Machine Learning to speed up Computational Fluid Dynamics engineering simulations for built environments: A review

Clément Caron, Philippe Lauret, Alain Bastide

Building and Environment 267 (2025) 112229

Contents lists available at ScienceDirect

Building and Environment

journal homepage: www.elsevier.com/locate/buildenv

Machine Learning to speed up Computational Fluid Dynamics engineering simulations for built environments: A review

Clément Caron [a,b,*], Philippe Lauret [a], Alain Bastide [a]

[a] Department of Sustainable Built Environment, PIMENT lab, University of Reunion, 117 rue du Général Ailleret, Le Tampon, 97430, La Réunion, France
[b] INTEGRALE INGENIERIE, 4 bis rue Fond Générèse, Saint-Gilles-Les-Hauts, 97435, La Réunion, France

ARTICLE INFO

Keywords:
Artificial intelligence
Machine learning
Computational fluid dynamics
Built environment

ABSTRACT

Computational fluid dynamics (CFD) is a valuable tool in designing built environments, enhancing comfort, health, energy efficiency, and safety in both indoor and outdoor applications. Nevertheless, the time required for CFD computations still needs to be reduced for engineering studies. Recent advances in machine learning (ML) techniques offer a promising avenue for developing fast-running data-driven models for physics-related phenomena. As scientific machine learning (SciML) research increasingly focuses on efficiently coupling ML and CFD techniques, this literature review highlights the growing number of applications in the built environment field to accelerate CFD simulations. This work aims to identify emerging trends and challenges in incorporating ML techniques into built environment flow simulations to foster further advancements in this domain. The prevailing approaches are direct surrogate modeling and reduced-order models (ROMs). Both approaches increasingly rely on deep learning architectures based on neural networks. The reviewed studies reported computational time gains of several orders of magnitude in specific scenarios while maintaining reasonable accuracy. However, several challenges remain, such as improving models' generalizability and interpretability, enhancing methodology scalability, and reducing the computational cost of developing the models. Efforts are underway to address more complex cases with advanced SciML techniques. Notably, incorporating physics into the learning process and hybridizing CFD solvers with data-driven models merit further investigation. The exploration of these approaches represents a crucial step toward the deployment of reliable models that enable fast design for built environment engineering studies.

Contents

* Corresponding author at: Department of Sustainable Built Environment, PIMENT lab, University of Reunion, 117 rue du Général Ailleret, Le Tampon, 97430, La Réunion, France.
E-mail address: clement.caron@univ-reunion.fr (C. Caron).

https://doi.org/10.1016/j.buildenv.2024.112229
Received 3 June 2024; Received in revised form 6 September 2024; Accepted 23 October 2024
Available online 8 November 2024

Nomenclature

AE	Autoencoder
AE-CNN	Convolutional autoencoder
AI	Artificial intelligence
CFD	Computational fluid dynamics
CNN	Convolutional neural network
DMD	Dynamic mode decomposition
DNS	Direct numerical simulation
FCNN	Fully connected neural network
FNO	Fourier neural operator
GAN	Generative adversarial network
GNN	Graph neural network
GPU	Graphics processing unit
IAQ	Indoor air quality
LES	Large eddy simulation
LSTM	Long short-term memory
ML	Machine learning
NIROM	Non-intrusive reduced-order model
PINN	Physics-informed neural network
POD	Proper orthogonal decomposition
RANS	Reynolds-averaged Navier–Stokes
RBF	Radial basis function
RNN	Recurrent neural network
ROM	Reduced-order model
SciML	Scientific machine learning
SVM	Support vector machine
XGBoost	Extreme gradient boosting

1. Introduction

Computational fluid dynamics (CFD) techniques have been developed for decades to analyze complex fluid-related phenomena based on computer simulations. This numerical approach has great potential, especially for industrial needs, because it helps avoid or limit experimental procedures. Indeed, conducting experiments can be costly, time-consuming, difficult to set up, and challenging to monitor accurately. In addition, CFD methods can produce much more refined results than hand calculations or simplified models. It makes CFD a highly valuable complementary tool [1–3].

Since computing capacities continuously grow, CFD is becoming more affordable and widely used. Scientists tackle increasingly sophisticated systems with greater accuracy, improving our understanding of advanced physical phenomena. Simulations support the design of complex systems and provide key elements for decision-making or performance optimization. Nevertheless, while CFD codes have come a long way, they are still maturing and evolving. There are many aspects in which CFD methods can be improved, including turbulence modeling, multiphysics problems, and numerical methods [4,5]. A major challenge is the computational time needed to address large-scale real-world problems.

Building design is not exempt from the computational cost bottleneck [6,7], as engineering studies often involve large computational domains and turbulent flows. Providing exact computational times to support this statement is difficult due to the various factors involved, such as the specific problem, modeling assumptions, the CFD code used, and the computing resources available. However, it is possible to provide a practical range of time for building engineering studies, from hours to weeks, for a single simulation [8–11].

Artificial intelligence (AI), for its part, has experienced significant growth in the last few years. The plentiful data and increased computing power have revealed the full potential of machine learning (ML) algorithms [12,13]. Besides their ability to recognize patterns in data and accurately approximate complex functions, ML models may offer a major advantage in computation time compared to more conventional algorithms, such as high-fidelity physics-based models [14]. Although the training phase of ML models can be time-consuming, they are very efficient at making new predictions (i.e., the inference time is short).

Investigating how to combine CFD tools with AI to accelerate computation is a natural progression [14–16]. Nevertheless, coupling physics-based numerical models with ML presents many challenges that must be overcome [17]. The research question of how AI can accelerate CFD simulations while maintaining result reliability is still open. One straightforward solution is to substitute physical models with ML models (i.e., direct surrogate modeling), but this strategy may result in non-physical solutions and poor generalization [18]. Research is underway to enhance CFD with AI and democratize data-driven models for real-world fluid dynamics challenges [16,18–20]. The main objectives are to enforce underlying physics and increase model generalizability, robustness, and data efficiency.

Accelerated CFD computation would be a significant stride toward improving building design. To follow this path, we propose a review of recent advances and attempts to accelerate the built environment CFD simulations with ML algorithms. We present an up-to-date synthesis that has not yet been covered in the existing literature to the best of our knowledge. Given the fast-growing nature of ML and the numerous uncoordinated approaches to its application, we consider this work to be necessary. Therefore, the three main objectives of this review are as follows. First, (i) we determine emerging trends and challenges to accelerate built environment CFD simulations using ML techniques. Then, (ii) we identify advancements beyond the building domain to bridge the gap between scientific machine learning (SciML) and built

environment engineering. Finally, (iii) this review has a pedagogical scope, aiming to connect the seemingly distant worlds of the built environment and ML.

In this review, Section 2 provides an overview of CFD usage for built environment engineering. The challenge of CFD computational cost is discussed. Section 3 is dedicated to ML basics, outlining the key concepts to understand SciML trends. Our emphasis on deep learning is deliberate, as it is prevalent in the reviewed articles. Section 4 focuses on current strategies to accelerate CFD with ML algorithms. Thus, the preceding sections provide a foundation for an in-depth examination of current research efforts to accelerate built environment CFD simulations with ML. This specific review is conducted in Section 5 prior to the conclusions.

2. Computational fluid dynamics for building engineering

2.1. Main CFD applications for buildings

Most CFD applications for built environments relate to comfort, health, energy efficiency, or safety [6]. They cover indoor (Fig. 1) and outdoor (Fig. 2) flows, operating at different scales. In indoor applications, the focus is on air movement within a building, considering factors such as ventilation, temperature gradients, and the presence of pollutants. Ensuring the thermal comfort of occupants and maintaining satisfactory indoor air quality (IAQ) are key priorities. In addition, anticipating smoke propagation is critical for fire risk assessment. Outdoor simulations investigate the external airflow behavior in built environments. Such analysis can yield significant insights into various aspects of urban design, including pedestrian comfort, natural ventilation potential, wind load, urban thermal environment, and pollutant dispersion. CFD tools can assist in studying all of these topics [6,21]. In particular, CFD can accurately identify flow patterns, which has the potential to enhance building design.

2.2. Computational time issue of building CFD simulations despite simplifications

2.2.1. Traditional CFD approaches for building engineering

The standard numerical methods for addressing a CFD problem involve two main stages: preprocessing and solving [2]. In particular, the region of interest is meshed during the preprocessing phase (see, e.g., Fig. 1), and then the underlying equations are discretized into a system of algebraic equations that must be solved. Since solving the governing equations in their full complexity is often unnecessary and inefficient, the modeler is responsible for deciding which part of the physics can be ignored because it has little impact on the analysis results. For most building applications, the air can be treated as an isothermal incompressible Newtonian fluid. The equations are adapted, typically based on the Boussinesq approximation [3,22], to cover cases involving heat transfers, e.g., comfort studies. Ultimately, a more comprehensive version of the equations and supplementary intricate models are necessary to address complex flows, such as those encountered in fire safety studies [23].

The most natural approach to solving a CFD case is through *direct numerical simulation (DNS)*. Using fine meshes and small time steps enables the solver to capture all the scales of the flow, offering the possibility of unique fundamental insights. However, for turbulent large-scale real-world problems, this method becomes computationally prohibitive. Therefore, DNS is not part of the engineering toolbox for built environment studies.

Building simulations traditionally rely on the use of *Reynolds-averaged Navier–Stokes (RANS)* and *large eddy simulation (LES)* approaches [7,26], which incorporate models to avoid solving all the turbulent scales. RANS approach decomposes the quantities of interest (typically the velocity, pressure, or temperature) into a mean and a fluctuating component. The resulting equations allow us to solve the mean flow

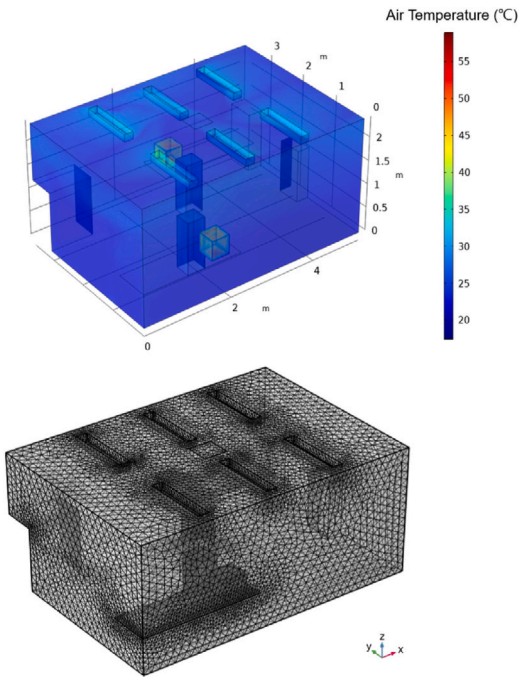


Fig. 1. Example of a CFD indoor application: simulation result (top) and mesh (bottom) for a room.
Source: Extracted from Luo et al. [24].

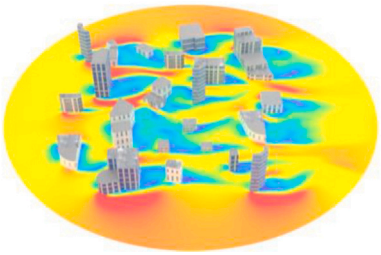


Fig. 2. Example of a CFD outdoor application: urban wind field.
Source: Extracted from Shao et al. [25].

while modeling the influence of turbulent fluctuations. On the other hand, in LES, a spatial filtering operation is applied to the quantities involved in the governing equations. Large eddies of the fluid are explicitly resolved, whereas the dissipative effect of the small scales (under filter size, usually the grid size) is modeled. Thus, LES can provide time-dependent solutions to turbulent flows at a fraction of the DNS computational cost. Several other techniques mix LES and RANS strategies. However, they have yet to be widely used for building engineering.

There is no doubt that the LES strategy is capable of producing results that are more accurate than the RANS approach [7,27,28]. However, this comes at the cost of increased computation time. Since providing CFD results with extreme accuracy is often unnecessary for engineering purposes, RANS remains the most popular approach for building applications [7]. While it is common to assume that the mean airflow is stationary, it may be essential to account for fluctuations in some cases. For example, Yoshie et al. [29] admitted that LES would be desirable for achieving better accuracy in predicting the wind environment around buildings. Ultimately, the choice of the numerical strategy depends on the speed-accuracy trade-off.

2.2.2. Computational time bottleneck

Even with simplified approaches such as RANS or LES, computational time remains a significant bottleneck for CFD engineering

studies. For an order of magnitude, a single LES simulation for built environment applications can take several days, even with a dedicated supercomputer [11]. Assessing numerous configurations is beyond the capabilities of conventional computational resources. Furthermore, Morozova et al. [30] estimated that the growth of computational resources is unlikely to be sufficient to make CFD a viable option for routine use in building applications. Therefore, reducing CFD computational costs is a major future challenge for the CFD community [4,6,30,31]. The advent of fast CFD methods could transform the decision-making process, enabling parametric and optimization studies for more efficient designs. This perspective provides the main motivation for investigating innovative methodologies, such as machine learning, which represents a promising avenue for accelerating built environment CFD simulations [4,30].

2.3. Replacing CFD with simplified models

Various simplified approaches are being developed and improved to produce faster flow simulations [32,33]. For example, Xu et al. [32] mentioned the porous media model [34,35], fast fluid dynamics [36,37], multizone model [38], lattice Boltzmann method [39,40], or coarse-grid CFD [41]. Regardless of the method, model simplifications inevitably reduce accuracy for some configurations. Nevertheless, when appropriately used, simplified models remain powerful approaches to achieving fast calculations and delivering valuable insights, although they cannot entirely replace traditional CFD methods. For example, Seifert et al. [42] demonstrated that the simplified macroscopic method frequently employed for wind-driven ventilation analysis can yield satisfactory results for engineering applications. However, outcomes can be dramatically inaccurate depending on building porosity and wind direction. In another context, Hodges et al. [43] emphasized that fire zone models cannot predict complex configurations and may lack the necessary resolution for hazard analysis.

