents [12], where accountability is unclear.

From this set of issues, we can draw the following conclusions: the data-dependency of AI models results in models that only reflect the content of the training dataset, which may contain inaccuracies, the decision process is often opaque, individual outcomes can be unreliable and alignment with human ethics is not guaranteed. For all these reasons, we argue that a purely data-driven and domain-agnostic approach is no longer sufficient to ensure the development of safe and reliable AI models.

Nonetheless, it is evident that the benefits of data-driven AI are extraordinary and have opened new avenues previously considered inaccessible. Early commercial AI systems based on symbolic approaches used explicit rules, making them inherently controllable, interpretable, and transparent, but they failed to trigger a true technological revolution. A notable example is MYCIN [13], a medical tool developed in the 1970s to assist diagnosis with reasonable accuracy, which, despite its potential, remained confined to academics. The diffusion of AI only came with the advent of ML and the domain-agnostic use of data, which enabled its adoption across diverse domains.

This makes it clear that a return to less data-dependent and fully interpretable models is not the right path. We must address the weaknesses, while maintaining the current advantages. In particular, to effectively address the weaknesses of current AI systems, future advancements should maintain, at least partially, the exploitation of data, but prioritize enhancing the models' reliability and trustworthiness. In our view, these two concepts (i.e., reliability and trustworthiness) capture key areas where AI still needs to improve, as they encompass different yet complementary aspects.

Reliability focuses on technical aspects, such as robustness, physical consistency, extrapolation capability, and resilience to adversarial attacks. Trustworthiness concerns fairness, bias mitigation, transparency, and alignment with ethical principles. It emphasizes control over the moral correctness of a model, via the dataset, learning process and model architecture. These two concepts encapsulate the core areas where research and development should concentrate in order to address the present limitations of AI.

But how can these objectives be practically achieved? The previously discussed challenges posed by AI can be tackled either independently or as a whole, following multiple approaches. Currently, numerous promising research directions are being explored. We will present a few notable examples to highlight potential avenues for progress.

To improve algorithm explainability, tools such as SHAP [5] and LIME [6] have been developed to interpret black-box models without modifying their internal structure. They are used to improve the understanding of models' decisions in a way that can be easily applied to existing systems, making them particularly versatile. This approach has already reached a sufficient level of maturity and has found success in real-world applications where interpretability is crucial, such as in the medical field [14, 15].

Another line of research focuses on robustness and safety. Notable studies are devoted to enhancing the adversarial resistance of ML models [16]. Other approaches introduce human-in-the-loop mechanisms to provide informed control over decisions [17], and fallback strategies to ensure safety in critical applications [18].

Significant research also addresses bias detection and mitigation [19, 20], and ethical alignment via techniques like reinforcement learning with human feedback [21]. Such techniques help build confidence in the models, encouraging their practical use.

[5] https://github.com/shap/shap
[6] https://github.com/marcotcr/lime

Another field, known as Knowledge-Informed ML, has rapidly captured the attention of the industry. It incorporates domain knowledge into ML models, aiming to tackle a broader set of their intrinsic issues; in fact, these techniques can increase accuracy, robustness, and consistency, while also improving interpretability of the model's inference mechanisms. Due to its potential, Knowledge-Informed ML is gaining traction in industry, but, while promising, it presents unresolved challenges, largely due to its recent introduction. First, Knowledge-Informed Models remain complex and resource-intensive. Their development demands both domain-specific expertise and ML proficiency, namely skills that are rarely found in a single professional profile. Moreover, building a Knowledge-Informed Model remains a highly application-specific task, and acquiring and encoding domain knowledge, as well as selecting the implementation strategy, is both non-trivial and costly. This limits the widespread adoption of Knowledge-Informed ML, unlike the more flexible, purely data-driven ML techniques.

However, despite all these challenges, Knowledge-Informed Models are seeing growing adoption in the industry and research, thanks to their potential to overcome the limitations of ML. As shown in Figure 1.2, according to Scopus[7] the number of publications related to knowledge-informed machine learning has been steadily increasing for years, with annual growth rates consistently above 36% since 2019.

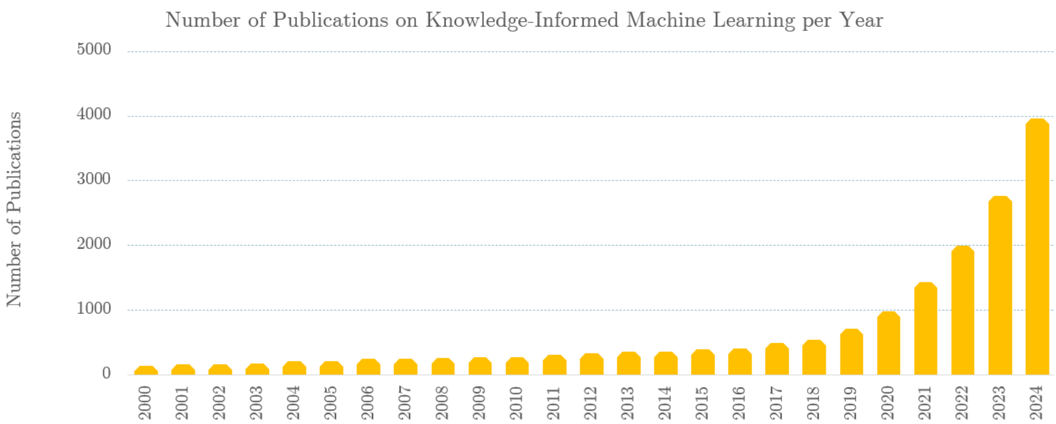

Fig. 1.2 Annual number of publications on Scopus[7] related to knowledge-informed machine learning, obtained through a keyword-based search using terms associated with knowledge-informed ML.

[7] https://www.scopus.com/

When compared to the overall annual growth in AI-related publications (Figure 1.3), it becomes clear that knowledge-informed ML is among the fastest-growing areas in the field of AI. Knowledge-Informed ML is increasingly

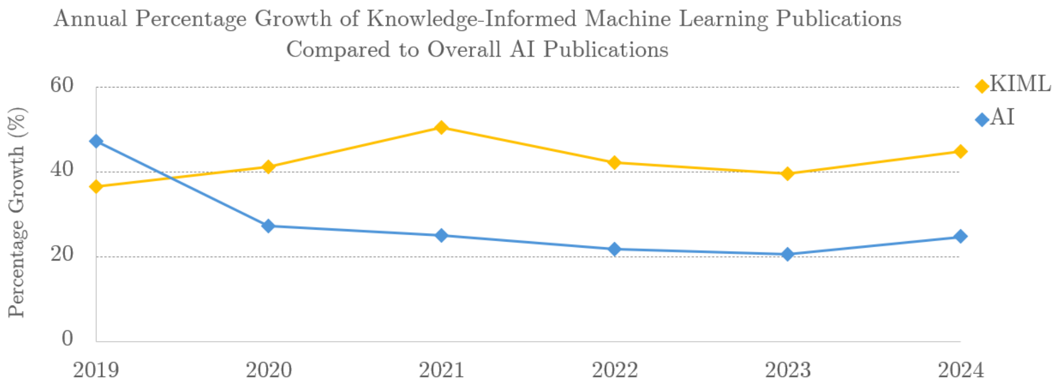

Fig. 1.3 Annual increase on Scopus[7] in publications related to knowledge-informed machine learning and general AI, obtained through a keyword-based search using terms associated with knowledge-informed ML and AI, respectively.

being applied in industry, as it offers effective solutions to the limitations of purely data-driven approaches-limitations that are particularly critical in many industrial contexts. The substantial investments industry [4] is making in AI are also driving the adoption of Knowledge-Informed ML, making it an especially attractive area of research.

We expect that the diffusion of Knowledge-Informed Models will be particularly evident in high-stakes sectors, where error tolerance is minimal or where decisions must follow strict, logical, and transparent chains of reasoning. Practical examples include deploying AI for operational control in nuclear plants, real-world applications involving collaborative robots, or domains where low-risk thresholds are mandated by regulation. In these fields, the diffusion of AI has been slowed by safety-critical concerns, and knowledge-informed approaches may offer a viable solution to overcome these limitations.

This is the context in which the present doctorate program is situated. As part of the National Ph.D. in AI, the funding for this Ph.D. was intended to develop *Trustworthy and Reliable AI in industry applications*. Various research paths could be taken to address this theme. All the motivations discussed earlier, combined with the availability of multidisciplinary expertise in physics, mechanics, and ML, directed the choice towards Knowledge-Informed ML as the core argument of this doctorate. We consider this approach as one of the most promising path toward achieving trustworthy and reliable AI.

Another factor that contributed to this choice is the vast, unexplored territory within this subject. This topic is still recent and largely uninvestigated, making it well-suited for research. The history of Knowledge-Informed ML is, indeed, relatively young. Broadly speaking, the first ideas of combining physics-based and statistical models date back to the 1990s, but their early development has been intermittent and slow. Initial implementations were simple combinations of models' outputs. Gradually over time, integration has become deeper, with physical constraints being incorporated directly into model architectures, with the first real wave of adoption emerging in the late 2010s and early 2020s. However, a solid and well-defined research field and a unified framework are still lacking. In fact, during the preliminary stages of this research, we realized that the current state of Knowledge-Informed ML is fragmented. Most models are highly application-specific, and comprehensive studies and classifications of their practical applications are missing in the literature. Existing surveys are typically domain-specific, and no common ground has been found so far. There have been attempts to develop a unified framework applicable to different domains, though they have been confined to specific knowledge-informed techniques. For example, Raissi et al. [22] proposed the first general framework for integrating physical laws into the training process of ML models, ushering in a new wave of research. This represented only an initial step, and substantial progress is still needed. We argue that, in order to foster wider adoption, models should be as general as possible and less reliant on specific domains, but rather to broad classes of tasks. The various knowledge-informed techniques must be encoded in a single framework, within which it is possible to leverage, along with data, different types of knowledge and handle various sets of problems.

To bridge the current gaps in the literature and lay the foundation for concrete research, the first part of this Ph.D. has been devoted to defining the state of the art, proposing a systematic classification, and abstracting the fundamental concepts of Knowledge-Informed ML. This foundational work is necessary for applying general principles deductively to specific applications.

Building on these foundations, the second part of this Ph.D. focuses on practical applications, both to test novel techniques and to apply existing methods to new industrial domains. As will be detailed later, during this Ph.D. program, Knowledge-Informed ML techniques have been explored under both predictive and generative paradigms, across diverse sectors such as energy, marine, chemical, and materials industries. This manuscript provides two representative examples of these applications. Finally, we attempted to outline what we believe to be the future perspectives and the open challenges that still remain in this field.

1.1 Key objectives and contributions

To provide clarity for the reader, we describe in more detail the objectives and contributions of this doctoral research. The core focus of this Ph.D. is on advancing the field of Knowledge-Informed ML. Two main goals guided this research:

- Define the state of the art: establish a clear understanding of the current state of Knowledge-Informed ML, identifying areas where it has already been applied and evaluating the outcomes achieved.
- Promote industrial adoption: drive the advancement of knowledge-informed techniques in industrial applications, with a focus on both predictive and generative AI models.

With these goals in mind, the research developed over the course of the Ph.D. has led to several concrete outcomes, which are summarized below as key contributions to the advancement of Knowledge-Informed ML:

- Review of techniques used in Knowledge-Informed ML: a comprehensive synthesis of existing literature to establish the state of the art and identify research gaps in this line of research.
- Introduce a formal analysis of traditional and Knowledge-Informed Predictive Models, generalizing their underlying concepts and classifying the various techniques found in current research.

- Review of industrial applications of Knowledge-Informed Predictive Models: an in-depth analysis of how predictive models have been applied in industrial contexts, highlighting key achievements and limitations.
- Exploration of new applications of Knowledge-Informed Predictive Models: development of novel techniques and integration of existing ones for applications in previously unexplored domains, specifically in the energy and maritime industries.
- Exploration of new applications of Knowledge-Informed Generative Models: investigation of case studies where generative models enhanced with domain knowledge provide innovative solutions, with specific applications in the materials and chemical industries.
- Laying the foundations for future work: offering insights and recommendations to support the continued development and industrial adoption of Knowledge-Informed AI.

The main contributions of this research are summarized in the following publications: [23] concerning the review of existing techniques, [24] regarding the development of Knowledge-Informed Predictive Models, and [25] focusing on Knowledge-Informed Generative Models. To support navigation, the next section outlines how these topics are structured in the manuscript.

1.2 Structure of the thesis

This thesis is structured to provide a comprehensive overview of research conducted during the doctoral program. The chapters are organized as follows:

- Chapter 2: this chapter introduces the state of the art of Knowledge-Informed ML, particularly focusing on predictive modeling. First, we synthesize existing reviews in the field and provide a formal analysis of established techniques and an illustrative example. Then, we present a detailed literature review of applications across various industrial sectors. This chapter mainly refers to the content of [23].
- Chapter 3: this chapter presents the application of a Knowledge-Informed Predictive Model to a case study in the energy industry, specifically focusing on the development of a physics-informed surrogate model for power

flow analysis, which is compared with traditional predictive methods. The contributions presented in this chapter have been published in [24].

- Chapter 4: this chapter explores the emerging potential of Knowledge-Informed Generative Models and illustrates it through a case study in material science, namely, the inverse design of mechanical metamaterials. The content of this chapter has been presented in [25].
- Chapter 5: there, we bring together the key insights from the previous chapters, offering a broader analysis of the contributions and outlining possible directions for future research in Knowledge-Informed ML.
- Chapter 6: this chapter summarizes the activities carried out during the Ph.D., such as the projects undertaken, publications, international activities, and teaching assistance.
- Chapter 7: the final chapter summarizes this manuscript and presents the concluding remarks.

Figure 1.4 presents a graphical index to provide a clearer understanding of the structure of this manuscript.

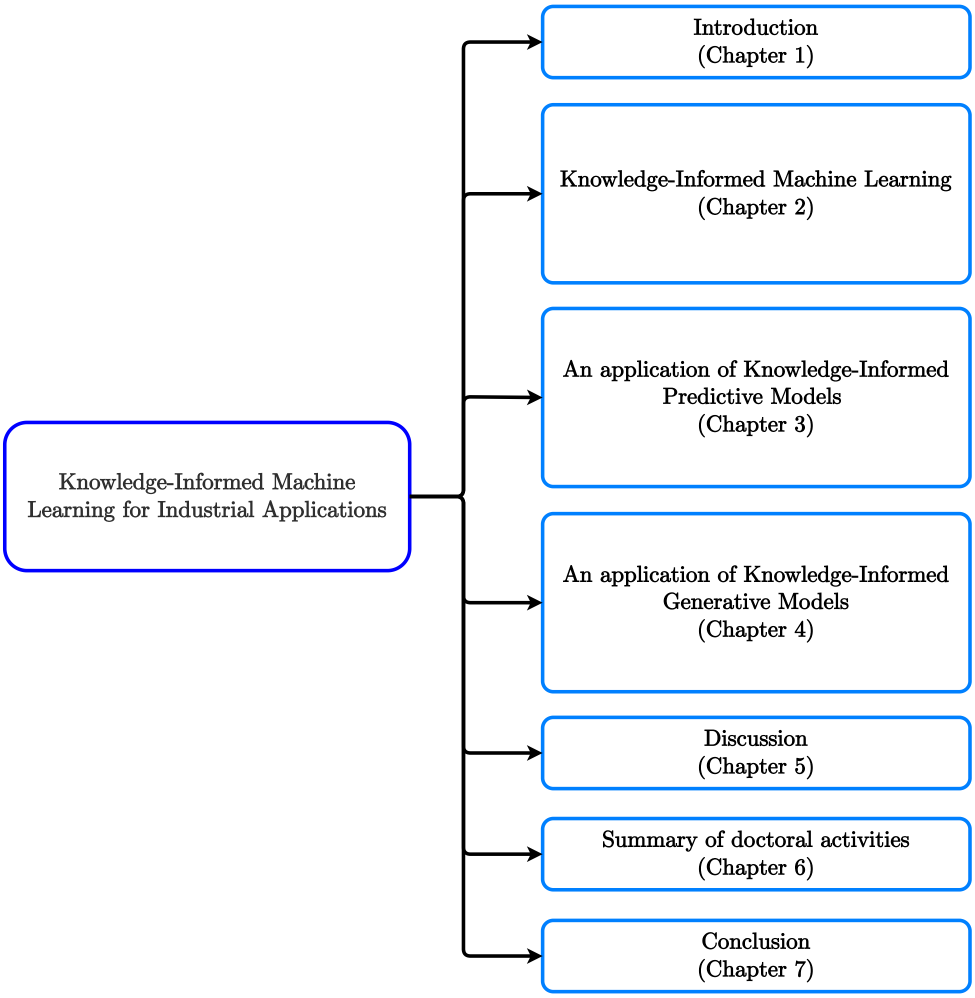

Fig. 1.4 Graphical index

Chapter 2

Knowledge-Informed Machine Learning

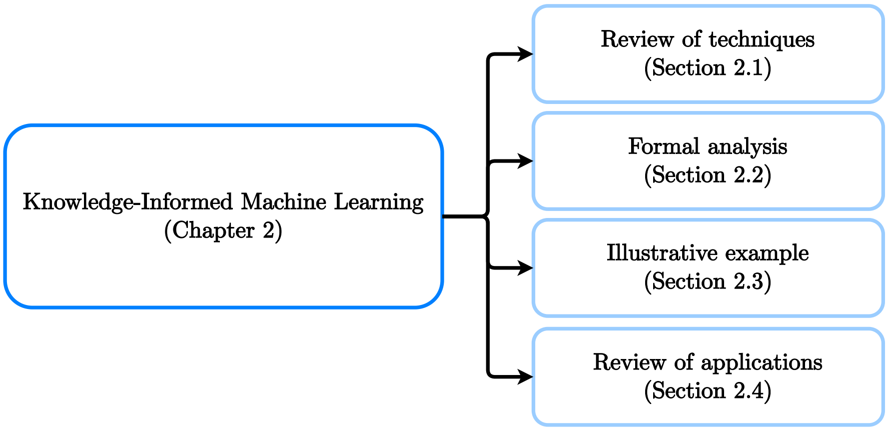

Fig. 2.1 Graphical index of Chapter 2

Knowledge-Informed ML encompasses an entire class of methods aimed at enriching ML models with various forms of domain knowledge. These approaches have proven particularly effective in improving several key aspects—such as robustness, physical consistency, and interpretability—and are therefore attracting increasing attention.

When we first approached the field of Knowledge-Informed ML, it immediately became clear that there was a lack of a solid and universally accepted foundation upon which to develop new applications and extend its scope. Our literature review revealed that, although many studies have moved in this direction, the efforts have been fragmented and varied. Hundreds of practical applications have enhanced data-driven models by incorporating knowledge, yet they often follow disparate frameworks with little convergence. A few studies have attempted to bring order to the field by proposing taxonomies or organizing principles. However, none of these works offer a comprehensive view: they often focus on a single domain, omit certain model types, or have become outdated due to recent advances. Therefore, the first objective of this doctoral research was to establish a coherent framework for understanding the current state of Knowledge-Informed ML research, with particular attention given to industry.

From our initial exploratory analysis, it emerged that the existing body of literature on Knowledge-Informed ML is focused almost exclusively on predictive models. This focus stems from the widespread industrial use of predictive modeling, which has historically relied on either domain knowledge or statistical data. Both approaches present significant limitations, and the industry's need to improve predictive performance has fueled the expansion of Knowledge-Informed ML in this area far more than in others, such as generative modeling. For this reason, the remainder of this chapter will focus exclusively on Knowledge-Informed Predictive Models. Nonetheless, a small but emerging line of research exists on integrating domain knowledge into generative models. Although still in its infancy and too limited to justify a full review, notes on the existing work in this area can be found in Chapter 4.

Once we defined the scope of our research, we began by identifying and analyzing the techniques proposed in the literature. This included all relevant surveys and theoretical papers that abstract and classify Knowledge-Informed ML approaches, grouping them by shared characteristics. From the literature, we were able to extract common concepts and outline a formal analysis to organize the wide array of existing predictive models. Following a common approach in the literature, our presentation compares Knowledge-Informed Models with traditional modeling paradigms (i.e., based purely on domain knowledge and based purely on data). Framing the models in this way helps to highlight their hybrid nature, which lies between first-principles approaches and statistical techniques.

Although most of the literature categorizes models into knowledge-based, data-driven, and hybrid types, a variety of terminologies are used. It is therefore helpful to clarify both the terminology found in the literature and the terminology adopted in this manuscript, through Table 2.1

Knowledge-Based Models	**Data-driven Models**	**Knowledge-Informed Models**
Full-Knowledge Models	Zero-Knowledge Models	Partial-Knowledge Models
Physical Models	Statistical Models	Physics-Informed Models
White-Box Models	ML models	Hybrid Models
Transparent Models	Black-Box Models	Theory-Guided Models
Theoretical Models		Gray-Box Models
First-Principle Models		

Table 2.1 Terminology of traditional and Knowledge-Informed Models

This chapter compares existing techniques, then for the sake of clarity, we will primarily use the terms Full-Knowledge Predictive Models, Zero-Knowledge Predictive Models, and Partial-Knowledge Predictive Models. In the remainder of the manuscript, we will adopt the more intuitive terms: Knowledge-Based, Data-Driven, and Knowledge-Informed Models.

Following the formal analysis, we present an illustrative example to clarify the concepts, using the classical engineering problem of the Mass-Spring-Damper system, approached through both traditional and knowledge-informed modeling methods.

Finally, to complete the state of the art overview, we review the industrial applications of Knowledge-Informed Models. This review gathers the main use cases across different sectors, aiming to identify the key research trends within each domain.

The content presented in this chapter reflects the research published in [23]. The structure of this chapter is summarized graphically in Figure 1, and the content of the sections is as follows:

- Section 2.1 examines existing survey papers on knowledge-informed predictive techniques, synthesizing insights from the literature to provide a high-level overview of these models. This review extracts general concepts of traditional and knowledge-informed approach, introducing a preliminary classification into three categories based on the degree of integration of domain knowledge: Full-Knowledge Predictive Models (FKPMs), Zero-Knowledge Predictive Models (ZKPMs), and Partial-Knowledge Predictive Models (PKPMs). The first rely solely on knowledge, the second solely on data, and the third, coinciding with the knowledge-informed techniques, leverage both knowledge and data

- Section 2.2 builds on this foundation with a formal analysis of the three identified categories. Each class of models is examined in detail, focusing on their defining characteristics, methodologies, and distinguishing features. Particular attention is given to PKPMs, as they constitute the core focus of this research. As will be detailed, PKPMs differ in how domain knowledge is utilized, resulting in a further subdivision into three integration strategies: Pre-Processing, In-Processing, and Post-Processing.
- Section 2.3 presents an illustrative example. We compare the three different kinds of predictive model to a Mass-Spring-Damper problem, to put in evidence the characteristics of each design choice. This section demonstrates the concrete advantages of a PKPM, as well as the major efforts its development requires.
- Section 2.4 reviews practical applications of predictive models in industry across various sectors, namely extraction, chemical, manufacturing, transportation, energy, and construction. Each sector is analyzed separately to illustrate how these models address its specific challenges.

2.1 Review of techniques

In the existing literature, various reviews partially address the topic of models combining physics and statistical modeling. This section examines these reviews to help readers trace relevant studies and gain a clearer understanding of the subject. It is important to note that the presented research primarily focuses on PKPMs, but since the literature often discusses them in relation to traditional FKPMs and ZKPMs, we also included the latest developments in these approaches.

The study of the current work began with the extraction of all papers complying with specific criteria from the scientific database Scopus[1]. Regarding the time window, we selected a timespan based on insights from the literature. According to Xu et al. [26], Physics-Informed Regularization, representing the most recent wave of advancements in predictive modeling, emerged around 2016. The early literature on Knowledge-Informed Predictive Models, up to 2016, has been comprehensively reviewed in the survey by Rai and Sahu [27]. For these reasons, our analysis started from 2013, to also capture precursor works that may have been overlooked in [27].

For what concerns the keywords to be searched, we applied the following criteria:

- Papers containing keywords related to predictive modeling (see [23] for more information) in their title, abstract, or keywords.
- Papers containing the words “review,” “overview,” or “survey” in their title, abstract, or keywords.

The initial search returned over 500 papers. To refine the selection, additional criteria were applied:

- Papers published from 2013 to 2021 were included if cited ≥30 times.
- Papers from 2022 if cited ≥20 times.
- Papers from 2023 if cited ≥10 times.
- Papers from 2024 if cited ≥1 time.

The list also included papers cited in relevant works identified during the review. This selection was narrowed down through an individual examination of each paper, yielding the final list summarized in Table 2.2, where we report: the work reference, year of publication, time span covered by the work, analyzed FKPMs, ZKPMs, and PKPMs, presence of a formal analysis, presence of illustrative examples, and domain of application.

From the analysis summarized in Table 2.2, it is possible to derive some observations about the work already conducted on this subject. All reviews, despite using different terminologies, agree on categorizing predictive models into three main types: FKPMs, ZKPMs, and PKPMs. Every review defines FKPMs as methods that infer first-principles-based relationships between

[1] https://www.scopus.com/

Review	Year	Timespan	FK	ZK	PK	Formal Analysis	Practical Example	Domain
[29]	2013	Unspecified	Computational Fluid Dynamics, Zonal approach, Nodal approach	Multiple Linear Regression, Genetic Algorithm, Neural network, Support vector machine	Hybrid models	Yes	No	Buildings' Energy
[32]	2014	1992-2013	Linear/Non-linear Ordinary/Partial differential equations, Linear/Non-linear algebraic equations	Neural networks, Partial least squares, Splines, Kernels	Serial, Parallel	Partial (PK)	No	Chemical, Biochemical
[33]	2014	Unspecified	White-Box	Black-Box	Gray-Box	No	No	Buildings' Energy
[34]	2016	2000-2016	White-Box	Black-Box	Gray-Box, Hybrid models	No	No	Vehicle Fuel Consumption
[35]	2017	Unspecified	No	No	Theory-guided design, Theory-guided learning, Theory-guided refinement, Learning hybrid models of theory and data-science, Augmenting theory-based models	Partial (PK)	No	Science
[30]	2018	Unspecified	White-Box	Black-Box	Gray-Box (Parallel, Serial, Mixed parallel/serial)	Yes	No	Chemical, Petroleum, Energy
[36]	2018	2007-2017	White-Box	Black-Box	Gray-Box	No	No	Buildings' Energy
[27]	2020	Unspecified	Physics equation based CPS, State Machine Based Cyber-Physical System	Data-driven CPS	Physics-based Pre-Processing, Physics-based Network Architectures, Physics-based Regularization, Miscellaneous	Yes	No	General
[37]	2020	Unspecified	Simulation	ML	Simulation-assisted ML, ML assisted simulation	No	No	General
[38]	2020	Unspecified	No	No	Serial, Parallel	No	No	Chemical
[39]	2020	Unspecified	White-Box	Black-Box (Neural networks, Support vector machine, Random forest, Gradient boosting, Multiple linear regression)	Grey-Box	No	No	Buildings' Energy
[40]	2020	Unspecified	No	No	Physics-guided loss function, Physics-guided initialization, Physics-guided design of architecture, Residual modeling, Hybrid physics-ML models	No	No	General
[41]	2021	Unspecified	No	No	Grey-Box	Yes	No	Buildings' Energy

[42]	2021	1992-2020	Mechanistic models	Data-driven models (Neural networks, Support vector machine, Multivariate Adaptive Regression Splines, Latent Variable Models)	Serial, Parallel, Surrogate, Alternative models	No	No	Chemical
[43]	2021	Unspecified	No	No	Knowledge source, Knowledge representation, Knowledge integration way (training data, hypothesis set, learning algorithm, final hypothesis)	Partial (PK)	No	General
[44]	2021	Unspecified	No	No	Observational bias, Inductive bias, Learning bias	Partial (PK)	Yes	General
[45]	2021	Unspecified	White-Box	Black-Box	Grey-Box	No	No	Batteries
[46]	2022	Unspecified	No	No	Mechanistic feature processing, Physics-informed model development, Data-driven discovery	Partial (PK)	No	Manufacturing
[47]	2022	2014-2021	No	No	Data, cost function, initialization, run time, architecture	No	No	Civil Engineering
[48]	2022	Unspecified	No	No	ML compliments science (Inverse models, direct hybrid models, reduced order models, uncertainty quantification, discover laws), Science compliments ML (design, learning, refinement)	No	No	Chemical
[49]	2022	Unspecified	No	No	Physics-informed loss function, Physics-informed initialization, Physics-informed design architecture, Hybrid physics-DL models	Partial (PK)	No	Power Systems
[28]	2022	Unspecified	Physics-based models	Data-driven models	Physics-informed ML (data, modeling, loss function), ML assisted simulation, Explainable Artificial Intelligence	Yes	No	Manufacturing
[50]	2022	Unspecified	No	Gaussian Process, Neural networks	Hybrid sub-modeling, Physics-informed ML, Model calibration	Partial (ZK, PK)	Yes	General
[51]	2022	2018-2022	No	No	Physics-informed neural networks	Partial (PK)	No	Science
[52]	2022	2001-2021	White-Box	Black-Box (Statistical and ML models)	Grey-Box (Black-Box based on White-Box and White-Box based on Black-Box)	No	No	Ship Fuel Consumption
[53]	2022	Unspecified	No	No	Physics-informed data, architecture, loss function, optimization, inference	Partial (PK)	No	General

[31]	2022	2002-2022	Physics-based (Electrochemical, Equivalent circuit, Semi-empirical)	No	Grey-Box (Data-driven assisted physical models, Physics-guided data-driven)	Partial (FK, PK)	No	Batteries
[54]	2022	Unspecified	No	No	Physics-guided loss function, Physics-guided initialization, Physics-guided design of architecture, Hybrid physics-ML models	Partial (PK)	No	Engineering, Environmental Systems
[26]	2023	2016-2022	No	No	Physics-informed loss function, Physics-informed architecture	Partial (PK)	No	General

Table 2.2 Related works in chronological order

physical variables [28]. FKPMs can be based on physics equations (such as algebraic equations, ordinary differential equations, or partial differential equations) or State-Machines (Mealy or Moore machines) [27]. Comprehensive analyses of FKPMs compared to PKPMs exist in specific works [27], while others focus on domain-specific applications [28–31].

Despite the differing descriptions of ZKPMs in the literature, they are widely recognized as *models relying entirely on data* [27]. These models include a variety of ML techniques that fall under different classification schemes, such as shallow and deep learning models, frequentist and Bayesian approaches, and classification or regression frameworks [23]. In the literature, it is possible to find general reviews comparing ZKPMs and PKPMs [27], as well as surveys focused on specific domains [28–30, 50].

In contrast to the previous ones, the concept of PKPMs is addressed quite differently across various reviews, reflecting the specific focus areas and the PKPM categories considered. Generally speaking, there is consensus that PKPMs are models that combine FKPMs and ZKPMs, even though there is no comprehensive analysis of their methods and applications. What can be noticed is a clear temporal evolution of PKPMs. In fact, early integrations were often limited to simple combinations of inputs and outputs. There is a tendency in the literature to reserve the terms *Hybrid Models* or *Gray-Box Models* for these earlier approaches. Over time, however, they have evolved into more sophisticated and tightly coupled frameworks, in which internal structures of traditional models are manipulated according to specific needs.

This review has allowed us to reconstruct the fragmented history of PKPMs. The very first idea of merging FKPMs and ZKPMs dates back to the 1990s. Zendehboudi et al. [30] assert that Joerding and Meador [55] were the first to propose leveraging a priori knowledge in the implementation of neural networks, and the work of Psichogios and Ungar [56] is the very first application of Hybrid Models. According to Von Stosch et al. [32], Agarwal [57] did the very first theoretical study on Hybrid Models. During the early 2010s, precursor works, done by Foucquier et al. [29] and Von Stosch et al. [32], reviewed the parallel and serial paradigms of hybrid modeling. Recent advancements have introduced concepts such as *Theory-Guided Data Science* [35] where scientific theories are seamlessly integrated into data-driven models [26]. It is generally accepted [44, 43, 51, 53, 46, 26] that the last breakthrough in the field of PKPMs is the work of Raissi et al. [22, 58, 7], that proposed the adoption of modified loss functions in neural networks (ZKPMs) to incorporate differential equations (FKPMs) during the training phase.

The very first review of PKPMs was introduced by von Rueden et al. [43]. The authors defined a PKPM as the combination of a FKPM (i.e., a domain knowledge source) and a technique of integrating it inside the ZKPM (i.e., a data-driven model). The successive work of Karniadakis et al. [44] remains the most popular review on PKPMs, where the authors described three approaches for informing a ZKPM using a FKPM: observational, learning, and inductive biases. Other significant contributions include the survey by Rai and Sahu [27], which explores the development of Cyber-Physical Systems using FKPMs, ZKPMs, and PKPMs, and Wang et al. [28], which examines recent trends in hybrid modeling for Smart Manufacturing.

Despite existing research, several gaps persist, such as the lack of comprehensive classifications and unified taxonomies, and the absence of generalized concepts, which remain fragmented and tailored to specific application fields. Hence, during the first part of the doctoral work, these gaps were addressed by developing a unified taxonomy and providing straightforward examples to highlight the advantages and disadvantages of each model type.

