AIAA Journal
Vol. 60, No. 10, October 2022

Deep Learning Surrogate for the Temporal Propagation and Scattering of Acoustic Waves

Antonio Alguacil*
Université de Sherbrooke, Sherbrooke, Quebec J1K 2R1, Canada
Michael Bauerheim†
Institut Supérieur de l'Aéronautique et de l'Espace, ISAE-Supaero, Toulouse, France
Marc C. Jacob‡
École Centrale de Lyon, Université de Lyon, 69130 Ecully, France
and
Stéphane Moreau§
Université de Sherbrooke, Sherbrooke, Quebec, J1K 2R1, Canada

https://doi.org/10.2514/1.J061495

A deep learning surrogate for the direct numerical temporal prediction of two-dimensional acoustic waves propagation and scattering with obstacles is developed through an autoregressive spatiotemporal convolutional neural network. A single database of high-fidelity lattice Boltzmann simulations is employed in the training of the network, achieving accurate predictions for long simulation times for a variety of test cases representative of bounded and unbounded configurations. The capacity of the network to extrapolate outside the manifold of examples seen during the training phase is demonstrated by obtaining accurate acoustic predictions for relevant applications, such as the scattering of acoustic waves on an airfoil trailing edge, an engine nacelle, or an in-duct propagation. The study focuses on the influence of three main design parameters that allow rolling out accurate and stable long-term predictions: 1) the choice of a dataset-related characteristic time, 2) the normalization of the input data, and 3) the number of input temporal frames into the neural network. The results show that for the optimum choice of design parameters, the presented data-driven model is able to systematically obtain low-error prediction at a lower computational cost than the reference high-fidelity computational code.

Nomenclature

b_p	=	Gaussian pulse half-width
$\boldsymbol{c}$	=	Gaussian pulse center
c_0	=	speed of sound
D	=	domain size
$\mathcal{L}$	=	loss function
N_{nodes}	=	number of nodes in per spatial direction
r_c, R	=	cylinder radii
$\boldsymbol{u}$	=	fluctuating velocity vector
$\mathcal{U}$	=	uniform probability distribution function
$\boldsymbol{x}$	=	spatial coordinate vector
$\boldsymbol{x}_c$	=	cylinder center
$\boldsymbol{x}_p$	=	Gaussian pulse center
$\boldsymbol{x}_r$	=	Rectangle origin
Δt	=	time step
Δx	=	grid spacing
ε	=	amplitude of initial Gaussian source
ε_p	=	amplitude of initial Gaussian pulse
θ	=	neural network parameters
ρ	=	fluctuating density
σ	=	standard deviation
τ_D	=	nondimensional time based on the domain size
ϕ	=	acoustic potential

I. Introduction

THE propagation of acoustic waves in complex media constitutes a challenging task in the context of aerodynamically generated noise. The presence of complex boundaries, mean flow, or background medium inhomogeneities affects the trajectory of propagating waves through known physical mechanisms such as scattering, dispersion, reflection, or absorption. Taking into account such phenomena is required to design quiet aeronautical devices; however, creating such predesign tools with a low cost is still an open challenge. To do so, two types of approaches exist for predicting noise propagation [1]: on the one hand, direct numerical computational aeroacoustics simultaneously calculate both acoustic sources and the subsequent wave propagation [2]. Direct computations yield highly accurate results but are highly demanding in terms of computational costs. On the other hand, hybrid methods constitute a more affordable way to perform aeroacoustic predictions. These methods rely on the separate computations of the source region and the propagation into the far field. The propagation step can be achieved either through semianalytical means (acoustic analogies) or fully numerical approaches (linearized Euler equations or acoustic perturbation equations).¶

The coupling between the source and the propagation regions can become cumbersome, especially when considering the complex media effects on propagation. For example, acoustic analogies for nonhomogeneous media (e.g., Lilley's analogy for sheared mean flows [3]) require an additional effort in the computation of the associated Green's

Presented as Paper 2021-2239 at the AIAA Aviation 2021 Forum, Virtual Event, August 2–6, 2021; received 29 November 2021; revision received 29 April 2022; accepted for publication 17 May 2022; published online 10 August 2022.

*Ph.D. Student, Mechanical Engineering Department; also Département Aérodynamique, Énergétique et Propulsion, Institut Supérieur de l'Aéronautique et de l'Espace ISAE-Supaero, BP 54032, F-31055 Toulouse Cedex 4, France; antonio.alguacil-cabrerizo@isae-supaero.fr.

†Associate Professor, Département Aérodynamique, Énergétique et Propulsion, Institut Supérieur de l'Aéronautique et de l'Espace ISAE-Supaero, BP 54032, F-31055 Toulouse Cedex 4, France; also Mechanical Engineering Department, Université de Sherbrooke, J1K 2R1 Sherbrooke, Quebec, Canada.