Simplified models have been the subject of extensive study over the past few years. On the other hand, there has recently been a proliferation of studies on machine learning approaches that may offer a breakthrough in accelerating simulations. New turnkey ML tools are beginning to emerge, particularly for urban wind modeling [44–46]. Their objective is to revolutionize early-stage design with instantaneous predictions. However, the lack of detail provided about the performance and the ML methods used under the hood limits the ability to make a comprehensive evaluation. We believe that a review should help clarify recent advances. Thus, our study focuses on ML techniques, potentially hybridized with traditional approaches, to expedite built environment CFD simulations.

3. Machine learning overview

3.1. Machine learning approaches

Machine learning leverages data to gain insight into the behavior of systems, opening up a range of potential applications across scientific fields [47,48]. The ML framework is powerful in that it requires no prior knowledge of the laws of the underlying system. Furthermore, a model can generate predictions almost instantaneously once the learning phase is complete. Returning to our main topic, this advantage may be exploited to accelerate computations in fluid mechanics.

ML algorithms are traditionally divided into three categories, depending on the information available in the data and the nature of the knowledge to be extracted. (i) In *supervised learning*, a model is trained to predict a specific outcome directly provided by the dataset, i.e., the labels. The learning algorithm aims to minimize a loss function that quantifies the discrepancy between the model predictions and the target values over the training data (Fig. 3). (ii) In contrast, *unsupervised learning* extracts information from unlabeled data about the underlying probability densities using techniques such as clustering

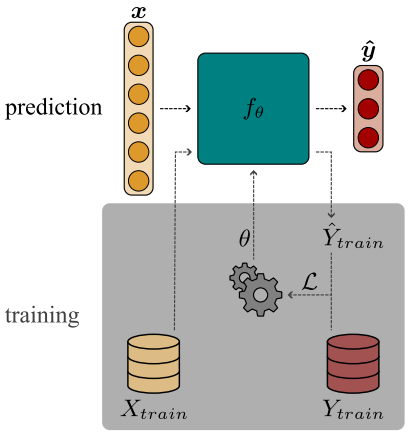


Fig. 3. Schematic of supervised learning. The model maps inputs $\boldsymbol{x}$ to outputs $\hat{\boldsymbol{y}} = f_\theta(\boldsymbol{x}) \approx \boldsymbol{y}$. Training data (X_{train}, Y_{train}) are used to identify the optimal parameters θ for the best possible approximation of f according to the metric $\mathcal{L}$.

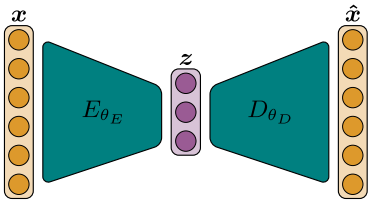


Fig. 4. Schematic of the encoder–decoder procedure describing the dimensionality reduction. The encoder E and the decoder D are trained to compress data while preserving maximum information ($\hat{\boldsymbol{x}} \approx \boldsymbol{x}$).

or dimensionality reduction. In particular, the latter is instrumental in machine learning for fluid mechanics [15]. An encoder–decoder scheme compresses data while retaining essential information (Fig. 4). (iii) As the name suggests, *semisupervised learning* bridges unsupervised and supervised methods, addressing situations where not all data are labeled. Techniques such as generative models or reinforcement learning, found in fluid dynamics applications, usually fall into this category [15]. A common feature is that these algorithms generate data to improve their own performance.

A myriad of ML algorithms are available [47,48], including linear models, decision trees, support vector machines (SVMs), and neural networks, but no single algorithm is universally superior. Although ML techniques are not novel, the advancements have been substantial in recent years. This progress can be mainly attributed to the significant increase in computational power and accessible data.

3.2. Focus on deep learning

Nowadays, *deep learning* is undoubtedly the most popular machine learning approach [13]. In practice, deep learning architectures are based on *deep neural networks*, which are capable of approximating any continuous function [49,50]. The capacity of neural networks to handle large datasets [51,52] and their efficiency in being trained on graphics processing units (GPUs) give them a distinct advantage, contributing to their popularity. Researchers now have the means to train much deeper models than was conceivable a few decades ago [13], pushing back the limits of neural networks. A notable breakthrough is that deep learning can process raw data, avoiding the need for feature engineering to create helpful model inputs [12]. Nevertheless, it is worth remembering that even if deep learning is very efficient in a large panel of tasks, classical data-driven methods should not be discarded. Complex models also come with difficulties such as overfitting, lack of generalization, bad interpretability, and a heavy training process.

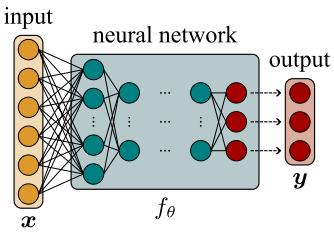


Fig. 5. Schematic of a fully connected neural network (FCNN). Neurons combine input components linearly and apply a final nonlinear activation function to provide a scalar output. They are arranged in layers, which are connected to form an FCNN.

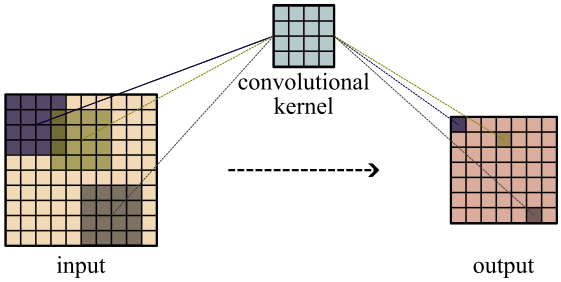


Fig. 6. 2D discrete convolution illustration. 4 × 4 kernel, no padding, stride of 1. The convolutional kernel applies the same sequence of operations to all the patches of the input matrix. Thus, convolutional neural networks (CNNs) can recognize the same local pattern at different locations with a relatively small number of parameters.

3.2.1. Neural networks architectures

A *fully connected neural network* (FCNN) represents the most basic neural network architecture for deep learning (Fig. 5) [47]. However, it is not always the optimal choice, primarily due to the significant increase in the number of parameters with the network size. As a consequence, sparsely connected networks emerged and proved to be very effective [12].

Convolutional neural networks (CNNs) [12,53] rely on convolutional layers that perform discrete convolutions for filtering (Fig. 6). They have become a widely adopted architecture across different domains because they are easier to train and have superior generalization capabilities compared to FCNNs. *Recurrent neural networks* (RNNs) [12,54] successfully handle data sequences such as time series. By design, they retain information about previous elements of a sequence when producing an output (Fig. 7). However, RNNs are challenging to train because the gradient tends to vanish or explode during optimization. Long short-term memory networks (LSTMs) [55], which include an explicit memory to address the difficulty of long-term information retention, have been demonstrated to be more efficient for long time series. The trend has recently shifted from RNNs to *transformers*, based on the attention mechanism [56], to handle data sequences. *Graph neural networks* (GNNs) [57] are designed to apply neural networks to problems formulated in graph form. This framework represents an exciting approach for physics-based problems, as an unstructured mesh can be seen as a graph [58,59]. Usually, a message-passing mechanism is implemented to allow information to flow between graph elements, i.e., nodes and edges. The graph is then iteratively updated based on the information received from their neighboring elements.

This paragraph only provides a narrow perspective of the extensive possibilities attainable within this highly adaptable framework. Deep learning is a fast-growing field, resulting in an incredible variety of ideas being grafted onto this general concept. As an illustration, CNNs have been extended to graphs by introducing *diffusion-convolutional neural networks* [60]. These have subsequently been integrated into an RNN architecture for spatiotemporal forecasting tasks [61,62].

3.2.2. Neural networks for physics

The field of *scientific machine learning* emerged at the intersection of scientific computing and machine learning, intending to leverage data science to improve numerical modeling [18,63,64]. A typical objective of SciML is to construct models consistent with both the data and the governing equations. Integrating physics at various stages of the ML process, including network architecture and loss function, ensures that the model adheres to physical principles with varying degrees of certainty. Two anticipated advantages are the enhancement of model generalization capabilities and a reduction in the required training data.

One of the most well-known techniques in SciML is undoubtedly the promising *physics-informed neural network* (PINN) framework [65], specifically designed to incorporate physics equations during training. Equation residuals and boundary constraints are typically added to the loss function (Fig. 8). While the loss is statistically minimized, there is no guarantee that zero residuals will be achieved. Despite the widespread enthusiasm for the PINNs in the literature, Chuang and Barba [66] argued that the PINN framework is not yet mature enough to handle real-world problems and replace, for example, conventional CFD solvers.

Neural operator architectures [67], e.g., deep operator networks (DeepONets) [68] or Fourier neural operators (FNOs) [69], are gaining increasing attention for solving partial differential equations [18]. Neural networks learn the underlying operator, i.e., mapping between function spaces, rather than a discrete representation. This produces a resolution-invariant solution. Although neural operators are the leading algorithms in many real-time inference applications, they exhibit substantial limitations when applied to large multiphysics problems [18].

Other approaches encode the underlying physics into the neural networks [18,70,71]. In particular, the *neural ordinary differential equations* (NeuralODEs) [71] architecture combines a standard differential equation solver with a neural network. This powerful methodology for modeling ordinary differential equations has been extended in numerous ways [72–74]. These physics-encoded strategies are likely to be more difficult to train but have the potential to accelerate scientific computing significantly [18].

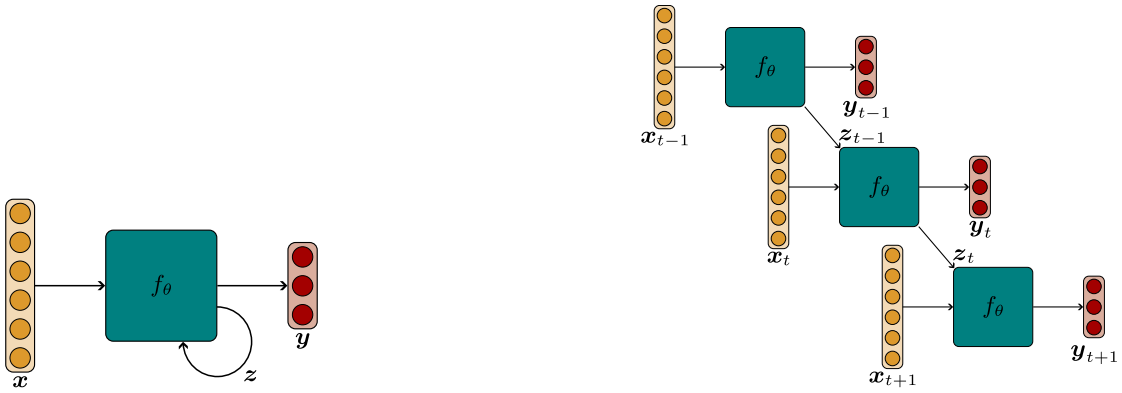
4. Accelerating CFD with ML

In the wake of recent advances in machine learning, the fluid mechanics community is actively investigating ways to leverage this potential to improve CFD [15–17,75]. Vinuesa and Brunton [16] proposed a three-category classification of the applications of ML to enhance CFD solvers (Fig. 9): (i) accelerating direct numerical simulations, (ii) improving turbulence models, and (iii) developing reduced-order models. Each of these three lines of research can accelerate calculations, which is the focus of this paper.

From an industrial perspective, data science is impacting the design of complex engineering systems through the implementation of *digital twins* [76,77]. ML models can enrich these virtual representations of physical entities, providing rapid and reliable predictions based on simulations and experiments. Consequently, novel paradigms are being investigated to generate digital twins capable of delivering up to real-time predictions without additional resource-intensive CFD simulations [78–80].

4.1. Accelerate high-fidelity flow simulations

The simplest approach is to build an ML-based surrogate model to replace an entire CFD simulation. While effective in specific applications, this method tends to generalize poorly [18]. Incorporating physical knowledge to guide or constrain the ML model (see Section 3.2.2) represents a promising strategy for enhancing model performance and ensuring the generation of physically consistent predictions. That is why research efforts are shifting toward data-driven models that enforce underlying physics [19,75].



(a) RNN (b) Unfolded RNN

Fig. 7. Schematic of a recurrent neural network (RNN). 7(a) shows the usual recursive representation. 7(b) illustrates the unfolded graph in time on a forward pass. For each element t of the sequence, the RNN provides an output that depends implicitly on all the previous elements.

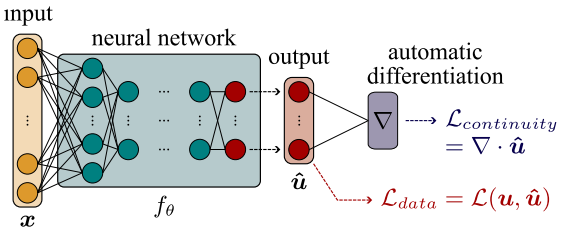


Fig. 8. Physics-informed neural network (PINN) illustration. In this example, the neural network predicts a velocity field $\hat{u}$ from which the classical supervised loss $\mathcal{L}_{data}$ and the physics-based continuity loss $\mathcal{L}_{continuity}$ are computed. Both losses are combined during the training process.

Vinuesa and Brunton [17] correctly pointed out that CFD solvers have been optimized for decades, while ML-CFD coupling is still a new frontier. How to combine ML and CFD efficiently remains unclear, but it seems unrealistic to completely replace CFD with ML shortly. A more practical approach would be to discover how to make them work together. *Hybrid methods* generally result in less drastic improvements in computational time than purely data-driven approaches. However, they demonstrate enhanced robustness and generalizability. For instance, Jeon et al. [81] implemented a cross-coupling strategy to accelerate long-term unsteady simulations. The alternating sequence ML-CFD successfully controlled the error accumulation in successive ML model predictions. List et al. [82] showed that unrolling training trajectories over time strongly influences the prediction accuracy for dynamical systems, i.e., *solver-in-the-loop* approach. The study also emphasized the importance of learning a correction, whereby the model learns to correct a classical numerical solver rather than supplanting its role. Another popular approach is to coarsely mesh the domain, which dramatically influences the computational time, and use ML models to mitigate the loss of accuracy [83–86]. This technique is known as *super-resolution*. Costly CFD solver subcomponents, such as the Poisson solver for incompressible fluids, are also reasonable targets for ML models. For example, the model predictions can improve preconditioning techniques or serve as an initial guess for iterative linear solvers [87–91].