2.2 Formal analysis

After reviewing the surveys on the recent advancements of predictive models, we can generalize the concepts found in the literature and established industrial applications. This effort aims to define a formal analysis capable of abstracting and systematizing the distinctive features of FKPMs, ZKPMs, and PKPMs. In particular, we refer to formal analysis as the systematic examination of the model logical structure to understand its behavior under various conditions, identify assumptions, limitations, and biases, and to select optimal parameters and structures. The discussion begins with a definition of the general concepts underlying predictive models, and then introduces a formal analysis of the three approaches.

2.2.1 Predictive models

Let us consider a physical system $\mathtt{S}$ that, given an input X from an input space $\mathcal{X}$, produces an output Y in the output space $\mathcal{Y}$. A *predictive model* $\mathtt{M}$ is an approximation of $\mathtt{S}$ that, provided with only X, can produce $\tilde{Y} \in \mathcal{Y}$ such that $Y \approx \tilde{Y}$ according to a chosen metric $\mathscr{M}$. This concept is illustrated in Figure 2.2.

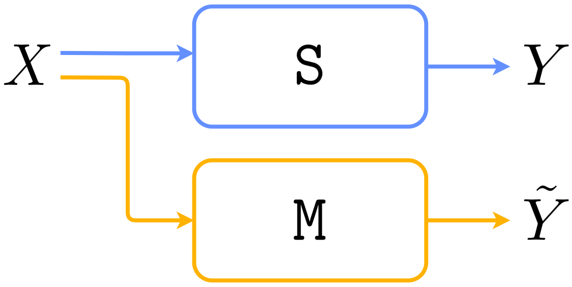

Fig. 2.2 Predictive model

The spaces $\mathcal{X}$ and $\mathcal{Y}$ can take various forms, such as real-valued vectors, graphs, trees, images, natural language, or mixtures of these. The system $\mathtt{S}$ may be deterministic or non-deterministic. It is worth noting that it is not strictly required for a causal link to exist between X and Y, as correlation alone can be sufficient. Typically, X contains elements that are relatively easy or inexpensive to measure, while Y often includes quantities that are difficult, costly, or impossible to obtain. In these scenarios, the construction of $\mathtt{M}$ provides estimates of Y that would not be possible otherwise. Strictly speaking, $\mathtt{M}$ is defined as a model f, chosen from a set of possible models $\mathcal{F}$ and depending on a set of parameters θ. To construct $\mathtt{M}$, one can resort to FKPMs, ZKPMs, and PKPMs, depending on the available information sources.

Information sources for building predictive models

In order to design and build $\mathtt{M}$, different sources of information may be available. Four primary sources can be identified:

- Domain knowledge: Pre-existing understanding of the underlying principles governing $\mathtt{S}$. This may include physical laws providing partial or complete descriptions of the relationships between inputs and outputs, invariances in the input-output relationship under certain conditions, or constraints on the outputs.

- Measurements: These refer to direct data acquisition related to the system characteristics. For example, one can measure the actual weight of a body or other physical quantities. Such empirical data can be critical for the construction of FKPMs, as they require precise information about $\mathtt{S}$. It is important to note that, in this manuscript, measurements do not correspond to the usual input/output pairs used in datasets (which are discussed below), but rather to intrinsic parameters or physical quantities of the system that can be directly observed or measured. This conceptual distinction between measurements and examples will be further clarified in Section 2.3 through an example.
- Examples of input/output relations (i.e., datasets): Typically required by ZKPMs, datasets $\mathcal{D}_n$ containing pairs (X_i, Y_i) help infer the relationship between inputs and outputs without relying on domain knowledge. If outputs are unavailable, one may resort to unlabeled datasets $\mathcal{U}_m = \{X_1, \dots, X_m\}$, which can still hold value for certain tasks.
- Experience: This source is the most challenging to formalize. Experience informs many decisions: in FKPMs, it may guide the simplification of certain effects; in ZKPMs, it might influence the choice of learning algorithms or model architectures; and in PKPMs, it can guide the blending of FKPMs and ZKPMs. Unlike examples, which provide explicit data points from which models can learn, experience embodies implicit knowledge accumulated through practice or prior exposure, often without being directly encoded in datasets. Essentially, experience reduces the degree of freedom in the decision-making process by leveraging prior know-how [43].

These sources can be present simultaneously or not, and their availability determines whether it is possible to construct FKPMs, ZKPMs, or PKPMs.

2.2.2 Full-Knowledge Predictive Models

FKPMs are only based on first principles and have historically been the most common approach before the digital era. They generally require all four types of information source [59]. Figure 2.3 presents an abstraction of the FKPMs' framework, highlighting the information sources involved in the construction of f.

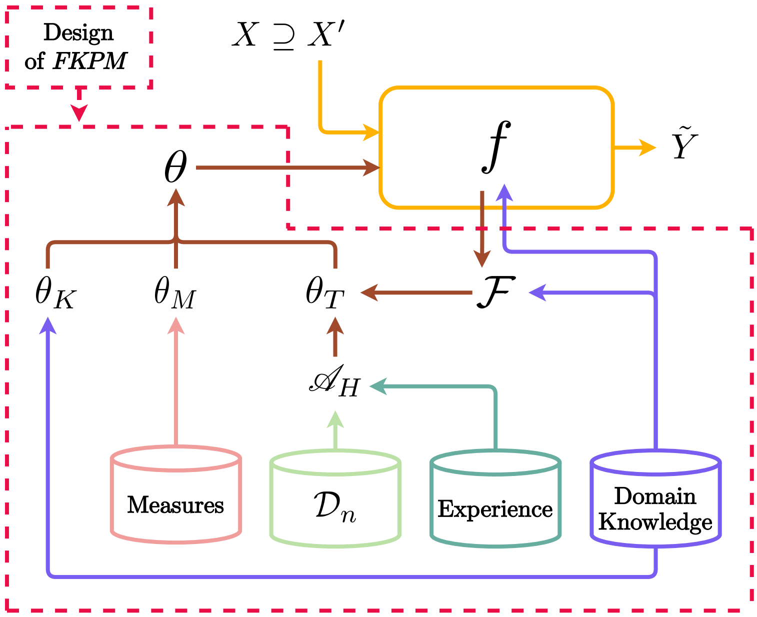

Fig. 2.3 FKPMs

The structure of FKPMs features the following key aspects:

- The construction of f typically uses only a subset $X' \subseteq X$ of the available dataset. This is because modeling relations between Y and some data in X may be too complex, unnecessary, or negligible.
- The functional form of f (e.g., linear or differential equations) results from the domain knowledge. The simplest systems involve formulas that can be solved in closed form, whereas more complex ones require solving sets of equations iteratively. In other cases, the function f can be progressively constructed using a series of interconnected blocks, providing the model with a hierarchical structure and enhancing its interpretability.
- The parameters θ of f can be classified into θ_K, θ_M, and θ_D. Parameters θ_K are set by domain knowledge (e.g., physical constants), θ_M come from measurements (e.g., the mass of a machine), and θ_D are tuned by an algorithm that exploits a dataset $\mathcal{D}_n$, the domain knowledge, and experience.

Practically speaking, defining the functional form of f in a FKPMs can take a long time and may be heavily application-dependent [59]. Moreover, the tuning of the parameters θ_D requires special attention. These parameters can be found via grid search or using an optimization problem, that can be convex or non-convex. In case of optimization, the optimizer features specific hyperparameters that can be tuned using experience, coupled with grid search algorithms [60]. In general, the best parameters θ_D^* minimize the prediction error on $\mathcal{D}_n$:

$$\theta_D^* = \arg\min_{\theta_D \in \mathcal{S}} \mathtt{E}(\theta_D, \mathcal{D}_n) \tag{2.1}$$

where $\mathcal{S}$ denotes the search space (induced by domain knowledge or experience), and $\mathtt{E}(\theta, \mathcal{D}_n)$ represents the error, measured according to a specific metric and determined by the particular choice of θ_D over the dataset $\mathcal{D}_n$.

Depending on the application, the required level of accuracy, the available input data, the extent of prior research on $\mathtt{S}$, and the computational resources, the range of possible FKPMs is extremely broad [59]. It includes all physical laws, differential equations solved either in closed form or through numerical methods, and multiphysics models. It is evident that each individual FKPM is closely dependent on the specific application. Unlike ZKPMs, it is difficult to identify a common model schema across FKPMs, as their variability is significantly greater and does not allow for a uniform framework applicable to all, or even to most, cases.

General advantages of FKPMs include:

- They need relatively few data points [42].
- They perform reasonably well in interpolation and provide physically plausible predictions in extrapolation [61].
- They are explainable and interpretable by construction [62].
- They can sometimes be computationally efficient when used for predictions [29].

Disadvantages of FKPMs include:

- They require extensive human effort and time to develop [59].
- They may not fully exploit all available data [63].
- Their accuracy may be limited in modeling highly complex phenomena [29].

- In some cases, making predictions can be computationally expensive (e.g., solving partial differential equations) [64].

2.2.3 Zero-Knowledge Predictive Models

ZKPMs require only datasets and experience [65], they do not rely on domain knowledge or direct measurements, and are typically application-independent [65]. ZKPMs became popular with the advent of big data, powerful computing resources, and ML [42]. The general schema of a ZKPM is illustrated in Figure 2.4.

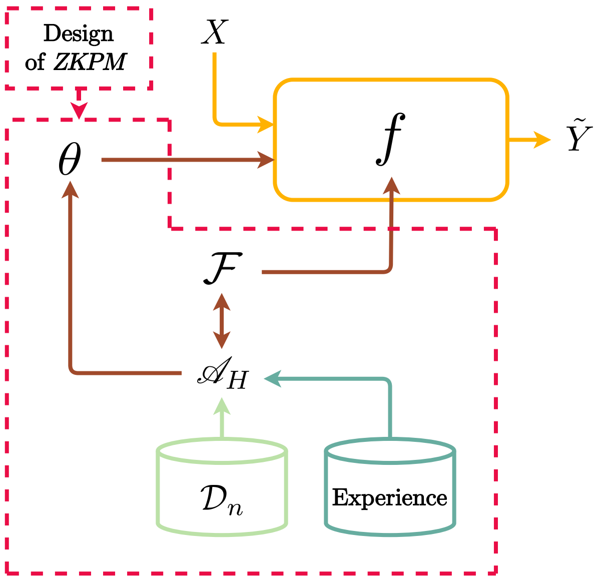

Fig. 2.4 ZKPMs

We can summarize the aspects of ZKPMs as:

- The model f typically uses all available inputs X. Several input sources can be leveraged, such as scalars, vectors, time series, images, graphs, and natural language. It should be noted that ZKPMs do not discriminate between causality and correlation, rising the risk of leveraging spurious correlations.
- The functional form of f depends on the selected ML algorithm. Many different ML algorithms and families of algorithms exist in the literature [66].

The most common families[2] are Kernel methods [66], Ensemble methods [67], Neural networks [68],and Bayesian approaches [69]. Typically the functional form of f of ZKPMs is monolithic with no intermediate blocks between inputs and outputs. This reduces the model interpretability.

- The model f depends on hyperparameters and parameters. The hyperparameters regulate the model architecture (e.g., linear, non-linear, kerneled, convolutive, attention, and recursive) and are intrinsically more influential on the model performance. Given the hyperparameters, the parameters tune the model to the specific data. Since no domain knowledge is available, the selection of hyperparameters and parameters, as well as the quality assessment relies only on data. The standard solution involves assuming that the dataset $\mathcal{D}_n$ is independent and identically distributed (i.i.d.) and splitting it (one or multiple times) into three subsets: training $\mathcal{L}$, validation $\mathcal{V}$, and test $\mathcal{T}$. $\mathcal{L}$ is used to optimize the parameters θ that define f through an algorithm $\mathscr{A}_H$, which is characterized by its hyperparameters H. Optimization problems are often convex, allowing efficient identification of the optimal point, or at least differentiable, enabling the use of gradient descent to find local minima efficiently. Note that there always exist combinations of $\mathscr{A}$ and H that achieve an arbitrarily low error on $\mathcal{L}$, which is undesirable because such models tend to memorize $\mathcal{L}$ (over-fitting) rather than generalizing to unseen data. Then, $\mathcal{V}$ is used to optimize the selection of $\mathscr{A}$ and its hyperparameters H. These optimization problems are generally non-convex and globally non-differentiable, requiring methods such as grid search over possible algorithm choices combined with local optimization techniques for hyperparameter tuning. However, if too many options for $\mathscr{A}$ and H are tested, there is a risk of selecting a combination that minimizes error only on $\mathcal{V}$, compromising generalization to unseen data (over-validation). Finally, $\mathcal{T}$, a new and independent subset of data, is used to evaluate the performance of the model f generated by the best combination of $\mathscr{A}$ and H using $\mathcal{L} \cup \mathcal{V}$. This error serves as a statistically unbiased estimate of the model performance on previously unseen data since $\mathcal{T}$ is used only once.

[2]See also the top performing models in the popular Kaggle www.kaggle.com ML competition website.

The choice of the ML algorithm $\mathscr{A}$ is non-trivial. The so-called *No-Free-Lunch theorem* [70] ensures that there is no way to determine a priori the best ML algorithms to use for a specific application. In general, we can distinguish two main approaches to ML. On the one hand, shallow models directly manipulate X to generate Y and can be used when the structure of X is considered a good representation for making effective predictions (e.g., tables, dataframes). On the other hand, Deep Learning models (also known as *Representation Learning*) are employed when it is necessary to first learn a good representation of the input data X from the data themselves, and then use this representation to generate effective predictions. This method is particularly useful when data are formatted as images or natural language. Deep Learning also opened the way to procedures like Transfer Learning with pre-trained[3] and foundation[4] models (i.e., reuse the same representation for multiple tasks).

For ZKPMs, it is possible to define a general model construction schema applicable to most of them. The training process of a ZKPM (i.e., the tuning of the parameters θ) mostly relies on the principle of Empirical Risk Minimization (ERM) [66]. Training a model according to ERM means finding the function that fits a training set, searching in a set of functions $\mathcal{F}$ carefully tuned during the model selection phase [71]. ERM selects a function based on a measure of the error it produces on the data, possibly combined with one or more regularization terms. Regularization allows to trade off errors in the training data and the model complexity. The regularization terms can be explicitly expressed in the loss, or also implicitly, for instance through the functional form (e.g., convolutions or transformers) or the optimization (e.g., dropout, early stopping).

To formally express these concepts, we can define a ML algorithm as the process of finding the optimal parameters θ, corresponding to a function $f \in \mathcal{F}$, as follows:

$$f^* = \arg\min_{f \in \mathcal{F}} \mathtt{E}(f, \mathcal{D}_n) + \mathtt{C}(f) \tag{2.2}$$

where $\mathcal{F}$ is the space of models induced by the choice of the algorithms, $\mathtt{E}(f, \mathcal{D}_n)$ is the error term, measured with reference to the desired metric, determined

[3] https://modelzoo.co
[4] https://huggingface.co

by the specific f over the dataset $\mathcal{D}_n$ and $\mathtt{C}(f)$ is a measure of the model complexity.

Given a set of possible configurations of algorithms and hyperparameters $\mathcal{A}_H$, the best combination $\mathscr{A}^*_{H^*}$ can be found as:

$$\mathscr{A}^*_{H^*}: \quad \arg\min_{\mathscr{A}_H \in \mathcal{A}_{\mathcal{H}}} \frac{1}{r}\sum_{i=1}^{r} \mathtt{E}(\mathscr{A}_H(\mathcal{L}^i), \mathcal{V}^i) \tag{2.3}$$

where $\mathtt{E}(f, \mathcal{V}^i)$ is the desired error metric for f computed using the data in $\mathcal{V}^i$, and $\mathscr{A}_H(\mathcal{L}^i)$ is a model f built with the algorithm $\mathscr{A}$ with its set of hyperparameters H and with the data $\mathcal{L}^i$. Note that the best algorithm and the associated set of hyperparameters $\mathscr{A}^*_{H^*}$ are evaluated on a dataset $\mathcal{V}^i$ that is independent of the training set $\mathcal{L}^i$.

Finally, to evaluate the performance of the final model $f^* = \mathscr{A}^*_{H^*}(\mathcal{D}_n)$, we can use:

$$\mathtt{E}(f^*) = \frac{1}{r}\sum_{i=1}^{r} \mathtt{E}(\mathscr{A}^*_{H^*}(\mathcal{L}^i \cup \mathcal{V}^i), \mathcal{T}^i) \tag{2.4}$$

We highlight that $\mathtt{E}(f^*)$ is an unbiased estimator of the true performance, since the data in $\mathcal{L}^i \cup \mathcal{V}^i$ are independent from the ones in $\mathcal{T}^i$ [71].

The choice of the most effective ZKPMs depends on factors such as the application domain, desired accuracy, input data characteristics, data scientists' expertise, and available computational resources.

Advantages of ZKPMs include:

- They require minimal human intervention needed to design, build, and test the model [65].
- They can exploit large amounts of data in various formats [65].
- If large datasets are available, they perform very well on average in real-world applications [65].
- They work effectively in interpolation scenarios [38].
- They usually require low computational power for inference [65].

Disadvantages of ZKPMs include:

- They require large amounts of data; deep models need huge datasets if no pretrained resources are available [27].
- Training can be computationally expensive [65].

- They generally provide poor extrapolation capabilities [38].
- They can produce predictions that lack physical plausibility [44].
- They are often not explainable or interpretable, although mitigation strategies exist [62].

For these reasons, ZKPMs are widely used, both as standalone predictive tools and as faster surrogates of more expensive models (e.g., complex simulations), allowing fast predictions suitable for optimization and decision-making scenarios [42].

2.2.4 Partial-Knowledge Predictive Models

PKPMs represent a promising frontier in predictive modeling. They combine the strengths of FKPMs and ZKPMs. On the one hand, they address the main limitation of FKPMs in terms of computational demands for accurate predictions and suboptimal performance, particularly in interpolation, where ZKPMs typically perform better. On the other hand, they also mitigate the limitations of ZKPMs, including their need for large datasets, limited physical plausibility, and weaker extrapolation performance, where FKPMs are more effective [35]. PKPMs need all four sources of information previously introduced, as represented in the general schema in Figure 2.5.

We can delineate the key aspects of PKPMs as:

- Similar to ZKPMs, PKPMs usually leverage on all the available data X. Domain knowledge can be incorporated to reduce spurious correlations and guide the learning process. Typically, PKPMs can be seen as ZKPMs informed by domain-specific knowledge. This leads to a distinction depending on when this knowledge is integrated into the learning process of the ZKPMs (i.e., before, during, and after). We refer to these categories as:
  - Pre-Processing: this class includes transforming inputs, datasets, or guiding algorithm choices before applying the ZKPM, to improve the model's ability to extract information. For example, Serial Hybrid PKPMs generate an initial output using a FKPM, which is subsequently used as input to a ZKPM to model aspects where domain knowledge is lacking. Domain knowledge can also inform

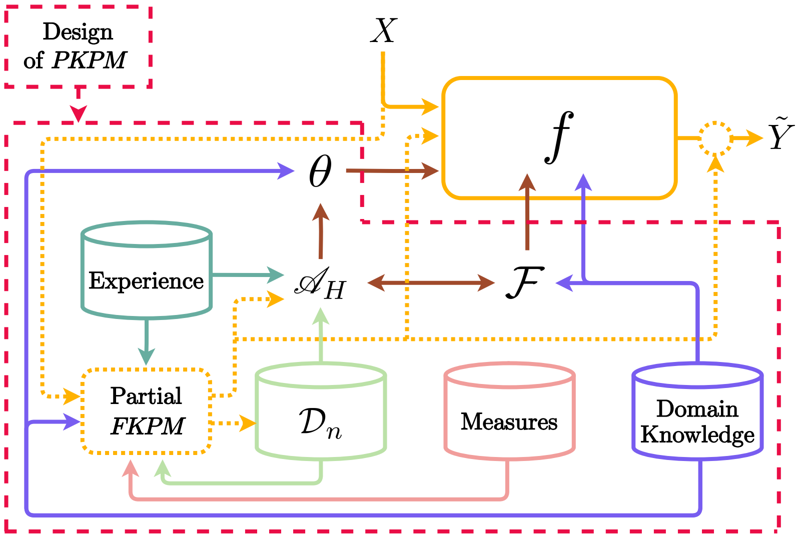

Fig. 2.5 PKPMs

the design of the ZKPM architecture itself. In other cases, input features are pre-processed or transformed according to principles derived from expert knowledge. In general, these strategies serve to improve data quality, guide feature engineering, and enrich or clean the data prior to training, thereby enhancing the model's predictive accuracy and overall reliability.

- In-Processing: in this case, the domain knowledge is embedded directly into the learning mechanism of a ZKPM, aiming to steer the training toward a more consistent result. This may involve adjusting the model space, adding regularizers that encode known physical laws, or using certain constraints that respect domain knowledge. Typically, this method requires us to formalize the mathematical form of the domain knowledge. This technique preserves the advantageous properties of ZKPMs, such as convexity and differentiability, and ensures that specific knowledge contributes to model performance. We highlight that this strategy is not just focused on the average model behavior, but rather on its point-wise performance.
- Post-Processing: it adjusts the model's outputs to meet domain-specific requirements or constraints, ensuring logical consistency and physically plausible predictions. For instance, the output of a

ZKPM can be used to replace the portion of modeling where domain knowledge is lacking, and this output then serves as input to a FKPM. Other approaches involve manipulating the output of the ZKPM to align it with specific logical constraints or domain principles. In general, these methods do not alter the internal structure of the ZKPM itself. They are particularly valuable in sensitive domains where logical consistency and adherence to expert knowledge are of critical importance.

We emphasize that the distinction between these three classes of techniques is not always clear. In the following, we will analyze the three types of PKPMs in greater depth.

- Finally, PKPMs involve a set of parameters and hyperparameters. Since PKPMs integrate both FKPMs and ZKPMs, they are defined by the parameters of FKPMs as well as the parameters and hyperparameters of ZKPMs.

We can trace a general overview of their main advantages and disadvantages of PKPMs compared to traditional approaches:

- They can handle various inputs and large datasets, as ZKPMs do [44].
- They can encapsulate varying degrees of domain knowledge, even when such knowledge is insufficient to construct a fully specified FKPM [43].
- They require less data than pure ZKPMs, but more than FKPMs [42].
- They usually need less human intervention than FKPMs, but more than ZKPMs [43].
- Construction is computationally intense, usually less than FKPM development and slightly more than ZKPMs [44].
- They generally provide fast and cheap predictions, similar to ZKPMs [65].
- They perform well both in interpolation (better than FKPMs and usually better than ZKPMs) and in extrapolation (comparably or better than FKPMs) [44].
- They are more physically plausible than ZKPMs, though not always as physically grounded as FKPMs [44].
- They achieve good average performance (comparably to ZKPMs) and often match or surpass FKPMs point-wise [22].

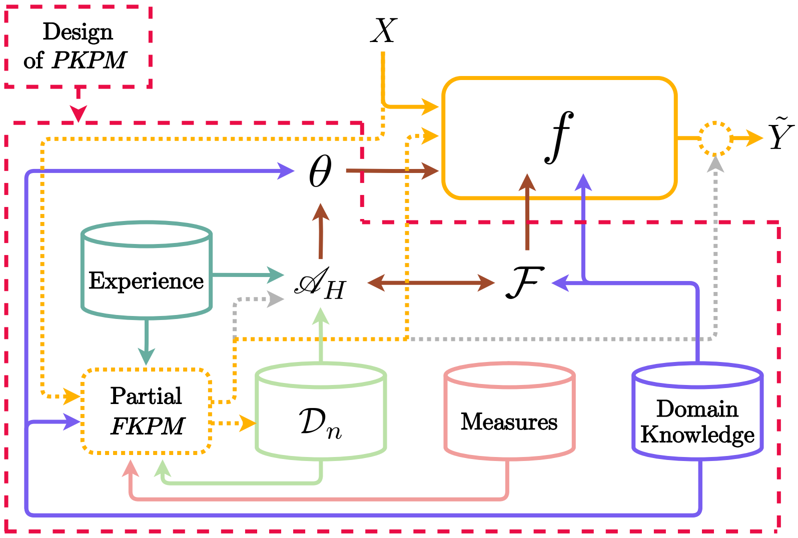

Fig. 2.6 PIPMs: Pre-Processing

- They improve explainability compared to ZKPMs, though still may not be fully interpretable [72].

Let us now analyze in detail the features of each type of PKPM.

Pre-Processing

In PKPMs, Pre-Processing enhances model performance by incorporating domain knowledge to inform ZKPMs before the learning process. Figure 2.6 illustrates this, with non-relevant parts for Pre-Processing grayed out.

For Pre-Processing, it is possible to identify three groups of techniques. The first one consists of methods that modify the input X. It involves cleaning and transforming the data to ensure quality and relevance [73]. Feature engineering, enrichment, and reduction are crucial steps, where knowledge can be leveraged to optimize computational efficiency, capture physical attributes, or to increase model interpretability [74]. We can formalize this methods as:

$$X \to \phi(X) \tag{2.5}$$

where $\phi : \mathcal{X} \to \Phi$ maps the input to a domain-informed space.

The second group involves modifying the dataset $\mathcal{D}_n$. It focuses on selecting and enriching data, even during the design of experiments phase, in order to ensure it accurately represents the studied phenomena [75]. This includes, for example, recognizing which data are of higher quality due to external factors [76] or enrich the dataset using FKPMs to generate more data [24]. Formally:

$$\mathcal{D}_n \rightarrow \mathcal{D}'_{n'} \tag{2.6}$$

representing transformations such as removal, stratification, and augmentation.

The third group aims to guide the ZKPMs selection. It involves choosing algorithms or hyperparameters of the ZKPM, informed by the domain knowledge [43]. For example, interpretable models are prioritized in safety-critical situations, while deep models may suit large structured datasets. In general, a mix of domain knowledge and experience can be extremely useful in constructing an informed ZKPM. This process is expressed as:

$$\mathscr{A}_H \rightarrow \mathscr{A}'_{H'} \tag{2.7}$$

signifying modifications in choices during performance assessment.

In general, the Pre-Processing framework prepares the models and the data for learning, aligning them with physical insights and domain constraints to enable accurate and interpretable PKPMs.

In-Processing PKPMs

In-Processing methods enhance predictive model performance by embedding domain knowledge into the learning process of a ZKPM (see Figure 2.7, where the parts that are not relevant for In-Processing PKPMs are in gray). They typically require the domain knowledge to be encoded in mathematical form, even if the theoretical model only partially describes the phenomena. In-Processing methods act on both the functional form and in the way the parameters are actually found. The learning process can be guided in several ways [43]. One method, introduced by Raissi et al. [22] and named Physics-Informed Neural Network (PINN), incorporates a regularization term that penalizes discrepancies with respect to the physical law. This concept can also be leveraged

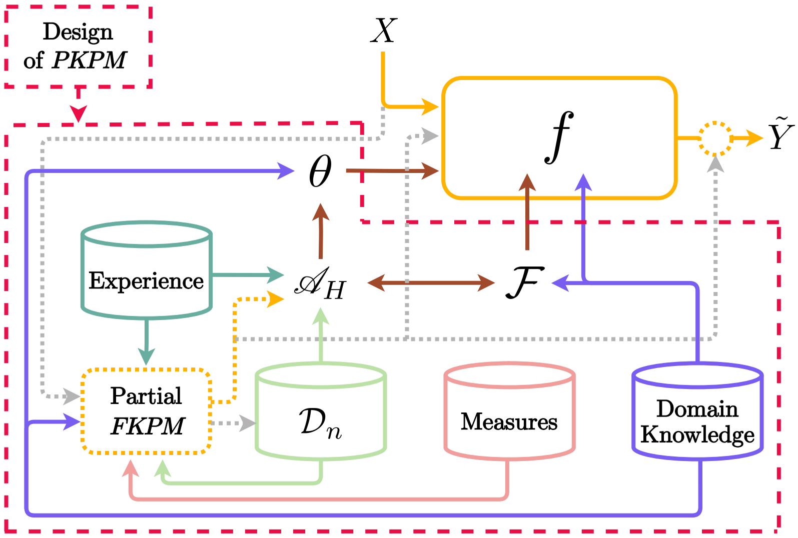

Fig. 2.7 PIPMs: In-Processing

in presence of complex systems, described by intricate differential equations, where simulators exist but are too slow for practical use. In such a situation, surrogate models can be developed leveraging both on simulator outputs and a physics-guided penalty term [22]. This method applies a soft constraint that guides the learning process and can also leverage partial knowledge or even hints on the physical behavior [43].

Knowledge can be exploited also during performance tuning. For instance, preference can be given to models that align more closely with domain knowledge, are simpler, or more explainable.

In-Processing methods can be formalized as:

$$\mathcal{F} \rightarrow \mathcal{F}' \tag{2.8}$$

$$f^* = \arg\min_{f \in \mathcal{F}'} \mathtt{E}(f, \mathcal{D}_n) + \mathtt{C}(f) + \mathtt{K}(f) \tag{2.9}$$

where $\mathcal{F}'$ is the adjusted function space and $\mathtt{K}(f)$ represents the regularization term embedding domain knowledge.

In summary, the In-Processing strategy within PKPMs fosters the training of models that are compliant with the domain knowledge.

Post-Processing PKPMs

Post-Processing refines ZKPMs outputs using domain-specific knowledge (see Figure 2.8, where the parts that are not relevant for Post-Processing PKPMs are in gray).

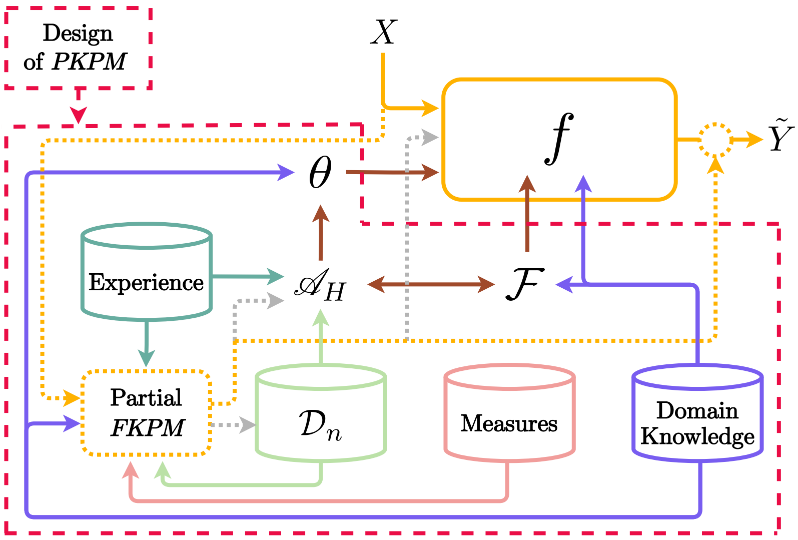

Fig. 2.8 PIPMs: Post-Processing

In particular, Post-Processing enhances data-driven predictions by enforcing domain-specific constraints and relationships. This is crucial when predictions must guarantee certain characteristics [77]. This final step can refine predictions to maintain compliance with established physical principles or operational norms, enhancing both reliability and acceptance among practitioners [77] [77].

Post-Processing techniques can also be useful in cases where FKPMs are interpretable but require inputs that are either difficult to measure or depend on unknown phenomena. In these scenarios, ZKPMs can complement them by providing estimations of missing physical modeling [42].

Another common way to implement Post-Processing are the so called Parallel Hybrid Models, where the outputs of a FKPM and a ZKPM are combined to improve the predictions [42].

Post-Processing is formalized as:

$$f(X) \to \psi(f(X)) \tag{2.10}$$

where $\psi : \mathcal{Y} \to \mathcal{Y}$ represents a function embedding domain knowledge to modify the ZKPM outputs.

A notable advantage of Post-Processing is its ability to adapt existing ZKPMs without altering their core structure [42]. This makes it a practical approach for integrating additional constraints or refinements into running systems. By applying Post-Processing, practitioners can ensure predictions not only meet accuracy requirements but also adhere to ethical, physical, and practical constraints, enhancing trust and relevance in real-world applications [72].

2.3 Illustrative example

In this section, the concepts of FKPMs, ZKPMs, and PKPMs will be illustrated through an example, demonstrating their practical use and highlighting advantages and disadvantages.

The chosen example is the dynamic analysis of a Mass-Spring-Damper system operating without external forces. The example is tackled using all three approaches, and their performance is evaluated across various scenarios. The structure of the presentation is as follows.

- Section 2.3.1 describes the Mass-Spring-Damper setting.
- Section 2.3.2 describes the different scenarios considered in this example.
- Section 2.3.3 outlines the methodology used to generate the dataset.
- Section 2.3.4 describes the FKPM, ZKPM, and PKPM used in this example.
- Section 2.3.5 details the evaluation metrics employed.
- Section 2.3.6 presents and discusses the results.

2.3.1 The Mass-Spring-Damper System

Consider a Mass-Spring-Damper system [78], depicted in Figure 2.9, in which no external forces are applied.

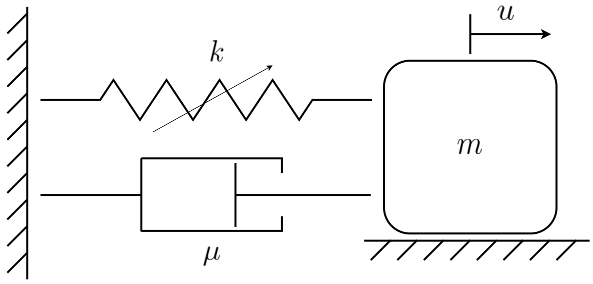

Fig. 2.9 Mass-Spring-Damper system.

This system models a damped harmonic oscillator, commonly used in physics and engineering. It features an oscillating object of mass m, a spring with constant k providing a restoring force, and a damper with coefficient μ representing resistive damping forces. In this setup, the mass is initially displaced in position u_0, and then allowed to oscillate freely over time.

Let us assume the displacement $u(t)$ of the mass can be measured at various time points $t \in [0, t_m]$. The objective is to predict the position of the mass not only within this time range (i.e., interpolation) but also beyond it, up to a final time t_f (i.e., extrapolation).

The measured points compose a dataset $\mathcal{D}_n = \{(t_1, u(t_1)), \ldots, (t_n, u(t_n))\}$, which, for demonstration purposes, is synthetically generated. The input space is time, and the output space is the corresponding displacement.

The modeling of a Mass-Spring-Damper system can be approached in several ways. Historically, physical models (i.e., FKPMs) have proven effective in accurately describing such systems. Several FKPMs can be formulated for this system, varying in complexity and accuracy. A commonly used FKPM is given by a second-order linear differential equation [78]:

$$m \frac{d^2 u(t)}{dt^2} + \mu \frac{du(t)}{dt} + k_0 u(t) = 0 \tag{2.11}$$

In this equation:

- $m\frac{d^2u(t)}{dt^2}$ models the inertial force, as the product between of the mass (measured in $[kg]$) and the acceleration $\frac{d^2u(t)}{dt^2}$ (measured in $[m/s^2]$).
- $\mu\frac{du(t)}{dt}$ represents the damping force, as the product between the damping factor μ (measured in $[Ns/m]$) and the velocity $\frac{du(t)}{dt}$ (measured in $[m/s]$).
- $k_0u(t)$ is the restoring force, as the product between of the the elastic constant k_0 (measured in $[N/m]$) and the displacement $u(t)$ (measured in $[m]$).

When linear approximations are insufficient, a more complex model may be used [78]:

$$m\frac{d^2u(t)}{dt^2}+\mu\frac{du(t)}{dt}+k_1u(t)+k_2u^3(t)=0 \tag{2.12}$$

Here, $k_1u(t)$ captures the linear restoring force, while $k_2u^3(t)$ introduces a non-linear component. Constants k_1 and k_2 (measured in $[{}^{N}/{}_{m}]$ and $[{}^{N}/{}_{m^3}]$, respectively) reflect the strengths of the respective effects.

Although even more accurate physical models exist (e.g., involving non-linear damping), their inclusion is beyond the scope of this section.

In practice, even with a considerable amount of physical knowledge, modeling such a system is far from trivial. First of all, not all physical parameters are necessarily known, but many often need to be measured or estimated. Furthermore, FKPMs may inaccurately represent reality. For instance, a linear elastic force may be assumed for simplicity, even though the system's elasticity might be better described by a non-linear law. Given these complications, FKPMs are not the only modeling approach adopted; as we will see, the same problem can also be tackled using ZKPMs and PKPMs.

For the purposes of this example, we consider different conditions under which each model type is evaluated. We assume that Eqns.(2.11) and (2.12), with all parameters known, represent the Ground Truth of different scenarios. The datasets were synthetically generated from these equations. In particular, Eq.(2.12) serves as a more refined model compared to Eq.(2.11), allowing us to demonstrate both a case where the FKPM correctly models the physics and one where part of the physics is inaccurately represented.

2.3.2 Scenarios: Surrogation and Modeling

In order to assess the effectiveness of each class of methods in different conditions, we consider two distinct and practically relevant scenarios: surrogation and modeling. For each scenario, we examine two cases with different measurement precisions, resulting in a total of four case studies. Each case presented is characterized by a different dataset and is approached using all three model types.

Surrogation scenario

The surrogation scenario considers the case in which we do not have access to measurements of the real-world system, but only to a costly approximation of it (e.g., software simulations, model scale tests). Therefore, we need to create a model to simplify output predictions. In our example, the costly approximation of the real system is represented by the solution of Eq. (2.11). For this scenario, we analyze two different cases:

- The case in which we can measure everything without noise (denoted by the superscript $(2.11), 0$). This represents a typical use case of surrogates for software simulators.
- The case in which measurements are noisy (denoted by the superscript $(2.11), \sigma$). This is a common scenario involving model-scale tests.

In such conditions, it may be necessary to surrogate the existing model to reduce the prediction time, the computational cost (for software simulators) or the experimental cost (for model scale test).

Modeling scenario

This setting analyzes the case in which data are measured directly from a real physical system. In this example, we assume that the physical system is governed by Eq. (2.12), while during the model construction phase, only Eq. (2.11) is known to the modeler. Analyzing this condition allows us to evaluate modeling errors due to physical assumptions.

Then, the datasets of the modeling scenario are obtained by the solution of Eq. (2.12). Two measurement precision cases are considered:

- The case of highly precise measurements (denoted by the superscript $(2.12), 0$).
- The case of noisy measurements (denoted by the superscript $(2.12), \sigma$).

This scenario allows us to evaluate how models approximate a real system by assessing their behavior both in extreme cases with high-fidelity measurements and when the data are noisy.

2.3.3 Data generation

This section outlines the data generation process used to construct the datasets corresponding to each case. As previously mentioned, all the datasets were synthetically generated. In particular, we defined a Ground Truth model (i.e., Eq. (2.11) for surrogation and Eq.(2.12) for modeling), solved it via numerical integration, and sampled data points from the resulting solutions.

According to Section 2.3.2, we varied two aspects to create different conditions: the first is the model used to create the dataset and the second is level of measurement noise. Specifically, for the model, we used:

- The linear model defined in Eq. (2.11).
- The non-linear model defined in Eq. (2.12).

Regarding measurement quality, we sampled:

- Clean measurements.
- Measurements corrupted by adding Gaussian noise.

The solution of the differential equations Eqns. (2.11) and (2.12) were numerically obtained using Euler's method [79], imposing an initial condition $u(0) = u_0$. For each of the two solutions, we sampled n equally spaced points within the range $[0, t_m]$, where $t_m < t_f$. This resulted in the first two datasets. Then, two additional datasets were generated by adding Gaussian noise $\mathcal{N}(0, \sigma^2)$ to the original ones. In summary, these are the four available datasets:

- $\mathcal{D}_n^{(2.11),0} = \{(t, u(t)) | t \in \{0, \frac{t_m}{n-1}, \cdots, t_m\}\}$ with $u(t)$ the solution of Eq. (2.11). This dataset was used for the surrogation case without noise.

- $\mathcal{D}_n^{(2.11),\sigma} = \{(t, u(t) + \mathcal{N}(0, \sigma^2)) | t \in \{0, \frac{t_m}{n-1}, \cdots, t_m\}\}$ with $u(t)$ the solution of Eq. (2.11). This dataset was used for the surrogation case with noise.
- $\mathcal{D}_n^{(2.12),0} = \{(t, u(t)) | t \in \{0, \frac{t_m}{n-1}, \cdots, t_m\}\}$ with $u(t)$ the solution of Eq. (2.12). This dataset was used for the modeling case without noise.
- $\mathcal{D}_n^{(2.12),\sigma} = \{(t, u(t) + \mathcal{N}(0, \sigma^2)) | t \in \{0, \frac{t_m}{n-1}, \cdots, t_m\}\}$ with $u(t)$ the solution of Eq. (2.12). This dataset was used for the modeling case with noise.

Note that the noisy datasets depend on a single realization of a random variable. As we will show later, to avoid relying on a single realization, the results will be averaged over 30 different dataset realizations. Table 2.3 reports the parameter values used in the simulation process.

Parameter	**Eq.**(2.11)	**Eq.**(2.12)
m	1	
u_0	1	
t_f	1	
t_m	$t_f/3$	
σ	0.1	
μ	3	
k_0	250	–
k_1	–	230
k_2	–	3

Table 2.3 Parameters used for data generation.

Figure 2.10 provides a visual comparison of the four datasets and shows the solutions obtained from linear and non-linear models. For noisy datasets, the figure shows a single noise realization.

2.3.4 Modeling approaches

This section outlines the implementation of the three modeling strategies (i.e., FKPMs, ZKPMs, and PKPMs) applied to the Mass-Spring-Damper system described above.

These are the available information during the models' construction:

- Physical knowledge about the system (i.e., Eq. (2.11)).

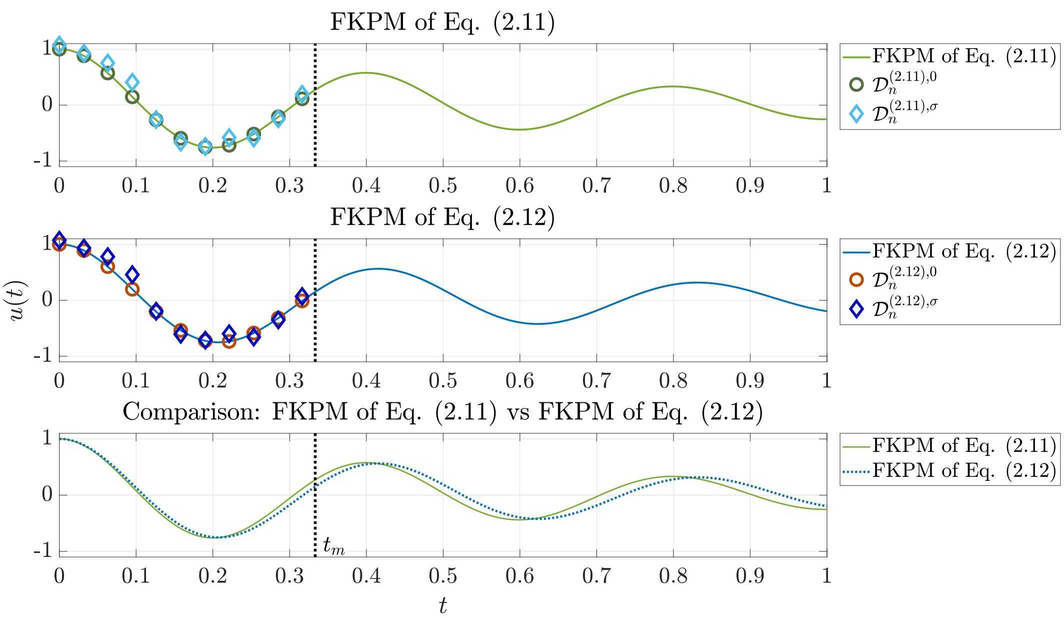

Fig. 2.10 Solutions of FKPMs and generated datasets with and without noise.

- Measured system parameters (i.e., mass).
- A dataset consisting of time-displacement pairs (i.e., $\mathcal{D}_n^{(2.11),0}$, $\mathcal{D}_n^{(2.11),\sigma}$, $\mathcal{D}_n^{(2.12),0}$, or $\mathcal{D}_n^{(2.12),\sigma}$).
- Expert knowledge or assumptions.

FKPM

To construct a FKPM, one begins by formulating the governing equations using known physical principles and generating the functional form of the FKPM. In this illustrative example, for all cases, let us assume that when constructing the FKPM, Eq.(2.11) is known and serves as an approximation of the system dynamics; however some of its parameters are unknown. We also assume that a solver using Euler's method is available, as it is common in practice. Conversely, Eq.(2.12) is assumed to be unknown. This differs from the data generation phase, during which both Eqns.(2.11) and (2.12), along with their parameters, were fully known, as they were used to simulate the Ground Truth. In fact, the purpose of this example is to describe the practical construction of a FKPM, in which part of the physics (e.g., laws, parameters) may be unknown or partially incorrect.

In real scenarios, some parameters, such as the mass m, may be statically measured. Others (e.g., damping and stiffness coefficients) must be estimated by fitting the model to the data. In this example, u_0, μ, k_0, k_1, and k_2 are assumed to be tuned. A straightforward method involves grid search: for a range of possible parameters' values, the Eq. (2.11) is solved using Euler's method, and the combination minimizing the prediction error on the dataset is selected. In this study, parameters were searched over the range $[1,100]$ using 45 logarithmically spaced values. Increasing the resolution of the grid search used for parameter tuning may further improve FKPM performance (particularly in noise-free settings, the error would tend to zero). However, such exhaustive searches are typically infeasible in real-world applications, due to computational constraints.

ZKPM

ZKPMs are purely data-driven and make no assumptions about system physics. The construction of a ZKPM requires one to choose the ML algorithms and then tune its hyperparameters just based on the data, for example, with a grid search hold-out method. The experience of the data scientist may help in reducing the searching space of algorithms and hyperparameters. This example uses a polynomial regression model of degree p over the normalized time interval $[0,1]$:

$$f(x) = \sum_{i=0}^{p} w_i x^i \tag{2.13}$$

where w_i are the model coefficients. The degree p is a hyperparameter that needs to be tuned. The training loss is defined as the squared error:

$$\ell(f(x), y) = (y - f(x))^2 \tag{2.14}$$

and regularization is added based on the model's curvature [80]:

$$\mathtt{C}(f) = \int_0^1 \left(\frac{d^2 f(x)}{dx^2} \right)^2 dx = \boldsymbol{w}^T C \boldsymbol{w} \tag{2.15}$$

where

$$C_{i,j} = \begin{cases} 0 & \text{If } j < 2 \vee j < 2 \\ \frac{i(i-1)j(j-1)}{i+j-3} & \text{Otherwise} \end{cases}, \quad i,j \in \{0, \cdots, p\} \tag{2.16}$$

The training objective becomes:

$$\min_{\boldsymbol{w}} \quad \|X\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda \boldsymbol{w}' C \boldsymbol{w} \tag{2.17}$$

where

$$\boldsymbol{w} = \begin{bmatrix} w_0 \\ \vdots \\ w_p \end{bmatrix} \quad X = \begin{bmatrix} x_1^0 & \cdots & x_1^p \\ \vdots & \ddots & \vdots \\ x_n^0 & \cdots & x_n^p \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{2.18}$$

and λ is a hyperparameter. This problem has a closed-form solution:

$$\boldsymbol{w} = (X^T X + \lambda C)^+ X^T \boldsymbol{y} \tag{2.19}$$

where $(X'X + \lambda M)^+$ is the Moore–Penrose inverse of $(X'X + \lambda M)$. Hyperparameters p and λ are tuned via grid search hold-out methods (leave-one-out cross-validation). The values of p range from 0 to 30, while λ is selected from $[10^{-6}, 10^6]$ using a logarithmic grid of 65 points.

PKPM

The PKPM combines physics-based constraints with data-driven learning. In this example, these two sources of information (i.e., physical knowledge and data) can be fully leveraged in building the PKPM. Similarly to FKPMs, during the construction of the PKPM, Eq. (2.11) is known, while Eq. (2.12) is unknown.

The PKPM must be constructed to fit the data, as the ZKPM does, while also leveraging on the Eq. (2.11). When physics is expressed by a mathematically encoded equation like Eq. (2.11), the state of the art PKPM approach is to introduce a penalty term into the ZKPM formulation of Eq.(2.17). This forces the function $f(x)$ (i.e., the approximation of $u(y)$) to follow the principle of Eq. (2.11). There are several methods for defining the penalty term. Some common approaches involve sampling different values of x (called *collocation* *points*), evaluating the point-wise violation of Eq. (2.11) using a specific metric, and using the sum of these violations as the penalty term. In other cases, the integral of the violation over the function domain can be solved in closed form and applied as a metric for the overall violation of the physical equation. Typically, the second method compels the model to adhere to the physical model across the entire domain and uses a single term, leading to lower computational effort. Therefore, it is preferable when applicable, as is the case in our example. More formally:

$$\mathtt{P}(f) = \int_0^1 \left(m\frac{d^2 f(x)}{dx^2} + \mu\frac{df(x)}{dx} + k_0 f(x) \right)^2 dx = \boldsymbol{w}' P \boldsymbol{w} \tag{2.20}$$

where

$$P = k_0^2 P_1 + \mu^2 P_2 + m^2 C + k_0\mu(P_3 + P_3') + \mu m(P_4 + P_4') + k_0 m(P_5 + P_5') \tag{2.21}$$

with

$$P_{1,i,j} = \frac{1}{i+j+1} \qquad P_{2,i,j} = \begin{cases} 0 & \text{If } i < 1 \vee j < 1 \\ \frac{ij}{i+j-1} & \text{Otherwise} \end{cases}$$
$$P_{3,i,j} = \begin{cases} 0 & \text{If } i < 1 \\ \frac{i}{i+j} & \text{Otherwise} \end{cases} \qquad P_{4,i,j} = \begin{cases} 0 & \text{If } i < 2 \vee j < 1 \\ \frac{i(i-1)j}{i+j-2} & \text{Otherwise} \end{cases}$$
$$P_{5,i,j} = \begin{cases} 0 & \text{If } i < 2 \\ \frac{i(i-1)}{i+j-1} & \text{Otherwise} \end{cases} \tag{2.22}$$

where $i, j \in \{0, \cdots, p\}$.

ZKPM of Eq. (2.17) is modified applying the new penalty term $\mathtt{P}(f)$:

$$\min_{\boldsymbol{w}} \quad \|X\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda_1 \boldsymbol{w}' C \boldsymbol{w} + \lambda_2 \boldsymbol{w}' P \boldsymbol{w} \tag{2.23}$$

Also in this case, the minimization problem (2.23) can be solved in closed form:

$$\boldsymbol{w} = (X'X + \lambda_1 C + \lambda_2 P)^{+} X' \boldsymbol{y} \tag{2.24}$$

Hyperparameters p, λ_1, and λ_2 are optimized using leave-one-out cross-validation. In particular, λ_1 and λ_2 have been searched in the rage $[10^{-6}, 10^{+6}]$ with a grid of 65 points equally spaced in logarithmic scale and the p hyperparameter in $[0, 1, \cdots, 30]$. The parameter m is known, while μ and k_0 have been set according to the scenario considered. To simulate typical real-world conditions, we assume that the parameters μ and k_0 are known for surrogation cases, while for modeling, they are tuned in the same manner as in the FKPM approach.

2.3.5 Evaluation criteria

This section defines the evaluation strategy used to compare the performance of FKPMs, ZKPMs, and PKPMs across all scenarios.

We adopt three metrics:

- Training time: the computational time required to build the model, excluding any prior knowledge acquisition or data collection time.
- Prediction time: the time taken to generate a single prediction given a new input value.
- Accuracy: assessed via the Mean Absolute Error (MAE) between the model prediction and the true output. Ground Truth values are taken from the same model used to generate the data in each scenario (i.e., Eq. (2.11) for surrogation and Eq. (2.12) for modeling).

To evaluate accuracy in greater detail, we distinguish between:

- Interpolation error, computed over the interval $[0, t_m]$, where training data are available.
- Extrapolation error, computed over the interval $(t_m, t_f]$, where predictions must generalize beyond the observed data.

2.3.6 Results and discussion

This section presents the results obtained from FKPMs, ZKPMs, and PKPMs. For consistency and fair comparison, the number of data points used for training differs slightly between models: FKPMs are trained on $n = 11$ points (reflecting use in low-data regimes), whereas ZKPMs and PKPMs are trained on $n = 22$ points (where data-driven learning is usually preferred).

The summary of the results is reported as follows.

		FKPM	ZKPM	PKPM
		(n=11)	(n=22)	
MAE	**Interpolation**	0.0176	0.00179	0.00143
	Extrapolation	0.0374	2.888	0.0253
Time	**Training**	$\sim 160s$	$\sim 5s$	$\sim 170s$
	Predicting	$\sim 10^{-3}ms$	$\sim 10^{-4}ms$	$\sim 10^{-4}ms$

Table 2.4 Results for the Surrogation scenario using $\mathcal{D}_n^{(2.11),0}$: training time, prediction time, interpolation error, and extrapolation error of FKPM (with n=11), ZKPM, and PKPM (with n=22).

- Table 2.4 and Figure 2.11 present the results for the case of surrogation with noise-free data (dataset $\mathcal{D}_n^{(2.11),0}$).
- Table 2.5 and Figure 2.12 present the results for the case of surrogation with noisy data (dataset $\mathcal{D}_n^{(2.11),\sigma}$).
- Table 2.6 and Figure 2.13 present the results for the case of modeling with noise-free data (dataset $\mathcal{D}_n^{(2.12),0}$).
- Table 2.7 and Figure 2.14 present the results for the case of modeling with noisy data (dataset $\mathcal{D}_n^{(2.12),\sigma}$).

In particular, Tables 2.4-2.7 show the MAE (in interpolation and extrapolation) and the Time (training and test) of each model. In order to remove the dependency from the single noise realization, the results of models dependent on a noisy dataset ($\mathcal{D}_n^{(2.11),\sigma}$. and $\mathcal{D}_n^{(2.12),\sigma}$.) have been averaged over 30 repetition of the experiments.

Figures 2.11-2.14, instead, report the data points leveraged to build the predictive models, and the plots of Ground Truth (i.e., the solution of Eq. (2.11) for the surrogation scenario and Eq. (2.12) for the modeling scenario) and each predictive model (i.e., the FKPM, the ZKPM, and the PKPM). For cases involving a noisy dataset, only a single representative realization of the noise has been reported.

From the results, it is possible to extract the following key findings:

- FKPMs exhibit physically consistent results across both interpolation and extrapolation but depend heavily on the granularity of the parameter tuning.

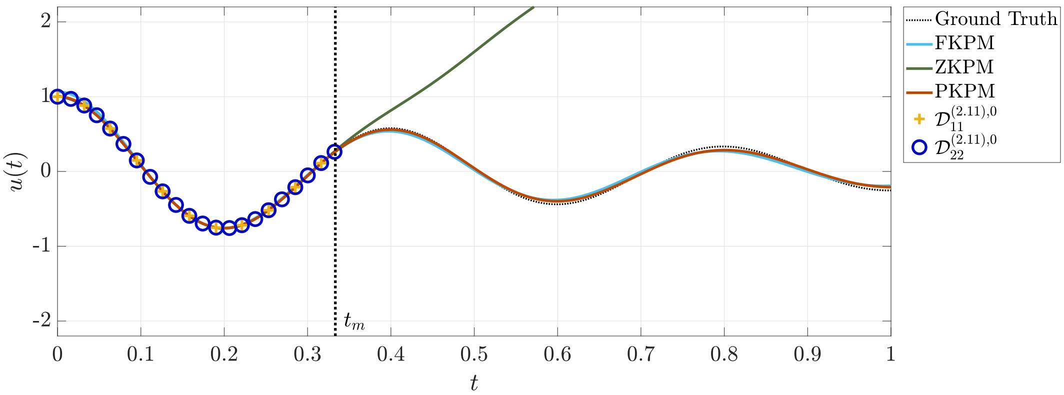

Fig. 2.11 Results for the Surrogation scenario using $\mathcal{D}_n^{(2.11),0}$: Ground Truth, $\mathcal{D}_{11}^{(2.11),0}$, $\mathcal{D}_{22}^{(2.11),0}$, and the prediction of the FKPM, ZKPM, and PKPM.

		FKPM	**ZKPM**	**PKPM**
		(n=11)	(n=22)	
MAE	**Interpolation**	0.0394 ± 0.0165	0.0390 ± 0.0051	0.0343 ± 0.0091
	Extrapolation	0.0662 ± 0.0487	3.1649 ± 0.358	0.0287 ± 0.0038
Time	**Training**	$\sim 160s$	$\sim 5s$	$\sim 170s$
	Predicting	$\sim 10^{-3}ms$	$\sim 10^{-4}ms$	$\sim 10^{-4}ms$

Table 2.5 Results for the Surrogation scenario using $\mathcal{D}_n^{(2.11),\sigma}$: mean and standard deviation (across 30 repetitions of the experiment) of training time, prediction time, interpolation error, and extrapolation error of FKPM (with n=11), ZKPM, and PKPM (with n=22).

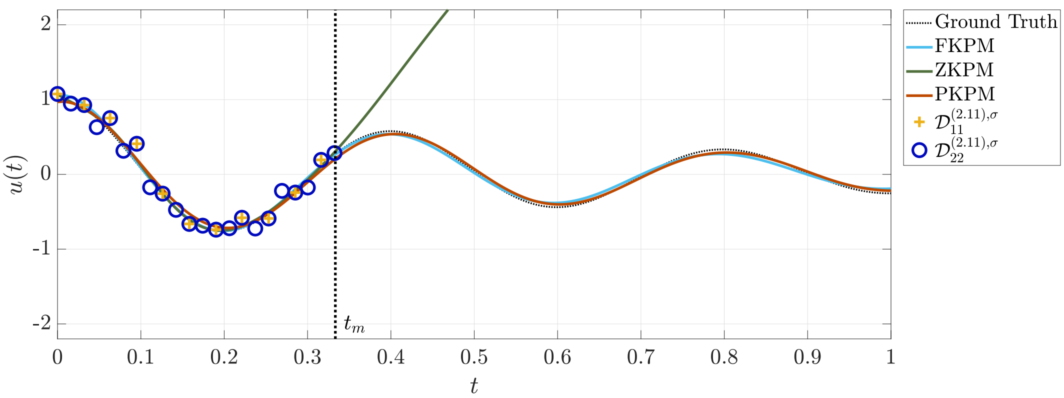

Fig. 2.12 Results for the Surrogation scenario using $\mathcal{D}_n^{(2.11),\sigma}$: Ground Truth, $\mathcal{D}_{11}^{(2.11),\sigma}$, $\mathcal{D}_{22}^{(2.11),\sigma}$, and the prediction of the FKPM, ZKPM, and PKPM for a single realization of the noise.

		FKPM	ZKPM	PKPM
		(n=11)	(n=22)	
MAE	Interpolation	0.0233	0.0020	0.0012
	Extrapolation	0.0528	2.8706	0.0233
Time	Training	$\sim 160s$	$\sim 5s$	$\sim 330s$
	Predicting	$\sim 10^{-3}ms$	$\sim 10^{-4}ms$	$\sim 10^{-4}ms$

Table 2.6 Results for the Modeling scenario using $\mathcal{D}_n^{(2.12),0}$: training time, prediction time, interpolation error, and extrapolation error of FKPM (with n=11), ZKPM, and PKPM (with n=22).

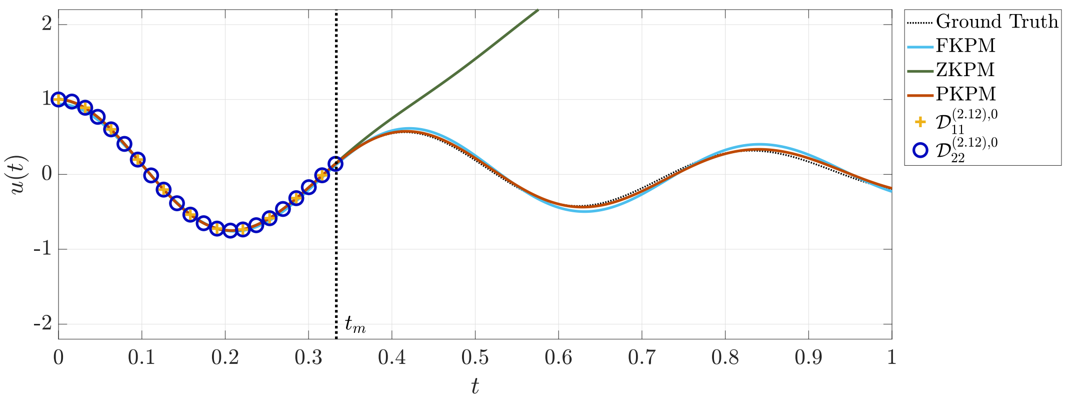

Fig. 2.13 Results for the Modeling scenario using $\mathcal{D}_n^{(2.12),0}$: Ground Truth, $\mathcal{D}_{11}^{(2.12),0}$, $\mathcal{D}_{22}^{(2.12),0}$, and the prediction of the FKPM, ZKPM, and PKPM.

		FKPM	ZKPM	PKPM
		(n=11)	(n=22)	
MAE	Interpolation	0.0556 ± 0.0039	$0.0403 \pm 0.0.0060$	0.0391 ± 0.0108
	Extrapolation	0.1045 ± 0.0175	3.0229 ± 0.3624	0.0962 ± 0.0257
Time	Training	$\sim 160s$	$\sim 5s$	$\sim 330s$
	Predicting	$\sim 10^{-3}ms$	$\sim 10^{-4}ms$	$\sim 10^{-4}ms$

Table 2.7 Results for the Modeling scenario using $\mathcal{D}_n^{(2.12),\sigma}$: mean and standard deviation (across 30 repetitions of the experiment) of training time, prediction time, interpolation error, and extrapolation error of FKPM (with n=11), ZKPM, and PKPM (with n=22).

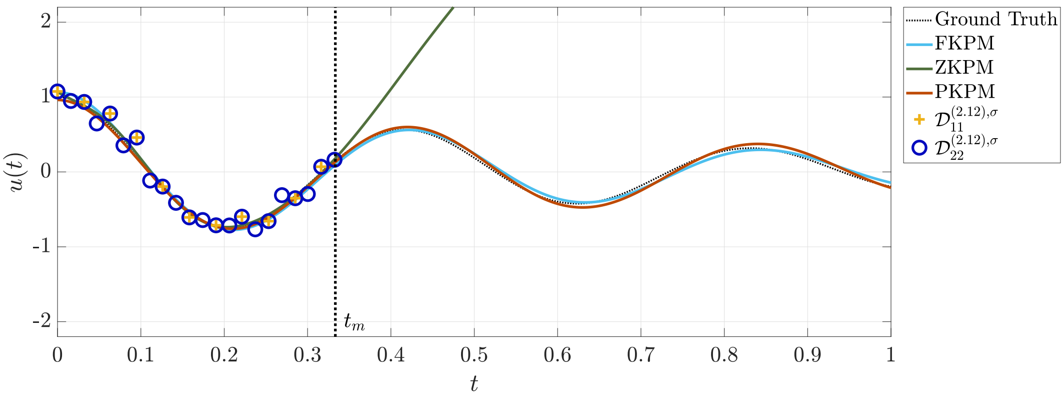

Fig. 2.14 Results for the Modeling scenario using $\mathcal{D}_n^{(2.12),\sigma}$: Ground Truth, $\mathcal{D}_{11}^{(2.12),\sigma}$, $\mathcal{D}_{22}^{(2.12),\sigma}$, and the prediction of the FKPM, ZKPM, and PKPM for a single realization of the noise.

- ZKPMs achieve better performance than FKPM in interpolation, but fail in extrapolation, yielding unrealistic predictions outside the training interval.
- PKPMs consistently outperform both FKPMs and ZKPMs across all metrics, demonstrating strong generalization and physical plausibility.
- Noise degrades models' accuracy, but does not alter the relative performance hierarchy among the approaches.
- All models have high training time (it is strictly related to the number of points selected in the grid search algorithms).
- PKPMs are the most computationally expensive due to their combined learning objective.
- Prediction times for ZKPMs and PKPMs are lower than FKPMs by one order of magnitude.

These outcomes align with theoretical concepts discussed in Section 2.2, and validate the practical benefits of incorporating physical knowledge into data-driven learning.

2.4 Review of industrial applications

This section examines the use of PKPMs across various industrial sectors. The focus is on six primary domains: extraction, chemical, manufacturing, transportation, energy, and construction.

To carry out this review, a targeted search within a defined time window was conducted using selected keywords in the academic database of Scopus[1]. The search timespan starts from 2013 onward, according to Section 2.1. Because the initial keyword-based search retrieved a considerable number of publications, additional selection criteria were employed based on the significance and impact of the works. A minimum citation threshold was applied to ensure that only the most influential contributions are discussed.

Within each of the six domains mentioned, we provide a summary table and discusses the relevant results. The summary tables include details such as the work's reference information, year of publication, PKPM type (Pre-, In-, or Post-Processing), industry sector, the data and knowledge available or generated, any associated FKPM, the ZKPM adopted, and key outcomes achieved.

2.4.1 Extraction

The extraction sector delivers fundamental raw materials and fuels to many industrial operations. It includes activities such as mining, drilling, and refining, all of which are critical for producing metals, minerals, and energy resources like oil and gas. These materials serve as the foundation of contemporary manufacturing and infrastructure development worldwide. Improving extraction processes while concurrently addressing sustainability concerns is necessary to ensure long-term industrial viability and environmental protection. Increasingly, predictive models like FKPM, ZKPM, and PKPMs are employed to optimize extraction operations, anticipate resource availability, forecast subsurface composition, and enhance safety measures.

Numerical simulation is among the most common FKPM approaches. Previous research has reviewed the application of FKPMs to mining processes [81], as well as related areas including mineral processing [82] and tunneling [83].

Many publications thoroughly discuss ZKPM applications in the extraction industry. For instance, there are works devoted to the oil and gas sector [84] and the mineral industry [85]. With regard to PKPMs, although some studies have tangentially addressed their use in the extraction domain [30], a comprehensive review is currently lacking. During the doctorate program, this gap was filled by summarizing the emerging trends in PKPMs applications within this sector.