In addition to the aforementioned methods, other initiatives indirectly reduce calculation times by targeting the simulation preprocessing steps. For example, estimating realistic boundary conditions (e.g., inflow) may reduce the size of the computational domain and affect computation time [75,92]. Another promising avenue for acceleration is automatic mesh generation, a particularly time-consuming process requiring expert knowledge [18,93,94].

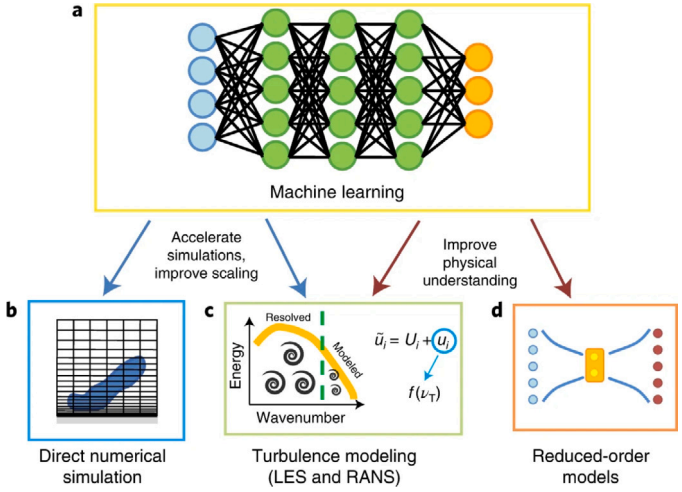


Fig. 9. Summary of some of the most relevant areas where machine learning (ML) can enhance computational fluid dynamics (CFD), according to Vinuesa and Brunton [16]. In the present review, we focus on ML to accelerate simulations. We highlight that these three categories (b, c, and d) can lead to faster CFD calculations.
Source: Figure extracted from Vinuesa and Brunton [16].

4.2. Enhance turbulence modeling with ML

As stated in Section 2.2, engineers mainly rely on RANS and LES approaches. However, turbulence closure models can oversimplify the flow behavior and be difficult to tune for a specific use case. Hence, turbulence modeling is one of the major areas of research in ML-enhanced CFD [15–17,75]. Data-driven algorithms can assist in calibrating turbulence models to a particular application, learn part of a closure term, or even provide entirely new ML-based turbulence models [95,96]. Given the high cost of LES for wall-bounded flows, which are common in engineering, some attempts have been made to develop data-driven wall models for LES [97–99]. More generally, ML is a powerful and promising approach for all kinds of fluid flow underlying models, e.g., turbulent, reacting, or porous media flows.

ML models are not primarily designed to reduce computational time in these cases. Nevertheless, improved accuracy can broaden the scope of applicability for lower-cost simulations, such as RANS or wall-modeled LES. It should be noted that hybrid ML-CFD strategies (presented in Section 4.1) can also speed up approaches such as RANS and LES. For example, Obiols-Sales et al. [100] successfully coupled RANS steady simulations with a CNN to accelerate the convergence.

4.3. Reduced-order models and flow control

Reduced-order models (ROMs) [101] offer a low-dimensional, low-fidelity representation of complex systems. The first step is identifying an appropriate low-dimensional basis, which can be achieved through a range of dimensionality reduction techniques. Then, the objective is to capture the system dynamics expressed in the reduced basis. ROMs have the potential to decrease fluid mechanics computations up to real-time flow approximation. However, this speed-up comes at the cost of a limited scope of application. Indeed, ROMs are usually designed for a specific flow configuration and may not generalize well [75].

Proper orthogonal decomposition (POD) [102] is the most widely used technique in fluid dynamics for finding a low-dimensional orthogonal spatial basis from flow snapshots. The basis components, designated as *modes*, have the dimension of the studied domain. This linear method is physically interpretable, which helps improve our physical understanding of complex flows [103]. POD modes are typically ordered by energy content, and the first few modes show the main flow structures (Fig. 10). One advantage of POD is that it allows for selecting the number of modes included on a reduced basis, which adjusts the trade-off between simulation fidelity and computational speed. The Galerkin projection is a popular choice for projecting the physical governing equations onto the modes. This procedure yields a new system that can be solved faster than the full-order model. However, the process is intrusive, i.e., it requires explicit knowledge of the governing equations.

Dynamic mode decomposition (DMD) [105] is another common approach to building a low-dimensional flow dynamics model. This model-free data-driven technique captures the main flow dynamics into a collection of non-orthogonal modes and constructs a linear model describing the main flow evolution in time. DMD and POD are complementary approaches but share some similarities. They rely on singular value decomposition to build the best linear model fitting the data. However, DMD associates each mode with a temporal frequency, which is particularly relevant for periodic systems. Numerous extensions to these methods (e.g., higher-order DMD [106], SINDy [107]) have been implemented to improve the approximation capabilities of reduced-order models.

ML progress offers new opportunities for improving reduced-order modeling strategies [75]. A contemporary approach is to train a neural network to map high-dimensional flow snapshots to a low-dimensional latent space in a non-intrusive manner, using, e.g., *autoencoders* (AEs) [108]. *Non-intrusive reduced-order models* (NIROMs) can solve the main limitations of classical intrusive approaches. (i) NIROMs avoid the need for cumbersome CFD code manipulations during the projection step. (ii) They may learn a more compact data representation using non-orthogonal and nonlinear encoding. (iii) Linear methods such as POD and DMD may encounter difficulties in handling nonlinearities. Modern ML techniques can also be instrumental in capturing the system dynamics of the flow on a reduced basis. Neural networks with specific architectures, such as LSTMs, are trained to predict the nonlinear temporal evolution of a flow.

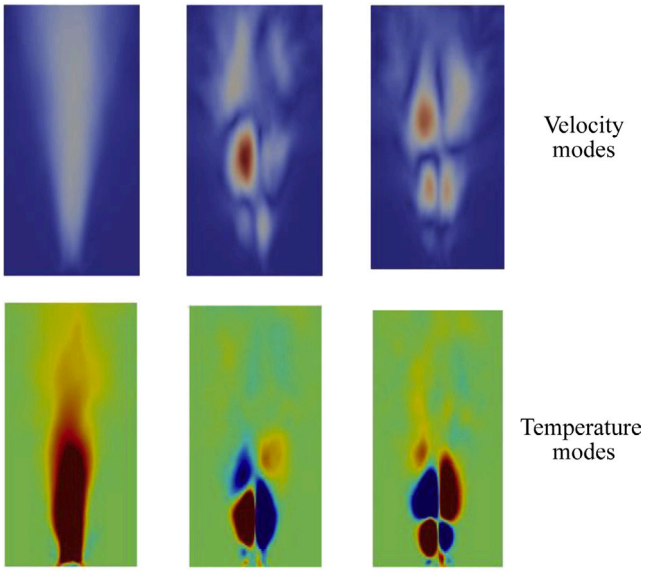


Fig. 10. Lattimer et al. [104] used the proper orthogonal decomposition (POD) to create a reduced-order model (ROM) of a buoyant plume simulation. This figure shows the contribution of the first velocity and temperature modes (1, 2, and 3 from left to right) to the ROM. Each mode adds different levels of coherent structures.
Source: Figure modified from Lattimer et al. [104].

POD, DMD, and other ROMs are ideal candidates for tackling fluid flow control tasks, where model responsiveness is critical. It should also be noted that the *deep reinforcement learning* framework [109], which brings deep neural networks into the reinforcement learning [110] process, is also well-suited to complex control issues such as those related to fluid dynamics [111].

5. AI-accelerated CFD simulations for built environments

In their detailed review, Calzolari and Liu [20] investigated the potential of deep learning for computational fluid dynamics analysis in built environment applications. They observed that most studies employed machine learning to substitute costly CFD simulations and achieve rapid predictions. Hence, they concentrated on deep learning surrogate modeling when reviewing built environment studies. The authors emphasized that some promising methods have yet to be fully explored in this field, such as super-resolution, turbulence model enhancement, or physics-informed deep learning. We share this perspective and believe bridging the gap between the built environment field and advanced ML methods is essential.

Besides, a complementary and updated review is necessary. The current section comprehensively analyzes recent literature to identify emerging trends in surrogate modeling for accelerating built environment studies. We then focus on the developments in reduced-order models. Finally, we describe attempts to integrate ML into CFD solvers with hybrid strategies.

5.1. Direct surrogate modeling for built environment CFD applications

Westermann and Evins [112] pointed out that surrogate modeling is already a prevalent trend in current research on building performance

simulation and optimization. Their literature review concentrated on using ML surrogates to improve the design of sustainable buildings. Although the conclusions drawn are relevant for CFD surrogate models, it is worth noting that almost all the listed surrogate models target building simulation software that does not include CFD codes. The authors identified four stages in the building design process that can benefit from fast surrogate models: (i) conceptual design, (ii) design optimization, (iii) sensitivity analysis, and (iv) uncertainty analysis. The review highlighted promising accelerations for building design optimization tasks, reducing computation time by up to 80% without compromising the optimum quality. However, challenges remain to be addressed, particularly regarding generalizability, interpretability, and the computational cost of creating surrogate models [20,112].

As illustrated in Table 1, many recent studies have applied ML techniques to replace heavy built environment CFD simulations completely. We can observe that all building applications (indoor and outdoor) are part of this trend.

5.1.1. Surrogates for high-level metrics

CFD simulations are often conducted to obtain a high-level metric of interest, either to optimize its value or to verify that it meets specific criteria. These metrics can correspond to global indices, such as thermal comfort level, pollution concentration, or air change efficiency. Training data track the evolution of high-level metrics under various simulation conditions and can then be used to learn the response surface. Therefore, the models typically learn functions that map geometric features and physical quantities describing the problem (i.e., the inputs) to some figures of merit aggregating information produced by a CFD simulation (i.e., the outputs). The most popular models for this task are undoubtedly fully connected neural networks [113–115,118,120] and tree-based algorithms [116,117,119].

Concerning indoor studies, Tian et al. [113] and Li et al. [114] trained FCNNs to predict indoor air quality, thermal comfort, and energy performance in a room. Output values are indicators such as the room energy consumption, CO_2 concentration, air age, and thermal comfort predicted mean vote. In a related development, Ding and Lam [116] proposed to infer a novel index for estimating cross-ventilation potential. Similarly, outdoor CFD simulations can predict wind characteristics [117,119,120] or pollutant concentration [118], which are common targets for surrogate models.

5.1.2. Surrogates for spatial fields

Some studies attempt to predict spatial information rather than a high-level metric to obtain more detailed results, often relevant for engineering purposes.

Kim et al. [121] and Shin et al. [122] employed CNNs to make spatial predictions regarding IAQ from images. Zhou and Ooka [123,124] and Quang et al. [126] demonstrated that FCNNs can estimate the velocity and temperature distributions for non-isothermal cases across the entire domain of interest. Hodges et al. [43], Lattimer et al. [104], and Su et al. [127] developed surrogate models to estimate spatially resolved temperature, velocity, or smoke visibility slices within a building. Transpose convolutional neural networks, well-suited for upsampling tasks, were trained to generate images from high-level fire scenario descriptions in less than a second. They demonstrated promising generalization capabilities with acceptable accuracy for fire safety engineering.

Regarding outdoor applications, Javanroodi et al. [131] used FCNNs to estimate the local wind speeds or air temperatures in arbitrary urban morphologies, achieving satisfactory accuracy. Peng et al. [129] made pioneering efforts to adapt FNOs to a 3D dynamic microclimate simulation at an urban scale. This FNO approach significantly accelerated simulations with an error rate of less than 1% and satisfactory generalizability across different wind directions. Shao et al. [25] proposed a novel physics-informed graph neural network (i.e., combining GNN and PINN approaches) for urban wind field fast predictions (Fig. 11). They claimed that their model met the requirements of low cost, repeatability, reliable precision, scalability, and high performance. Nevertheless, they acknowledged the necessity to extend their work to more complex scenarios. Huang et al. [130] utilized generative adversarial networks (GANs) [135] as surrogate models to accelerate urban design, which they achieved by a factor of more than 100 in their research. They targeted multiple fields of interest, such as pedestrian-level wind, thermal comfort, and accumulated solar radiation. Glumac et al. [11] and Lamberti and Gorlé [128] employed ML models to determine the pressure coefficients on building facades. They utilized a multi-fidelity approach, incorporating low-fidelity RANS and high-fidelity LES data.

Lin et al. [134] used FCNNs to predict pollutant dispersion in an ideal urban environment. With a relative error of NO concentration below 20%, they showed that a surrogate model can provide valuable insights within a few seconds. Similarly, Jurado et al. [133] trained several state-of-the-art CNN-based deep learning architectures to predict airborne pollutant dispersion. They achieved the best results using the multiResUnet architecture [136]. Pedro Souza de Oliveira et al. [132] focused on predicting the concentration field associated with 2D atmospheric dispersion cases. The authors obtained promising results, even when considering the influence of wind.

5.1.3. Observed trends in direct surrogate modeling

Computational time savings. Surrogate models provide almost instantaneous predictions, replacing time-consuming CFD simulations and thereby offering tremendous computational time savings. Authors often claim that models are faster by several orders of magnitude. However, it is crucial to counterbalance this advantage with the model's generalization capabilities and the time required for data generation and model training, including hyper-parameter optimization.

Many simulations must be run to populate the learning database, usually from dozens [113,114,134] to thousands [43,116,133]. A best practice is to balance database size with model performance. Most studies employed RANS solvers to generate data within a reasonable time frame, resulting in models with a limited degree of accuracy.

If the model is tailored to a particular situation, the entire learning process must be repeated for each new study [11,119,124–126,128]. Although the gains in computation time may be less impressive, this retraining strategy can still be effective when many calls to the surrogate model are necessary, such as for optimization purposes. Mortezazadeh et al. [119] demonstrated that a training database with an angle step of 5° for wind direction could be sufficient for training a model predicting the wind power potential over a specific urban area. In their case, they needed 24 h of computation to generate the database. Zhou and Ooka [124] and Quang et al. [126] fairly displayed the computation time comparison between a classical CFD approach and the surrogate model approach, including data generation and model training. Glumac et al. [11] and Wei and Ooka [125] stated that a promising way to reduce the retraining cost is using pre-trained models, i.e., *transfer learning*.