For the extraction industry, a specific search query was employed that includes relevant keywords (see [23] for more information) related to both the extractive industry and PKPMs. Because this search yielded over 1400 documents, a further filtering based on citation thresholds was applied (papers cited, according to Scopus, by $\geq$25 papers for works published from 2013 and 2021, by $\geq$15 papers for works published in 2022, by $\geq$10 papers for works published in 2023, and by $\geq$1 papers for works published in 2024).

This analysis indicates that the most prevalent PKPMs applications in extraction focus on fossil fuel production, particularly subsurface porosity and reservoir modeling. While fully known numerical simulations traditionally address these problems and numerous commercial tools exist [86], they are computationally demanding. Thus, practitioners seek more practical methods, such as PKPMs [86]. Graph-based PKPMs, which incorporate domain knowledge and rely on historical data for parameter estimation, are widely used for reservoir modeling [86–91]. These techniques have been validated across multiple scenarios, achieving accuracies comparable to established numerical simulators but with substantially lower computational times [91, 92]. Additional PKPMs approaches include employing efficient surrogates through Neural Operators, PINNs, or the use of FKPM outputs to train surrogate models [93–96]. An overview of the reviewed papers is provided in Table 2.8.

2.4.2 Chemical

The chemical industry encompasses the transformation of basic feedstock like oil, air, water, metals, and minerals into thousands of chemical products. It includes subdomains such as agrochemicals, pharmaceuticals, polymers, and oleochemicals, each serving specific market needs. A prominent challenge in the chemical sector is sustainability, requiring a reduction in carbon footprints,

Paper	Year	PKPM	Application Domain	Available and/or Generated Data	Prior Knowledge and/or FKPM	ZKPM	Results & Comparison between FKPM, ZKPM, and PKPM
[87]	2016	In (parameters tuning)	Reservoir waterflooding	(i) Synthetically generated data (1050 days simulated using Eclipse 100). (ii) Experimental data (5800 samples)	1-dimensional connective flow units model	Interior-point method	(i)With PKPM, NPV increased more than 3 times. PKPM is comparable to FKPM but more efficient (ii) With PKPM, +20% oil production rate and −4% water cut. No comparisons vs ZKPM
[86]	2016	In (parameters tuning)	Reservoir waterflooding	Field data (1925 days)	1-dimensional connective flow units model	Interior-point method	Graphs show that the PKPM can have an accuracy similar to FKPM simulations, but with superior efficiency. No comparisons vs ZKPM
[88]	2017	In (parameters tuning)	Reservoir waterflooding	Synthetically generated data (field example trained with simulations of 800 days)	1-dimensional connective flow units model	Ensemble smoother with multiple data assimilation	Graphs show that the PKPM can have an accuracy similar to FKPM simulations, but with superior efficiency. PKPM outperforms a previous capacitance-resistance PKPM. No comparisons vs ZKPM
[89]	2018	In (parameters tuning)	Reservoir waterflooding	Synthetically generated data (field example trained with Eclipse 100 simulations of 2050 days)	1-dimensional connective flow units model	Ensemble smoother with multiple data assimilation	In the field test, optimization with PKPM increases NPV of 40%. No comparisons vs FKPM and ZKPM
[90]	2018	In (parameters tuning)	Unconventional reservoir flow modeling	Field data (300 days)	Diffusive diagnostic function	Ensemble smoother with multiple data assimilation	Graphs show that the PKPM can have a good accuracy on the testset
[97]	2020	Post	Hydraulic fracturing	Synthetically generated data (2841 snapshot of 251 points each)	Fracture propagation PKN model	Neural network	Graphs shows that PKPM and ZKPM perform similarly in interpolation. PKPM is much better in extrapolation. FKPM can predict the trend, but with low accuracy.
[93]	2020	Pre & In (surrogate)	Reservoir simulation	Synthetically generated data (5700 data points)	AD-GPRS full-order simulations and Darcy's model	Neural network	Tested in 100 cases, PKPM can predict the production with an average error of 0.14 and is 1000 times faster than the FKPM simulation. No comparisons vs ZKPM

[98]	2021	Pre	Carbonate rock permeability estimation	Experimental data (1100 samples of 100x100x160 voxels) and data from literature (400 samples). Ground-truth porosity computed using Lattice Boltzmann method	Pore network model	Linear regression, Support vector regression, Gradient boosting, Random forest, Deep neural network, Convolutional neural network	PKPMs tested varying the architecture. PKPMs give a best value of R^2 of 0.87, FKPM 0.15. No comparisons vs ZKPM
[94]	2021	Pre (surrogate)	Reservoir pressure management	Synthetically generated data (variable number of samples)	Theis model	Neural network	PKPM can predict overpressure with $RMSE < 0.1 mmH_2O$. No comparisons vs FKPM and ZKPM.
[95]	2022	In(surrogate)	Porous media gas drainage	Synthetically generated data (11 saturation profiles made of 100 points each)	Buckley-Leverett model	Neural network	The best PKPM gives a $MEAresidual$ of 0.01. ZKPM neural network is comparable in interpolation, but accuracy drops in extrapolation. No comparisons vs FKPM
[96]	2022	In(surrogate)	Subsurface oil/water two-phase flow modeling	Synthetically generated data (1000 samples simulated with SGeMS's sequential Gaussian simulation)	two-phase underground oil/water flow 2D model	Fourier neural operator network	Tested in reservoir saturation prediction the proposed PKPM reduces $RMSE$ by more than 46% with respect previous PKPMs. No comparisons vs FKPM and ZKPM
[91]	2023	In (parameters tuning)	Reservoir waterflooding	Synthetically generated data (field example trained with Eclipse 100 simulations of 2400 days)	1-dimensional connective flow units model	Ensemble smoother with multiple data assimilation	Results of PKPM are comparable to FKPM (Eclipse 100) and it is 2 order of magnitude more efficient. No comparisons vs ZKPM.

Table 2.8 Papers about PKPMs in extraction industry

minimized chemical waste, and the incorporation of renewable resources. The industry is increasingly shifting toward green chemistry and biotechnology, focusing on environment-friendly and sustainable processes and products. Predictive models are essential for this progress, as they optimize production processes, forecast demand, improve material properties, and accelerate innovation, thereby reducing costs and driving forward industrial advancements.

The earliest FKPMs in the chemical industry date back to the 1960s [42]. Detailed reviews on FKPMs are available in the literature [99–101], as well as exhaustive reviews on ZKPMs [102–107] . Several thorough reviews also address PKPMs applications in the chemical industry [32, 42, 30, 38, 48], but they need to be complemented by more recent contributions not covered in these studies. This work aims to integrate the literature, focusing on papers published after 2020 that have not been included in previous reviews, thus offering an updated perspective on PKPMs advances in this area.

The search targeted specific keywords related to predictive models in the chemical industry (see [23] for more information). This search produced a more than 1000 of studies, which were refined by applying further citation filters (papers cited, according to Scopus, by ≥15 papers for works published from 2020 and 2022, by ≥10 papers for works published in 2023, and by ≥1 papers for works published in 2024).

In the chemical industry, PKPMs are predominantly applied to process modeling, targeting improvements in the production of chemicals, materials, and pharmaceuticals, as well as wastewater treatment [108–112]. Another relevant application field is property prediction for various substances [74, 113, 114]. The most common PKPMs approach involves hybrid combining a FKPM model with a separate ZKPM. In some cases, ZKPM outputs are post-processed by FKPMs [115, 116, 110, 117], while in others, the ZKPM input is pre-processed using a FKPM [111]. Another strategy is integrating FKPM and ZKPM outputs to produce a final predictions [109]. More recent methods use integrated approaches, adjusting model architectures or loss functions [112, 108, 112]. Other techniques include physics-informed feature engineering [74], zero-knowledge neural networks to estimate parameters of a FKPM [113], or multi-fidelity neural networks that combine low-fidelity FKPM

Paper	Year	PKPM	Application Domain	Available and/or Generated Data	Prior Knowledge and/or FKPM	ZKPM	Results & Comparison between FKPM, ZKPM, and PKPM
[74]	2021	Pre	Shape memory alloys design	Data of existing alloys (528 from literature and 26 from authors' lab)	Thermodynamics and kinetics of phase transformations	Gaussian process	PKPM is more accurate than ZKPM in predicting $\tilde{T}$ and T. No comparisons vs FKPM
[113]	2021	Post	Viscosity prediction	Data extracted from SciGlass database (17584 data of 847 oxide liquids)	MYEGA viscosity equation	Neural network	PKPM predicts testset with $R^2 = 0.97$. No comparisons vs FKPM and ZKPM
[109]	2021	Post	Semi-batch polymerization reactors	Synthetically generated (125 batches)	Physical equations (e.g., mass and energy balances)	Symbolic regression	PKPM is qualitatively more accurate than the simplified FKPM. No comparisons vs ZKPM
[115]	2021	Post	Chromatographic processes	Synthetically generated data (17 BT curves) and experimental data (17 BT curves)	Lumped kinetic model	Neural network	PKPM 3 times more accurate than FKPM in interpolation and extrapolation. No comparisons vs ZKPM
[116]	2021	Post	Czochralski silicon single crystal growth process	Experimental data (5000 sets of crystal growth, continuously selected at intervals of 5 data points)	Hydrodynamic and geometric model	LSTM network	Compared 3 different LSTMs inside PKPM. M-LSTM-HW is the most accurate in all the tests. No comparisons vs FKPM and ZKPM
[110]	2021	Post	Yeast astaxanthin production	Experimental data (quantity not specified)	Kinetic model	Gaussian process	PKPM has higher accuracy and lower uncertainty than FKPM, for all the predicted quantities. No comparisons vs ZKPM
[108]	2021	In(surrogate)	Catalytic CO_2 methanation	Initial and boundary conditions and synthetically generated data (Inverse problem. Different sized datasets)	Governing equations	Neural network	Once trained, PKPM is more efficient than FKPM (numerical solution). Error $< 0.3\%$ in inverse problem (parameters identification).
[117]	2022	Post	β-carotene production	Experimental data (quantity not specified)	Kinetic model	Neural network	PKPM is more accurate than FKPM, for all the predicted quantities. No comparisons vs ZKPM
[111]	2022	Pre	Wastewater treatment	Data from a full-scale WWTP (1248 data after pre-treatment)	Activated sludge model	LSTM network	PKPM is more accurate than ZKPM (MSE $-22,5\%$) and FKPM (MSE $-93,0\%$)
[118]	2022	In & Post	Industrial fermentation process	Experimental data (from a sponsoring industry, quantity not specified)	Kinetic model	Neural network	PKPM is more accurate than FKPM, for all the predicted quantities. No comparisons vs ZKPM
[114]	2023	Pre	Nanofluids viscosity prediction	Experimental data (1425 data of 19 class of nanofluids)	Viscosity theoretical models (Einstein, Batchelor, Brinkman, Masoumi, and Udawattha)	Neural network	PKPM is more accurate than ZKPM ($R^2 = 0.991$ vs $R^2 = 0.977$). No comparisons vs FKPM

[112]	2023	Pre & In	(i) Semi-batch reactor. (ii) Wastewater treatment	(i) Synthetically generated data (500 data from numerical solution). (ii) Data from literature (1345 data)	Chemical equations	Neural network and Recurrent neural network	Tested several structures of PKPM and compared with ZKPMs. PKPMs outperform ZKPMs in all the tests. No comparisons vs FKPM

Table 2.9 Papers about PKPMs in chemical industry

outputs with high-fidelity experimental data [114]. Table 2.9 summarizes the reviewed articles.

2.4.3 Manufacturing

Manufacturing transforms raw materials into finished goods through processes like design, machining, fabrication, and assembly, across various industrial segments. Challenges in the manufacturing sector include improving product design, optimizing efficiency, ensuring quality, and managing supply chain disruptions. Predictive modeling is crucial here, as it leverages physical knowledge and data analytics to boost productivity, anticipate demand, optimize inventory, and enhance operations.

FKPMs in manufacturing are often associated with numerical simulations, with comprehensive reviews available [119] and sector-specific overviews [120]. Several reviews discuss the use of ZKPMs in manufacturing [121–123]. Some reviews on PKPMs address smart manufacturing [28, 46] and additive manufacturing [31]. However, a broader overview encompassing various manufacturing subdomains has been missing so far.

For the manufacturing sector, a dedicated search query was used including keywords related to manufacturing (see [23] for more information). This search returned more than 2900 papers, which were then narrowed down by applying citation filters (papers cited, according to Scopus, by $\geq$50 papers for works published from 2013 and 2021, by $\geq$30 papers for works published in 2022, by $\geq$10 papers for works published in 2023, and by $\geq$1 papers for works published in 2024).

The selected studies indicate that PKPMs are applied to various areas such as design, quality prediction, quality control, treatments, machining, and performance prediction [124–131]. In advanced technologies, such as additive manufacturing, PKPMs support tasks like melt pool modeling [132], defect prediction [126, 133], and fatigue life forecasting [134–136]. Applications are also notable in composite materials[137, 129]. PINNs are prevalent [127, 129, 138, 139, 128, 131, 125, 124, 135], alongside approaches involving feature preprocessing [133] and surrogate modeling to reduce computational demand [126, 137]. PKPMs primarily enhance computational efficiency and reduce data

Paper	Year	PKPM	Application Domain	Available and/or Generated Data	Prior Knowledge and/or FKPM	ZKPM	Results & Comparison between FKPM, ZKPM, and PKPM
[126]	2020	Pre (surrogate)	Additive manufacturing defect prediction	Experimental (320 hatches) and synthetically generated data (3200 simulation blocks)	Numerical simulation of plastic damage model	Neural network	In the LPBF case, the $F - Score$ for bulk and overhang is 87.5% and 79.6% for PKPM and ZKPM respectively. No comparisons vs FKPM.
[137]	2020	Pre (surrogate)	Composite materials damage characterization	Synthetically generated (10000 simulations of LS-DYNA) and experimental data (3 ICT tests)	Numerical simulation of plastic damage model	Neural network	The $RMSE$ of PKPM with respect to FKPM is $1.9kJ^{kJ}/_{m^2}$ for fracture energy. No comparisons vs ZKPM.
[127]	2020	In(surrogate)	Surface cracks ultrasound quantification	Experimental data (240x240 grid, 1024 time instants, 3 different wave-crack orientations)	Acoustic wave equation	Neural network	Inverse problem. With 20% of data PKPM can predict sound speed with an error of 1%. No comparisons vs FKPM and ZKPM
[132]	2021	Pre & In	Additive manufacturing melt pool modeling	Synthetically generated data (quantity not specified)	Momentum, mass and energy conservation	Neural network	PKPM results are comparable to numerical FKPMs. No comparisons vs ZKPM
[133]	2021	Pre	Additive manufactured components porosity analysis	Experimental data (549 3D-printed components)	Physical parameters affecting the process	Linear regression, Gaussian process, Support vector regression	Accuracy of PKPMs and ZKPMs are comparable, but PKPMs are independent from the specific machine parameters. No comparisons vs FKPM
[129]	2021	In(surrogate)	Composite materials curing process	Initial and boundary conditions	Thermodynamics equations	Neural network	Max error of PKPM with respect to FKPM is $0.97°C$. PKPM can extrapolate better than ZKPM
[138]	2021	In(surrogate)	Component elastodynamics	Initial and boundary conditions	Elastodynamics equations	Neural network	Qualitative comparison of PKPMs with different hyperparameters and reference FKPM numerical solution. No comparisons vs ZKPM
[139]	2021	Pre & In (surrogate)	Composite materials curing process	Initial and boundary conditions	Heat transfer law and degree of cure equation	Neural network	PKPM compared to the FKPM numerical simulation. PKPM 10 times faster and error $< 1.6K$ and 0.007 of temperature and degree of cure respectively. No comparisons vs ZKPM
[128]	2022	Pre & In (surrogate)	Internal structure and defects of components	Different data for each case (initial and boundary conditions, internal samples)	Mechanical laws (different assumptions for different cases)	Neural network	Inverse problem. PKPM predicts the defect parameters with a max error of 6.8%. No comparisons vs FKPM and ZKPM.

[130]	2022	Pre & In	Machining tool wear	Experimental data (3 datasets of 315 data)	Tool wear empirical equation	Neural network	PKPM predicts the wear with an $RMSE$ of 3.17, 3.01, and 7.58 in x, y, and z respectively. PKPM is better than all the tested ZKPMs, no comparisons vs FKPM
[134]	2022	Pre & In	Additive manufactured components fatigue	Data from literature (12 samples)	Fracture mechanics model	Neural network	In the test, PKPM and ZKPM give $R^2 = 0.591$ and $R^2 = 0.322$ respectively. No comparisons vs FKPM
[131]	2023	In	Fatigue life prediction	Data from literature (3 materials: WIRE 85 samples, 2024-T4 252 samples, AAW 200 samples)	Fatigue model	Neural network	Measure of the physical consistency: PKPM and ZKPM has a minimum value of -0.99 and -0.53. No comparisons vs FKPM
[140]	2023	In(surrogate)	Finite-strain plasticity modeling	Initial and boundary conditions	Finite-strain elastoplasticity model	Neural network	3 step loading test: PKPM gives an error of 3.58% with respect to the FKPM. No comparisons vs ZKPM
[125]	2023	In(surrogate)	Topological optimization	Boundary conditions	Deep energy method	Neural network	Images show a good agreement between PKPM and FKPM. PKPM is less sensitive to the number of nodes
[124]	2023	In(surrogate)	Topological optimization	Boundary conditions	Total potential energy model	Neural network	In the 3D example the difference between PKPM and FKPM is less than 5%. No comparisons vs ZKPM
[135]	2023	Pre & In	Additive manufactured components fatigue	Data from literature (561)	Fracture mechanics model	Neural network	PKPM more interpretable than ZKPM. PKPM gives an MSE 20% lower than ZKPM. No comparisons vs FKPM
[136]	2023	(i) Pre (ii) Post	Additive manufactured components fatigue	Experimental data (62 tests)	Paris' law, $\delta\sigma$ model	Neural network, Support vector regression	Several PKPM architectures tested. $RMSE$ given by all PKPMs is less than FKPMs and ZKPMs

Table 2.10 Papers about PKPMs in manufacturing industry

dependency compared to traditional approaches. Additionally, PKPMs have been employed for solving inverse problem (i.e., parameter identification) [128] and improving model generalization [133]. See Table 2.10 for a summary.

2.4.4 Transportation

The transportation industry underpins global commerce, involving the movement of goods and passengers. Key sectors include automotive, marine, aerospace, and railway, as well as traffic management and logistics. Predictive models help forecast trends, optimize operations, enhance sustainability, and reduce risks. In the automotive industry, predictive models support demand forecasting, maintenance planning, and autonomous driving systems development. In the marine sector, they improve vessel routing, predict weather patterns, and strengthen navigation safety. In aerospace, predictive modeling drives better aircraft design, operational optimization, and maintenance scheduling. Within the rail domain, these models facilitate efficient scheduling, maintenance planning, and infrastructure management, ensuring reliable and eco-friendly rail networks.

General reviews of FKPM applications in transportation are available [141], as are numerous surveys on ZKPM use [142, 143]. Work on PKPMs in this field remains scarce. Some reviews exist for specialized areas like fuel consumption [34, 52] or ship motion [144], but no overview spans the entire transportation sector.

To address this, a targeted search was conducted (see [23] for more information), which returned more than 1200 papers and was subsequently filtered by citation thresholds (papers cited, according to Scopus, by $\geq$25 papers for works published from 2013 and 2021, by $\geq$15 papers for works published in 2022, by $\geq$10 papers for works published in 2023, and by $\geq$1 papers for works published in 2024).

The findings reveal significant exploration of PKPMs in the marine sector for ship operations optimization. Studies address fuel consumption reduction [145–147], ship behavior modeling [148–151], and integrating emerging propulsion solutions [152]. PKPMs are also used to model battery behavior for vehicle applications [153, 154] and to analyze cumulative aircraft damage [155, 156].

Paper	Year	PKPM	Application Domain	Available and/or Generated Data	Prior Knowledge and/or FKPM	ZKPM	Results & Comparison between FKPM, ZKPM, and PKPM
[145]	2015	In	Ship efficiency estimation	Performance evaluated with several different dataset sizes	Ship fuel consumption model	Kernel method	The best PKPM predicts the fuel consumption with $MAPE = 1.49\%$, FKPM with 20.95% and ZKPM (best) with 1.71%
[149]	2015	In (parameters tuning)	Ship maneuvering model	Experimental data (quantity not specified)	Dynamical model	Least-squares support vector regression	Qualitatively PKPM is comparable to ZKPM but less accurate than FKPM
[148]	2015	In (parameters tuning)	Ship maneuvering model	Synthetically generated data (graphs shows a simulation of about 400s)	Dynamical model	ϵ support vector regression	PKPM is more accurate than FKPM and ZKPM in all the tests
[146]	2017	(i) Pre (ii) In	Ship trim optimization for fuel consumption	Performance evaluated with several different sized datasets	Ship fuel consumption model	Regularized least squares, lasso regression, and random forest	Fuel consumptions $MAPE$: FKPM 20.95%, best ZKPM 1.95%, best PKPM Pre-Processing 0.83%, , best PKPM In-Processing 0.95%
[147]	2019	In (parameters tuning)	Ship fuel consumption forecast	Data sampled in 7 years of ship operation (daily sampling)	Fuel consumption model	Genetic algorithm	The proposed PKPM predicts the consumption with $R^2 = 0.90$ (a previous PKPM with $R^2 = 0.88$). No comparisons vs FKPM and ZKPM
[61]	2020	Pre	Propeller noise prediction	Experimental data (258 tests in the cavitation tunnel of University of Genoa)	Cavitation vortex and high-frequency noise models	Neural network	In the interpolation test MAE of PKPM is -20% and -82% with respect to ZKPM and FKPM respectively. In the extrapolation test MAE of PKPM is -29% and -69% with respect to ZKPM and FKPM respectively.
[155]	2020	Pre & In	Wings cumulative damage modeling	Synthetically generated data (15 planes, 174000 cycles each)	Paris' law	Recurrent neural network	The accuracy of PKPM on the test set is 96.55%. No comparisons vs FKPM and ZKPM
[159]	2020	In	Traffic state estimation	Synthetically generated data (1000 samples extracted from a 500x500 spatiotemporal grid)	Lighthill-Whitham-Richards model	Neural network	Eulerian test accuracy: PKPM 73.7% and ZKPM 37%. Lagrangian test accuracy: PKPM 79% and ZKPM 86.1%. PKPM $>$ 2 times faster than ZKPM. No comparisons vs FKPM
[158]	2021	In	Port state control	Total of 1,974 records (Hong Kong port)	Monotonicity regarding ship flag/recognized organization/company performance	Extreme gradient boosting	PKPM improves by 20% the current scheduling method. PKPM outperforms the ZKPM XGBoost

[150]	2021	Pre	Ship maneuvering model	Synthetically generated (70 simulations of 600 samples) and experimental data (quantity not specified)	Hydrodynamic model	Neural network	In simulation PKPM predicts the trajectory with an error 90% lower than FKPM. No comparisons vs ZKPM
[153]	2021	Pre & In	Unmanned aerial vehicle batteries modeling	Data from NASA's prognostic Center of Excellence Data Repository (36 discharge curves)	Electrochemical model	Recurrent neural network	Tested on random loading cases, PKPM can predict voltage with $RMSE < 0,088\%$ for 7 batteries out of 8. No comparisons vs FKPM and ZKPM
[154]	2023	Pre	Vehicle batteries heat generation estimation	Virtual battery data for five standard driving tests	Single particle model with thermodynamics	LSTM network	Comparison of different architectures of PKPM. The best can predict heat generation with an $RMSE = 1.428$ for the WLTP standard. No comparisons vs FKPM and ZKPM
[156]	2023	Pre	Aerospace alloys fatigue modeling	Experimental data (500 samples)	Basquin model, Smith-Watson-Topper model, Neuber's rule	Support vector machine, random forest, extreme gradient boosting	PKPM improves by 2% the predictions that lie in the ± 2 error bands. Graphs shows that the MSE of PKPM is about 30% lower than FKPM. No comparisons vs ZKPM
[152]	2023	Pre	Safety assessment of solid oxide fuel cell for ships	Synthetically generated and experimental data (4392 samples)	Fuel cell electrochemical model	Gradient boosting	In predicting leaks, PKPM has an accuracy 11% better than ZKPM. No comparisons vs FKPM
[151]	2024	In & Post	Ship speed prediction	Experimental data (from two ships 63000 and 8600 samples respectively)	P-v curve in calm water	Neural network, extreme gradient boosting	PKPM is 30% more accurate than ZKPM. No comparisons vs FKPM

Table 2.11 Papers about PKPMs in transportation industry

Common PKPMs strategies include physics-inspired model structures with data-driven parameter estimation [148, 149, 147] and physics-informed regularization [145, 157] and architechtures [155, 154]. These approaches primarily enhance accuracy and allow knowledge-based constraints [158], as well as reduce computational time [159]. Table 2.11 presents the reviewed papers.

2.4.5 Energy

The energy industry underlies modern civilization, powering cities and industries. It involves the production, storage, and distribution of energy. This sector increasingly relies on nuclear and renewable sources (e.g., solar, wind, hydro) alongside traditional fossil fuels. Together with the management of the distribution of energy generated from these source, the energy storage technologies (batteries, pumped hydro, hydrogen fuel cells) are becoming essential for balancing supply and demand, ensuring grid stability, and integrating intermittent renewables. Predictive modeling supports energy production forecasting, consumption patterns, component maintenance scheduling, and overall operational efficiency.

Several reviews cover FKPM applications in energy [160–162]. There are also studies on the use of ZKPM in smart energy management [163] and other specific fields like renewables and batteries [164, 165]. For PKPMs, there is a growing interest in batteries and renewables [45, 31]. [30] focus on parallel and serial combinations of FKPMs and ZKPMs, while [49] is devoted to PKPMs for power systems.

The literature analysis identified relevant papers by using specific keywords (see [23] for more information), yielding a total of 2900 items. Then, papers and was filtered by citation (papers cited, according to Scopus, by $\geq$30 papers for works published from 2013 and 2021, by $\geq$20 papers for works published in 2022, by $\geq$10 papers for works published in 2023, and by $\geq$1 papers for works published in 2024).

Many works address batteries and renewables [45], in particular battery state-of-charge and state-of-health estimation [166–170], wind turbine maintenance [171, 172], photovoltaic production forecast [173], as well as fuel cell [174], and nuclear reactor [175] modeling. The most common type of PKPMs in

Paper	Year	PKPM	Application Domain	Available and/or Generated Data	Prior Knowledge and/or FKPM	ZKPM	Results & Comparison between FKPM, ZKPM, and PKPM
[166]	2017	Post	Modeling SOC and temperature of batteries	Experimental data (PCD, HPPC e CCD tests. Quantity not specified)	Equivalent circuit and thermal models	Neural network	PKPM tested varying conditions and hyperparameters. PKPM validated for practical battery management systems. No comparisons vs FKPM and ZKPM
[178]	2020	Post	Critical heat flux	Data from literature (1865 test cases)	Liu model	Neural network, Random forest	$rRMSE$ is 5.5%, 30.9%, and 13.6% for the best PKPM, FKPM, and ZKPM respectively
[171]	2020	Pre & In	Wind turbine bearings	Data from literature (NREL database, hourly from 2007 to 2013) and synthetically generated (bearing life curves)	Bearing models	LSTM network	PKPM qualitatively has a good adherence with the bearing damage curve. No comparisons vs FKPM and ZKPM
[167]	2021	Pre (surrogate)	Battery electrode-level state estimation	Synthetically generated data (75 simulations of power profiles)	Electrochemical and thermal models	LSTM network	The maximum value of $RMSPE$ for all the quantities predicted by PKPM is 2.93% (without noise). Improvement between 75% and 95% with respect to ZKPM. No comparisons vs FKPM
[176]	2021	Pre	Prediction of polarization curves of solid oxide fuel cell anodes	Data from literature (quantity not specified)	Fundamental conservation laws of mass and electrical charge	Neural network	PKPM has a MSE that is 63% lower than FKPM. No comparisons vs ZKPM
[172]	2022	Pre & In	Wind turbine bearings	Data from literature (NREL database, hourly from 2007 to 2013) and synthetically generated (bearing life curves)	Bearing models	LSTM network	PKPM trained with different sized datasets. Best $RMSE$ for grease damage is 0.021. Qualitatively better than ZKPM. No comparisons vs FKPM
[175]	2022	Pre (surrogate)	Nuclear reactor modeling	Synthetically generated data (18480 samples)	Neutronic field simulation	Decision tree, k-nearest-neighbors	In the forward case, the PKPMs have a reconstruction error of 2.7% with respect to the FKPM simulation. PKPM is 3 orders of magnitude faster. No comparisons vs ZKPM
[168]	2022	Pre	Battery health management	Data from NASA Ames Prognostics Center of Excellence (9 different datasets)	Calendar and cycle aging models	LSTM network	PKPM more accurate than FKPM and ZKPM (Best $MAPE$: PKPM 0.92%, FKPM 3.59%, and ZKPM 4.12%)
[173]	2023	Pre	Photovoltaic power forecasting	Experimental data (from 14 plants) and numerical (weather predictions)	PV model chain	Neural network	With data sampled for more than 1 year, PKPM more accurate than FKPM. No comparisons vs ZKPM

[170]	2023	Pre	Battery health prognosis	(i) Data from NASA (first 90 life cycles of 3 batteries and first 60 life cycles of 1 battery). (ii) Experimental data (1400 life cycles)	Pseudo 2-dimensional electrochemical model	LSTM network	(i) PKPM gives $R^2 > 0.711$. (ii) PKPM gives $R^2 > 0.989$. No comparisons vs FKPM and ZKPM
[169]	2023	Pre & Post	Battery voltage modeling	(i) Synthetically generated data (DFN model). (ii) Experimental data (commercial battery). Quantity not specified	(i) Single particle model with thermal dynamics. (ii) Non-linear double capacitor equivalent circuit model	Neural network	(i) PKPM more accurate than FKPM in all the tests. (ii) PKPM more accurate than FKPM and ZKPM in all the tests
[179]	2024	Pre	Power system load margin prediction	(i) Synthetically generated data (1000 samples from nominal IEEE 68-bus). (ii) Synthetically generated data (1000 samples from nominal BIPS)	Load margin assessment model	Neural network	PKPM more accurate and robust than ZKPM. PKPM is slower than ZKPM in training. Similar in prediction times. No comparisons vs FKPM
[174]	2024	Pre (surrogate)	Solid oxide fuel cells operating condition optimization	Synthetically generated data (50000 simulated data)	Local overpotential model	Neural network	PKPM 10000 times faster than FKPM. Error 0.513% and 2.86% with respect to FKPM and experiments, respectively. No comparisons vs ZKPM
[177]	2024	Pre	Battery remaining useful life prediction	Data from literature (72 data after cleaning)	Constant current charging process model	Neural network	PKPM gives a $MARE = 3.19\%$, 44% less than ZKPM. No comparisons vs FKPM

Table 2.12 Papers about PKPMs in energy industry

the examined literature is the feeding of ZKPMs with additional features outputted by FKPMs [168, 173, 169, 170, 176, 177]. The second most adopted PKPMs is the surrogation of heavy computational FKPMs with more efficient PKPMs [167, 175, 174]. Table 2.12 summarizes the reviewed papers.

2.4.6 Construction

The construction industry includes diverse activities from residential and commercial building, to infrastructure development. Energy efficiency in buildings is increasingly critical, driven by environmental and regulatory factors, and involves implementing advanced insulation, efficient lighting, and renewable energy sources. Predictive modeling is crucial for optimizing resource use, improving project planning, mitigating risks, and ensuring sustainable building life cycles.

We found extensive reviews of FKPM applications to building industry [180], as well as works on structural optimization in civil engineering [181]. For ZKPMs, some reviews analyze the building life cycle management [182], while other surveys study structural health monitoring [183]. Several reviews examine PKPMs in building efficiency [29, 33, 36, 39, 41], and there are broader perspectives on civil engineering [47]. Here, an updated perspective on PKPMs in construction is presented, excluding previously covered works.

We used a dedicated search string (see [23] for more information), resulting in more than 1800 articles. Then we screened the results by applying citation filters (papers cited, according to Scopus, by $\geq$40 papers for works published from 2013 and 2021, by $\geq$20 papers for works published in 2022, by $\geq$10 papers for works published in 2023, and by $\geq$1 papers for works published in 2024).

Our findings show that most PKPMs address building efficiency and thermal modeling [184–189], along with structural integrity [190–192] and soil-structure interaction [193, 194]. Common approaches involve incorporating physics-based regularization into the loss function [190, 191, 187, 188, 194], or using physics-guided structures with data-driven parameter fitting [184, 189]. Most of these classes of methods have been discussed in previous surveys [29, 33, 36, 39, 41]. Table 2.13 summarizes the reviewed studies.