High-level metrics versus spatial fields. Focusing the learning effort on the high-level goal appears to be an effective strategy. The models demonstrated good accuracy, which is highly valuable, especially for the early stages of built environment studies that may not require high precision. In light of these readings, the weak points are the following. (i) *The generalization capabilities.* Most studies focused on a limited use case with strong hypotheses, such as simple geometries. (ii) *Lack of interpretability and investigation possibilities.* Since the model outputs are high-level indicators, much of the valuable information from a traditional CFD simulation is lost. In practice, extracting the full flow characteristics is often crucial, at least for consistency checks. (iii) *No uncertainty estimate.* For engineering purposes, exploiting back-box models without estimating uncertainty is inadvisable, particularly in extrapolation tasks.

With the advancement of ML methods, more studies attempted to predict spatial fields (see Table 1). This approach is appealing as it

Table 1
Recent contributions from the built environment field using machine learning for CFD direct surrogate modeling.

Reference	Building application		Study focus	Machine learning model	Physics incorporated	Predicted target type[a]		Prediction scope[b]	
	Indoor	Out-door				High-level metrics	Spatial fields	Local	Global
Tian et al. [113]	•		IAQ, thermal comfort, and energy performance. Single room. Stratified environment (heating and cooling).	FCNN		•			•
Li et al. [114]	•		IAQ, thermal comfort level, and energy consumption. Single room. Smart environmental control.	FCNN		•			•
Shen et al. [115]	•		Inverse tracing of fire source location. Single room. The forward model predicts the temperature distribution.	FCNN		•	• (forward)		•
Ding and Lam [116]	•	•	Cross-ventilation potential. High-density cities.	Linear regression, gradient boosting		•			•
He et al. [117]		•	Pedestrian-level wind environment. Low-Velocity Areas. Isolated rectangular buildings.	Extra trees regressor		•			•
Wai and Yu [118]		•	Vehicle-derived airborne particulate dispersion within a compact high-rise-built environment.	FCNN		•			•
Mortezazadeh et al. [119]		•	Wind power potential of a cluster of roof-mounted wind turbines.	Random forest		•			•
Higgins and Stathopoulos [120]		•	Wind velocity for wind turbine placement in urban areas.	FCNN		•		•	
Kim et al. [121]	•		Optimize portable air cleaner placement in a room.	CNN			•	•	
Shin et al. [122]	•		IAQ. Mean age of air. Reefer containers.	FCNN, CNN			•		•
Zhou and Ooka [123]	•		2D indoor environment, including both velocity and temperature distributions.	FCNN			•		•
Zhou and Ooka [124]	•		Indoor airflow prediction.	FCNN			•		•
Wei and Ooka [125]	•		Reconstruct an indoor airflow field based on an experimental dataset.	FCNN	PINN		•	•	
Quang et al. [126]	•		Indoor airflow distribution in multi-storey buildings.	FCNN			•		•
Hodges et al. [43] Lattimer et al. [104]	•		Spatially resolved temperatures and velocities. Fire safety.	CNN			•		•
Su et al. [127]	•		Smoke motion and the Available Safe Egress Time in an atrium. Fire safety.	CNN			•		•
Lamberti and Gorlé [128]		•	Pressure coefficients on a highrise building.	FCNN			•	•	
Glumac et al. [11]		•	Wind pressures and wind loads on high-rise buildings.	SVM, FCNN, random forest, gradient boosting			•	•	
Shao et al. [25]		•	Urban wind field.	FCNN	PINN, GNN		•	•	
Peng et al. [129]		•	Urban wind field.	FNO	Neural operator		•		•
Huang et al. [130]		•	Environmental performance-driven urban design.	GAN			•		•
Javanroodi et al. [131]		•	Climate variables in arbitrary urban morphologies, with a focus on extreme climate conditions.	FCNN			•	•	
Pedro Souza de Oliveira et al. [132]		•	Gas dispersion.	FCNN			•	•	
Jurado et al. [133]		•	Airborne pollutant dispersion.	CNN			•		•
Lin et al. [134]		•	Pollutant dispersion in the near-wake of building. Ideal urban environment.	FCNN			•		•

Abbreviations
CNN: Convolutional Neural Network, **FCNN**: Fully Connected Neural Network, **FNO**: Fourier Neural Operator, **GAN**: Generative Adversarial Network, **GNN**: Graph Neural Network, **IAQ**: Indoor Air Quality, **PINN**: Physics Informed Neural Network, **SVM**: Support Vector Machine.

[a] Distinction between models predicting (i) aggregated metrics of interest (i.e., high-level metrics) or (ii) spatial distributions of quantities of interest (i.e., spatial fields).

[b] Distinction between models providing predictions for (i) a specific subpart of the domain of interest (i.e., local) or (ii) the whole domain at once (i.e., global).

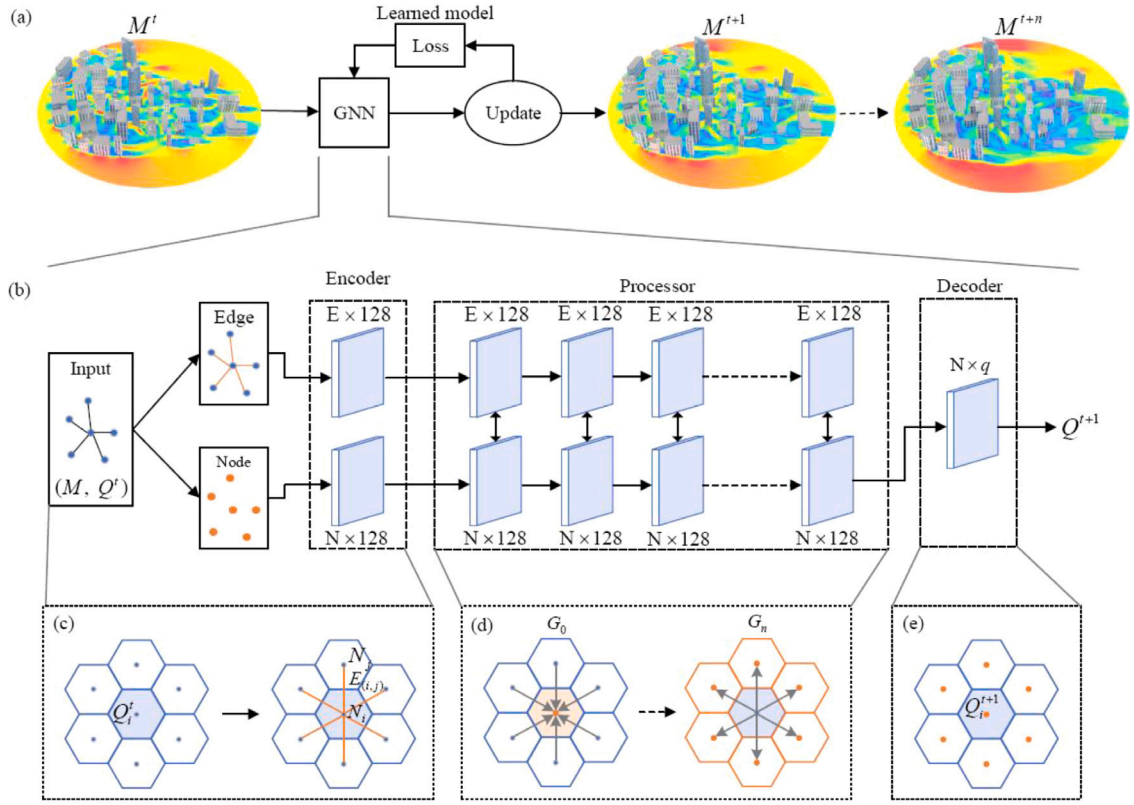


Fig. 11. PIGNN-CFD architecture proposed by Shao et al. [25]. A graph neural network (GNN) iteratively computes the next state M^t (a). An encoder generates a latent graph G_0 from the input graph (c), which is updated through n rounds of learned message-passing steps (d) and then decoded to produce the output Q at the next time step (e). In the GNN architecture, trained neural networks process a subset of the graph (i.e., local approach) but are applied to the entire domain to make global predictions. Therefore, their model scales up to arbitrary large-scale urban scenes.
Source: Figure extracted from Shao et al. [25].

addresses the lack of investigation possibilities. Huang et al. [130] demonstrated the importance of obtaining global information (i.e., spatial fields) to optimize urban blocks (Fig. 12). Indeed, the GAN-based field surrogate outperformed the FCNN-based average metric surrogate for this optimization task.

From our perspective, spatial field surrogate modeling is a crucial approach for expanding the use of ML in CFD engineering studies. Although the results are promising, the models' ability to generalize remains a significant concern. Furthermore, we noticed that most studies focused on 2D cases or predicted 2D slices, suggesting that the approach has difficulty scaling to 3D.

Global versus local approach. As Table 1 depicts, ML-CFD studies can be divided into two main approaches based on their prediction scope: global and local.

On one hand, the *global* approach provides predictions for the entire domain of interest simultaneously. High-level metric surrogate models (see Section 5.1.1) mainly relied on global approaches, given that the outputs correspond to some metrics that summarize the global simulation. The spatial field surrogates (see Section 5.1.2) that provided global predictions either generated outputs from features [43,104,127] or maintained the spatial arrangement between inputs and outputs [122,129,130,133] (see, e.g., Fig. 12a). However, if the dimensions of the model inputs or outputs depend on the domain size, (i) the model is probably tailored to a specific domain size, and (ii) the learning procedure is unlikely to work on a vast domain. Therefore, they may exhibit deficiencies in generalization and scalability, as evidenced by the following examples. (i) It is impossible to generalize to new domains without retraining while using an FCNN to predict a spatial field with a global approach [123,124,126]. (ii) Complex deep learning architectures were involved in handling the problem's dimensionality, e.g., CNN [43,104,122,127,133], FNO [129], and GAN [130]. (iii) Peng et al. [129] faced GPU memory limitations in their work on 3D urban-scale simulations.

On the other hand, the *local* approach provides predictions for a subpart of the domain (Fig. 13). Applying the same model to different subparts of the domain makes it possible to generate a prediction map. Thus, it is particularly well-suited for spatial field predictions. As for the global approach, local predictions relied on handcrafted features [11,128,131] or raw data from the local flow field [25,132] as inputs. This local treatment restricts the problem dimensionality, i.e., input and output sizes are usually reasonable, which brings some major advantages. (i) Model architecture is simpler, (ii) the model predictions scale up to any domain size, and (iii) learning is data-efficient because one global simulation provides many local samples. Intuitively, compared to the global approach, one potential drawback is that inputs may not provide sufficient information to solve the local problem, i.e., the problem is ill-posed. Also, if all local predictions are made independently, it may create a lack of overall coherence in the predicted field. Pedro Souza de Oliveira et al. [132] demonstrated that the local approach can be employed for both unsteady and steady dispersion of pollutants. For a given cell, they gathered information from this cell and its neighborhood as input for an FCNN to predict the next cell state (Fig. 13). Shao et al. [25] used a GNN to predict the next state at the node level (Fig. 11). With this approach, their model scales up to arbitrary large-scale urban scenes. We noticed that some studies included the position as an input, i.e., the global coordinates related to the local prediction to perform [11,25,125,131]. In our view, this choice may not align well with creating a versatile local model as the global position information could lead to overfitting.

Transient simulation predictions. Most of the reviewed studies treated their problem in a classical manner, assuming a steady flow. However,

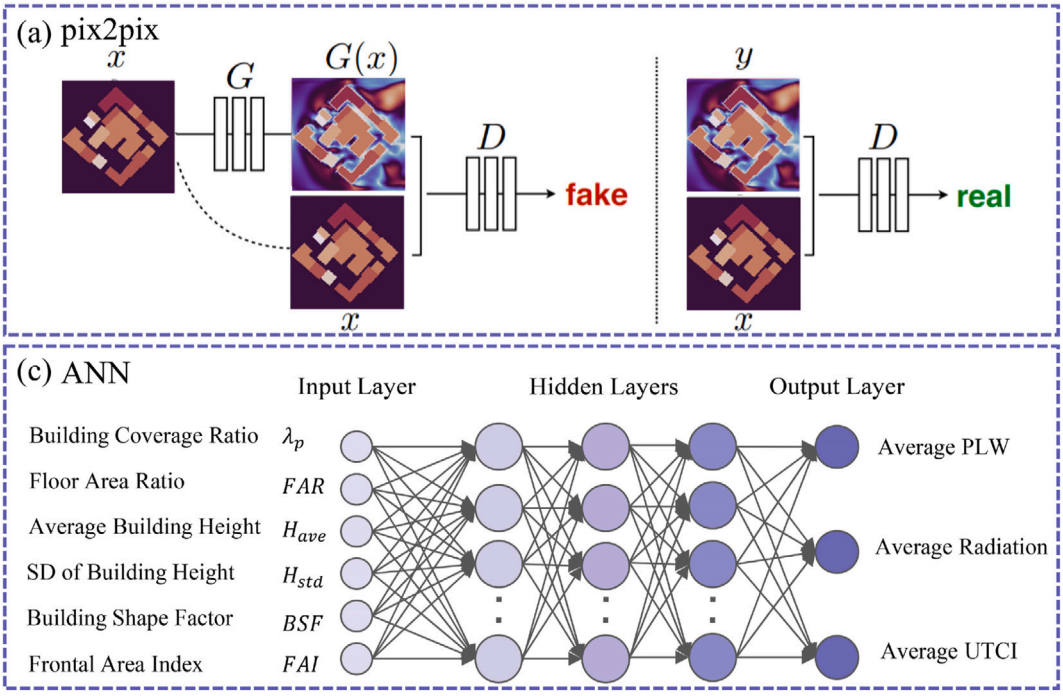


Fig. 12. Two of the surrogate models proposed by Huang et al. [130] in their study are presented here. A comparison was conducted between a generative model predicting the fields of interest (a) and a fully connected neural network predicting high-level metrics (c). In (a), *G* denotes the generator, and *D* is the discriminator. These are deep neural network architectures involving convolutional layers. In (c), the neural network takes six morphological indicators as inputs to predict averaging values of the environmental field. In their article, the results of this comparison underscored the importance of predicting information on the spatial distribution of environmental indicators.
Source: Figure modified from Huang et al. [130].