Paper	Year	PKPM	Application Domain	Available and/or Generated Data	Prior Knowledge and/or FKPM	ZKPM	Results & Comparison between FKPM, ZKPM, and PKPM
[184]	2018	In (parameters tuning)	Building thermal modeling	Experimental data (measurements of 31 days)	RC model	Genetic algorithm and pattern search algorithm	The best FKPM, ZKPM, and PKPM give an average MAE of 0.7°C, 0.4°C, and 1.0°C respectively
[190]	2020	In	Building seismic response modeling	(i) Synthetically generated data (100 seismic sequences of 1001 data). (ii) Experimental data (23 seismic events)	Equations of motion	Neural network	(i) In the worst scenario PKPM has a correlation factor $r = 0.61$, ZKPM 0.37. (ii) The error of PKPM in predicting the displacement is $< 5\%$ with a confidence interval $> 93\%$. No comparisons vs FKPM
[191]	2021	Pre & In	Structural damage identification	(i) Synthetically generated data (100 simulations of damage cases). (ii) Experimental data (60 shaking tests)	Mechanics of structures	Neural network	(i) 97.5% of the errors of PKPM are < 0.2. 22.3% of errors of ZKPM are > 1. (ii) 93.3% of the errors of PKPM are < 0.1. 20% of errors of ZKPM are > 0.2. No comparisons vs FKPM
[185]	2022	Pre & In	Building thermal modeling	Synthetically generated data (3672 simulation samples)	Heat transfer law	Multiple linear regression	PKPM has a mean $RMSE$ of 81.312, the best ZKPM 92.836. No comparisons vs FKPM
[186]	2022	Pre & In	Building thermal modeling	Experimental data (training set of 370 days of measurements)	Laws of thermodynamics	Linear regression	In the test PKPM gives $MAE < 0.1$, while ZKPM $MAE > 0.1$. No comparisons vs FKPM
[187]	2022	Pre & In	Building thermal modeling	Experimental data (125 days, 48 samples per day)	RC model	Neural network	PKPM predicts T with and error $<$ 0.5°C, $60 - 70\%$ better than ZKPM. No comparisons vs FKPM
[188]	2023	In	Building thermal modeling	Experimental data (236 samples for each scenario)	RC model	Neural network	In the worst scenario PKPM predicts T with $MAE = 0.25$, ZKPM with 0.47. No comparisons vs FKPM
[193]	2024	Post	Tunnel settlement analysis	(i) Synthetically generated data (simulations of 10 load cases). (ii) Experimental data (12 sets of measurements available)	Multi-beam model	Neural network	(i) PKPM predicts foundation modulus with $R^2 > 0.973$ (ii) PKPM predicts settlement with $R^2 = 0.895$. No comparisons vs FKPM and ZKPM.
[192]	2024	Post	Structure full-field response modeling	(i) Synthetically generated data (time series of 12000 samples). (ii) Synthetically generated data (time series of 11360 samples)	Impulse response matrix	Convolutional neural network	(i) PKPM predicts the displacement with $RMSE = 1.41\%$, FKPM 4.37%. (ii) PKPM predicts the displacement with $RMSE = 3.61\%$. No comparisons vs ZKPM

[194]	2024	In	Structure interaction with soil	Boundary conditions	Laterally loaded pile governing equations	Neural network	PKPM is comparable to FKPM FEM simulation, but 3000 times faster. No comparisons vs ZKPM
[189]	2024	In (parameters tuning)	Building thermal modeling	Synthetically generated data (measurements of 37 days in summer and 37 in winter)	Single-zone building model	Trust Region Reflective algorithm	PKPM compared to ZKPM in different scenarios. Results are comparable. PKPM requires less data. No comparisons vs FKPM

Table 2.13 Papers about PKPMs in construction industry

Chapter 3

An application of Knowledge-Informed Predictive Models

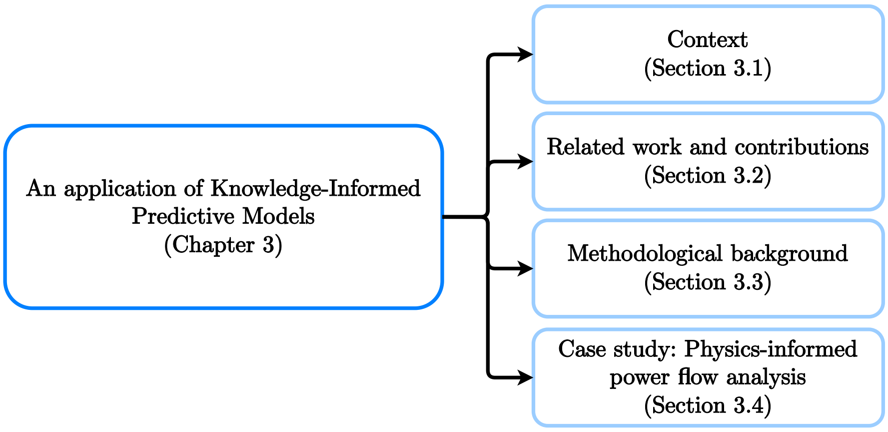

Fig. 3.1 Graphical index of Chapter 3

In the previous chapter, we discussed the state of the art of Knowledge-Informed Models, and reviewed the practical applications already adopted by industry. Given that they are, to date, the only models with a sufficient level of maturity to have been applied in industrial contexts, our review was limited to Knowledge-Informed Predictive Models (also referred to as Partial-Knowledge Predictive Models). To complete the picture of these models, it is now necessary to analyze the deployment in real-world cases. Therefore, this chapter introduces a practical example of a Knowledge-Informed Predictive Model, detailing its development and presenting a fair comparison with traditional techniques.

3.1 Context

Practical applications of Knowledge-Informed Predictive Models represent the core of the doctoral program. Two main projects were undertaken: one in the energy and one in the maritime industries. Here, we will present the project related to the energy industry, namely the application of Knowledge-Informed Predictive Models to the power flow analysis of an electrical grid. As for the maritime application, the project is still ongoing, and we leave this discussion to future work.

The choice to apply our framework to the power flow analysis stems from the growing attention this field is receiving, due to the increasing complexity of electrical grids and the need for effective tools to control such systems. Power flow analysis aims to determine voltage and current (both in magnitude and phase) in an electrical grid over time, given load conditions, power injections, and the grid topology. This problem, which emerged with the spread of electrical networks, is not new, and power distributors have long developed models capable of managing grid dynamics. However, with the recent introduction of decentralized renewable energy sources, storage systems, microgrids, and growing variability in demand, power grids are becoming increasingly complex, and traditional models are no longer sufficient for fast and accurate analysis. More than ever, real-time knowledge of the grid state is essential to manage renewable resources and their intermittent production, coordinate the interaction between traditional and decentralized sources, and optimize distribution. Today, energy providers face the need to create new and more complex models to adapt to this evolving complexity, which translates into significant investments in research and development.

This case study appears to be a perfect match for the application of the Knowledge-Informed Models. Indeed, the physics of the problem is well known and these physical models are proven to align with practical observations. Moreover, reliable software tools for iteratively solving the problem already exist, and there are large databases of historical data and several benchmarks available. All the ingredients are present to build a data-driven model enriched with domain knowledge. To demonstrate the feasibility of using these models in real-world applications, we chose the IEEE 57-bus case study. This network is a widely used benchmark. It features a medium-complexity network, allowing for meaningful comparisons on a non-trivial case and prepare scalability to larger systems.

The presentation of this project is organized as follows:

- Section 3.2 provides an overview of the related work, introducing traditional models for power flow analysis, their strengths and weaknesses, and existing attempts to combine physics and historical data to build better models.
- Section 3.3 introduces the techniques used in this work, to help the reader follow every step of the experiment.
- Section 3.4 presents the actual case study, defining the setting and discussing the results.

The experiments for this project were conducted on the DelftBlue supercomputer at the Delft High-Performance Computing Center, and the results have been published in [24].

3.2 Related work and contributions

Regarding the modeling of electrical grids, the physical principles behind power flows in these are well known and closely match real-world observations [195]. As we will describe in more detail in Section 3.3.1, power flow analysis can be modeled as a system of non-linear equations dependent on the injected power, the loads, and the topology. Traditionally, the solution of the system is obtained using numerical iterative methods, which are highly accurate but not always guaranteed to converge, and require significant time and computational resources [196]. This often prevents real-time predictions which are necessary for control tasks. Examples of solving techniques include the Newton-Raphson method, Gauss-Seidel method, and Fast Decoupled Load Flow [197]. All these methods rely solely on physical and domain knowledge and thus fall under the category of FKPMs, as defined in the previous chapter.

A more efficient solution involves collecting a dataset of historical data (either real or generated by solving a set of conditions using an iterative method) and training a data-driven model (referred to as ZKPM in the previous chapter), able to predict the voltage state of the network based on the power injected or requested at the nodes. Examples have been proposed using Least Squares and Bayesian Linear Regression [198], Koopman Operator Methods [199], Support Vector Machines [200], Radial Basis Function Networks [201], Graph Convolutional Networks [202], and Stacked Denoising Auto-Encoders [203]. Generally speaking, these methods require a sufficient amount of historical data and a time-consuming training phase, which can typically be performed offline. This class of methods proves extremely useful for real-time applications because, once trained, their inference time is significantly shorter compared to iterative methods. Nonetheless, the quality of the results from data-driven models tends to be low and potentially physically inconsistent.

In this context, Knowledge-Informed Predictive Models (called PKPM in the previous chapter) emerge as a promising approach, combining the efficiency of data-driven models with improved accuracy and, most importantly, enforcing compliance with physical laws to ensure physically reliable results. In the past, a few attempts have been made to develop Knowledge-Informed Predictive Models [49], but these only partially incorporate physical knowledge and do not provide a fair comparison with competing models. For instance, the paper by Hu et al. [204] modifies the architecture to account for Kirchhoff's laws and system topology (i.e., Pre-Processing); Müller et al. [205] use physical laws to generate new features (i.e., Pre-Processing); Yang et al. [206] propose a basic Physics-Informed Regularization term (i.e., In-Processing).

Our work aims to bridge the gap in previous research by proposing an extensive integration of physical knowledge through regularization and physically-consistent extension of the set of collocation points (i.e., the points where the physical law is evaluated). Moreover, for the first time, a fair comparison between knowledge-informed and traditional techniques is provided. Since the type of knowledge that is leveraged is a set of physical laws, we can define this method as a Physics-Informed Predictive Model.

3.3 Methodological background

This section introduces the key concepts necessary to understand the steps of the presented project. Specifically, we will cover:

- Power flow problem (Section 3.3.1).
- Physics-Informed Regularization and collocation points (Section 3.3.2).

3.3.1 Power flow problem

This section introduces the physical model that describes the behavior of a power grid. A power grid can be modeled as an undirected graph $\mathtt{G}(\mathfrak{N}, \mathfrak{E})$, in which $\mathfrak{N}$ is the set of nodes (buses) and $\mathfrak{E}$ represents the set of edges (branches) connecting them. At each node $n \in \mathfrak{N}$, it is possible to identify the following quantities, defined as positive when injected into the node [207]:

- Active power injection P_n.
- Reactive power injection Q_n.
- Voltage V_n, composed of a real component V_n^R and imaginary component V_n^I.
- Current injection I_n, composed of a real component I_n^R and imaginary component I_n^I.

Active and reactive power injections are typically easier to measure in practice, while voltage and current require to be estimated. Therefore, the power flow analysis aims to find the value of V_n^R, V_n^I, I_n^R, and I_n^Iat each node $n \in \mathfrak{N}$.

The grid model can be built by leveraging electric engineering principles. First, we can introduce the admittance matrix [208], defined as $Y = Y^{br} + Y^{sh}$, where Y^{br} and Y^{sh} are the branch and shunt matrices, respectively. The branch matrix can be built by modeling each edge (i,k) as a π-equivalent branch model. Each network line is transformed into the equivalent series of impedance $Z^{\mathfrak{E}}i,k = R^{\mathfrak{E}}i,k + jX^{\mathfrak{E}}i,k$ and the corresponding admittance $Y^{\mathfrak{E}}i,k = (Z^{\mathfrak{E}})^{-1}_{i,k}$. The branch admittance matrix Y^{br} becomes:

$$Y^{br}_{i,k} = \begin{cases} \sum_{w:(i,w)\in\mathfrak{E}} Y^{\mathfrak{E}}_{i,w} & \text{if } i = k \\ -Y_{i,k} & \text{if } i \neq k \text{ and } (i,k) \in \mathfrak{E} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

To build the shunt admittance matrix Y^{sh}, shunt-connected components, such as capacitors or inductors, are modeled as fixed impedances to ground at a node. The admittance of a shunt element i at bus $n \in \mathfrak{N}$ is given by $Y_n^{\mathfrak{N}} = G_n^{\mathfrak{N}} + jB_n^{\mathfrak{N}}$, where $G_n^{\mathfrak{N}}$ and $B_n^{\mathfrak{N}}$ represent the shunt conductance and susceptance, respectively. Then, Y^{sh} is defined as:

$$Y_{i,k}^{sh} = \begin{cases} Y_i^{\mathfrak{N}} & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Given the number of nodes N_{nodes}, the matrix Y has dimensions $N_{nodes} \times N_{nodes}$, where each row corresponds to a specific node n. Y contains information about the grid topology and its electrical characteristics.

To model the power exchange, each generator connected at node $n \in \mathfrak{N}$ is modeled as a complex power injection $S_n^g = P_n^g + jQ_n^g$, where P_n^g and Q_n^g are the active and reactive power supplied. Similarly, power loads are modeled as active and reactive power consumption, P_n^d and Q_n^d, such that $S_n^d = P_n^d + jQ_n^d$.

The power flow equations, which describe the steady-state sinusoidal behavior of the network, are expressed as:

$$I = YV \tag{3.3}$$

$$S = V \odot I^H \tag{3.4}$$

where I and V are the vectors of bus currents and voltages, respectively, and $\odot$ denotes the element-wise (Hadamard) product.

The power balance at each bus is enforced by equating the net injected power to the sum of the generator and load injections, leading to the nodal power balance equations:

$$\begin{aligned} P_n + P_n^d + P_n^g = 0 \\ Q_n + Q_n^d + Q_n^g = 0 \end{aligned} \tag{3.5}$$

for all $n \in \mathfrak{N}$.

We can also write, in terms of voltage:

$$V_n \;=\; V_n^R + jV_n^I, \;=\; |V_n|(\cos(\angle V_n) + j\sin(\angle V_n)) \tag{3.6}$$

yielding a formulation in terms of the real and imaginary components of voltage:

$$
\begin{aligned}
P_n &= \sum_{k \in N} G_{n,k}(V_n^R V_k^R + V_n^I V_k^I) + B_{n,k}(V_n^I V_k^R - V_n^R V_k^I) \\
Q_n &= \sum_{k \in N} G_{n,k}(V_n^I V_k^R - V_n^R V_k^I) - B_{n,k}(V_n^R V_k^R + V_n^I V_k^I)
\end{aligned} \tag{3.7}
$$

Alternatively, using the voltage magnitude and phase angle:

$$
\begin{aligned}
P_n &= \textstyle\sum_{k \in N} |V_n||V_k| \left((G_{n,k} + jB_{n,k})e^{-j(\angle V_n - \angle V_k)}\right)^R \\
Q_n &= -\textstyle\sum_{k \in N} |V_n||V_k| \left((G_{n,k} + jB_{n,k})e^{-j(\angle V_n - \angle V_k)}\right)^I
\end{aligned} \tag{3.8}
$$

Or, in terms of the real and imaginary parts of voltage and current:

$$
\begin{aligned}
P_n &= V_n^R I_n^R + V_n^I I_n^I \\
Q_n &= -V_n^R I_n^I + V_n^I I_n^R
\end{aligned} \tag{3.9}
$$

for all $n \in \mathfrak{N}$.

These non-linear equations (i.e., (3.7), (3.8), and (3.9)) can be solved using various iterative numerical techniques [197]. For this work, the open-source tools MATPOWER [209] was used. Typically, iterative numerical solvers yield to accurate results, but present two main drawbacks. First, the solution process is time-consuming and often prevents real-time predictions. The second negative aspect is the convergence of the solution which is not guaranteed for ill-conditioned problems. These considerations make the research of alternative methods particularly important.

3.3.2 Physics-Informed Regularization

In Section 2.2.4, we provided a theoretical overview of the most common techniques for integrating knowledge into a data-driven model. We emphasized that no technique is universally superior; rather, the most appropriate method varies depending on the specific case, the ML architecture, and the type of knowledge available.

As we will explain in Section 3.4, the structure of this problem suggests adopting a neural network as the ML architecture. We have also shown that, in this setting, knowledge is expressed in the form of physical laws. With this type of model and this form of knowledge, the most suitable integration technique, based on the authors' understanding, is to introduce a penalty term into the model loss function, as proposed in Eq. (2.9) of Section 2.2.4.

This method can be explained in more detail by considering a conventional ML model, characterized by a set of parameters θ, which receives certain inputs $\mathbf{x}$ and produces the corresponding outputs $\mathbf{y}$. Given a dataset $\mathcal{D}$ of historical data, it is possible to define a loss function L as a measure of the error made by the model on the training set. Typically, the definition of L includes a regularization term that penalizes overly complex functions, promoting better generalization. Training the model means solving the following objective function:

$$\hat{\theta} = \arg\min_{\theta} \mathrm{L}(\theta, \mathcal{D}) \tag{3.10}$$

The introduction of a further physics-guided penalty term L_{phy} to L, leads to a new loss term L', defined as

$$\mathrm{L}' = \mathrm{L} + \lambda \mathrm{L}_{phy} \tag{3.11}$$

The multiplying factor λ is an additional hyperparameter, weighing the physical constraint component against the rest of the loss. L_{phy} represents a measure of how the model violates a physical principles. This approach, first introduced by Raissi et al. [22] for solving differential equations, is effective when the physical law is well mathematically defined, as it guides the model training toward functions that are both consistent with the data and physically plausible. The constraint is not imposed in a hard manner, but its strength is rather adjusted through the factor λ.

The definition of L_{phy} depends on the application. In the simplest form, when the physical law can be written in the form $\Phi(x, y) = 0$, it is possible to use the sum of the squared values of Φ evaluated over a set of points as the penalty term. This can be expressed as

$$\mathrm{L}_{phy} = \sum_{i=1}^{N_c} \|\Phi(x_i, y(x_i))\|^2 \tag{3.12}$$

x_i are the so-called collocation points, $y(x_i)$ is the model output at these points, and N_c is the total number of collocation points. Practically speaking, these are the points in which the physical constraint is imposed.

In more complex cases, the physical equation might depend on derivatives of the output with respect to the input. This scenario can be handled using automatic differentiation tools [7].

The choice of collocation points introduces a new hyperparameter, which plays a crucial role in model performance. Overall constraint satisfaction, training convergence, and computational cost, they all strictly depend on this choice. With all other parameters fixed, a small and sparsely distributed set of collocation points results in low computational cost and easier convergence, but the physical law may only be satisfied locally, without generalization. Conversely, a very dense grid might ensure better constraint satisfaction, but can lead to convergence issues and higher computational cost.

When the input is made up of spatio-temporal coordinates, practitioners typically use equally spaced grids or optimized grids (e.g., for a better convergence or a finer result in regions of interest). However, when the input quantities are not spatio-temporal coordinates (such as electrical powers, as in our case), the selection of the collocation points requires further attention. One common solution to this problem is to start from the input values present in the training set and apply small perturbations to generate new, distinct points while keeping them within the same range of interest. This is the strategy adopted in our project: the set of collocation points was obtained from the training set input data, and then augmented by adding N_{aug} new points created by introducing Gaussian noise $\mathcal{N}_S(\sigma)$ to the existing ones. The amount of noise added must ensure that the new points are sufficiently different, yet still lie within the same range as the original inputs. This guarantees that the physical equation is evaluated over a consistent and relevant domain. The number N_{aug} of added points and the σ factor, which scales the noise magnitude, are treated as hyperparameters to be optimized.

This completes the introduction of preliminary concepts that will be combined together in the following section, in the development of a real model.

3.4 Case study: Physics-informed power flow analysis

This section introduces the case study, namely the application of physics-informed techniques to power flow analysis. We have already discussed the importance of power flow analysis in the management of modern electrical grids, as well as the increasing complexity that calls for the development of more efficient methods. We will now bring together the concepts presented in Section 3.3 to outline the application to the IEEE 57-bus case study.

3.4.1 Experimental setup

The IEEE 57-bus system is a classic case study introduced to evaluate the performance of different models in a realistic scenario. It represents an approximation of the American Electric Power system (in the U.S. Midwest) as it was in the 1960s. Over the years, it has been used as a benchmark for iterative solvers, making it a suitable case study for our application.

This network consists of 57 buses, 7 generators, and 42 loads. The topology of the model is shown in Figure 3.2. The power flow analysis of this problem involves determining the voltage and current states (both magnitude and phase), given the input power conditions at the nodes. The resolution of this problem using physics-based models and iterative methods is already well established. However, each modification to the input set requires a new iterative solution, which takes a non-negligible amount of time. For real-time applications, alternative models, such as surrogates, need to be developed.

The goal of our project is to develop a surrogate model using a new physics-informed technique and compare it with traditional techniques, both physics-based and data-driven. The ultimate objective is to produce a model that is accurate, physically consistent, and, most importantly, efficient.

First, we will define the dataset, which for a surrogate model is obtained through multiple solutions of the accurate but inefficient model to be replaced. In our case, the reference iterative tool is MATPOWER [209], which was used to create the dataset. The generated dataset consists of 10000 samples, containing

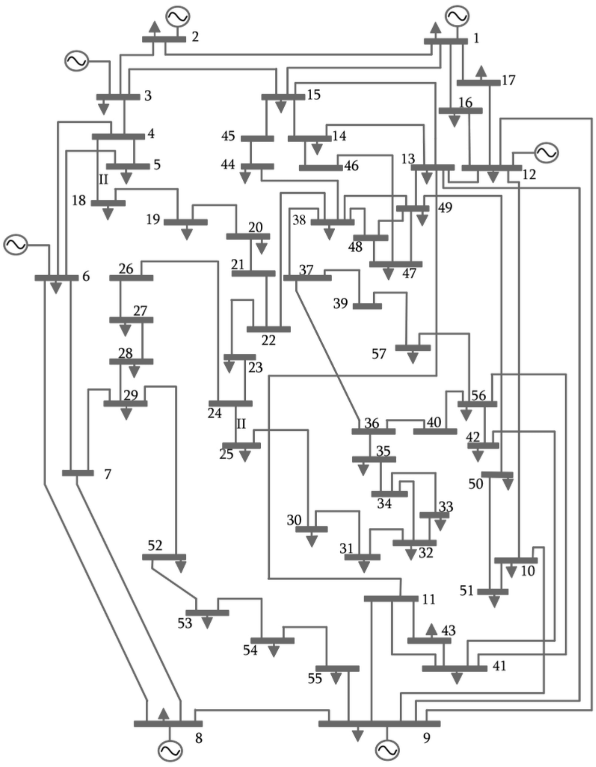

Fig. 3.2 IEEE 57-bus

the nodal voltage and current values (i.e., V_n^R, V_n^I, I_n^R, and I_n^I $\forall n \in \mathfrak{N}$) , and their corresponding nodal power values (i.e., P_n and Q_n $\forall n \in \mathfrak{N}$). To work with more realistic data, we added random noise of ± 0.2 per unit, using a power base of $100MVA$.

Having defined the physical model and the dataset created from it, we will analyze the architecture of the surrogates, starting from a simple data-driven model and then introducing the proposed physics-informed adaptation. Finally, we will evaluate the results, in order to assess performance in terms of accuracy and efficiency.

3.4.2 Method implementation

A physical model is available that can be solved iteratively with accurate but computationally expensive results. Using this model, we created a dataset $\mathcal{D}$

of historical samples of power, voltage, and current. Now, it is possible to introduce the surrogate models.

In this setting, the surrogate model can be framed as a multi-output regression problem [66]. More formally, given an input space $\mathcal{X} = \mathbb{R}^{n_x}$ and the associated output space $\mathcal{Y} = \mathbb{R}^{n_y}$, we define a model $f_\theta : \mathcal{X} \to \mathcal{Y}$, with specific parameters θ and trained using an algorithm $\mathscr{A}_H$ with specific hyperparameters H. f_θ is built to approximate the underlying relation between input and output data.

Given the multi-output structure of our problem, a neural network is the most natural choice. In this case, using a deep neural network does not seem to be the best choice, since the system topology is fixed and the vector $\boldsymbol{x} \in \mathbb{R}^{n_x}$, containing P_n and Q_n $\forall n \in \mathfrak{N}$, is already a good representation of the input. Therefore, based on these considerations and prior experience, we chose a fully connected hidden-layer neural network. The chosen architecture consists of a first non-linear block that learns a representation, which is then passed to a linear block that outputs the prediction, handling the multi-output structure in a computationally efficient way. A more detailed refinement of the neural network architecture (e.g., by slightly increasing the model's complexity) could potentially improve performance. Nevertheless, this level of detail falls outside the scope of the present study, which focuses on demonstrating the advantages of integrating knowledge into the model while keeping other conditions constant. Then, the proposed architecture can be formalized as:

$$f_{\theta=\{W^o,W^h\}}(\boldsymbol{x}) = W^o \varphi(W^h \boldsymbol{x}) \tag{3.13}$$

where $W^o \in \mathbb{R}^{n_y \times n_h}$, $W^h \in \mathbb{R}^{n_h \times n_x}$ (for a total of $n_y n_h + n_h n_x$ parameters), φ is the activation function, and n_h is the number of hidden units.

The dataset has been split into training, validation, and testing subsets, so that all quantities associated with a single condition were assigned entirely to a single subset. The model was trained using the ADAM optimizer, with a batch size of 128 and a learning rate scheduler.

As for the Physics-Informed Model, the underlying architecture is the same as that of the data-driven model. This is necessary to ensure a fair comparison between the two models. Indeed, in the literature, it is common to find comparisons between physics-informed and data-driven techniques using different architectures [205, 204]. Even when the architectures are the same, hyperparameter tuning is sometimes performed only for one model and then directly transferred to the other [206]. In this study, we explicitly aim to make the comparison as fair as possible.

The Physics-Informed Model differs from the data-driven model by the introduction of a penalization term in the loss function:

$$\mathrm{L}_{phy}(\theta) = \sum_{S\in\mathcal{D}}\Big[\|I(f_\theta(S)) - YV(f_\theta(S))\| + \|S - V(f_\theta(S))\odot I^H(f_\theta(S))\|\Big] \tag{3.14}$$

where S is each input sample from the dataset. In this case, the collocation points are represented by the set of S itself. However, as discussed in Section 3.3.2, it is possible to extend the collocation points by adding noise $\mathcal{N}_S(\sigma)$ to existing ones. We can thus introduce another loss function, which includes new collocation points:

$$\begin{aligned}\mathrm{L}_{phy}(\theta) = &\tfrac{1}{2}\sum_{S\in\mathcal{D}}\Big[\|I(f_\theta(S)) - YV(f_\theta(S))\| + \|S - V(f_\theta(S))\odot I^H(f_\theta(S))\|\Big]\\ &+ \tfrac{1}{2}\sum_{\tilde{S}\in\{S+\mathcal{N}_S(\sigma):S\in\mathcal{D}\}}\Big[\|I(f_\theta(\tilde{S})) - YV(f_\theta(\tilde{S}))\| + \|\tilde{S} - V(f_\theta(\tilde{S}))\odot I^H(f_\theta(\tilde{S}))\|\Big]\end{aligned} \tag{3.15}$$

The number of added samples is adjustable and is effectively treated as a hyperparameter. Using this technique allows for improved performance even with low-cardinality datasets.

In this study, we evaluated the use of both regularization terms, namely Eqns. (3.14) and (3.15).

For the sake of clarity, from now on, we will refer to the models using the following names:

- INT: the iterative numerical solution of the physical model.
- DDT: the data-driven model.
- $PIDDT^0$: the Physics-Informed Model using the regularizer expressed in Eq. (3.14).
- $PIDDT^\sigma$: the Physics-Informed Model using the regularizer expressed in Eq. (3.15).

For each of the proposed surrogate models, hyperparameter tuning was performed separately. The optimal hyperparameters are reported in Table 3.1.

Hyperparameter	**DDT**	**PIDDT^0**	**PIDDT$^\sigma$**
Number of resamplings n_r		1	
Dataset splitting $\|\mathcal{L}\|, \|\mathcal{V}\|, \|\mathcal{T}\|$		$70\%, 15\%, 15\%$	
Number of hidden neurons n_h	1291	3162	3162
Activation function φ	`relu`	`relu`	`relu`
Gradient descent scheduler initial value ρ_0	10^{-4}	10^{-4}	10^{-4}
Gradient descent scheduler factor ρ_f	0.90	0.95	0.95
Gradient descent scheduler threshold ρ_t	10^{-4}	10^{-6}	10^{-6}
Gradient descent scheduler epochs number ρ_e	10	20	20
Number of epochs n_e	2000	5000	5000
λ scheduler initial value λ_0	-	10^{-10}	10^{-10}
λ scheduler number of changes λ_c	-	7	7
λ scheduler factor λ_f	-	100	100
Noise intensity σ	-	-	0.01

Table 3.1 Optimal configuration of the hyperparameters

3.4.3 Results and discussion

In this section, we evaluate the performance of the introduced models.

The output of INT is considered the Ground Truth against which the accuracy of the other models is assessed. Accuracy is expressed in terms of $MAPE$, averaged across the different outputs (V_n^R, V_n^I, I_n^R, and I_n^I $\forall n \in \mathfrak{N}$), and is evaluated on unseen data from the test set.

Table 3.2 presents the results of each model, also showing the improvement achieved by each of these. Table 3.3 reports the accuracy results for each of the four output groups (V_n^R, V_n^I, I_n^R, and I_n^I), showing both $MAPE$ and the Coefficient of Determination R^2. As an additional evaluation of accuracy, Figure 3.3 shows the distribution of the absolute error across the different models.

The second metric used to assess performance is efficiency. Table 3.4 reports the time required by each model to complete a full prediction of all outputs. These results are obtained running the experiments on a machine that hosts 238 Compute nodes with a total of 476 Intel XEON E5-6248R 24C 3.0GHz CPUs and 192 GB of Memory per node and 10 GPU nodes with a total of 40 NVIDIA Tesla V100S 32GB GPUs and 256 GB of memory per node.

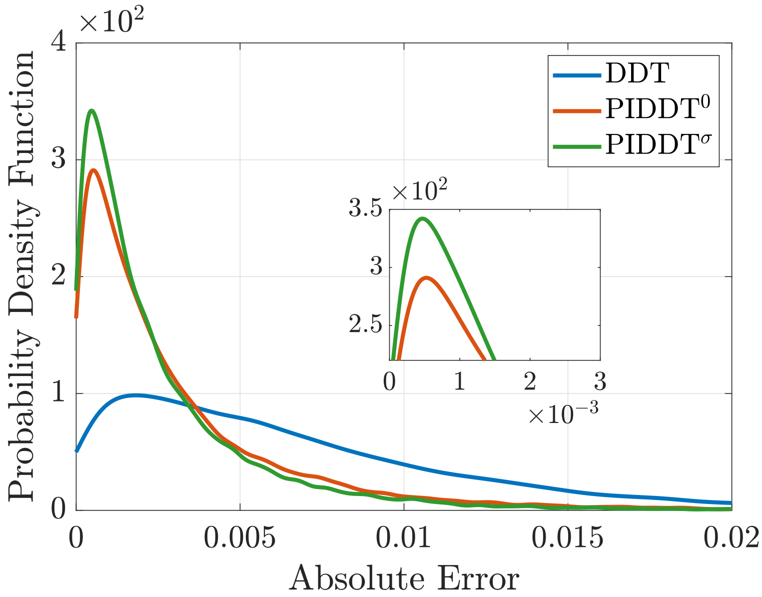

Fig. 3.3 Distribution of the absolute error of the DDT, the $PIDDT^0$, and the $PIDDT^\sigma$ against the INT

MAPE **DDT**	**MAPE** **PIDDT^0**	**% Impr.** **DDT** **vs** **PIDDT^0**	**MAPE** **PIDDT$^\sigma$**	**% Impr.** **PIDDT^0** **vs** **PIDDT$^\sigma$**
0.23	0.12	48	0.11	2

Table 3.2 $MAPE$ and percentage of improvement over the different outputs of the DDT, the $PIDDT^0$, and the $PIDDT^\sigma$ against the INT

From the results obtained, several insights can be drawn. The improvement introduced by the Physics-Informed Models is evident in Table 3.2, where it is also evident that the augmentation of collocation points leads to further benefits. That the Physics-Informed Models achieve lower errors is also clearly shown in Figure 3.3. Table 3.3 indicates that performance is comparable across all four output groups.

With regard to efficiency, the inference time data in Table 3.4 clearly show that the iterative model is not suitable for real-time applications. In contrast, the prediction time required by the data-driven and Physics-Informed Models is much lower and compatible with controls that use real-time predictions. Moreover, the prediction time of DDT and $PDDT$s is substantially comparable,

MAPE			
Output	**DDT**	**PIDDT^0**	**PIDDT$^\sigma$**
V^R	0.2181	0.0910	0.0886
V^I	0.1443	0.0525	0.0499
I^R	0.2390	0.1500	0.1480
I^I	0.3259	0.1839	0.1824
R^2			
Output	**DDT**	**PIDDT^0**	**PIDDT$^\sigma$**
V^R	0.9821	0.9823	0.9823
V^I	0.9823	0.9824	0.9824
I^R	0.9996	0.9998	0.9998
I^I	0.9990	0.9996	0.9996

Table 3.3 $MAPE$ and R^2 of the DDT, the $PIDDT^0$, and the $PIDDT^\sigma$ against the INT for each group of the outputs

INT	DDT	PIDDT^0	PIDDT$^\sigma$
15 s	0.09 ± 0.01 ms	0.09 ± 0.01 ms	0.11 ± 0.01 ms

Table 3.4 Prediction time of all the outputs for the INT, the DDT, the $PIDDT^0$, and the $PIDDT^\sigma$.

with a slight increase in the case of $PDDT^\sigma$. This demonstrates that all surrogate models are suitable for real-time applications.

Under these conditions, it is evident that Physics-Informed Models are advantageous: they offer significantly greater efficiency than physical models, and compared to data-driven models, they maintain similar efficiency while achieving higher accuracy.

This experiment clearly confirms what was introduced in Chapter 2: Physics-Informed Models require an additional construction step, but lead to improved model accuracy, with inference efficiency comparable to that of data-driven approaches.

This case study represents the first step of a broader project aimed at expanding the network size and scaling the model to larger grids.

Chapter 4

An application of Knowledge-Informed Generative Models

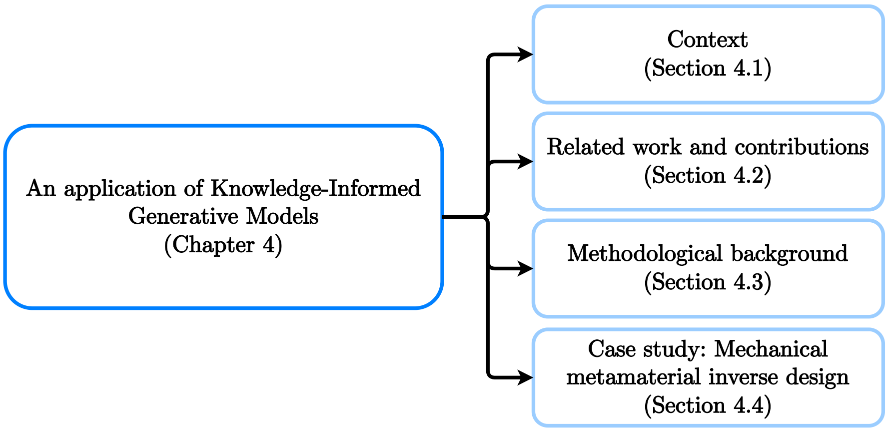

Fig. 4.1 Graphical index of Chapter 4

Up to this point, we have focused primarily on predictive models, but the integration of domain knowledge can also be extended to generative models, a field that is currently experiencing a significant surge in popularity. In this chapter, we discuss early contributions from the literature in this direction and present our contribution in a practical application.

4.1 Context

Lately, generative models have become a key player in ML, making it possible to create new samples similar to the training data. More formally, a generative model is defined as a model capable of producing samples that belong to the same probability distribution as the training set. Some of the earliest popular techniques were Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) [210]. A GAN works by training two neural networks: a generator that creates synthetic data, and a discriminator that learns to distinguish between real and generated samples [211]. This adversarial process drives the generator to produce increasingly realistic outputs. On the other hand, a VAE learns to encode input data into a latent representation and then decodes it back, using probabilistic methods to ensure that the reconstructed data belongs to the training set's probability distribution [212]. Both GANs and VAEs showed significant results for several tasks, such as generating images, transferring styles between them, reconstructing areas of the samples, and enhancing their resolution.

More recently, Diffusion Models (DMs) have gained significant attention. This new technique is robust and capable of producing really high-quality results. Unlike GANs and VAEs, DMs start with random noise and gradually refine it over multiple steps to create meaningful samples. In the specific task of generating images, DMs have actually outperformed GANs and VAEs, delivering better quality, more diversity, and more realistic outputs [210].

A major step forward came with the development of Stable DMs, a variation of DMs that works more efficiently by operating in a compressed space. This not only speeds the process up but also makes it possible to generate large, coherent, high-resolution samples.

Practically speaking, generative models hold significant potential for applications in industry. They have been already employed for improving operational efficiency, minimizing downtime, and automated design [213, 214]. One of the reasons generative models are particularly appealing is their potential use in inverse design, that is, the design of a sample with predefined, desired properties. This type of problem is traditionally tackled using a trial-and-error approach, which is highly inefficient Bastek and Kochmann [215]; the use of generative models can substantially enhance the effectiveness of the process.

However, generative models still have some limitations. Although they are already highly useful, experience shows that their reliability still has considerable room for improvement. Exploring new techniques to make generative models more trustworthy and reliable is of great interest, especially in industry. As with predictive models, one possible way to make their outputs more robust and consistent is to ensure their alignment with external knowledge. Extending the concepts introduced in the previous chapters, we refer to these techniques as *Knowledge-Informed Generative Models*. We believe that this approach addresses an important aspect of the general discussion on Knowledge-Informed Models and, for this reason, it has been the subject of dedicated research efforts throughout this Ph.D..

It is essential to start by noting that Knowledge-Informed Generative Models have not yet reached a sufficient level of maturity and diffusion to support a detailed review. Nevertheless, the first prototypes are beginning to emerge in the research community and we argue it is worthwhile to dedicate a part of this chapter to reviewing the progress made so far.

After a brief introduction, the main goal of this chapter is to illustrate a practical use case that is directly applicable to industry. The selected demonstration is the inverse design of mechanical metamaterials, that is, engineered structures that derive their mechanical properties from the geometry of their micro-architecture. In particular, the choice of an inverse design problem allows us to highlight the strengths of generative models in solving such tasks.

After carefully reviewing the existing literature, we opted to use DMs as our base generative model. In fact, the DM pipeline not only produces highly realistic results, but also provides a promising framework for embedding domain-specific knowledge. As we will demonstrate, its inherent flexibility enables DMs to be *informed* through the guided manipulation of intermediate samples during the generation process.

The structure of this chapter includes the following sections:

- Section 4.2 discusses the existing research on Knowledge-Informed Generative Models and points out our contributions.

- Section 4.3 introduces the methods and concepts applied during the development of the proposed application.
- Section 4.4 presents the experimental setup and discusses the results.

This project was carried out during a visiting period at the RAISE Lab of the University of Virginia, and the results have been accepted at NeurIPS 2025 [25].

4.2 Related work and contributions

In the literature, some ongoing research efforts attempt to constrain the samples produced by generative models through domain-specific knowledge. Nonetheless, to the best of the author's knowledge, no one of these has already spread in the industrial sector. Several types of constraints have been considered by the research community, such as physical laws, domain experience, and also external data-driven proxy functions [216].

The simplest way to enforce a constraint is to correct the generated sample through a post-processing operation [216, 217]. With these methods, the corrected sample satisfies the constraint, but, at the same time, it can be significantly different from the originally generated output, and potentially out of the training set's probability distribution. This motivates the search for more refined methods of integration.

Recently, model conditioning techniques are gaining popularity, with Classifier-Free and Classifier-Based Methods. Classifier-Free Methods [218] integrate a conditioning variable $\boldsymbol{c}$ into the denoising process, while Classifier-Based Methods [210] steer the sampling toward a specific class y_i by leveraging on the estimation $p(y|\boldsymbol{x}_t)$ of an external classifier. Other affine methods have also been explored, such as Training-Free Guidance [219] and Conditional Stable DMs [220]. In general, model conditioning does not ensure the constraint compliance, even if it shows interesting results in driving the generation toward the desired direction [221, 210].

Other methods have been explored to enforce hard restrictions, but they are limited to specific frameworks, such as linear [222] and convex [223] constraint sets. Another approach is represented by the Projected Diffusion Method, proposed by Christopher et al. [216], which modifies the diffusion process by projecting the intermediate samples onto *the closest sample that satisfies a given constraint*. This promising technique has shown accurate results and ensures the constraint satisfaction [216]. It has proven to work well in various fields, such as microstructure image generation (by constraining porosity), body motion (by enforcing physical consistency through physical differential equations), and path planning (by constraining path feasibility). However, the Projected Diffusion Method is still limited to the research domain. At the current state of the art, the exploration of Projected Diffusion has been limited to the image space and for a limited set of cases, even if it appears promising for the industrial context, since it can potentially handle a wide range of constraints and ensure high-quality and feasible samples.

As a result of these considerations, we chose this framework and we aim to advance the limits of the results obtained by Christopher et al. [216]. The first contribution of our work is to extend Projected Diffusion to Stable DMs. The second contribution comes from incorporating constraints, in the form of non-differentiable simulators, which are a common tool in industry. The third contribution is the application to a new industrial scenario.

4.3 Methodological background

In this section, we introduce concepts and techniques used in the case study, with the aim of facilitating the reader's understanding of the subject. Sections 4.3.1 and 4.3.2 provide a brief introduction to DMs and Stable DMs. Sections 4.3.3 - 4.3.6 describe the constraining methods. Finally, Section 4.3.7 gives a introductory description of the Finite-Element Method.

4.3.1 Diffusion Denoising Probabilistic Models

DMs are a class of generative models that learn to generate data by simulating a stochastic process that gradually transforms structured data into noise and then reverses this process to recover the original distribution. The key idea behind DMs is to train a neural network to remove noise step by step, enabling the generation of realistic samples from simple noise distributions. The generative diffusion approach counts several different methods; this work focuses on the class of Diffusion Denoising Probabilistic Models (DDPMs) [224].

A generic DM is based on a Markov Chain $\{\boldsymbol{x}_t\}_{t=0}^{T}$ of progressively noisy samples. In particular, $\boldsymbol{x}_0$ represents the original and $\boldsymbol{x}_T$ the entirely corrupted data sample. The evolution relies on a Gaussian diffusion process, described as:

$$p(\boldsymbol{x}_0) = \int p(\boldsymbol{x}_T) \prod_{t=1}^{T} p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\, d\boldsymbol{x}_{1:T} \tag{4.1}$$

The procedure can be split into *forward* and *reverse* phases. During the forward phase, the data undergo a gradual corruption process by adding Gaussian noise at each step t. The transition kernel governing this process is given by

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}) \tag{4.2}$$

where each $\beta_t \in (0,1)$ constitutes the noise schedule $\{\beta_t\}_{t=1}^{T}$. This schedule is designed so that the terminal distribution $p(\boldsymbol{x}_T)$ approximates a Gaussian distribution. The model training process builds a neural network $\epsilon_\theta(\boldsymbol{x}_t, t)$ (often referred to as *denoiser*) that can estimate the noise added during the forward phase at each perturbation step. The network is optimized by minimizing the mean squared error between the actual noise and the predicted noise:

$$\min_\theta \mathbb{E}_{t\sim[1,T],\, \boldsymbol{x}_0,\, \epsilon\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})} \left[\|\epsilon - \epsilon_\theta(\boldsymbol{x}_t, t)\|_2^2 \right] \tag{4.3}$$

Once trained, the network $\epsilon_\theta(\boldsymbol{x}_t, t)$ can be used to estimate the added noise and iteratively reconstruct data samples from the noisy distribution $p(\boldsymbol{x}_T)$. This process is often called *reverse phase*, *denoising*, or *sampling*. At each step, the network can predict the transition $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ and generate a cleaner sample. In particular, DDPMs learn to remove noise by leveraging the score function of the probability density, namely the gradient of the logarithm of the probability density. Operationally, the denoising process of DDPMs employ Stochastic Gradient Langevin Dynamics (SGLD), featuring the following update step:

$$\boldsymbol{x}_t^{i+1} = \boldsymbol{x}_t^i + \gamma_t \nabla_{\boldsymbol{x}_t^i} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) + \sqrt{2\gamma_t}\boldsymbol{\eta} \tag{4.4}$$

where γ_t is the step size, η is standard normal, and the term $\sqrt{2\gamma_t}\boldsymbol{\eta}$ is added to avoid a deterministic behavior [224]. The update step moves the sample towards the training set probability distribution, until a clean sample is obtained.

4.3.2 Stable Diffusion

Stable DMs build upon DDPMs by performing the diffusion process in a lower-dimensional latent space rather than directly in the original data space [220]. This approach relies on a pre-trained encoder-decoder architecture, where the encoder E maps high-dimensional image data to a latent representation $\mathbf{z}_t$, while the decoder D reconstructs the image after the diffusion process has been applied in the latent space.

$$\min_{\theta} \mathbb{E}_{t\sim[1,T],\, \mathbf{z}_t\sim \mathrm{E}(\boldsymbol{x}),\epsilon\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t)\|_2^2\right] \tag{4.5}$$

The loss function remains consistent with that of standard DDPMs, with the key difference that the Stable DM is trained to remove noise in the latent space instead of the original data space. Notably, the denoiser is trained independently of the decoder, as its loss is computed entirely within the latent space. Then, the update step can be rewritten as:

$$\mathbf{z}_t^{i+1} = \mathbf{z}_t^i + \gamma_t \nabla_{\mathbf{z}_t^i} \log q(\mathbf{z}_t|\mathbf{z}_0) + \sqrt{2\gamma_t}\boldsymbol{\eta} \tag{4.6}$$

The sampling process outputs the clean latent sample $\mathbf{z}_0$, while the final generated sample is obtained by passing it through the decoder:

$$\boldsymbol{x}_0 = \mathrm{D}(\mathbf{z}_0) \tag{4.7}$$

4.3.3 Projection Operator

The Projected Diffusion method approaches the sampling process as a constrained optimization problem [216]:

$$\underset{\boldsymbol{x}_T,\dots,\boldsymbol{x}_1}{\text{minimize}} \sum_{t=T,\dots,1} -\log q(\boldsymbol{x}_t|\boldsymbol{x}_0) \tag{4.8a}$$

$$\text{s.t.:} \quad \boldsymbol{x}_T,\dots,\boldsymbol{x}_0 \in \mathbf{C} \tag{4.8b}$$

where $\mathbf{C}$ is the constraint set. Note that the condition $\boldsymbol{x}_T, ..., \boldsymbol{x}_0 \in \mathbf{C}$ can be rewritten as $\mathbf{g}(\boldsymbol{x}_t) = 0$, where $\mathbf{g}$ is a function that estimates the distance between $\boldsymbol{x}$ and the constraint set $\mathbf{C}$.

If a sample does not satisfy the condition, it is possible to define a Projection Operator that corrects the sample. The Projection Operator introduced by Christopher et al. [216] is defined as:

$$\mathcal{P}_{\mathbf{C}}(\boldsymbol{x}) = \underset{\boldsymbol{y} \in \mathbf{C}}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \tag{4.9}$$

Given a sample $\boldsymbol{x}$, $\mathcal{P}_{\mathbf{C}}(\boldsymbol{x})$ provides the nearest sample that complies with the constraint set $\mathbf{C}$. Hence, for a DM sampling process, it is possible to restore feasibility at each step through:

$$\boldsymbol{x}_t^{i+1} = \mathcal{P}_{\mathbf{C}}\left(\boldsymbol{x}_t^i + \gamma_t \nabla_{\boldsymbol{x}_t^i} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) + \sqrt{2\gamma_t}\boldsymbol{\eta}\right) \tag{4.10}$$

This is defined *Projected Sampling*, and ensures that each $\boldsymbol{x}_t^i$ satisfies the constraint.

This technique has proven to be effective for constraining samples generated by image space DMs [216]. However, because the constraints typically exist in the decoded sample space, the Projection Operator is not directly applicable to Stable DMs. In fact, enforcing constraints directly within the latent space presents a challenge because the latent representation lacks an explicit correspondence to the features of the decoded sample. The key to overcoming this difficulty lies in recognizing that, although constraints may not be explicitly represented in the latent space, their fulfillment can still be assessed throughout the diffusion process, by decoding the sample via the differentiable decoder D, and evaluating its feasibility in the decoded image space. In Section 4.3.5 we will show that the gradient of the constraint function, evaluated in the decoded space, can be used to iteratively refine the latent representation at any stage of the diffusion process, ensuring compliance with the desired constraints.

4.3.4 Proximity Operator

The concept of Projection Operator can further be extended to include a broader range of constraints. This is particularly useful when the constraint is a target value of a certain property, from which we want our corrected sample to differ by no more than an arbitrary threshold. In this case, we can define a constraint function $\mathbf{g}$ that measures how much the sample deviates from the target value, and a Proximity Operator:

$$\mathbf{prox}_{\lambda\mathbf{g}(\boldsymbol{x}_t)} = \arg\min_{\boldsymbol{y}} \left\{ \mathbf{g}(\boldsymbol{y}) + \tfrac{1}{2\lambda} \|\boldsymbol{y} - \boldsymbol{x}_t\|_2^2 \right\} \tag{4.11}$$

$\mathbf{prox}_{\lambda\mathbf{g}(\boldsymbol{x}_t)}$ estimates the sample that minimizes the weighted sum of $\mathbf{g}$ and the distance with respect to the original sample. The weighting term λ can be treated as a hyperparameter of the problem.

This formulation can be viewed as a generalization of the Projection Operator, where the function $\mathbf{g}(\boldsymbol{y})$ is defined such that it yields 0 when $\boldsymbol{y}$ satisfies the constraints, and infinity otherwise.

Similarly to Projected Diffusion, the update step can be rewritten including the Proximity Operator as:

$$\boldsymbol{x}_t^{i+1} = \mathbf{prox}_{\lambda\mathbf{g}(\boldsymbol{x}_t)} \left(\boldsymbol{x}_t^i + \gamma_t \nabla_{\boldsymbol{x}_t^i} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) + \sqrt{2\gamma_t}\boldsymbol{\eta} \right) \tag{4.12}$$

in which the sample is corrected at each step.

The Proximity Operator enables the use of a wider set of constraints, if compared to the Projection Operator proposed in [216]. However, since the constraint function $\mathbf{g}(\mathbf{z_t})$ is not meaningful in the latent space, the Proximity Operator retains the same limitation as the Projection Operator, meaning it cannot be applied directly to Steble DM.

4.3.5 Latent Correction Algorithm

The primary goal of this research is extending previous results to Stable DMs. To do that, the Proximity Operator should be integrated in latent sampling process:

$$\mathbf{z}_t^{i+1} = \mathbf{prox}_{\lambda\mathbf{g}(\mathbf{z_t})} \left(\mathbf{z}_t^i + \gamma_t \nabla_{\mathbf{z}_t^i} \log q(\mathbf{z}_t|\mathbf{z}_0) + \sqrt{2\gamma_t}\boldsymbol{\eta} \right) \tag{4.13}$$

As anticipated, the constraint evaluation is meaningful only in the decoded space; therefore the operator $\mathbf{prox}_{\lambda\mathbf{g}(\mathbf{z_t})}$ cannot be applied directly in the latent space. The simplest solution appears to be decoding the latent representation, applying the projection in the decoded space, and re-encoding the corrected sample. Unfortunately, the encoding process is lossy, meaning that the corrected latent sample may not contain some of the original information.

The solution proposed in our work is to decode the sample $\mathrm{D}(\mathbf{z}_t^i)$ and compute its corrected version $\mathbf{prox}_{\lambda\mathbf{g}(\mathbf{z_t})}(\mathrm{D}(\mathbf{z}_t^i))$. Then, we can compute the gradient of the norm of the difference between $\mathbf{prox}_{\lambda\mathbf{g}(\mathbf{z_t})}(\mathrm{D}(\mathbf{z}_t^i))$ and $\mathrm{D}(\mathbf{z}_t^i)$, with respect to the latent sample:

$$\nabla_{\mathbf{z}_t^i}(||\mathbf{prox}_{\lambda\mathbf{g}(\mathbf{z_t})}(\mathrm{D}(\mathbf{z}_t^i)) - \mathrm{D}(\mathbf{z}_t^i)||_2), \tag{4.14}$$

At this point, we have everything we need to correct the sample directly in the latent space though a gradient descent approach. The latent sample is iteratively corrected until feasibility is ensured, meaning that the constraint error is below a certain threshold. The proposed update step is the following:

$$\mathbf{z}_t^{i+1} = \mathbf{z}_t^i - \gamma_p \nabla_{\mathbf{z}_t^i}(||\mathbf{prox}_{\lambda\mathbf{g}(\mathbf{z_t})}(\mathrm{D}(\mathbf{z}_t^i)) - \mathrm{D}(\mathbf{z}_t^i)||_2) \tag{4.15}$$

where γ_p is the step size.

We highlight that also the correction process $\mathbf{prox}_{\lambda\mathbf{g}(\mathbf{z_t})}(\mathrm{D}(\mathbf{z}_t^i))$ frequently requires an iterative approach. Then, it is worth introducing a distinction between the *Outer Minimizer*, which refers to the iterative correction of the latent sample, and the *Inner Minimizer*, referring to the process of computing $\mathbf{prox}_{\lambda\mathbf{g}(\mathbf{z_t})}(\mathrm{D}(\mathbf{z}_t^i))$.

4.3.6 Constraining functions

Here, we analyze the practical construction of the Inner Minimizer. First of all, it is important to mention that the Proximal Operator is application-specific and strictly depends on the formulation of the constraining function $\mathbf{g}(\boldsymbol{x}_t)$. In some cases, the constraint function $\mathbf{g}(\boldsymbol{x}_t)$ is differentiable, convex, and the Proximal Operator can be computed in closed form, allowing the corrected sample to be obtained directly by applying the operator, without the iterative process. Unfortunately, this situation is uncommon in practical applications. In fact, in most cases, the constraint is a desired property that has an implicit mathematical form, such as a differential equation. Then, the function

$\mathbf{g}(\boldsymbol{x}_t)$ cannot be directly implemented in the minimization process to compute $\mathbf{prox}_{\lambda\mathbf{g}(\mathbf{z_t})}(\mathrm{D}(\mathbf{z}_t^i))$. In industry, this is a common scenario, where complex phenomena are often described by partial differential equations, typically solved through computer-aided simulations.

Since this Ph.D. program is focused on industrial applications, this work shows how to integrate a Finite-Element Method into the process, as a non-differentiable model to estimate the compliance to the desired constraint. As we will describe, a Differential Particle Optimizer (DPO) [225, 226] has been used, capable of transforming the outputs of a black-box function (i.e., the Finite-Element Method) into a smooth approximate function. The proposed algorithm is capable of computing an estimation of $\mathbf{prox}_{\lambda\mathbf{g}(\mathbf{z_t})}(\mathrm{D}(\mathbf{z}_t^i))$, which can then be used by the Outer Minimizer.

4.3.7 Finite-Element Method for structural applications

To complete the picture, we provide a brief introduction to the Finite-Element Method (FEM). The FEM is a powerful numerical technique widely used across industry to model and analyze complex systems described by partial differential equations. The base principle behind the FEM is discretizing continuous domains into smaller sub-regions and approximating unknown fields through simple functions at nodal points. FEMs are widely used in many engineering fields, such as structural analysis, heat transfer analysis, and crash simulations. In our project, we leverage FEMs for the structural analysis of complex shapes.

Structural analysis is a key discipline in mechanical engineering that focuses on understanding how structures respond to external forces. The response of the structure includes the stresses and strains that develop within the body. Structural analysis is governed by the fundamental equations of continuum mechanics, specifically equilibrium equations, constitutive laws, and compatibility conditions. Different theories and assumptions provide formulations that link stresses $\boldsymbol{\sigma}$, strains $\boldsymbol{\varepsilon}$, body forces $\boldsymbol{b}$, displacements $\mathbf{u}$, and surface traction $\bar{\mathbf{t}}$. From the Principle of Virtual Work for a static elastic problem, it is possible to derive the following fundamental equation in weak form [227]:

$$\int_{\Omega} \boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\, d\Omega = \int_{\Omega} \boldsymbol{b} \cdot \delta\mathbf{u}\, d\Omega + \int_{\Gamma_t} \bar{\mathbf{t}} \cdot \delta\mathbf{u}\, d\Gamma \tag{4.16}$$

where $\delta\boldsymbol{\varepsilon}$ is the virtual strain, $\delta\mathbf{u}$ is the virtual displacement, Ω is the body domain, and Γ_t is the body boundary.

This expression, by its nature, is very difficult to handle. The FEM converts the expression into a system of many algebraic equations that can be solved by a computer [227].

In more detail, in this method, the component volume is *meshed*, meaning it is divided into small sub-regions (i.e., *discretized*), and the quantities of interest are approximated using simple functions (i.e., *shape functions*) at the nodal points. The combination of all the nodal functions, leads to a system of equations, which can be expressed in matrix form. For example, for linear systems can be expressed as:

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F} \tag{4.17}$$

where $\boldsymbol{K}$ is the global stiffness matrix, $\boldsymbol{u}$ is the nodal displacement vector, and $\boldsymbol{F}$ is the force vector.

For non-linear problems (e.g., considering plasticity), it is common to adopt the following formulation [228, 229]:

$$\boldsymbol{K}_T \Delta\boldsymbol{u} = \boldsymbol{R} \tag{4.18}$$

where $\boldsymbol{K}_T$ is the tangent stiffness matrix, $\Delta\boldsymbol{u}$ is the displacement increment, and $\boldsymbol{R}$ is the residual force vector. The nodal solution can be found using an incremental-iterative approach, typically the Newton-Raphson method [229].

The core of FEM analysis lies in the mesh formulation [227]. In fact, overly coarse meshes lead to approximate and inaccurate results. Conversely, excessively fine meshes result in huge matrices and often require prohibitive computation times. The regularity of the elements' shapes also has a significant impact on the quality of the results [227]. Finding the right trade-off and building a good mesh is the responsibility of practitioners, who typically increase node density, which in turn increases details, in areas of greater interest or with more complex geometries, while keeping a coarser mesh wherever possible.

The solution of a structural analysis with FEMs is rarely carried out by manually assembling the matrix equations. Instead, it is most commonly done through the graphical interfaces offered by commercial software (e.g., Abaqus), which handle the construction and solution of the matrix problem under the hood.

This concludes the introduction of the concepts required to understand the presented study case. In the following section, we will demonstrate how these methods can be effectively combined to solve the inverse design problem of mechanical metamaterials.

4.4 Case study: Mechanical metamaterial inverse design

Mechanical metamaterials are engineered structures designed to achieve specific non-linear stress-strain curves by tailoring their micro-architecture (see Figure 4.2). Unlike conventional materials, metamaterials derive their mechanical properties from geometry rather than composition, allowing for tunable stiffness. Typically, they feature a base cellular structure that is periodically repeated throughout the entire component. When subjected to an external force, the structure deforms according to its shape, producing a characteristic mechanical response.

The interest in mechanical metamaterials is growing in sectors in which customized mechanical responses are crucial for performance and safety [215]. For instance, enhancing the interaction between a robotic hand and an object may require a mechanical response, such as adaptive compliance, which can be achieved through metamaterials. Another example is the specific stress-strain profile of a body, designed for vibration isolation. Many other use cases are multiplying, in response to the growing demand for products with customized properties [215].

However, designing the micro-architecture that ensures the desired mechanical properties is a complex task. Engineering a metamaterial requires precise control over phenomena such as buckling, contact interactions, and large-strain deformations, features that are inherently non-linear and highly sensitive to minor variations in design parameters. Conventional design methodologies often rely on iterative trial-and-error strategies, which are computationally expensive and do not necessarily yield optimal solutions [215]. Motivated by this,

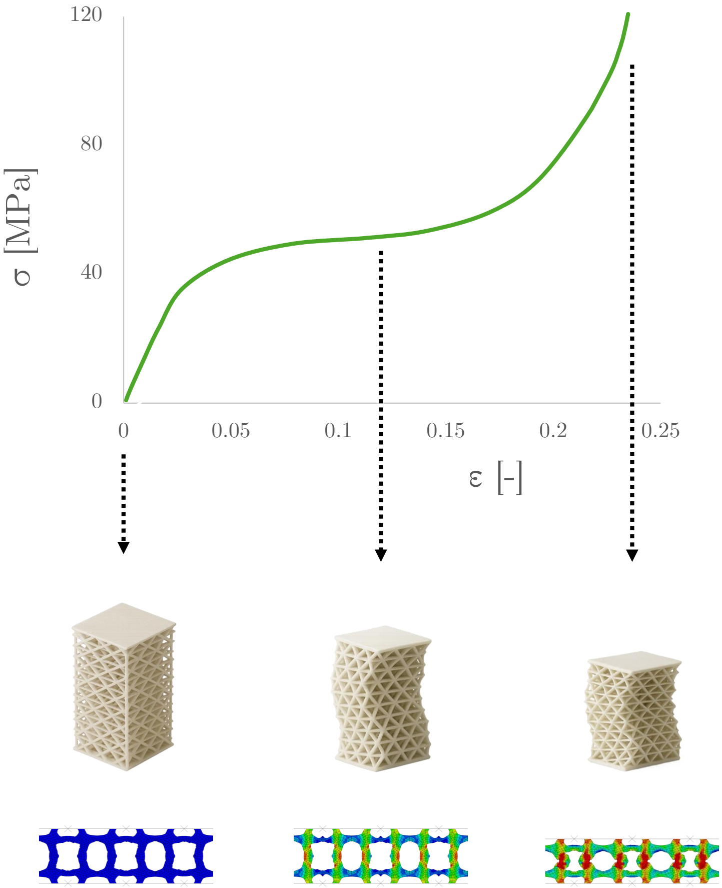

Fig. 4.2 Example of a stress-strain curve of a mechanical metamaterial and corresponding deformation

we propose to exploit the potential of generative models for inverse problem solving, aiming to generate designs with targeted mechanical properties.

To prove the potential of the proposed model, the results are compared against two baselines:

- An unconstrained prompt-conditional Stable DM.
- The video-generation DM architecture proposed by [215], which represents the state of the art in metamaterial inverse design.

4.4.1 Experimental setup

In this research, a constrained Stable DM has been implemented to generate a metamaterial cellular structure that closely replicates a desired stress-strain response.

The training dataset was sourced from Bastek and Kochmann [215], containing around 50,000 periodic stochastic cellular structures subjected to large-strain compression. The dataset includes full-field stress and strain results obtained from simulations of a compression test under strain-controlled loading, ranging from 0 to 20%. From the original dataset, only topological data have been used in the context of this project. In particular, we trained the DM using the binary images 128×128, representing the shape of the microstructure.

The evaluation of the stress-strain curve of new microstructures has been done using a FEM, implemented in Abaqus [230]. Since the problem is invariant with respect to scale, geometric parameters are treated as dimensionless, with stress values expressed in megapascals (MPa).

Abaqus simulations have been managed through a Python program. A fixed loading cycle was defined, identical to the one used by the creators of the dataset [215], in which the shape, subjected to plane strain conditions, undergoes strain-controlled deformation from 0 to 20%. We fixed the parameters for Abaqus' automatic meshing, boundary conditions, as well as the elastic, plastic, and contact material properties. With this setting, the Python program takes as input a 128×128 binary image representing the structure, converts it into a geometry interpretable by Abaqus, and launches the simulation with the predefined settings. Then, from the simulation output, the stress-strain curve is extracted and used both for the constraining phase and the evaluation phase.

4.4.2 Method implementation

Our goal is to constrain the generation process, to ensure the similarity between a target stress-strain curve and the response curve of the generated cellular structure. To obtain that, we follow the Latent Correction Algorithm, introduced in Section 4.3.5.

First, we need to identify an operator that computes the corrected sample, which we can implement in the Outer Minimizer. In this case, we do not know a function that projects the intermediate sample $\boldsymbol{x}_t = \mathrm{D}(\mathbf{z}_t^i)$ into the correct one, so we must rely on an Inner Minimizer, which, through a series of iterations, will give us the correct sample.

Here, the Inner Minimizer, implements a software simulator tool, defined as a generic operator A. The simulator takes a specific cellular structure $\boldsymbol{x}$ as input and produces as output a vector of points, representing the stress-strain curve $\boldsymbol{y}$.

$$\mathrm{A}(\boldsymbol{x}) = \boldsymbol{y} \tag{4.19}$$

A is a black-box model and is not differentiable, therefore the inner minimization process cannot follow a gradient descent approach.

However, we can define a function $\mathbf{g}$ that measures the discrepancy between the stress-strain curve $\boldsymbol{y}$ generated by the simulator for a given cellular structure and the target curve $\boldsymbol{y}_T$ (i.e., the desired mechanical property).

$$\mathbf{g}(\boldsymbol{x}) = ||\mathrm{A}(\boldsymbol{x}) - \boldsymbol{y}_T||^2 \tag{4.20}$$

The function $\mathbf{g}$ depends on the simulator, and it is still non-differentiable. To overcome this challenge, we can resort to DPO [225, 226].

DPO works by introducing controlled perturbations into the optimization variables, to create a smooth approximation of the objective function, allowing for the computation of an approximated gradient. More specifically, the process involves applying small random modifications to the input parameters, represented by the cellular shapes. For each perturbed cellular shape, the discrepancy $\mathbf{g}$ is evaluated through the simulator, and the shape is updated in the direction that maximizes the reduction of $\mathbf{g}$. This operation can also be interpreted as a Monte Carlo estimation of the gradient $\nabla_{\boldsymbol{x}_t}\mathbf{g}(\boldsymbol{x}_t)$, whose direction can be used to minimize $\mathbf{g}$.