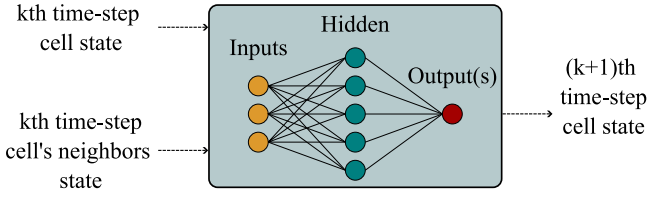


Fig. 13. Local approach described by Pedro Souza de Oliveira et al. [132]. The model uses only the local state as input to make a local prediction. In particular, this approach enables scaling across any domain without additional training.
Source: Figure redrawn from Pedro Souza de Oliveira et al. [132].

it may be important to simulate the temporal evolution. Hodges et al. [43] and Lattimer et al. [104] conducted unsteady fire simulations. Nevertheless, the models predicted temporally averaged outputs and were not intended for transient predictions. Shao et al. [25] pointed out that their physics-informed GNN maintained consistent predictions while taking approximately 20 times larger time steps. This advantage could be crucial in improving computing efficiency for unsteady problems. Peng et al. [129] and Pedro Souza de Oliveira et al. [132] tackled transient problems and exhibited strong predictive capabilities for a single time-step transition. However, the authors revealed the *error accumulation* associated with the ML sequential time step predictions. Hence, data-driven unsteady simulations that run over a long period will likely encounter stability issues.

Embedding physics. Although SciML is rapidly growing (see Section 3.2.2), efforts to incorporate physics into surrogate models for built environment studies are still limited. Wei and Ooka [125] and Shao et al. [25] added terms in the loss function to incorporate the boundary condition constraints or the residuals of the underlying equations, i.e., the PINN approach. Wei and Ooka [125] found out that PINNs yield results that adhere more closely to the laws of physics and are more robust when dealing with limited data. Peng et al. [129] adapted FNO for a realistic urban-scale simulation and demonstrated high accuracy. Several articles discussed the integration of physics into their future work [43,129,132,133], underscoring the importance of investigating this avenue.

5.2. Reduced-order models for built environment CFD applications

Reduced-order models are capable of providing predictions in near real-time for building engineering CFD studies by capturing main flow features (see Section 4.3). Masoumi-Verki et al. [137] reviewed recent advances in ROMs for predicting urban airflow and pollutant dispersion. They reported that the current focus is on NIROMs, which demonstrate superior potential for approximating real-world nonlinear wind flows. The authors also noted that the POD technique may not be the best choice for urban flow due to constantly changing boundary conditions, which can deteriorate POD performance.

Table 2 contains some recent studies that utilized ROMs for building-related applications. Machine learning techniques are deployed for the two primary stages of developing ROMs: reducing dimensionality and computing feature dynamics.

The classical POD approach has been extensively studied for modeling built environments but remains a standard for reducing dimensionality [137]. Lattimer et al. [104] demonstrated, in a simple 2D case (see Fig. 10), that POD can reasonably represent the global buoyant plume flow 2–3 orders of magnitude faster than CFD simulations. By combining POD with radial basis function (RBF) interpolation, Luo et al. [24] successfully obtained accurate and rapid predictions of indoor airflow. However, autoencoders exhibited better results for both indoor and outdoor studies [138,139]. Indeed, Masoumi-Verki et al. [137] confirmed that advanced nonlinear methods based on neural networks are promising for the dimensionality reduction step. Upon further study of AEs, it becomes clear that fully connected AEs do not scale well to the problem dimensionality [137]. Convolutional autoencoders (AE-CNN) are better suited for complex fluid dynamics problems, such as those related to urban areas [139–141]. Masoumi-Verki et al. [141] improved

Table 2
Recent contributions from the built environment field using reduced-order models for CFD.

Reference	Building application and model focus	Dimensionality reduction	Reduced basis coefficient determination
Lattimer et al. [104]	*Indoor*. Velocity and temperature fields. 2D buoyancy-driven plumes from fires.	POD	PDE projection (intrusive)
Luo et al. [24]	*Indoor*. Air velocity and temperature fields. 3D non-uniform environment with multiple boundary conditions.	POD	RBF
Wang et al. [138]	*Indoor*. Air velocity field. 2D room with variable inlet velocity.	POD AE-FCNN	FCNN
Xiang et al. [139]	*Outdoor*. Urban airflow (wind field) with dynamic boundary conditions. 2D-slice at the pedestrian level.	POD AE-FCNN AE-CNN	XGBoost
Xiang et al. [140]	*Outdoor*. 3D urban airflow (wind field) with dynamic boundary conditions.	AE-CNN	XGBoost
Masoumi-Verki et al. [141]	*Outdoor*. Urban airflow. Turbulent airflow field in the wake region of an isolated high-rise building, located in an unstable thermal stratification condition.	AE-CNN Multi-scale AE-CNN Self-attention AE-CNN	Parallel LSTM
Masoumi-Verki et al. [142]	*Outdoor*. Urban airflow. Turbulent airflow field in the wake region of an isolated high-rise building.	WGAN-AAE	Parallel bidirectional LSTM

Abbreviations
AAE: Adversarial Autoencoder, **AE**: Autoencoder, **CNN**: Convolutional Neural Network, **FCNN**: Fully Connected Neural Network, **LSTM**: Long Short-Term Memory, **PDE**: Partial Differential Equation, **POD**: Proper Orthogonal Decomposition, **RBF**: Radial Basis Function, **WGAN**: Wasserstein Generative Adversarial Network, **XGBoost**: Extreme Gradient Boosting.

the AE-CNN approach by considering the multi-scale nature of the flow. The authors explained that convolutional layers only capture local correlations among data and may be limited for non-local correlations. To overcome this limitation, they implemented a multi-scale convolutional autoencoder and a self-attention convolutional autoencoder. The study revealed that the self-attention AE-CNN model outperformed the other models in reconstructing the velocity field of a simplified urban geometry. In a similar scenario, Masoumi-Verki et al. [142] demonstrated that adversarial autoencoders performed better than the self-attention AE-CNN in predicting the turbulent kinetic energy of the flow field. Therefore, when POD is inefficient for addressing complex airflows, advanced neural network approaches such as adversarial autoencoder or AE-CNN may successfully reduce dimensionality with reasonable accuracy.

Neural network techniques are also increasingly involved in computing feature dynamics in reduced space. Luo et al. [24] emphasized the limited interpolation ability of RBF and suggested using better interpolators, such as neural networks or SVMs. Recurrent neural networks (e.g., LSTM) or temporal CNN [143] have been successfully used for outdoor urban airflows [137].

Table 2 highlights that advanced machine learning regression techniques such as extreme gradient boosting [144] or LSTMs are already used for built environment applications.

The computational time gains reported for the proposed ROMs are remarkable in the reviewed papers. These models demonstrated reductions in time ranging from 2 to 6 orders of magnitude compared to CFD simulations. However, as with direct surrogate modeling (cf. Section 5.1.3), these results must be balanced with the model's generalization capabilities and the time required to create the dataset and to train the model. Nevertheless, there is considerable interest in ROM-based control systems that make numerous calls to the model.

While the ROM procedure is powerful, it is essential to recall that ROMs are typically tuned to a specific situation. As Luo et al. [24] stated, the ROM needs to be rebuilt if there are any changes to the geometry or other aspects within the domain. Nevertheless, studies have shown that ROMs can adapt to different boundary conditions [24, 138–140]. Even if a ROM must be rebuilt, transfer learning techniques appear promising to accelerate this process [24,142].

Although neural network-based NIROMs have great potential, challenges still need to be addressed. The main weaknesses include the computational cost of training, the physical interpretation of the model, the embedding of physics, and the stability and robustness of the model [137]. Physical interpretation, in particular, is a major challenge for engineering purposes. Classical ROMs (e.g., POD-Galerkin) have the significant advantage of being physically interpretable (see Section 4.3), unlike the black-box models, which have the potential for improving ROM accuracy. Incorporating more physics into these data-driven models is crucial to improve their performance and interpretability. Efforts are already underway to tackle these challenges [137], but they must continue to enable the use of ROMs in increasingly complex and realistic situations.

5.3. Hybrid approaches for built environment CFD applications

Although machine learning models can drastically reduce computation, they cannot claim to be as reliable and accurate as CFD simulations. This statement is particularly evident when dealing with long-term simulations, as pure ML predictions can rapidly diverge from physical reality over time (see 5.1.3). Furthermore, as previously discussed, model generalization capabilities are often limited. Simulation reliability is of paramount importance in engineering. It may be acceptable to sacrifice accuracy for computation time for early design, but it is still important to quantify the error, which is challenging. Therefore, relying on an ML model for extrapolation tasks can be risky. To address these limitations, Westermann and Evins [112] discussed *gray-box methods*, which do not rely solely on data. This approach may offer a better compromise between black-box ML models and classical physics-based simulations. Research is underway to hybridize the two approaches, bridging the gap between pure-ML and pure-CFD methods. As stated in Section 4.1, this may lead to more modest computational gains but with valuable physical consistency.

Several built environment studies explored hybrid approaches. Quang et al. [126] noted that in their future work, it would be essential to investigate whether ML models can accelerate steady-state simulations when used as a warm-up setting for high-accuracy CFD simulations. This strategy could converge faster with excellent accuracy.

Saboori et al. [145] developed an ML-based method to predict the flow behavior and thermal pattern of a large pore-scale porous

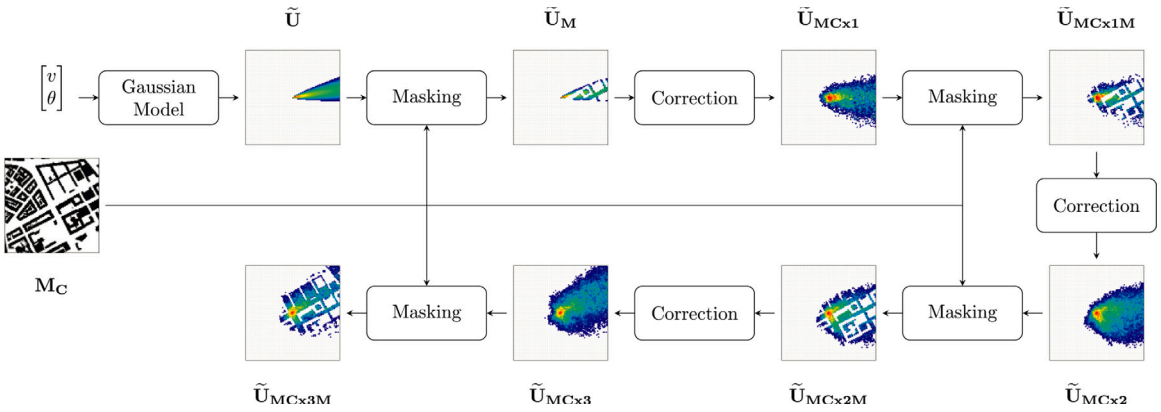


Fig. 14. Example of a hybrid strategy developed by Mendil et al. [146] for outdoor pollutant dispersion. The deep neural network is involved in the correction stages. The prediction is initiated with a Gaussian plume model, and masking stages ensure spatial constraints.
Source: Figure extracted from Mendil et al. [146].

media Trombe wall incorporating phase change materials. They used high-fidelity CFD results at the micro-scale to train recurrent neural networks. These models enabled them to scale up to large-domain predictions. Thus, they performed long-term simulations and optimized the Trombe wall composition, which is beyond the capabilities of standard numerical methods. We posit that a promising short-term avenue is the data-driven approximation of costly subcomponents, which are subsequently integrated into a classical physics-based simulation.

Mendil et al. [146] introduced a novel hybrid strategy for predicting outdoor pollutant dispersion in 2D sections of urban areas (Fig. 14). This approach combined a simple, fast, and interpretable Gaussian plume model with masking and correction blocks. The masking operation enforced spatial constraints, while the correction stage mitigated inconsistencies using a deep neural network trained on synthetic data. The resulting approach was instantaneous, physically interpretable, and generalizable to any simple urban geometry with reasonable accuracy. Although there are ways to improve the methodology (e.g., 3D predictions, rigorous mass conservation, more complex conditions), this article demonstrated that purely data-driven models are not the only means of enhancing CFD with ML.

More generally, using ML models to learn corrections of a physical model is a promising approach (see Section 4.1). In this line, Waibel et al. [147] used inexpensive physics-based fast fluid dynamics calculations as inputs for ML models to predict wind pressure on the facade of high-rise buildings. They hypothesized that such coupled methods could improve simplified model fidelity while keeping simulation times low. We concur with their view that physics-based models will not be superseded shortly and that they will play a pivotal role in integrating ML into building engineering. In a separate approach, Wang et al. [148] proposed a data-driven framework to discover a nonlinear Reynolds stress correction model for steady RANS simulations of urban airflows. This correction could enhance prediction accuracy and extend the applicability of the cost-effective RANS approach to a broader range of engineering scenarios.

5.4. Synthesis, discussion, and future challenges

5.4.1. Emerging trends

The reviewed articles highlight that a significant body of research has explored the potential of ML to accelerate CFD calculations within the context of the built environment. Table 3 provides a summary of the observed trends.

Thus far, the main focus has been to employ ML-based direct surrogate modeling to substitute entire CFD simulations. Surrogate models typically target high-level metrics of interest, but we have also observed numerous attempts to predict spatial fields. Spatial field surrogates can provide more accurate and detailed estimations but often necessitate complex deep learning architectures. While direct surrogate modeling may be the most straightforward to implement, it has been documented that significant challenges are associated with models' generalizability and interpretability. In particular, model accuracy will likely decline when extrapolation tasks are involved. However, it is essential to acknowledge that direct surrogate models can be sufficiently accurate in a constrained and well-defined context, offering significant time savings, potentially by several orders of magnitude.

Reduced-order models have demonstrated the ability to deliver significant computational time gains while maintaining reasonable accuracy for specific flow configurations. Recently, there has been a shift in focus toward nonlinear and non-intrusive ROMs. Advanced ML methods are becoming increasingly prevalent for dimensionality reduction (e.g., convolutional autoencoders) and feature dynamics estimation (e.g., recurrent neural networks or extreme gradient boosting).