Using an approximation of $\nabla_{\boldsymbol{x}_t}\mathbf{g}(\boldsymbol{x}_t)$, it is possible to follow an iterative process that stops only when the discrepancy falls below a certain threshold. At that point, the sample is considered corrected, and the outer minimization process can start. While some correctors can operate effectively even on highly noisy samples (i.e., during the early steps of the denoising process), in this case, the use of a FEM model requires the sample to be sufficiently clean. For this reason, the correction can only begin after approximately 70% of the denoising has been completed.

By introducing this method, our approach remains applicable also in presence of black-box models, where conventional gradient-based optimization techniques fail.

We highlight that the proposed algorithm requires the setting of several hyperparameters. These are the selected values:

- Denoising steps: 25.
- Denoising step at which DPO starts: 24.
- Initial perturbation size: 7 px.
- Perturbation decay schedule: –1 px/step.
- Number of random perturbations per step: 4.

Note that a fine optimization of the hyperparameters falls outside the scope of this presentation in the current stage of research.

4.4.3 Results and discussion

An example of the projection process is presented in Figure 4.3. The potential of the proposed model to produce stress-strain curves that closely match the target is clearly illustrated at the bottom of Figure 4.3. We also highlight that, by increasing the number of steps and tuning the DPO parameters, it is possible to achieve arbitrarily small errors. Naturally, this comes at the cost of increased computational effort.

A comparison with the baseline methods is provided in Figure 4.4, both in interpolation (i.e., when the target curve falls within the stress range covered by the training set) and in extrapolation (i.e., when the target is outside this range). The picture displays the stress-strain curves along with the corresponding MSE against the target curve, for our model (after 5 DPO iterations) and for the two baseline methods. This shows how integrating the simulator within the optimization loop enables our model to extrapolate beyond the training dataset. This characteristic is of fundamental importance in the context of inverse design, as the desired properties often deviate significantly

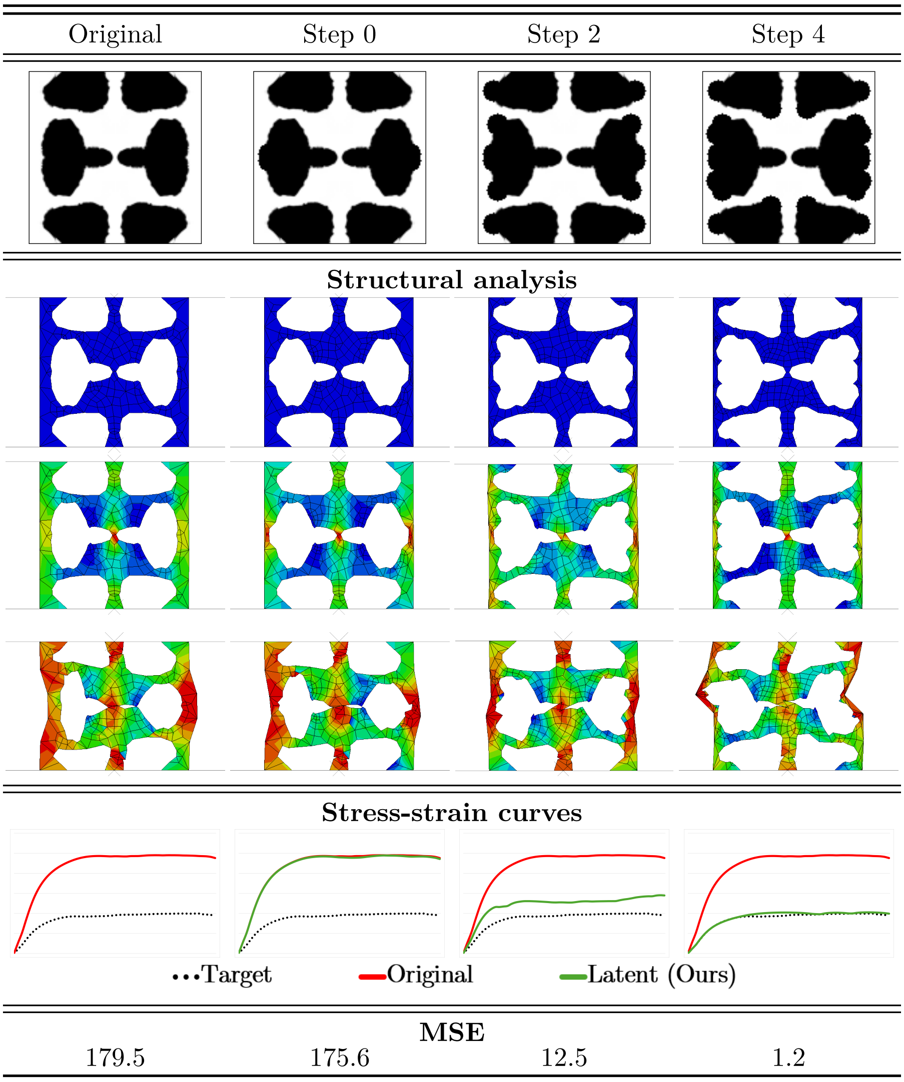

Fig. 4.3 Successive steps of the DPO algorithm. For each step, the image reports the corresponding stress-strain curve and the MSE with respect to the target curve

from those represented in the dataset. It is also important to consider the dependency of the Latent Correction Algorithm on the quality of the initial sample, prior to correction. For this reason, an analysis was conducted on 20 different samples, to have a consistent measure of MAE with respect to the target curve. Also in this case, the results of the proposed model are evaluated after 5 DPO iterations. The results are compared to the baselines and reported in Table 4.1. It can be observed that, even after only 5 iterations, our model outperforms the baselines. Latent Correction Algorithm reports

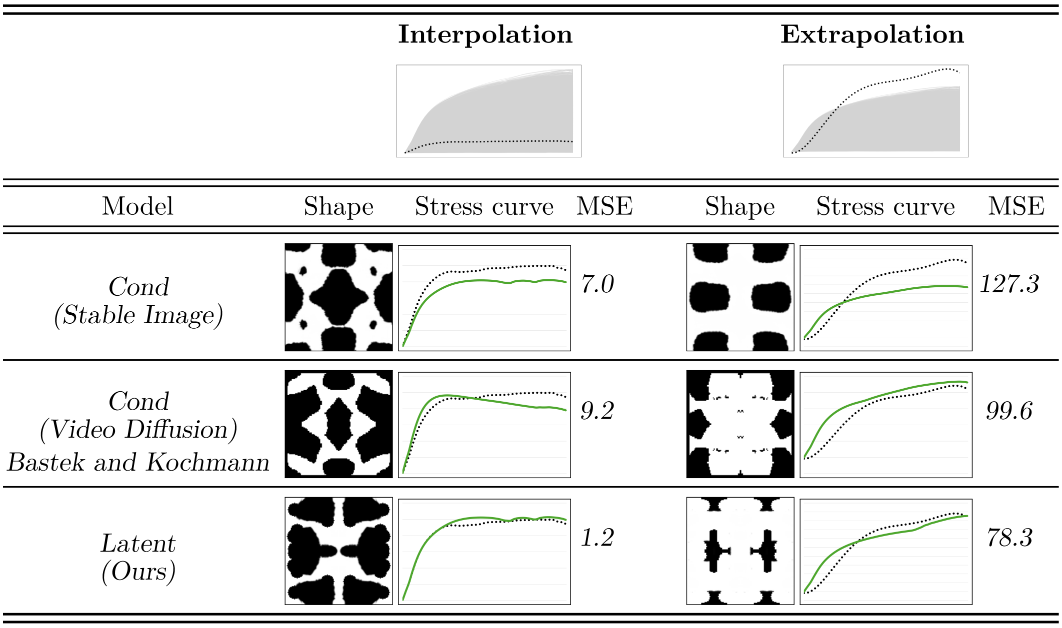

Fig. 4.4 Comparison between the proposed model (after 5 DPO iterations) and the baselines in interpolation and in extrapolation. The picture shows results in terms of stress-strain curve and MSE with respect to the target curve

Model	MSE	Invalid Shapes
Conditional	7.1 ± 4.5	55%
Bastek and Kochmann [215]	6.4 ± 4.6	20%
Latent (Ours)	$\mathbf{1.4 \pm 0.6}$	**5%**

Table 4.1 Comparison of MSE with respect to the target stress-strain response and fraction of physically invalid shapes

a 4.6× improvement over the state of the art model (i.e., [215]) and a 5.1× improvement over the conditional Stable DM in terms of MSE. The same table also reports the number of physically admissible samples (i.e., those exhibiting internal continuity and proper boundary adhesion). Once again, our model achieves the best performance. Notably, more than half of the samples generated by the prompt-conditional model are not physically valid.

These results empirically validate the superiority the proposed method when strict compliance with a constraint is required.

Finally, we provide a brief note on the computational cost of our method. The sampling time for the diffusion model is approximately 5 seconds (measured on an NVIDIA A100-SXM4-80GB), while each Abaqus simulation takes on average around 30 seconds (measured on an Intel Core i7-8550U CPU). This implies that the total time required to produce an output heavily depends on the number of iterations (and thus simulations) performed. We conclude that, while the proposed model is capable of achieving arbitrarily low error, it is crucial to find a balance between accuracy and computational efficiency.

Chapter 5

Discussion

In the previous sections, we examined the core components of our research. In this section, we discuss more in general what has been achieved throughout the Ph.D. journey. For the sake of clarity, we structure the discussion into: main contributions and observations from our research, current limitations, future perspectives, and finally, a series of take-home messages summarizing this work.

Main contributions and observations: This Ph.D. work contributed to the field of Knowledge-Informed Models in two significant ways: the first comprehensive definition of the state of the art and the application to novel practical cases.

In the absence of a comprehensive survey on the topic, we began our work with a review of the state of the art. Within the Knowledge-Informed ML framework, only predictive modeling has seen sustained research, making it the focus of our literature review. As part of this effort, we analyzed previous surveys on Knowledge-Informed Predictive Models, including both theoretical works and major industrial applications, in order to gain a broad perspective on the research directions across different sectors. We used the insights from the review to generalize the concepts behind the existing methods and propose a formal analysis of predictive models, illustrated through an example. This work represents the first comprehensive survey of Knowledge-Informed Predictive

Models that includes all recent developments and spans the full range of industrial sectors.

The second contribution of this Ph.D. concerns the dissemination of Knowledge-Informed Models in real-world applications. In line with our goal of promoting their use in industry, we selected applications closely related to industrial domains, particularly in the energy, marine, materials, and chemical sectors. We targeted underexplored fields and addressed both the predictive framework (by combining existing techniques) and the generative one (by proposing new approaches). In this manuscript, we covered the two most representative applications: one in the predictive domain (Chapter 3) and one in the generative domain (Chapter 4). However, we also contributed to other projects. In the predictive domain, in addition to the physics-informed power flow analysis presented in Chapter 3, we worked on a marine application, specifically the development of a Knowledge-Informed Model to study cavitation around a hydrofoil for naval propellers design. Both projects were carried out in collaboration with Delft University of Technology. In particular, the data collection for the marine project took place during a 5-month visiting period. Regarding generative applications, we successfully applied Knowledge-Informed Models to the materials industry (i.e., metamaterial design, reported in Chapter 4), chemical industry (i.e., microstructures with desired porosity), and secondary studies (i.e., copyright-safe generation). The work on generative models was conducted in collaboration with the University of Virginia, where a 6-month visit period was conducted.

The initial objectives of this Ph.D. were achieved, both in terms of establishing the state of the art and translating knowledge into practical applications. However, these projects are not ended. The extension of the power flow analysis model is ongoing, and we are actively working on scaling the results to larger grids. The naval project on cavitation required significant time for data collection, and model development is still underway. On the generative models side, we are working to expand the integration of FEM structural models for sample correction. Finally, we are working on extending Knowledge-Informed Generative Models for microstructures with specific porosity to other properties (e.g., composition).

However, at this stage of our work, we can draw some observations. Starting with Knowledge-Informed Predictive Models, we observed that they do live up to their promise of enhancing the performance of traditional models, particularly in terms of consistency, robustness, efficiency, and transparency. In both the reviewed and the implemented applications, a clear performance improvement was observed, although, as we will detail later, this comes at the cost of increased development cost and time. We highlighted how Knowledge-Informed Predictive Models are rapidly expanding in industry and how they are continuously evolving, leading to a greater integration of physical and data-driven models. The benefits of these techniques are highly context-dependent, yet some general trends in their advantages and disadvantages can be identified. These are summarized in Table 5.1, in comparison with traditional models.

	Knowledge-Based	**Data-driven**	**Knowledge-Informed**
Information sources			
Leverage knowledge	↑↑↑	-	↑↑↑
Leverage data	↑	↑↑↑	↑↑↑
Leverage experience	↑↑↑	↑↑	↑↑↑
Leverage measured quantities	↑↑↑	-	↑↑↑
Modeling cost			
Human intervention	↑↑↑	↑	↑↑↑
Data dependency	↑	↑↑↑	↑↑
Construction cost	↑↑↑	↑↑	↑↑↑
Training cost	↑↑	↑↑↑	↑↑↑
Prediction cost	↑↑↑	↑	↑
Predictive quality and reliability			
Interpolation performance	↑↑	↑↑↑	↑↑↑
Extrapolation performance	↑↑	↑	↑↑
Physical consistency guarantee	↑↑↑	-	↑↑
Explainability	↑↑↑	↑	↑↑

Table 5.1 Predictive models performance recap. The reported performance are representative averages and may vary in specific real-world cases

As for Knowledge-Informed Generative Models, we found them to be relatively new, although they are gaining increasing attention, in line with the broader rise of generative AI. We demonstrated that Knowledge-Informed Generative Models can enforce physical consistency in generated samples and solve specific problems, such as inverse design. However, it is clear that this field still requires further research and development before it can reach widespread adoption.

Overall, it can be concluded that Knowledge-Informed Models offer clear advantages, albeit at a non-negligible cost.

Limitations: Despite the rapid expansion, it is clear that the predictive models that leverage knowledge-informing techniques currently remain quite limited. For generative models, their use is restricted to a few academic research projects. These restrictions are mainly due to the novelty of these approaches, which brings unresolved or expensive-to-resolve challenges. We believe that future developments will help overcome these limitations; however, at present, several issues remain and were encountered during the course of our work.

The major limitation we encountered is the need for multidisciplinarity. Developing Knowledge-Informed Models requires solid ML skills and deep domain expertise, along with the ability to encode that knowledge into a usable format. Professionals who possess both sets of skills are extremely rare, and there is currently no standard educational pathway to train such profiles. We expect such profiles to emerge in the future in response to market demands, but, currently, collaboration between different domains is the only option. While collaboration can be fruitful, it requires each expert to have sufficient awareness of the other's field, which is difficult to develop. Furthermore, encoding domain knowledge in a format suitable for Knowledge-Informed Models can be difficult, even for experts. Physical knowledge is not always described by well-defined mathematical equations, but in practice, often comes from practitioners' experience or sets of logical rules that require considerable effort to translate into mathematical language. This complicates the construction of Knowledge-Informed Models and limits the choice of the proper technique. All these limitations, combined with the wide variety of available techniques, make the development of Knowledge-Informed Models a complex and time-consuming task. No unified framework or flowchart currently exists to guide the selection of the most suitable approach, and trial and error remains the predominant strategy. In addition to that, practitioners often struggle to estimate the potential benefits of these models in advance.

Beyond the building challenges, another obstacle to the spread of Knowledge-Informed Models is the lack of suitable development conditions. This can occur when data cannot be collected, or when physical knowledge is limited or absent, requiring sole reliance on data. In some other cases, statistical approaches work well enough or remain the best available option. In other situations, Knowledge-Informed Models are discarded because their development costs outweigh the benefits, even when the necessary components are available. This is the case when, for example, encoding knowledge requires significant effort, adding constraints makes training overly complex, or when it is not feasible to train specialized personnel.

One way to overcome these limitations is to build an accessible, unified framework that encompasses multiple techniques and can solve various classes of similar problems. This would make implementation easier, requiring practitioners only to match their problem to a class of problems and use the framework to solve it. A similar approach has been taken with PINNs for solving differential equations, but this concept must be expanded to a broader class of problems and made available via code libraries or commercial software. We expect that industry will play a crucial role in addressing these limitations through targeted investment and development.

Future Perspectives: In our view, Knowledge-Informed Models represent one of the most promising paths to improve the trustworthiness and reliability of ML models. As shown, in several fields the use of Knowledge-Informed Models is already the most advantageous choice, though their adoption remains limited if compared to the range of potential applications. Likely, their limitations will diminish in the near future, and the research advancements will reduce development costs, encouraging adoption in areas where these models are not yet viable. We anticipate that the growing success of these models in specific fields will facilitate their use in many more sectors. As adoption increases, the market will demand skilled professionals and practical tools, which in turn will further stimulate the field. Knowledge-Informed ML will also be a key component in future AI paradigms: just as the rise of generative AI was soon followed by Knowledge-Informed Generative Models, we expect knowledge-informed approaches to be rapidly integrated into upcoming AI trends.

From a model evolution perspective, we expect the trend of deeper integration between data-driven and physical models to continue. Where initially combinations were limited to inputs and outputs, we now observe physics being embedded within the internal structure of data-driven models. We believe this will deepen further, potentially including the integration of data-driven architectures within complex physical models. In the near term, however, we expect development to focus more on expanding into industrial sectors where Knowledge-Informed Models have not yet been tested, rather than creating new techniques. This is because many existing techniques are now mature enough to provide significant advantages to several domains.

Our work fits within the broader research journey towards greater maturity and dissemination of Knowledge-Informed Models, and we believe that our projects have contributed to advancing the field, both horizontally (by building solid foundations that abstract existing techniques) and vertically (by developing new techniques and exploring new applications), and will play a role in future developments in this domain.

Key Messages: To conclude, we present the key takeaways we aim to convey with this manuscript:

- Knowledge-informed models are ML models that integrate some form of domain knowledge.
- Currently, Knowledge-Informed Models are mainly applied in predictive paradigms (Knowledge-Informed Predictive Models).
- Knowledge-informed Predictive Models can enhance the performance of purely physics-based or purely data-driven models. They improve robustness, physical consistency, and extrapolation capabilities, while also increasing efficiency, interpretability, and user trust.
- They offer new, efficient solutions to recurring industrial problems, such as surrogation and inverse problems.
- Industry is particularly receptive. Different industrial sectors employ different techniques according to their needs.
- Knowledge-Informed Predictive Models are still in their early stages and limited to a few applications. Significant adoption began only in the past five years (even if initial ideas date back to the 1990s).

- Knowledge-Informed Models are also emerging in the generative domain. Although still rare, they are attracting attention for solving specific challenges, like inverse design.
- In general, Knowledge-Informed Models entail higher development costs and time, and require interdisciplinary expertise.
- There is still a long way to go in disseminating knowledge, developing new techniques, and improving existing models. The work done in this Ph.D. represents a step forward in advancing understanding in this field.

Chapter 6

Summary of doctoral activities

This chapter provides a schematic summary of the academic activities carried out during the Ph.D. program. These include participation in research projects, scientific dissemination, teaching support, and international mobility.

Research Projects :

- Research and literature review on the state of the art of Knowledge-Informed ML models. This work was carried out under the supervision of Profs. D. Anguita and L. Oneto from the University of Genoa, in collaboration with Prof. A. Coraddu from Delft University of Technology, and with the support of Ph.D. Student G. Parodi.
- Development of a Knowledge-Informed Predictive Model, applied in the energy industry and compared with traditional models. This project was conducted under the supervision of Profs. D. Anguita, L. Oneto, and M. Robba from the University of Genoa, in collaboration with Prof. A. Coraddu from Delft University of Technology and researcher G. Ferro from the University of Genoa, and with the support of Ph.D. Student G. Parodi.
- Development of a Knowledge-Informed Generative Model, applied in the materials industry, chemical industry, and copyright. This work was conducted at the University of Virginia under the supervision of Prof. F. Fioretto from the University of Virginia, in collaboration with Profs.

D. Anguita and L. Oneto from the University of Genoa, and with the support of Ph.D. Student J. Christopher.

- Data collection campaign in the maritime industry, conducted at the cavitation tunnel of Delft University of Technology under the supervision of Profs. A. Coraddu and D. Fiscaletti from Delft University of Technology, in collaboration with Profs. D. Anguita and L. Oneto from the University of Genoa, and with the support of Ph.D. Student M. Costa.
- Development of a Knowledge-Informed Predictive Model using the data collected during the cavitation tunnel campaign at Delft University of Technology. This project was carried out under the supervision of Profs. D. Anguita and L. Oneto from the University of Genoa, in collaboration with Profs. A. Coraddu and D. Fiscaletti from Delft University of Technology.

Publications :

Journals:

- S. Zampini, G. Parodi, L. Oneto, A. Coraddu, and D. Anguita. A review on full-, zero-, and partial-knowledge based predictive models for industrial applications. Information Fusion, page 102996, 2025.([23]) - PUBLISHED

Conferences:

- G. Parodi, L. Oneto, G. Ferro, S. Zampini, M. Robba, D. Anguita, and A. Coraddu. Physics informed data driven techniques for power flow analysis. In IEEE Symposium Series on Computational Intelligence, 2023 ([24]) - PUBLISHED
- S. Zampini, J. K. Christopher, L. Oneto, D. Anguita, and F. Fioretto. Training-Free Constrained Generation With Stable Diffusion Models. NeurIPS 2025 ([25]) - IN PUBLICATION

Teaching Activities :

- Teaching assistant - Reti Logiche (UniGe) - Profs. L. Oneto, D. Anguita - A.Y. 24/25 - 6 CFU

International Mobility and Visiting Periods :

- Participation to 2023 IEEE Symposium Series on Computational Intelligence - Mexico City (Mexico) - Dec. 2023
- Visiting period at University of Virginia (ISSNAF-POLITO Project 2024) - Charlottesville (Virginia, USA) - May - Dec. 2024
- Visiting period at Delft University of Technology - Delft (Netherlands) - Feb. - Jun. 2025

Completed courses :

Table 6.1 presents the completed courses in both Hard and Soft skills.

Course	Location	Hours	Year	Type
Data mining concepts and algorithms	PoliTo	20	2022/23	H
Information visualization and visual analytics	PoliTo	20	2022/23	H
Machine learning for pattern recognition	PoliTo	20	2022/23	H
Safety Verification of Deep Neural Networks	UniGe	12	2022/23	H
Trustworthy Artificial Intelligence	UniGe	20	2022/23	H
Adversarial Machine Learning	UniGe	12	2022/23	H
Machine Learning and Data Analysis	UniGe	15 (66)	2022/23	H
Fall School 2023	PoliTo	30	2023/24	H
IBM Data Analyst	IBM Coursera	30 (60)	2023/24	H
Personal branding	PoliTo	1	2022/23	S
Design Thinking, Processes and Methods	Polito	2	2022/23	S
Project management	PoliTo	5	2022/23	S
Public speaking	PoliTo	5	2022/23	S
Writing in the Sciences	Stanford University Coursera	15	2023/24	S
Communication	PoliTo	5	2023/24	S
Navigating the hiring process: CV, tests, interview	PoliTo	2	2023/24	S
Research Ethics in Computer Science	PoliTo	20	2023/24	S
The new Internet Society: entering the black-box of digital innovations	PoliTo	6	2023/24	S
Time management	PoliTo	2	2023/24	S

Table 6.1 Completed courses in Hard (H) and Soft (S) skills (actual hours in parentheses)

Total certified hours (PoliTo):

- Hard Skills: 179h

- Soft Skills: 63h

Professional Activities :

- Mechanical Engineer - Freelance Consultant (2022-2024)
  Provided engineering consulting services in the mechanical field, supporting clients in design, analysis, and technical documentation.
- AI Engineer - Esaote SpA (2025)
  Working on the development and integration of artificial intelligence solutions for medical imaging technologies.

Chapter 7

Conclusions

This Ph.D. was centered on exploring Knowledge-Informed ML, which blends data-driven modeling with domain expertise. This encompasses a broad spectrum of techniques.

In predictive modeling, traditional models were based on domain knowledge, such as the expertise of practitioners or physical laws. While these models had a strong physical foundation, they often lacked accuracy. The introduction of data-driven models has overcome this limitation, but at the cost of interpretability and poor performance outside the training set's domain. Recently, research has begun to explore the possibility of combining the advantages of both approaches in what are known as Knowledge-Informed Predictive Models, or Partial-Knowledge Predictive Models. These models integrate the strengths of both traditional and data-driven approaches and are gaining significant traction in the industrial sector. The increase in accuracy, but more importantly in reliability and trustworthiness, makes these models highly attractive for fields that require transparency, explainability, and strict adherence to constraints.

At the same time, advancements in AI have led to the development of generative models, which can generate samples belonging to the probability distribution of a training set. Despite being a relatively recent development, generative models are already making their way into industrial applications, such as super-resolution and inverse design. A growing research trend is now focusing on incorporating domain knowledge into generative models, constraining the generated samples to ensure they meet specific properties. Though still in its early stages, this approach shows great promise for tasks such as inverse design, especially in fields where compliance with imposed constraints is crucial for high-risk applications.

The study of all these models represents the core focus of this Ph.D. research. This doctoral thesis first analyzed the state of the art of Knowledge-Informed Predictive Models. We presented a formal classification of traditional and knowledge-informed techniques, categorizing the latter into three different classes: Pre-Processing, In-Processing, and Post-Processing. This exposition is supported by an illustrative example, followed by an analysis of current industrial applications of these models.

Next, we introduced a practical application of a Knowledge-Informed Predictive Model. Specifically, we implemented an innovative technique in power flow analysis and demonstrated its superiority over traditional models.

Finally, we developed a practical Knowledge-Informed Generative Model designed for industrial applications and nearly ready for real-world deployment. In particular, we created a model for the inverse design of mechanical metamaterials, aiming to generate material geometries that yield a desired mechanical response. We showed how integrating theoretical knowledge into a data-driven generative model led to improvements over the current state of the art.

In summary, this thesis revisits the work undertaken throughout the Ph.D. journey to tackle the intricate challenge of trustworthiness and reliability in AI models. We approached this challenge by leveraging multidisciplinary expertise and incorporating physical knowledge into AI systems. Our goal was to push the boundaries of current knowledge and promote the industrial adoption of these emerging techniques. We actively contributed in this direction, addressing both methodological and application-oriented aspects. We believe that the progress made during this Ph.D. will serve as a valuable foundation for further advancements in the field.

References

[1] A. M. Turing. Computing machinery and intelligence. *Mind*, 1950.

[2] J. McCarthy, M. Minsky, N. Rochester, and C. Shannon. A proposal for the dartmouth summer research project on artificial intelligence, 1956. URL https://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html.

[3] B. Delipetrev, C. Tsinaraki, and U. Kostić. Ai watch, historical evolution of artificial intelligence. *Publications Office of the European Union*, 2020.

[4] AI Index Steering Committee. The ai index 2025 annual report, April 2025. URL https://aiindex.stanford.edu/report/.

[5] McKinsey & Company. The state of ai in 2024: Generative ai's breakout year, December 2024. URL https://www.mckinsey.com/featured-insights/generative-ai/the-state-of-ai-in-2024.

[6] S. Agarwal and S. Mishra. *Responsible AI*. Springer, 2021.

[7] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[8] D. Lazer, R. Kennedy, G. King, and A. Vespignani. The parable of google flu: traps in big data analysis. *science*, 343, 2014.

[9] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50(1):3–44, 2021.

[10] A. Fabris, N. Baranowska, M. J. Dennis, D. Graus, P. Hacker, J. Saldivar, F. Zuiderveen Borgesius, and A. J. Biega. Fairness and bias in algorithmic hiring: A multidisciplinary survey. *ACM Transactions on Intelligent Systems and Technology*, 16(1):1–54, 2025.

[11] S. Bekker. Fundamental rights in digital welfare states: The case of syri in the netherlands. In *Netherlands yearbook of international law 2019: Yearbooks in international law: History, function and future*, pages 289–307. Springer, 2020.

[12] F. M. Favarò, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju. Examining accident reports involving autonomous vehicles in california. *PLoS one*, 12, 2017.

[13] E. Shortliffe. *Computer-based medical consultations: MYCIN*, volume 2. Elsevier, 2012.

[14] F. Prendin, J. Pavan, G. Cappon, S. Del Favero, G. Sparacino, and A. Facchinetti. The importance of interpreting machine learning models for blood glucose prediction in diabetes: an analysis using shap. *Scientific reports*, 13(1):16865, 2023.

[15] R. O. Alabi, M. Elmusrati, I. Leivo, A. Almangush, and A. A. Mäkitie. Machine learning explainability in nasopharyngeal cancer survival using lime and shap. *Scientific Reports*, 13:8984, 2023.

[16] B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 2154–2156, 2018.

[17] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381, 2022.

[18] J. Yu and F. Luo. Fallback strategy for level 4+ automated driving system. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 156–162. IEEE, 2019.

[19] F. Fioretto, C. Tran, P. Van Hentenryck, and K. Zhu. Differential privacy and fairness in decisions and learning tasks: A survey. *arXiv preprint arXiv:2202.08187*, 2022.

[20] J. Ko, J. Ziani, S. Das, M. Williams, and F. Fioretto. Fairness issues and mitigations in (differentially private) socio-demographic data processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 28160–28167, 2025.

[21] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

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

[23] S. Zampini, G. Parodi, L. Oneto, A. Coraddu, and D. Anguita. A review on full-, zero-, and partial-knowledge based predictive models for industrial applications. *Information Fusion*, page 102996, 2025.

[24] G. Parodi, L. Oneto, G. Ferro, S. Zampini, M. Robba, D. Anguita, and A. Coraddu. Physics informed data driven techniques for power flow analysis. In *IEEE Symposium Series on Computational Intelligence*, 2023.

[25] S. Zampini, J. K. Christopher, L. Oneto, D. Anguita, and F. Fioretto. Training-free constrained generation with stable diffusion models. In *Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[26] Y. Xu, S. Kohtz, J. Boakye, P. Gardoni, and P. Wang. Physics-informed machine learning for reliability and systems safety applications: State of the art and challenges. *Reliability Engineering & System Safety*, 230: 108900, 2023.

[27] R. Rai and C. K. Sahu. Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (cps) focus. *IEEE Access*, 8:71050–71073, 2020.

[28] J. Wang, Y. Li, R. X Gao, and F. Zhang. Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability. *Journal of Manufacturing Systems*, 63:381–391, 2022.

[29] A. Foucquier, S. Robert, F. Suard, L. Stéphan, and A. Jay. State of the art in building modelling and energy performances prediction: A review. *Renewable and Sustainable Energy Reviews*, 23:272–288, 2013.

[30] S. Zendehboudi, N. Rezaei, and A. Lohi. Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review. *Applied energy*, 228:2539–2566, 2018.

[31] W. Guo, Z. Sun, S. B. Vilsen, J. Meng, and D. I. Stroe. Review of "grey box" lifetime modeling for lithium-ion battery: Combining physics and data-driven methods. *Journal of Energy Storage*, 56:105992, 2022.

[32] M. Von Stosch, R. Oliveira, J. Peres, and S. F. de Azevedo. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers & Chemical Engineering*, 60:86–101, 2014.

[33] X. Li and J. Wen. Review of building energy modeling for control and operation. *Renewable and Sustainable Energy Reviews*, 37:517–537, 2014.

[34] M. Zhou, H. Jin, and W. Wang. A review of vehicle fuel consumption models to evaluate eco-driving and eco-routing. *Transportation Research Part D: Transport and Environment*, 49:203–218, 2016.

[35] 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*, 29(10):2318–2331, 2017.

[36] A. Boodi, K. Beddiar, M. Benamour, Y. Amirat, and M. Benbouzid. Intelligent systems for building energy and occupant comfort optimization: A state of the art review and recommendations. *Energies*, 11(10):2604, 2018.

[37] 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*, 2020.

[38] S. Yang, P. Navarathna, S. Ghosh, and B. W. Bequette. Hybrid modeling in the era of smart manufacturing. *Computers & Chemical Engineering*, 140:106874, 2020.

[39] A. A. A. Gassar and S. H. Cha. Energy prediction techniques for large-scale buildings towards a sustainable built environment: A review. *Energy and Buildings*, 224:110238, 2020.

[40] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 2020.

[41] Y. Li, Z. O'Neill, L. Zhang, J. Chen, P. Im, and J. DeGraw. Grey-box modeling and application for building energy simulations-a critical review. *Renewable and Sustainable Energy Reviews*, 146:111174, 2021.

[42] J. Sansana, M. N. Joswiak, I. Castillo, Z. Wang, R. Rendall, L. H. Chiang, and M. S. Reis. Recent trends on hybrid modeling for industry 4.0. *Computers & Chemical Engineering*, 151:107365, 2021.

[43] 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 Schuecker J. Informed machine learning-a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2021.

[44] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3 (6):422–440, 2021.

[45] S. Tamilselvi, S. Gunasundari, N. Karuppiah, A. Razak RK, S. Madhusudan, V. M. Nagarajan, T. Sathish, M. Zubair M. Shamim, C. A. Saleel, and A. Afzal. A review on battery modelling techniques. *Sustainability*, 13(18):10042, 2021.