Hybrid strategies have emerged to enhance the reliability and accuracy of purely data-driven approaches. ML models have been trained to correct coarse physics-based computations or to substitute high-fidelity simulation subcomponents. These gray-box methods may offer a better compromise between black-box ML models and traditional physics-based simulations.

5.4.2. Challenges and future directions

Alongside the enthusiasm expressed for using ML techniques to accelerate built environment CFD simulations, we have highlighted the main challenges that will need to be addressed in future studies. We recommend that efforts be made to encourage interaction with the SciML community to fill the following gaps.

Reference training data. A direct comparison of the model performances across the studies was deliberately avoided, given the diverse nature of the use cases involved. Additionally, it has been documented that SciML studies often exhibit overly optimistic results [149], which creates confusion regarding the actual computational time savings and the most effective approaches. Therefore, open-source reference datasets related to building engineering must be released. These data will enable the fair comparison of proposed methodologies and foster the development of new initiatives. As supported by McGreivy and Hakim [149], the establishment of domain-specific challenge problems with clearly defined evaluation metrics and baselines would be instrumental in guiding research efforts toward impactful topics. The emergence of high-fidelity reference SciML datasets representing real-world phenomena [150–153] underscores the necessity for the building sector to align with this trend by bringing its own challenges to the forefront.

Scaling and computational cost. Most studies concentrated on 2D predictions and employed learning data generated through RANS simulations. This highlights the challenge in developing effective models quickly, given the time-consuming processes of data curation and

Table 3
Review synthesis. Observed trends in recent built environment literature for accelerating CFD simulations with ML. Key strengths, weaknesses, and recommendations based on our analysis.

Approach	Main ML techniques	Strengths	Common weaknesses	Recommendations (for weaknesses)
Direct surrogates (5.1)		• Instantaneous predictions • Easy implementation	• Generalizability (cannot extrapolate) • Interpretability	• Incorporate physics (improve generalizability) Retrain using transfer learning (avoid extrapolation) • Enforce physics
↳ High-level metrics (5.1.1)	FCNN, tree-based	• Simple ML architectures • One model for multiple geometries • Scalable	• Investigation possibilities	• Spatial field surrogates
↳ Spatial fields (5.1.2)	FCNN, CNN, FNO, GAN	• Detailed fields	• Complex architecture • Tailored to a specific domain • Scalability • Error accumulation (transient flows)	• Local approaches (helpful for the first three weaknesses) • Hybrid approaches
Reduced-order models (5.2)		• Instantaneous predictions	• Generalizability (specific flow configuration)	• Retrain using transfer learning
↳ Intrusive	POD / Galerkin	• Interpretability	• Limited accuracy (linear methods) • Intrusive (cumbersome)	• Nonlinear approaches • NIROMs
↳ Non-intrusive	POD, AE / RBF, FCNN, XGBoost, LSTM	• Non-intrusive • Better accuracy (nonlinear approaches)	• Interpretability	• Enforce physics
Hybrid (5.3) ↳ Subcomponent surrogate ↳ Low-cost model correction	Various	• Physics-based (enhanced accuracy, robustness, generalizability, interpretability)	• Complex implementation • Limited acceleration	

Abbreviations
AE: Autoencoder, **CNN**: Convolutional Neural Network, **FCNN**: Fully Connected Neural Network, **FNO**: Fourier Neural Operator, **GAN**: Generative Adversarial Network, **LSTM**: Long Short-Term Memory, **ML**: Machine Learning, **NIROM**: Non-intrusive reduced-order model, **POD**: Proper Orthogonal Decomposition, **RBF**: Radial Basis Function, **XGBoost**: Extreme Gradient Boosting.

model training. While reviewed studies have provided valuable proof of concept, future research must demonstrate the ability to scale up to real 3D problems and to learn from higher-fidelity data.

Transfer learning [154] has been proposed as a potential solution to reduce the cost of the training phase, but further investigation is required. The fundamental concept is to leverage pre-trained models to address a novel problem that is related to the original one, e.g., the same task with different domains. The new model may be trained with a reduced dataset [155], a shortened training period, and enhanced generalization capabilities. Therefore, transfer learning could enable the democratization of rapidly trained models dedicated to specific configurations, which are likely to be easier to train and more accurate than complex general-purpose models.

Local approaches (described in Section 5.1.3), which concentrate on subparts of the domain, offer an intriguing perspective for addressing the issues of scalability, model complexity, and data efficiency. While some studies have focused on cell-level predictions [81,132], other authors within the field of SciML have suggested dividing the domain into blocks [90,91], which may be a better compromise for tackling large-scale problems. Some sparsely connected network architectures can also be regarded as local approaches. For example, with fully convolutional networks [87,156] or GNNs [25], the domain size can be modified at inference time. However, they can be particularly memory-intensive for training [25] and inference [87]. Given the prevalence of large-scale domains in building studies, our recommendation is that local approaches be subjected to more rigorous examination, as they could facilitate the development of models that can be applied to industrial cases.

Physics incorporation and hybridization. Improving the data-efficiency, generalizability, and interpretability of ML models is necessary for engineering purposes. Additionally, a solution should be found to control ML model error accumulation for transient simulations. From our perspective, incorporating more physics into the training process and pushing for greater hybridization with numerical solvers are crucial steps in this direction. Promising approaches such as PINNs, GNNs, neural operators, or hybrid strategies have begun to be incorporated into recent built environment studies. However, they are still under-represented within the existing literature. These efforts need to be continued, combined, and expanded to produce more reliable models. Recent SciML advances (see, e.g., Section 4) must be disseminated to the built environment field.

6. Conclusions

The primary objective of this review is to examine the potential of machine learning algorithms to speed up computational fluid dynamics calculations for built environments. The current and prospective trends and challenges have been discussed.

Our review highlights that ML methodologies are widely investigated for accelerating built environment CFD. Currently, the most prevalent approaches are deep learning for direct surrogate modeling and reduced-order models, showcasing considerable reductions in computational time. Nonetheless, critical shortcomings related to model generalizability, scalability, interpretability, and pre-deployment computational cost must be addressed before widespread adoption. To overcome these challenges, we advocate for embracing emerging SciML trends. In particular, integrating physical principles into the training process and implementing robust hybrid ML-CFD solvers represent significant avenues for advancement. It is encouraging to note that several reviewed studies have progressed in this direction, yet greater effort is still required. In addition, transfer learning and local approaches have been identified as potentially instrumental to the acceleration of real-world CFD cases, and further investigation is recommended. Finally, we argue that it is of the utmost importance for the built environment community to propose challenge problems and provide reference open-source CFD data.

While this review has focused on ML techniques, claiming they constitute the sole or optimal method for reducing computational time would be fallacious. Nevertheless, we firmly believe that they will contribute to this goal. Current literature places significant emphasis on deep learning, but it is essential to acknowledge the efficacy of classical ML methods and recognize that replacing them with deep learning is not always appropriate.

In light of the present review, the integration of ML into built environment CFD is advancing toward providing faster, more accurate, reliable, and scalable models. This progress will benefit building design at preliminary and detailed stages, enabling the development of novel control and optimization strategies and paving the way for holistic building designs.

CRediT authorship contribution statement

Clément Caron: Writing – original draft, Writing – review & editing. **Philippe Lauret:** Writing – review & editing, Supervision. **Alain Bastide:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

Clément Caron reports financial support was provided by the French National Association of Technical Research (ANRT). Clément Caron reports financial support was provided by INTEGRALE INGENIERIE. Clément Caron reports a relationship with INTEGRALE INGENIERIE that includes: employment. Other authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was conducted as part of a PhD program supported by INTEGRALE INGENIERIE and the French National Association of Technical Research (ANRT) under Cifre agreement No. 2021/1543.

Data availability

No data was used for the research described in the article.

References

[1] J.D. Anderson, J. Wendt, Computational Fluid Dynamics, vol. 206, Springer, 1995.
[2] H.K. Versteeg, W. Malalasekera, An Introduction To Computational Fluid Dynamics: The Finite Volume Method, second ed. [nachdr.] ed., Pearson/Prentice Hall, Harlow, 2007.
[3] J.H. Ferziger, M. Perić, R.L. Street, Computational Methods for Fluid Dynamics, Springer International Publishing, Cham, 2020, http://dx.doi.org/10.1007/978-3-319-99693-6.
[4] A.K. Runchal, M.M. Rao, CFD of the future: Year 2025 and beyond, in: 50 Years of CFD in Engineering Sciences: A Commemorative Volume in Memory of D. Brian Spalding, 2020, pp. 779–795.
[5] M.M. Bhatti, M. Marin, A. Zeeshan, S.I. Abdelsalam, Editorial: Recent trends in computational fluid dynamics, Front. Phys. 8 (2020).
[6] Z. Zhai, Application of computational fluid dynamics in building design: Aspects and trends, Indoor Built Environ. 15 (2006) 305–313, http://dx.doi.org/10.1177/1420326X06067336.
[7] B. Blocken, LES over RANS in building simulation for outdoor and indoor applications: A foregone conclusion? Build. Simul. 11 (2018) 821–870, http://dx.doi.org/10.1007/s12273-018-0459-3.
[8] P.Z. Gao, S.L. Liu, W.K. Chow, N.K. Fong, Large eddy simulations for studying tunnel smoke ventilation, Tunn. Undergr. Space Technol. 19 (2004) 577–586, http://dx.doi.org/10.1016/j.tust.2004.01.005.
[9] S.R. Hanna, M.J. Brown, F.E. Camelli, S.T. Chan, W.J. Coirier, O.R. Hansen, A.H. Huber, S. Kim, R.M. Reynolds, Detailed simulations of atmospheric flow and dispersion in Downtown Manhattan: An application of five computational fluid dynamics models, Bull. Am. Meteorol. Soc. 87 (2006) 1713–1726, http://dx.doi.org/10.1175/BAMS-87-12-1713.
[10] J. Liu, J. Niu, CFD simulation of the wind environment around an isolated high-rise building: an evaluation of SRANS, LES and DES models, Build. Environ. 96 (2016) 91–106, http://dx.doi.org/10.1016/j.buildenv.2015.11.007.
[11] A.Š. Glumac, O. Jadhav, V. Despotović, B. Blocken, S.P. Bordas, A multi-fidelity wind surface pressure assessment via machine learning: A high-rise building case, Build. Environ. 234 (2023) 110135, http://dx.doi.org/10.1016/j.buildenv.2023.110135.
[12] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444, http://dx.doi.org/10.1038/nature14539.
[13] Y. Bengio, I. Goodfellow, A. Courville, Deep Learning, vol. 1, MIT press Cambridge, MA, USA, 2017.
[14] M. Frank, D. Drikakis, V. Charissis, Machine-learning methods for computational science and engineering, Computation 8 (2020) 15.
[15] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, Annu. Rev. Fluid Mech. 52 (2020) 477–508, http://dx.doi.org/10.1146/annurev-fluid-010719-060214.
[16] R. Vinuesa, S.L. Brunton, Enhancing computational fluid dynamics with machine learning, Nat. Comput. Sci. 2 (2022) 358–366, http://dx.doi.org/10.1038/s43588-022-00264-7.
[17] R. Vinuesa, S. Brunton, Emerging trends in machine learning for computational fluid dynamics, 2023, http://dx.doi.org/10.48550/arXiv.2211.15145, arXiv:2211.15145.
[18] S.A. Faroughi, N.M. Pawar, C. Fernandes, M. Raissi, S. Das, N.K. Kalantari, S. Kourosh Mahjour, Physics-guided, physics-informed, and physics-encoded neural networks and operators in scientific computing: Fluid and solid mechanics, J. Comput. Inf. Sci. Eng. 24 (2024) 040802, http://dx.doi.org/10.1115/1.4064449.
[19] N. Zehtabiyan-Rezaie, A. Iosifidis, M. Abkar, Data-driven fluid mechanics of wind farms: A review, 2022, arXiv:2203.13289 [physics].
[20] G. Calzolari, W. Liu, Deep learning to replace, improve, or aid CFD analysis in built environment applications: A review, Build. Environ. 206 (2021) 108315, http://dx.doi.org/10.1016/j.buildenv.2021.108315.
[21] N. Markatos, CFD in environmental engineering and risk analysis, Int. J. Chem. Chem. Eng. Syst. 8 (2023) 1–15.
[22] J. Boussinesq, Théorie Analytique de La Chaleur: Mise En Harmonie Avec La Thermodynamique et Avec La Théorie Mécanique de La Lumière, vol. 2, Gauthier-Villars, 1903.
[23] G.H. Yeoh, K.K. Yuen, Computational Fluid Dynamics in Fire Engineering: Theory, Modelling and Practice, Elsevier, Amsterdam; Boston, 2009.
[24] Y. Luo, D. Cui, Y. Song, Z. Tian, J. Fan, L. Zhang, Fast and accurate prediction of air temperature and velocity field in non-uniform indoor environment under complex boundaries, Build. Environ. 230 (2023) 109987, http://dx.doi.org/10.1016/j.buildenv.2023.109987.
[25] X. Shao, Z. Liu, S. Zhang, Z. Zhao, C. Hu, PIGNN-CFD: A physics-informed graph neural network for rapid predicting urban wind field defined on unstructured mesh, Build. Environ. 232 (2023) 110056, http://dx.doi.org/10.1016/j.buildenv.2023.110056.
[26] P. Sagaut, Large eddy simulation for incompressible flows: An introduction, in: Scientific Computation, third ed., Springer, Berlin ; New York, 2006.
[27] W. Rodi, Comparison of LES and RANS calculations of the flow around bluff bodies, J. Wind Eng. Ind. Aerodyn. 69–71 (1997) 55–75, http://dx.doi.org/10.1016/S0167-6105(97)00147-5.
[28] X. Zheng, H. Montazeri, B. Blocken, CFD simulations of wind flow and mean surface pressure for buildings with balconies: Comparison of RANS and LES, Build. Environ. 173 (2020) 106747, http://dx.doi.org/10.1016/j.buildenv.2020.106747.
[29] R. Yoshie, A. Mochida, Y. Tominaga, H. Kataoka, K. Harimoto, T. Nozu, T. Shirasawa, Cooperative project for CFD prediction of pedestrian wind environment in the Architectural Institute of Japan, J. Wind Eng. Ind. Aerodyn. 95 (2007) 1551–1578, http://dx.doi.org/10.1016/j.jweia.2007.02.023.
[30] N. Morozova, F.X. Trias, R. Capdevila, C.D. Pérez-Segarra, A. Oliva, On the feasibility of affordable high-fidelity CFD simulations for indoor environment design and control, Build. Environ. 184 (2020) 107144, http://dx.doi.org/10.1016/j.buildenv.2020.107144.
[31] A. Runchal, The Future of CFD and the CFD of the Future, vol. 4, 2012, http://dx.doi.org/10.1615/ICHMT.2012.CHT-12.10.
[32] X. Xu, Z. Gao, M. Zhang, A review of simplified numerical approaches for fast urban airflow simulation, Build. Environ. 234 (2023) 110200, http://dx.doi.org/10.1016/j.buildenv.2023.110200.
[33] W. Zuo, Q. Chen, Simulations of air distributions in buildings by FFD on GPU, HVAC & R Res. 16 (2010) 785–798, http://dx.doi.org/10.1080/10789669.2010.10390934.
[34] J. Hang, Y. Li, Wind conditions in idealized building clusters: Macroscopic simulations using a porous turbulence model, Bound.-Layer Meteorol. 136 (2010) 129–159.
[35] D. Getachew, W.J. Minkowycz, J.L. Lage, A modified form of the κ–ε model for turbulent flows of an incompressible fluid in porous media, Int. J. Heat Mass Transfer 43 (2000) 2909–2915.
[36] M. Jin, W. Zuo, Q. Chen, Simulating natural ventilation in and around buildings by fast fluid dynamics, Numer. Heat Transfer A 64 (2013) 273–289.
[37] W. Zuo, Q. Chen, Fast and informative flow simulations in a building by using fast fluid dynamics model on graphics processing unit, Build. Environ. 45 (2010) 747–757.
[38] R. Yao, Q. Luo, B. Li, A simplified mathematical model for urban microclimate simulation, Build. Environ. 46 (2011) 253–265.
[39] C. Obrecht, F. Kuznik, L. Merlier, J.-J. Roux, B. Tourancheau, Towards aeraulic simulations at urban scale using the lattice Boltzmann method, Environ. Fluid Mech. 15 (2015) 753–770.

[40] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Annu. Rev. Fluid Mech. 30 (1998) 329–364.

[41] H. Wang, Z. Zhai, Application of coarse-grid computational fluid dynamics on indoor environment modeling: Optimizing the trade-off between grid resolution and simulation accuracy, HVAC & R Res. 18 (2012) 915–933.

[42] J. Seifert, Y. Li, J. Axley, M. Rösler, Calculation of wind-driven cross ventilation in buildings with large openings, J. Wind Eng. Ind. Aerodyn. 94 (2006) 925–947, http://dx.doi.org/10.1016/j.jweia.2006.04.002.

[43] J.L. Hodges, B.Y. Lattimer, K.D. Luxbacher, Compartment fire predictions using transpose convolutional neural networks, Fire Saf. J. 108 (2019) 102854, http://dx.doi.org/10.1016/j.firesaf.2019.102854.

[44] A. Chronis, A. Aichinger, S. Düring, T. Galanos, T. Fink, O. Vesely, R. Koenig, INFRARED an intelligent framework for resilient design, 2020.

[45] L. Kabošová, A. Chronis, T. Galanos, D. Katunskỳ, Leveraging urban configurations for achieving wind comfort in cities, SIGgraDi (2021) 79–90.

[46] M. Gür, I. Karadag, Machine learning for pedestrian-level wind comfort analysis, Buildings 14 (2024) http://dx.doi.org/10.3390/buildings14061845.

[47] T. Hastie, J.H. Friedman, R. Tibshirani, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second ed., Springer, 2009.

[48] I.H. Witten, E. Frank, M.A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, third ed., in: Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, Burlington, MA, 2011.

[49] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (1989) 359–366, http://dx.doi.org/10.1016/0893-6080(89)90020-8.

[50] M. Leshno, V.Y. Lin, A. Pinkus, S. Schocken, Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, Neural Netw. 6 (1993) 861–867, http://dx.doi.org/10.1016/S0893-6080(05)80131-5.

[51] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Stat. 22 (1951) 400–407, arXiv:2236626.

[52] L. Bottou, F.E. Curtis, J. Nocedal, Optimization methods for large-scale machine learning, SIAM Rev. 60 (2018) 223–311, http://dx.doi.org/10.1137/16M1080173.

[53] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel, Handwritten digit recognition with a back-propagation network, Adv. Neural Inf. Process. Syst. 2 (1989).

[54] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, Ed. De Rumelhart and J. Mcclelland. 1. 1986, Biometrika 71 (1986) 6.

[55] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.

[56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[57] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (2021) 4–24, http://dx.doi.org/10.1109/TNNLS.2020.2978386.

[58] F. Alet, A.K. Jeewajee, M.B. Villalonga, A. Rodriguez, T. Lozano-Perez, L. Kaelbling, Graph Element Networks: Adaptive, structured computation and memory, in: Proceedings of the 36th International Conference on Machine Learning, PMLR, 2019, pp. 212–222.

[59] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, P. Battaglia, Learning to simulate complex physics with graph networks, in: Proceedings of the 37th International Conference on Machine Learning, PMLR, 2020, pp. 8459–8468.

[60] J. Atwood, D. Towsley, Diffusion-convolutional neural networks, Adv. Neural Inf. Process. Syst. 29 (2016).

[61] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: International Conference on Learning Representations, ICLR '18, 2018.

[62] T. Mallick, P. Balaprakash, E. Rask, J. Macfarlane, Graph-partitioning-based diffusion convolutional recurrent neural network for large-scale traffic forecasting, Transp. Res. Rec. 2674 (2020) 473–488, http://dx.doi.org/10.1177/0361198120930010.

[63] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (2021) 422–440, http://dx.doi.org/10.1038/s42254-021-00314-5.

[64] S. Cuomo, V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics–informed neural networks: where we are and what's next, J. Sci. Comput. 92 (2022) 88, http://dx.doi.org/10.1007/s10915-022-01939-z.

[65] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707, http://dx.doi.org/10.1016/j.jcp.2018.10.045.

[66] P.-Y. Chuang, L.A. Barba, Experience report of physics-informed neural networks in fluid simulations: Pitfalls and frustration, 2022, arXiv:2205.14249.

[67] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Graph kernel network for partial differential equations, 2020, https://arxiv.org/abs/2003.03485v1.

[68] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. 3 (2021) 218–229.

[69] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, 2020, arXiv preprint arXiv:2010.08895.

[70] C. Rao, H. Sun, Y. Liu, Hard encoding of physics for learning spatiotemporal dynamics, 2021, arXiv:2105.00557.

[71] R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, Neural ordinary differential equations, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, Curran Associates Inc., Red Hook, NY, USA, 2018, pp. 6572–6583.

[72] O. Owoyele, P. Pal, ChemNODE: A neural ordinary differential equations framework for efficient chemical kinetic solvers, Energy AI 7 (2022) 100118, http://dx.doi.org/10.1016/j.egyai.2021.100118.

[73] A.J. Linot, J.W. Burby, Q. Tang, P. Balaprakash, M.D. Graham, R. Maulik, Stabilized neural ordinary differential equations for long-time forecasting of dynamical systems, J. Comput. Phys. 474 (2023) 111838, http://dx.doi.org/10.1016/j.jcp.2022.111838.

[74] T. Kumar, A. Kumar, P. Pal, A posteriori evaluation of a physics-constrained neural ordinary differential equations approach coupled with CFD solver for modeling stiff chemical kinetics, 2024, arXiv:2312.00038.

[75] R. Vinuesa, S.L. Brunton, The potential of machine learning to enhance computational fluid dynamics, 2021, arXiv:2110.02085 [physics].

[76] E. Glaessgen, D. Stargel, The digital twin paradigm for future NASA and US Air Force vehicles, in: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, 2012, p. 1818.

[77] Q. Qi, F. Tao, Y. Zuo, D. Zhao, Digital twin service towards smart manufacturing, Procedia CIRP 72 (2018) 237–242.

[78] B. Buisson, D. Lakehal, Towards an integrated machine-learning framework for model evaluation and uncertainty quantification, Nucl. Eng. Des. 354 (2019) 110197, http://dx.doi.org/10.1016/j.nucengdes.2019.110197.

[79] R. Molinaro, J.-S. Singh, S. Catsoulis, C. Narayanan, D. Lakehal, Embedding data analytics and CFD into the digital twin concept, Comput. & Fluids 214 (2021) 104759, http://dx.doi.org/10.1016/j.compfluid.2020.104759.

[80] S. Catsoulis, J.-S. Singh, C. Narayanan, D. Lakehal, Integrating supervised learning and applied computational multi-fluid dynamics, Int. J. Multiph. Flow 157 (2022) 104221, http://dx.doi.org/10.1016/j.ijmultiphaseflow.2022.104221.

[81] J. Jeon, J. Lee, R. Vinuesa, S.J. Kim, Residual-based physics-informed transfer learning: A hybrid method for accelerating long-term CFD simulations via deep learning, Int. J. Heat Mass Transfer 220 (2024) 124900, http://dx.doi.org/10.1016/j.ijheatmasstransfer.2023.124900.

[82] B. List, L.-W. Chen, K. Bali, N. Thuerey, How temporal unrolling supports neural physics simulators, 2024, http://dx.doi.org/10.48550/arXiv.2402.12971, arXiv:2402.12971.

[83] K. Fukami, K. Fukagata, K. Taira, Super-resolution analysis via machine learning: A survey for fluid flows, Theor. Comput. Fluid Dyn. 37 (2023) 421–444, http://dx.doi.org/10.1007/s00162-023-00663-0.

[84] D. Kochkov, J.A. Smith, A. Alieva, Q. Wang, M.P. Brenner, S. Hoyer, Machine learning–accelerated computational fluid dynamics, Proc. Natl. Acad. Sci. 118 (2021) e2101784118, http://dx.doi.org/10.1073/pnas.2101784118.

[85] J. Pathak, M. Mustafa, K. Kashinath, E. Motheau, T. Kurth, M. Day, Using machine learning to augment coarse-grid computational fluid dynamics simulations, 2020, http://dx.doi.org/10.48550/arXiv.2010.00072, arXiv:2010.00072.

[86] B.N. Hanna, N.T. Dinh, R.W. Youngblood, I.A. Bolotnov, Machine-learning based error prediction approach for coarse-grid Computational Fluid Dynamics (CG-CFD), Prog. Nucl. Energy 118 (2020) 103140, http://dx.doi.org/10.1016/j.pnucene.2019.103140.

[87] E.A. Illarramendi, M. Bauerheim, B. Cuenot, Performance and accuracy assessments of an incompressible fluid solver coupled with a deep convolutional neural network, Data-Cent. Eng. 3 (2022) e2, http://dx.doi.org/10.1017/dce.2022.2.

[88] R. Chen, X. Jin, H. Li, A machine learning based solver for pressure Poisson equations, Theor. Appl. Mech. Lett. (2022) 100362, http://dx.doi.org/10.1016/j.taml.2022.100362.

[89] E. Zhang, A. Kahana, E. Turkel, R. Ranade, J. Pathak, G.E. Karniadakis, A hybrid iterative numerical transferable solver (HINTS) for PDEs based on deep operator network and relaxation methods, 2022, http://dx.doi.org/10.48550/arXiv.2208.13273, arXiv:2208.13273.

[90] P. Sousa, C.V. Rodrigues, A. Afonso, Enhancing CFD solver with Machine Learning techniques, Comput. Methods Appl. Mech. Engrg. 429 (2024) 117133, http://dx.doi.org/10.1016/j.cma.2024.117133.

[91] P. Sousa, A. Afonso, C. Veiga Rodrigues, Application of machine learning to model the pressure Poisson equation for fluid flow on generic geometries, Neural Comput. Appl. (2024) http://dx.doi.org/10.1007/s00521-024-09935-0.

[92] K. Fukami, Y. Nabae, K. Kawai, K. Fukagata, Synthetic turbulent inflow generator using machine learning, Phys. Rev. Fluids 4 (2019) 064603.

[93] Z. Zhang, Y. Wang, P.K. Jimack, H. Wang, MeshingNet: A new mesh generation method based on deep learning, in: International Conference on Computational Science, Springer, 2020, pp. 186–198.

[94] W. Tingfan, L. Xuejun, A. Wei, Z. Huang, L. Hongqiang, A mesh optimization method using machine learning technique and variational mesh adaptation, Chin. J. Aeronaut. 35 (2022) 27–41.

[95] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. 51 (2019) 357–377, http://dx.doi.org/10.1146/annurev-fluid-010518-040547.

[96] K. Duraisamy, Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence, Phys. Rev. Fluids 6 (2021) 050504, http://dx.doi.org/10.1103/PhysRevFluids.6.050504.

[97] X.I.A. Yang, S. Zafar, J.-X. Wang, H. Xiao, Predictive large-eddy-simulation wall modeling via physics-informed neural networks, Phys. Rev. Fluids 4 (2019) 034602, http://dx.doi.org/10.1103/PhysRevFluids.4.034602.

[98] A. Lozano-Durán, H.J. Bae, Machine learning building-block-flow wall model for large-eddy simulation, J. Fluid Mech. 963 (2023) A35, http://dx.doi.org/10.1017/jfm.2023.331.

[99] T. Kumar, P. Pal, S. Wu, A.C. Nunno, O. Owoyele, M.M. Joly, D. Tretiak, Assessment of machine learning wall modeling approaches for large eddy simulation of gas turbine film cooling flows: An a priori study, J. Eng. Gas Turbines Power 146 (2024) 081019, http://dx.doi.org/10.1115/1.4064556.

[100] O. Obiols-Sales, A. Vishnu, N. Malaya, A. Chandramowliswharan, CFDNet: A deep learning-based accelerator for fluid simulations, in: Proceedings of the 34th ACM International Conference on Supercomputing, ACM, Barcelona Spain, 2020, pp. 1–12, http://dx.doi.org/10.1145/3392717.3392772.