[46] M. Mozaffar, S. Liao, X. Xie, S. Saha, C. Park, J. Cao, W. K. Liu, and Z. Gan. Mechanistic artificial intelligence (mechanistic-ai) for modeling, design, and control of advanced manufacturing processes: Current state and perspectives. *Journal of Materials Processing Technology*, 302:117485, 2022.

[47] S. R. Vadyala, S. N. Betgeri, J. C. Matthews, and E. Matthews. A review of physics-based machine learning in civil engineering. *Results in Engineering*, 13:100316, 2022.

[48] N. Sharma and Y. A. Liu. A hybrid science-guided machine learning approach for modeling chemical processes: A review. *AIChE Journal*, 68 (5):e17609, 2022.

[49] B. Huang and J. Wang. Applications of physics-informed neural networks in power systems-a review. *IEEE Transactions on Power Systems*, 38(1): 572–588, 2022.

[50] W. Bradley, J. Kim, Z. Kilwein, L. Blakely, M. Eydenberg, J. Jalvin, C. Laird, and F. Boukouvala. Perspectives on the integration between first-principles and data-driven modeling. *Computers & Chemical Engineering*, 166:107898, 2022.

[51] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

[52] A. Fan, J. Yang, L. Yang, D. Wu, and N. Vladimir. A review of ship fuel consumption models. *Ocean engineering*, 264:112405, 2022.

[53] Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, H. Su, and J. Zhu. Physics-informed machine learning: A survey on problems, methods and applications. *arXiv preprint arXiv:2211.08064*, 2022.

[54] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Computing Surveys*, 55(4):1–37, 2022.

[55] W. H Joerding and J. L. Meador. Encoding a priori information in feedforward networks. *Neural Networks*, 4(6):847–856, 1991.

[56] D. C. Psichogios and L. H. Ungar. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511, 1992.

[57] M. Agarwal. Combining neural and conventional paradigms for modeling, prediction, and control. In *International Conference on Control Applications*, 1995.

[58] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. arxiv 2017. *arXiv preprint arXiv:1711.10566*, 2017.

[59] S. Moura. Chapter 1: Modeling and systems analysis. https://ecal.berkeley.edu/files/ce295/CH01-ModelingSystems.pdf, 2018. Accessed: 2024-01-28.

[60] A. B. Jiménez, J. L. Lázaro, and J. R. Dorronsoro. *Innovations in Hybrid Intelligent Systems*. Springer Berlin Heidelberg, 2007.

[61] L. Miglianti, F. Cipollini, L. Oneto, G. Tani, and Others. Predicting the cavitating marine propeller noise at design stage: A deep learning based approach. *Ocean Engineering*, 209:107481, 2020.

[62] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold, and P. M. Atkinson. Explainable artificial intelligence: an analytical review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(5):e1424, 2021.

[63] K. Sargsyan, H. N. Najm, and R. Ghanem. On the statistical calibration of physical models. *International Journal of Chemical Kinetics*, 47(4): 246–276, 2015.

[64] Y. Liu and G. Glass. Effects of mesh density on finite element analysis. Technical report, SAE Technical Paper, 2013.

[65] C. C. Aggarwal. *Data mining: the textbook*, volume 1. Springer, 2015.

[66] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[67] Z. H. Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[68] C. C. Aggarwal. *Neural networks and deep learning: a textbook*. Spinger, 2023.

[69] D. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[70] D. H. Wolpert. The supervised learning no-free-lunch theorems. In *Soft computing and industry*, 2002.

[71] L. Oneto. *Model Selection and Error Estimation in a Nutshell*. Springer, 2020.

[72] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke. Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8:42200–42216, 2020.

[73] D. Joenssen, A. S. Hada, and J. Lenz. Physics-informed imputation for data cleaning and pre-processing in robust smart manufacturing systems. *Procedia Computer Science*, 232:377–387, 2024.

[74] S. Liu, B. B. Kappes, B. Amin-ahmadi, O. Benafan, X. Zhang, and A. P. Stebner. Physics-informed machine learning for composition-process-property design: Shape memory alloy demonstration. *Applied Materials Today*, 22:100898, 2021.

[75] J. P. C. Kleijnen. Design of experiments: overview. In *2008 Winter Simulation Conference*, 2008.

[76] N. Gupta, H. Patel, S. Afzal, N. Panwar, and Others. Data quality toolkit: Automatic assessment of data quality and remediation for machine learning datasets. *arXiv preprint arXiv:2108.05935*, 2021.

[77] E. Giunchiglia, F. Imrie, M. Van Der Schaar, and T. Lukasiewicz. Machine learning with requirements: a manifesto. *arXiv preprint arXiv:2304.03674*, 2023.

[78] L. W. Cai. *Fundamentals of mechanical vibrations*. John Wiley & Sons, 2016.

[79] B. P. Flannery, W. H. Press, S. A. Teukolsky, and W. Vetterling. Numerical recipes in c. *Press Syndicate of the University of Cambridge, New York*, 24(78):36, 1992.

[80] L. Oneto, S. Ridella, and D. Anguita. Do we really need a new theory to understand over-parameterization? *Neurocomputing*, 543:126227, 2023.

[81] L. A. M. Castro, R. P. Bewick, and T. G. Carter. An overview of numerical modelling applied to deep mining. *Innovative numerical modelling in geomechanics*, pages 393–414, 2012.

[82] S. Rajendran, T. R. Vakamalla, and N. Mangadoddy. Numerical methods in mineral processing: an overview. *Mineral Processing*, pages 251–285, 2023.

[83] G. Beer, C. Dunser, and M. Noronha. Recent advances in the numerical simulation of tunnel excavation. In *ISRM Congress*, 2003.

[84] K. Balaji, M. Rabiei, V. Suicmez, C. H. Canbaz, and Others. Status of data-driven methods and their applications in oil and gas industry. In *SPE Europec featured at EAGE Conference and Exhibition?*, 2018.

[85] M. Jooshaki, A. Nad, and S. Michaux. A systematic review on the application of machine learning in exploiting mineralogical data in mining and mineral industry. *Minerals*, 11(8):816, 2021.

[86] H. Zhao, Z. Kang, X. Zhang, H. Sun, and Others. A physics-based data-driven numerical model for reservoir history matching and prediction with a field application. *SPE Journal*, 21(06):2175–2194, 2016.

[87] H. Zhao, Y. Li, S. Cui, G. Shang, A. C. Reynolds, Z. Guo, and H. A. Li. History matching and production optimization of water flooding based on a data-driven interwell numerical simulation model. *Journal of Natural Gas Science and Engineering*, 31:48–66, 2016.

[88] Z. Guo, A. C. Reynolds, and H. Zhao. A physics-based data-driven model for history-matching, prediction and characterization of waterflooding performance. In *SPE Reservoir Simulation Conference?*, 2017.

[89] Z. Guo, A. C. Reynolds, and H. Zhao. Waterflooding optimization with the insim-ft data-driven model. *Computational Geosciences*, 22:745–761, 2018.

[90] Y. Zhang, J. He, C. Yang, J. Xie, and Others. A physics-based data-driven model for history matching, prediction, and characterization of unconventional reservoirs. *SPE Journal*, 23(04):1105–1125, 2018.

[91] Y. Li and M. Onur. Insim-bhp: A physics-based data-driven reservoir model for history matching and forecasting with bottomhole pressure and production rate data under waterflooding. *Journal of Computational Physics*, 473:111714, 2023.

[92] Y. Li, F. O. Alpak, V. Jain, R. Lu, and M. Onur. History-matching and forecasting production rate and bottomhole pressure data using an enhanced physics-based data-driven simulator. *SPE Reservoir Evaluation & Engineering*, 26(03):957–974, 2023.

[93] Z. L. Jin, Y. Liu, and L. J. Durlofsky. Deep-learning-based surrogate model for reservoir simulation with time-varying well controls. *Journal of Petroleum Science and Engineering*, 192:107273, 2020.

[94] D. Robert Harp, D. O'Malley, B. Yan, and R. Pawar. On the feasibility of using physics-informed machine learning for underground reservoir pressure management. *Expert Systems with Applications*, 178:115006, 2021.

[95] M. M. Almajid and M. O. Abu-Al-Saud. Prediction of porous media fluid flow using physics informed neural networks. *Journal of Petroleum Science and Engineering*, 208:109205, 2022.

[96] K. Zhang, Y. Zuo, H. Zhao, X. Ma, and Others. Fourier neural operator for solving subsurface oil/water two-phase flow partial differential equation. *Spe Journal*, 27(03):1815–1830, 2022.

[97] M. S. F. Bangi and J. S. Kwon. Deep hybrid modeling of chemical process: Application to hydraulic fracturing. *Computers & Chemical Engineering*, 134:106696, 2020.

[98] M. Tembely, A. M. AlSumaiti, and W. S. Alameri. Machine and deep learning for estimating the permeability of complex carbonate rock from x-ray micro-computed tomography. *Energy Reports*, 7:1460–1472, 2021.

[99] B. W. Bequette. *Process Dynamics: Modeling Analysis and Simulation.* Prentice Hall Incorporated, 1998.

[100] I. T. Cameron and K. Hangos. *Process modelling and model analysis*. Elsevier, 2001.

[101] J. A. Moulijn, M. Makkee, and A. E. Van Diepen. *Chemical process technology*. John Wiley & Sons, 2013.

[102] Z. Ge, Z. Song, and F. Gao. Review of recent research on data-based process monitoring. *Industrial & Engineering Chemistry Research*, 52 (10):3543–3562, 2013.

[103] M. S Reis and G. Gins. Industrial process monitoring in the big data/industry 4.0 era: From detection to diagnosis, to prognosis. *Processes*, 5(3):35, 2017.

[104] M. S. Reis and P. M. Saraiva. Data-centric process systems engineering for the chemical industry 4.0. *Systems Engineering in the Fourth Industrial Revolution*, pages 137–159, 2019.

[105] B. Sanchez-Lengeling and A. Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. *Science*, 361(6400):360–365, 2018.

[106] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, and Others. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

[107] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715): 547–555, 2018.

[108] S. I. Ngo and Y. Lim. Solution and parameter identification of a fixed-bed reactor model for catalytic co2 methanation using physics-informed neural networks. *Catalysts*, 11(11):1304, 2021.

[109] L. Scheffold, T. Finkler, and U. Piechottka. Gray-box system modeling using symbolic regression and nonlinear model predictive control of a semibatch polymerization. *Computers & Chemical Engineering*, 146: 107204, 2021.

[110] F. Vega-Ramon, X. Zhu, T. R. Savage, P. Petsagkourakis, K. Jing, and D. Zhang. Kinetic and hybrid modeling for yeast astaxanthin production under uncertainty. *Biotechnology and Bioengineering*, 118(12):4854–4866, 2021.

[111] K. Li, H. Duan, L. Liu, R. Qiu, B. van den Akker, and Others. An integrated first principal and deep learning approach for modeling nitrous oxide emissions from wastewater treatment plants. *Environmental Science & Technology*, 56(4):2816–2826, 2022.

[112] T. Asrav and E. Aydin. Physics-informed recurrent neural networks and hyper-parameter optimization for dynamic process systems. *Computers & Chemical Engineering*, 173:108195, 2023.

[113] D. R. Cassar. Viscnet: Neural network for predicting the fragility index and the temperature-dependency of viscosity. *Acta materialia*, 206:116602, 2021.

[114] I. Chiniforooshan Esfahani. A data-driven physics-informed neural network for predicting the viscosity of nanofluids. *AIP Advances*, 13(2), 2023.

[115] H. Narayanan, T. Seidler, M. F. Luna, M. Sokolov, M. Morbidelli, and A. Butté. Hybrid models for the simulation and prediction of chromatographic processes for protein capture. *Journal of Chromatography A*, 1650:462248, 2021.

[116] J. Ren, D. Liu, and Y. Wan. Modeling and application of czochralski silicon single crystal growth process using hybrid model of data-driven and mechanism-based methodologies. *Journal of Process Control*, 104: 74–85, 2021.

[117] M. S. F. Bangi, K. Kao, and J. S. Kwon. Physics-informed neural networks for hybrid modeling of lab-scale batch fermentation for β-carotene production using saccharomyces cerevisiae. *Chemical Engineering Research and Design*, 179:415–423, 2022.

[118] P. Shah, M. Z. Sheriff, M. S. F. Bangi, C. Kravaris, J. S. Kwon, C. Botre, and J. Hirota. Deep neural network-based hybrid modeling and experimental validation for an industry-scale fermentation process: Identification of time-varying dependencies among parameters. *Chemical Engineering Journal*, 441:135643, 2022.

[119] N. Roussel, J. Spangenberg, J. Wallevik, and R. Wolfs. Numerical simulations of concrete processing: From standard formative casting to additive manufacturing. *Cement and Concrete Research*, 135:106075, 2020.

[120] M. C. Kuşhan, Ö. Poyraz, Y. Uzononat, and O. Sezan. Systematical review on the numerical simulations of laser powder bed additive manufacturing. *Sigma Journal of Engineering and Natural Sciences*, 36(4):1197–1214, 2018.

[121] T. Wuest, D. Weimer, C. Irgens, and K. Thoben. Machine learning in manufacturing: advantages, challenges, and applications. *Production & Manufacturing Research*, 4(1):23–45, 2016.

[122] Dmitry I. Rahul R., Manoj K. T. and Alexandre D. Machine learning in manufacturing and industry 4.0 applications. *International Journal of Production Research*, 59(16):4773–4778, 2021.

[123] A. Dogan and D. Birant. Machine learning and data mining in manufacturing. *Expert Systems with Applications*, 166:114060, 2021.

[124] H. Jeong, J. Bai, C. P. Batuwatta-Gamage, C. Rathnayaka, and Others. A physics-informed neural network-based topology optimization (pinnto) framework for structural optimization. *Engineering Structures*, 278: 115484, 2023.

[125] J. He, C. Chadha, S. Kushwaha, S. Koric, D. Abueidda, and I. Jasiuk. Deep energy method in topology optimization applications. *Acta Mechanica*, 234(4):1365–1379, 2023.

[126] A. Gaikwad, R. Yavari, M. Montazeri, K. Cole, L. Bian, and P. Rao. Toward the digital twin of additive manufacturing: Integrating thermal simulations, sensing, and analytics to detect process faults. *Iise Transactions*, 52(11):1204–1217, 2020.

[127] K. Shukla, P. C. Di Leoni, J. Blackshire, D. Sparkman, and G. E. Karniadakis. Physics-informed neural network for ultrasound nondestructive quantification of surface breaking cracks. *Journal of Nondestructive Evaluation*, 39:1–20, 2020.

[128] E. Zhang, M. Dao, G. E. Karniadakis, and S. Suresh. Analyses of internal structures and defects in materials using physics-informed neural networks. *Science advances*, 8(7):eabk0644, 2022.

[129] N. Zobeiry and K. D. Humfeld. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Engineering Applications of Artificial Intelligence*, 101:104232, 2021.

[130] Y. Li, J. Wang, Z. Huang, and R. X. Gao. Physics-informed meta learning for machining tool wear prediction. *Journal of Manufacturing Systems*, 62:17–27, 2022.

[131] T. Zhou, S. Jiang, T. Han, S. Zhu, and Y. Cai. A physically consistent framework for fatigue life prediction using probabilistic physics-informed neural network. *International Journal of Fatigue*, 166:107234, 2023.

[132] Q. Zhu, Z. Liu, and J. Yan. Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Computational Mechanics*, 67:619–635, 2021.

[133] R. Liu, S. Liu, and X. Zhang. A physics-informed machine learning model for porosity analysis in laser powder bed fusion additive manufacturing. *The International Journal of Advanced Manufacturing Technology*, 113 (7-8):1943–1958, 2021.

[134] E. Salvati, A. Tognan, L. Laurenti, M. Pelegatti, and F. De Bona. A defect-based physics-informed machine learning framework for fatigue finite life prediction in additive manufacturing. *Materials & Design*, 222: 111089, 2022.

[135] A. Ciampaglia, A. Tridello, D. S. Paolino, and F. Berto. Data driven method for predicting the effect of process parameters on the fatigue response of additive manufactured alsi10mg parts. *International journal of fatigue*, 170:107500, 2023.

[136] L. Wang, S. Zhu, C. Luo, D. Liao, and Q. Wang. Physics-guided machine learning frameworks for fatigue life prediction of am materials. *International Journal of Fatigue*, 172:107658, 2023.

[137] N. Zobeiry, J. Reiner, and R. Vaziri. Theory-guided machine learning for damage characterization of composites. *Composite Structures*, 246: 112407, 2020.

[138] C. Rao, H. Sun, and Y. Liu. Physics-informed deep learning for computational elastodynamics without labeled data. *Journal of Engineering Mechanics*, 147(8):04021043, 2021.

[139] S. A. Niaki, E. Haghighat, T. Campbell, A. Poursartip, and R. Vaziri. Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture. *Computer Methods in Applied Mechanics and Engineering*, 384:113959, 2021.

[140] S. Niu, E. Zhang, Y. Bazilevs, and V. Srivastava. Modeling finite-strain plasticity using physics-informed neural network and assessment of the network performance. *Journal of the Mechanics and Physics of Solids*, 172:105177, 2023.

[141] S. P. Shepherd. A review of system dynamics models applied in transportation. *Transportmetrica B: Transport Dynamics*, 2(2):83–105, 2014.

[142] K. Tsolaki, T. Vafeiadis, A. Nizamis, D. Ioannidis, and D. Tzovaras. Utilizing machine learning on freight transportation and logistics applications: A review. *ICT Express*, 2022.

[143] H. Nguyen, L. Kieu, T. Wen, and C. Cai. Deep learning methods in transportation domain: a review. *IET Intelligent Transport Systems*, 12 (9):998–1004, 2018.

[144] Giulia Cademartori, Luca Oneto, Federica Valdenazzi, Andrea Coraddu, Andrea Gambino, and Davide Anguita. A review on ship motions and quiescent periods prediction models. *Ocean Engineering*, 280:114822, 2023.

[145] A. Coraddu, L. Oneto, F. Baldi, and D. Anguita. Ship efficiency forecast based on sensors data collection: Improving numerical models through data analytics. In *OCEANS 2015-Genova*, 2015.

[146] A. Coraddu, L. Oneto, F. Baldi, and D. Anguita. Vessels fuel consumption forecast and trim optimisation: A data analytics perspective. *Ocean Engineering*, 130:351–370, 2017.

[147] L. Yang, G. Chen, N. G. M. Rytter, J. Zhao, and D. Yang. A genetic algorithm-based grey-box model for ship fuel consumption prediction towards sustainable shipping. *Annals of Operations Research*, pages 1–27, 2019.

[148] X. Wang, Z. Zou, X. Hou, and F. Xu. System identification modelling of ship manoeuvring motion based on ϵ-support vector regression. *Journal of Hydrodynamics*, 27(4):502–512, 2015.

[149] X. Wang, Z. Zou, L. Yu, and W. Cai. System identification modeling of ship manoeuvring motion in 4 degrees of freedom based on support vector machines. *China Ocean Engineering*, 29(4):519–534, 2015.

[150] T. Wang, G. Li, L. I. Hatledal, R. Skulstad, V. Æsøy, and H. Zhang. Incorporating approximate dynamics into data-driven calibrator: A representative model for ship maneuvering prediction. *IEEE Transactions on Industrial Informatics*, 18(3):1781–1789, 2021.

[151] X. Lang, D. Wu, and W. Mao. Physics-informed machine learning models for ship speed prediction. *Expert Systems with Applications*, 238:121877, 2024.

[152] T. Vairo, D. Cademartori, D. Clematis, M. P. Carpanese, and B. Fabiano. Solid oxide fuel cells for shipping: A machine learning model for early detection of hazardous system deviations. *Process Safety and Environmental Protection*, 172:184–194, 2023.

[153] R. G. Nascimento, M. Corbetta, C. S. Kulkarni, and F. A. C. Viana. Hybrid physics-informed neural networks for lithium-ion battery modeling and prognosis. *Journal of Power Sources*, 513:230526, 2021.

[154] H. Pang, L. Wu, J. Liu, X. Liu, and K. Liu. Physics-informed neural network approach for heat generation rate estimation of lithium-ion battery under various driving conditions. *Journal of Energy Chemistry*, 78:1–12, 2023.

[155] A. Dourado and F. A. C. Viana. Physics-informed neural networks for missing physics estimation in cumulative damage models: a case study in corrosion fatigue. *Journal of Computing and Information Science in Engineering*, 20(6):061007, 2020.

[156] W. Q. Hao, L. Tan, X. G. Yang, D. Q. Shi, M. L. Wang, G. L. Miao, and Y. S. Fan. A physics-informed machine learning approach for notch fatigue evaluation of alloys used in aerospace. *International Journal of Fatigue*, 170:107536, 2023.

[157] A. Coraddu, L. Oneto, D. Ilardi, S. Stoumpos, and G. Theotokatos. Marine dual fuel engines monitoring in the wild through weakly supervised data analytics. *Engineering Applications of Artificial Intelligence*, 100: 104179, 2021.

[158] R. Yan, S. Wang, J. Cao, and D. Sun. Shipping domain knowledge informed prediction and optimization in port state control. *Transportation Research Part B: Methodological*, 149:52–78, 2021.

[159] A. J. Huang and S. Agarwal. Physics informed deep learning for traffic state estimation. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020.

[160] S. Kakac, A. Pramuanjaroenkij, and X. Y. Zhou. A review of numerical modeling of solid oxide fuel cells. *International journal of hydrogen energy*, 32(7):761–786, 2007.

[161] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, and Others. Review of photovoltaic power forecasting. *Solar energy*, 136:78–111, 2016.

[162] F. Alobaid, N. Mertens, R. Starkloff, T. Lanz, and Others. Progress in dynamic simulation of thermal power plants. *Progress in energy and combustion science*, 59:79–162, 2017.

[163] K. Zhou, C. Fu, and S. Yang. Big data driven smart energy management: From big data to big insights. *Renewable and sustainable energy reviews*, 56:215–225, 2016.

[164] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng. A review of deep learning for renewable energy forecasting. *Energy Conversion and Management*, 198:111799, 2019.

[165] Y. Li, K. Liu, A. M. Foley, A. Zülke, M. Berecibar, and Others. Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review. *Renewable and sustainable energy reviews*, 113:109254, 2019.

[166] Q. Wang, Y. He, J. Shen, Z. Ma, and G. Zhong. A unified modeling framework for lithium-ion batteries: An artificial neural network based thermal coupled equivalent circuit model approach. *Energy*, 138:118–132, 2017.

[167] W. Li, J. Zhang, F. Ringbeck, D. Jöst, and Others. Physics-informed neural networks for electrode-level state estimation in lithium-ion batteries. *Journal of Power Sources*, 506:230034, 2021.

[168] J. Shi, A. Rivera, and D. Wu. Battery health management using physics-informed machine learning: Online degradation modeling and remaining useful life prediction. *Mechanical Systems and Signal Processing*, 179: 109347, 2022.

[169] H. Tu, S. Moura, Y. Wang, and H. Fang. Integrating physics-based modeling with machine learning for lithium-ion batteries. *Applied Energy*, 329:120289, 2023.

[170] B. Sun, J. Pan, Z. Wu, Q. Xia, and Others. Adaptive evolution enhanced physics-informed neural networks for time-variant health prognosis of lithium-ion batteries. *Journal of Power Sources*, 556:232432, 2023.

[171] Y. A Yucesan and F. A. C. Viana. A physics-informed neural network for wind turbine main bearing fatigue. *International Journal of Prognostics and Health Management*, 11(1), 2020.

[172] Y. A. Yucesan and F. A. C. Viana. A hybrid physics-informed neural network for main bearing fatigue prognosis under grease quality variation. *Mechanical Systems and Signal Processing*, 171:108875, 2022.

[173] M. J. Mayer. Benefits of physical and machine learning hybridization for photovoltaic power forecasting. *Renewable and Sustainable Energy Reviews*, 168:112772, 2022.

[174] H. Li, Y. Wang, J. Zhu, and M. Han. Performance map and operating condition optimization of industrial-size sofcs using a pinn surrogate model. *International Journal of Hydrogen Energy*, 50:1294–1307, 2024.

[175] H. Gong, S. Cheng, Z. Chen, and Q. Li. Data-enabled physics-informed machine learning for reduced-order modeling digital twin: application to nuclear reactor physics. *Nuclear Science and Engineering*, 196(6):668–693, 2022.

[176] M. Gnatowski, S. Buchaniec, and G. Brus. The prediction of the polarization curves of a solid oxide fuel cell anode with an artificial neural network supported numerical simulation. *International Journal of Hydrogen Energy*, 48(31):11823–11830, 2021.

[177] L. Ma, J. Tian, T. Zhang, Q. Guo, and C. Hu. Accurate and efficient remaining useful life prediction of batteries enabled by physics-informed machine learning. *Journal of Energy Chemistry*, 91:512–521, 2024.

[178] X. Zhao, K. Shirvan, R. K. Salko, and F. Guo. On the prediction of critical heat flux using a physics-informed machine learning-aided framework. *Applied Thermal Engineering*, 164:114540, 2020.

[179] M. E. C. Bento. Physics-guided neural network for load margin assessment of power systems. *IEEE Transactions on Power Systems*, 2023.

[180] H. Wang and Z. J. Zhai. Advances in building simulation and computational techniques: A review between 1987 and 2014. *Energy and Buildings*, 128:319–335, 2016.

[181] L. Mei and Q. Wang. Structural optimization in civil engineering: a literature review. *Buildings*, 11(2):66, 2021.

[182] T. Hong, Z. Wang, X. Luo, and W. Zhang. State-of-the-art on research and applications of machine learning in the building life cycle. *Energy and Buildings*, 212:109831, 2020.

[183] M. Azimi, A. D. Eslamlou, and G. Pekcan. Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. *Sensors*, 20(10):2778, 2020.

[184] K. Arendt, M. Jradi, H. R. Shaker, and C. Veje. Comparative analysis of white-, gray-and black-box models for thermal simulation of indoor environment: Teaching building case study. In *Building Performance Analysis Conference and SimBuild: Co-organized by ASHRAE and IBPSA-USA*, 2018.

[185] S. Chen, X. Zhou, G. Zhou, C. Fan, P. Ding, and Q. Chen. An online physical-based multiple linear regression model for building's hourly cooling load prediction. *Energy and Buildings*, 254:111574, 2022.

[186] F. Bünning, B. Huber, A. Schalbetter, A. Aboudonia, and Others. Physics-informed linear regression is competitive with two machine learning methods in residential building mpc. *Applied Energy*, 310:118491, 2022.

[187] G. Gokhale, B. Claessens, and C. Develder. Physics informed neural networks for control oriented thermal modeling of buildings. *Applied Energy*, 314:118852, 2022.

[188] Y. Chen, Q. Yang, Z. Chen, C. Yan, S. Zeng, and M. Dai. Physics-informed neural networks for building thermal modeling and demand response control. *Building and Environment*, 234:110149, 2023.

[189] J. Vivian, E. Prataviera, N. Gastaldello, and A. Zarrella. A comparison between grey-box models and neural networks for indoor air temperature prediction in buildings. *Journal of Building Engineering*, 84:108583, 2024.

[190] R. Zhang, Y. Liu, and H. Sun. Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling. *Engineering Structures*, 215:110704, 2020.

[191] Z. Zhang and C. Sun. Structural damage identification via physics-guided machine learning: a methodology integrating pattern recognition with finite element model updating. *Structural Health Monitoring*, 20(4): 1675–1688, 2021.

[192] Y. Li, P. Ni, L. Sun, and Y. Xia. Finite element model-informed deep learning for equivalent force estimation and full-field response calculation. *Mechanical Systems and Signal Processing*, 206:110892, 2024.

[193] S. He, W. Zhou, and C. Tang. Physics-informed neural networks for settlement analysis of the immersed tunnel of the hong kong–zhuhai–macau bridge. *International Journal of Geomechanics*, 24(1):04023241, 2024.

[194] W. Ouyang, G. Li, L. Chen, and S. Liu. Machine learning-based soil–structure interaction analysis of laterally loaded piles through physics-informed neural networks. *Acta Geotechnica*, pages 1–26, 2024.

[195] A. Primadianto and C. N. Lu. A review on distribution system state estimation. *IEEE Transactions on Power Systems*, 32(5):3875–3883, 2016.

[196] S. Iwamoto and Y. Tamura. A load flow calculation method for ill-conditioned power systems. *IEEE transactions on power apparatus and systems*, pages 1736–1743, 1981.

[197] M. Karimi, A. Shahriari, M. R. Aghamohammadi, H. Marzooghi, and V. Terzija. Application of newton-based load flow methods for determining steady-state condition of well and ill-conditioned power systems: A review. *International Journal of Electrical Power & Energy Systems*, 113:298–309, 2019.

[198] Y. Liu, N. Zhang, Y. Wang, J. Yang, and C. Kang. Data-driven power flow linearization: A regression approach. *IEEE Transactions on Smart Grid*, 10(3):2569–2580, 2018.

[199] L. Guo, Y. Zhang, X. Li, Z. Wang, Y. Liu, L. Bai, and C. Wang. Data-driven power flow calculation method: A lifting dimension linear regression approach. *IEEE Transactions on Power Systems*, 37(3):1798–1808, 2021.

[200] J. Yu, Y. Weng, and R. Rajagopal. Mapping rule estimation for power flow analysis in distribution grids. *arXiv preprint arXiv:1702.07948*, 2017.

[201] A. Karami and M.S. Mohammadi. Radial basis function neural network for power system load-flow. *International Journal of Electrical Power & Energy Systems*, 30(1):60–66, 2008.

[202] V. Bolz, J. Rueß, and A. Zell. Power flow approximation based on graph convolutional networks. In *IEEE international conference on machine learning and applications*, 2019.

[203] M. Xiang, J. Yu, Z. Yang, Y. Yang, H. Yu, and H. He. Probabilistic power flow with topology changes based on deep neural network. *International Journal of Electrical Power & Energy Systems*, 117:105650, 2020.

[204] X. Hu, H. Hu, S. Verma, and Z. Zhang. Physics-guided deep neural networks for power flow analysis. *IEEE Transactions on Power Systems*, 36(3):2082–2092, 2020.

[205] H. H. Müller, M. J. Rider, and C. A. Castro. Artificial neural networks for load flow and external equivalents studies. *Electric power systems research*, 80(9):1033–1041, 2010.

[206] Y. Yang, Z. Yang, J. Yu, B. Zhang, Y. Zhang, and H. Yu. Fast calculation of probabilistic power flow: A model-based deep learning approach. *IEEE Transactions on Smart Grid*, 11(3):2235–2244, 2019.

[207] W. H. Kersting. *Distribution system modeling and analysis*. CRC press, 2017.

[208] J. J. Grainger. *Power system analysis*. McGraw-Hill, 1999.

[209] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on power systems*, 26(1): 12–19, 2010.

[210] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[211] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[212] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[213] S. Sai, R. Sai, and V. Chamola. Generative ai for industry 5.0: Analyzing the impact of chatgpt, dalle, and other models. *IEEE Open Journal of the Communications Society*, 2024.

[214] L. Regenwetter, A. H. Nobari, and F. Ahmed. Deep generative models in engineering design: A review. *Journal of Mechanical Design*, 144(7): 071704, 2022.

[215] J. Bastek and D. M. Kochmann. Inverse design of nonlinear mechanical metamaterials via video denoising diffusion models. *Nature Machine Intelligence*, pages 1466–1475, 2023.

[216] Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[217] G. Giannone, A. Srivastava, O. Winther, and F. Ahmed. Aligning optimization trajectories with diffusion models for constrained design generation. *Advances in Neural Information Processing Systems*, 36: 51830–51861, 2023.

[218] J. Ho and T. Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[219] H. Ye, H. Lin, J. Han, M. Xu, S. Liu, Y. Liang, J. Ma, J. Zou, and S. Ermon. Tfg: Unified training-free guidance for diffusion models. *arXiv preprint arXiv:2409.15761*, 2024.

[220] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

[221] T. J. Wang, J. Zheng, P. Ma, Y. Du, B. Kim, A. Spielberg, J. Tenenbaum, C. Gan, and D. Rus. Diffusebot: Breeding soft robots with physics-augmented generative diffusion models. *Advances in Neural Information Processing Systems*, 36:44398–44423, 2023.

[222] T. Frerix, M. Nießner, and D. Cremers. Homogeneous linear inequality constraints for neural network activations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 748–749, 2020.

[223] G. Liu, T. Chen, E. Theodorou, and M. Tao. Mirror diffusion models for constrained and watermarked generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[224] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[225] Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J. Vert, and F. Bach. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020.

[226] J. Mandi, J. Kotary, S. Berden, M. Mulamba, V. Bucarey, T. Guns, and F. Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, pages 1623–1701, 2024.

[227] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005.

[228] J. C. Simo and T. J. R. Hughes. *Computational inelasticity*, volume 7. Springer Science & Business Media, 2006.

[229] A. Pandolfi. Analisi non lineari di strutture con il metodo degli elementi finiti, 2003. URL https://www.researchgate.net/publication/257495153_Corso_Analisi_non_lineari_di_strutture_con_il_metodo_degli_Elementi_Finiti.

[230] L. Börgesson. Abaqus. In *Developments in geotechnical engineering*, volume 79, pages 565–570. Elsevier, 1996.