[101] B.R. Noack, M. Morzynski, G. Tadmor, Reduced-Order Modelling for Flow Control, vol. 528, Springer Science & Business Media, 2011.

[102] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. 25 (1993) 539–575.

[103] K. Taira, S.L. Brunton, S.T.M. Dawson, C.W. Rowley, T. Colonius, B.J. McKeon, O.T. Schmidt, S. Gordeyev, V. Theofilis, L.S. Ukeiley, Modal analysis of fluid flows: An overview, AIAA J. 55 (2017) 4013–4041, http://dx.doi.org/10.2514/1.J056060.

[104] B. Lattimer, J. Hodges, A. Lattimer, Using machine learning in physics-based simulation of fire, Fire Saf. J. 114 (2020) 102991, http://dx.doi.org/10.1016/j.firesaf.2020.102991.

[105] P.J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. 656 (2010) 5–28, http://dx.doi.org/10.1017/S0022112010001217.

[106] S. Le Clainche, J.M. Vega, Higher order dynamic mode decomposition, SIAM J. Appl. Dyn. Syst. 16 (2017) 882–925, http://dx.doi.org/10.1137/15M1054924.

[107] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 113 (2016) 3932–3937, http://dx.doi.org/10.1073/pnas.1517384113, arXiv: https://www.pnas.org/doi/pdf/10.1073/pnas.1517384113.

[108] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, Icml '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 1096–1103, http://dx.doi.org/10.1145/1390156.1390294.

[109] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing atari with deep reinforcement learning, 2013, arXiv:1312.5602.

[110] C. Szepesvári, Algorithms for Reinforcement Learning, in: Synthesis Lectures on Artificial Intelligence and Machine Learning, Springer International Publishing, Cham, 2010, http://dx.doi.org/10.1007/978-3-031-01551-9.

[111] C. Vignon, J. Rabault, R. Vinuesa, Recent advances in applying deep reinforcement learning for flow control: Perspectives and future directions, Phys. Fluids 35 (2023) 031301, http://dx.doi.org/10.1063/5.0143913.

[112] P. Westermann, R. Evins, Surrogate modelling for sustainable building design – A review, Energy Build. 198 (2019) 170–186, http://dx.doi.org/10.1016/j.enbuild.2019.05.057.

[113] X. Tian, Y. Cheng, Z. Lin, Modelling indoor environment indicators using artificial neural network in the stratified environments, Build. Environ. 208 (2022) 108581, http://dx.doi.org/10.1016/j.buildenv.2021.108581.

[114] L. Li, Y. He, H. Zhang, J.C.H. Fung, A.K.H. Lau, Enhancing IAQ, thermal comfort, and energy efficiency through an adaptive multi-objective particle swarm optimizer-grey wolf optimization algorithm for smart environmental control, Build. Environ. 235 (2023) 110235, http://dx.doi.org/10.1016/j.buildenv.2023.110235.

[115] X. Shen, Z. Cao, H. Liu, B. Cong, F. Zhou, Y. Ma, X. Zou, S. Wei, Inverse tracing of fire source in a single room based on CFD simulation and deep learning, J. Build. Eng. 76 (2023) 107069, http://dx.doi.org/10.1016/j.jobe.2023.107069.

[116] C. Ding, K.P. Lam, Data-driven model for cross ventilation potential in high-density cities based on coupled CFD simulation and machine learning, Build. Environ. 165 (2019) 106394, http://dx.doi.org/10.1016/j.buildenv.2019.106394.

[117] Y. He, X.-H. Liu, H.-L. Zhang, W. Zheng, F.-Y. Zhao, M. Aurel Schnabel, Y. Mei, Hybrid framework for rapid evaluation of wind environment around buildings through parametric design, CFD simulation, image processing and machine learning, Sustainable Cities Soc. 73 (2021) 103092, http://dx.doi.org/10.1016/j.scs.2021.103092.

[118] K.-M. Wai, P.K.N. Yu, Application of a machine learning method for prediction of urban neighborhood-scale air pollution, Int. J. Environ. Res. Public Health 20 (2023) 2412, http://dx.doi.org/10.3390/ijerph20032412.

[119] M. Mortezazadeh, J. Zou, M. Hosseini, S. Yang, L. Wang, Estimating urban wind speeds and wind power potentials based on machine learning with city fast fluid dynamics training data, Atmosphere 13 (2022) 214, http://dx.doi.org/10.3390/atmos13020214.

[120] S. Higgins, T. Stathopoulos, Application of artificial intelligence to urban wind energy, Build. Environ. 197 (2021) 107848, http://dx.doi.org/10.1016/j.buildenv.2021.107848.

[121] N.K. Kim, D.H. Kang, B.W. Kim, H.W. Kang, Optimal location and performance prediction of portable air cleaner in composite room shapes using convolutional neural network, Build. Environ. 242 (2023) 110500, http://dx.doi.org/10.1016/j.buildenv.2023.110500.

[122] S. Shin, K. Baek, H. So, Rapid monitoring of indoor air quality for efficient HVAC systems using fully convolutional network deep learning model, Build. Environ. 234 (2023) 110191, http://dx.doi.org/10.1016/j.buildenv.2023.110191.

[123] Q. Zhou, R. Ooka, Influence of data preprocessing on neural network performance for reproducing CFD simulations of non-isothermal indoor airflow distribution, Energy Build. 230 (2021a) 110525, http://dx.doi.org/10.1016/j.enbuild.2020.110525.

[124] Q. Zhou, R. Ooka, Neural network for indoor airflow prediction with CFD database, J. Phys. Conf. Ser. 2069 (2021) 012154, http://dx.doi.org/10.1088/1742-6596/2069/1/012154.

[125] C. Wei, R. Ooka, Indoor airflow field reconstruction using physics-informed neural network, Build. Environ. 242 (2023) 110563, http://dx.doi.org/10.1016/j.buildenv.2023.110563.

[126] T.V. Quang, D.T. Doan, N.L. Phuong, G.Y. Yun, Data-driven prediction of indoor airflow distribution in naturally ventilated residential buildings using combined CFD simulation and machine learning (ML) approach, J. Build. Phys. 47 (2024) 439–471, http://dx.doi.org/10.1177/17442591231219025.

[127] L.-c. Su, X. Wu, X. Zhang, X. Huang, Smart performance-based design for building fire safety: prediction of smoke motion via AI, J. Build. Eng. 43 (2021) 102529, http://dx.doi.org/10.1016/j.jobe.2021.102529.

[128] G. Lamberti, C. Gorlé, A multi-fidelity machine learning framework to predict wind loads on buildings, J. Wind Eng. Ind. Aerodyn. 214 (2021) 104647, http://dx.doi.org/10.1016/j.jweia.2021.104647.

[129] W. Peng, S. Qin, S. Yang, J. Wang, X. Liu, L.L. Wang, Fourier neural operator for real-time simulation of 3D dynamic urban microclimate, Build. Environ. 248 (2024) 111063, http://dx.doi.org/10.1016/j.buildenv.2023.111063.

[130] C. Huang, G. Zhang, J. Yao, X. Wang, J.K. Calautit, C. Zhao, N. An, X. Peng, Accelerated environmental performance-driven urban design with generative adversarial network, Build. Environ. 224 (2022) 109575, http://dx.doi.org/10.1016/j.buildenv.2022.109575.

[131] K. Javanroodi, V.M. Nik, M.G. Giometto, J.-L. Scartezzini, Combining computational fluid dynamics and neural networks to characterize microclimate extremes: Learning the complex interactions between meso-climate and urban morphology, Sci. Total Environ. 829 (2022) 154223, http://dx.doi.org/10.1016/j.scitotenv.2022.154223.

[132] J. Pedro Souza de Oliveira, J. Victor Barbosa Alves, J. Neuenschwander Escosteguy Carneiro, R. de Andrade Medronho, L. Fernando Lopes Rodrigues Silva, Coupling a neural network technique with CFD simulations for predicting 2-d atmospheric dispersion analyzing wind and composition effects, J. Loss Prev. Process Ind. 80 (2022) 104930, http://dx.doi.org/10.1016/j.jlp.2022.104930.

[133] X. Jurado, N. Reiminger, M. Benmoussa, J. Vazquez, C. Wemmert, Deep learning methods evaluation to predict air quality based on Computational Fluid Dynamics, Expert Syst. Appl. 203 (2022) 117294, http://dx.doi.org/10.1016/j.eswa.2022.117294.

[134] X. Lin, Y. Fu, D.Z. Peng, C.-H. Liu, M. Chu, Z. Chen, F. Yang, T.K.T. Tse, C.Y. Li, X. Feng, CFD- and BPNN- based investigation and prediction of air pollutant dispersion in urban environment, Sustainable Cities Soc. 100 (2024) 105029, http://dx.doi.org/10.1016/j.scs.2023.105029.

[135] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).

[136] N. Ibtehaz, M.S. Rahman, MultiResUNet : Rethinking the U-Net architecture for multimodal biomedical image segmentation, Neural Netw. 121 (2020) 74–87, http://dx.doi.org/10.1016/j.neunet.2019.08.025.

[137] S. Masoumi-Verki, F. Haghighat, U. Eicker, A review of advances towards efficient reduced-order models (ROM) for predicting urban airflow and pollutant dispersion, Build. Environ. 216 (2022) 108966, http://dx.doi.org/10.1016/j.buildenv.2022.108966.

[138] S. Wang, X. Chen, P. Geyer, Feasibility analysis of pod and deep autoencoder for reduced order modelling indoor environment cfd prediction, in: Proceedings of Building Simulation 2023: 18th Conference of IBPSA, Vol. 18, IBPSA, Shanghai, China, 2023, pp. 758–766, http://dx.doi.org/10.26868/25222708.2023.1227.

[139] S. Xiang, X. Fu, J. Zhou, Y. Wang, Y. Zhang, X. Hu, J. Xu, H. Liu, J. Liu, J. Ma, S. Tao, Non-intrusive reduced order model of urban airflow with dynamic boundary conditions, Build. Environ. 187 (2021a) 107397, http://dx.doi.org/10.1016/j.buildenv.2020.107397.

[140] S. Xiang, J. Zhou, X. Fu, L. Zheng, Y. Wang, Y. Zhang, K. Yi, J. Liu, J. Ma, S. Tao, Fast simulation of high resolution urban wind fields at city scale, Urban Clim. 39 (2021b) 100941, http://dx.doi.org/10.1016/j.uclim.2021.100941.

[141] S. Masoumi-Verki, F. Haghighat, U. Eicker, Improving the performance of a CAE-based reduced-order model for predicting turbulent airflow field around an isolated high-rise building, Sustainable Cities Soc. 87 (2022) 104252, http://dx.doi.org/10.1016/j.scs.2022.104252.

[142] S. Masoumi-Verki, F. Haghighat, N. Bouguila, U. Eicker, The use of GANs and transfer learning in model-order reduction of turbulent wake of an isolated high-rise building, Build. Environ. 246 (2023) 110948, http://dx.doi.org/10.1016/j.buildenv.2023.110948.

[143] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, WaveNet: A generative model for raw audio, 2016, arXiv:1609.03499.

[144] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[145] T. Saboori, L. Zhao, M. Mesgarpour, S. Wongwises, O. Mahian, A novel macro-scale machine learning prediction based on high-fidelity CFD simulations: A case study on the pore-scale porous Trombe wall with phase change material capsulation, J. Build. Eng. 54 (2022) 104505, http://dx.doi.org/10.1016/j.jobe.2022.104505.

[146] M. Mendil, S. Leirens, P. Armand, C. Duchenne, Hazardous atmospheric dispersion in urban areas: A Deep Learning approach for emergency pollution forecast, Environ. Model. Softw. 152 (2022) 105387, http://dx.doi.org/10.1016/j.envsoft.2022.105387.

[147] C. Waibel, R. Zhang, T. Wortmann, Physics meets machine learning: Coupling FFD with regression models for wind pressure prediction on high-rise facades, 2021.

[148] Y. Wang, J. Li, W. Liu, S. Zhang, J. Dong, J. liu, Prediction of urban airflow fields around isolated high-rise buildings using data-driven non-linear correction models, Build. Environ. 246 (2023) 110894, http://dx.doi.org/10.1016/j.buildenv.2023.110894.

[149] N. McGreivy, A. Hakim, Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations, 2024, arXiv:2407.07218.

[150] F. Bonnet, A. Mazari, P. Cinnella, P. Gallinari, AIRFRANS: high fidelity computational fluid dynamics dataset for approximating reynolds-averaged-navier-stokes solutions, 2022.

[151] N. Ashton, D.C. Maddix, S. Gundry, P.M. Shabestari, AhmedML: High-fidelity computational fluid dynamics dataset for incompressible, low-speed bluff body aerodynamics, 2024, arXiv:2407.20801.

[152] N. Ashton, J.B. Angel, A.S. Ghate, G.K.W. Kenway, M.L. Wong, C. Kiris, A. Walle, D.C. Maddix, G. Page, WindsorML: High-fidelity computational fluid dynamics dataset for automotive aerodynamics, 2024, arXiv:2407.19320.

[153] N. Ashton, C. Mockett, M. Fuchs, L. Fliessbach, H. Hetmann, T. Knacke, N. Schonwald, V. Skaperdas, G. Fotiadis, A. Walle, B. Hupertz, D. Maddix, DrivAerML: High-fidelity computational fluid dynamics dataset for road-car external aerodynamics, 2024, arXiv:2408.11969.

[154] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (2010) 1345–1359, http://dx.doi.org/10.1109/TKDE.2009.191.

[155] Y. Chen, Z. Tong, Y. Zheng, H. Samuelson, L. Norford, Transfer learning with deep neural networks for model predictive control of HVAC and natural ventilation in smart buildings, J. Clean. Prod. 254 (2020) 119866, http://dx.doi.org/10.1016/j.jclepro.2019.119866.

[156] J. Tompson, K. Schlachter, P. Sprechmann, K. Perlin, Accelerating Eulerian fluid simulation with convolutional networks, 2017, arXiv:1607.03597.

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many pages to expect for this document.