‡Professor, Université de Lyon, Ecole Centrale de Lyon, Centre National de la Recherche Scientifique, Université Claude Bernard Lyon (Lyon, France), Institut National des Sciences Appliquées de Lyon (Lyon, France), Laboratoire de Mécanique des Fluides et d'Acoustique, Unité Mixte de Recherche 5509, F-69134 Ecully Cedex, France. Member AIAA.

§Professor, Mechanical Engineering Department. Lifetime Member AIAA.

¶Code and datasets are publicly available at https://gitlab.isae-supaero.fr/daep/dl-surrogate-acoustic-scattering.

function because of the presence of hydrodynamic modes and critical layers. The same observation applies for complex boundary conditions, as in the Ffowcs Williams and Hawkings analogy [4], where new integrals appear when accounting for the presence of moving surfaces. Thus, the more mean flow or boundaries effects are taken into account by the acoustic analogy, the more complex the calculation of the associated Green's function or its subsequent convolution with sources becomes. In the most complex cases, which are relevant for industrial application, the Green's function can no longer be fully analytically calculated (e.g., annular duct with swirling mean flow analogy [5]), and numerical methods must be employed instead. Therefore, in the general cases of complex mean flows or boundary conditions, propagation methods must rely on expensive numerical methods by using (for instance) high-order numerical schemes to ensure low dissipation and dispersion properties, which prevents these methods from being applied efficiently at a predesign stage.

In this context, using a reduced-order model or, more generally, a data-driven surrogate model can help drive down the computational cost of aeroacoustic prediction while maintaining a high accuracy level. The quick development of deep learning methods, with neural networks at their backbone, has led to applications in several fluid-mechanics problems. Examples can be found in turbulence modeling [6], prediction of aerodynamic flows [7], performance [8] and inverse design [9] of airfoils, identification of nonlinear unsteady loads [10], or prediction of compressible cavity flows [11].

Recent work by the authors [12,13] developed an alternative method to replicate, at lower cost, a high-fidelity numerical code for the propagation of acoustic waves using a surrogate model: instead of resolving the discretized propagation equations, a neural network was trained with examples of high-fidelity direct lattice–Boltzmann method (LBM) simulations, resulting in a data-driven convolutional neural network for acoustic wave propagation. It was found that such an approach managed to reproduce wave propagation in simple academic cases of two-dimensional (2-D) quiescent closed domains with reflecting walls. These promising results suggest that a similar approach might be applied in open-domain problems in the presence of complex boundary conditions (such as ducts or scattering objects). In particular, the method has demonstrated good generalization capabilities (e.g., being able to predict wave propagation of unseen source distribution) while diminishing classical numerical constraints [e.g., the Courant–Friedrichs–Lewy (CFL) number], allowing faster simulations without loss of accuracy.

Yet, the previous framework [12] implicitly encoded the boundary conditions of the problem, and was thus not applicable to more complex geometries and representative configurations. Several studies have dealt with the problem of encoding boundary condition information into neural network surrogates for modeling partial differential equations (PDEs). Some make use of built-in tools of convolutional networks, such as periodic padding to encode simple boundary conditions, such as periodic conditions [14]. Other works dealing with more complex geometries employ binary inputs [7] or signed distance functions [15] to encode the geometry information. However, such works only tackle steady-state problems, such as Reynolds-averaged Navier–Stokes predictions [7] or sound pressure level estimation [15].

Closer to the problem of interest, Lino et al. [16] predicted the full spatiotemporal propagation and scattering of seismic waves with a convolutional network. They employed a binary field to encode geometry information and used an autoregressive strategy to perform the temporal prediction. However, few quantitative results were shown: in particular, regarding the ability of the network to perform accurately outside the training dataset. Furthermore, it seems that they trained the network on different datasets, depending on the different types of boundary conditions. A general database capable of training the neural network for different types of boundaries conditions would be beneficial to such kinds of methods because it would avoid the cost of retraining the network each time. A typical example of such a pathology can be found in physics-informed neural networks (PINNs) [17], which need to be retrained for each new set of initial or boundary conditions. Some works, such as that by Moseley et al. [18], employed PINNs to solve the wave equation for complex background media [18] and parameterize the source input position in order to avoid retraining the model for each new set of initial conditions. Other works, such as that of Hasegawa et al. [19], used a combination of spatial (convolutional autoencoder) and recurrent neural networks (long short-term memory), creating a reduced-order model for the prediction of low-Reynolds-number unsteady flows around bluff bodies. Again, a binary field allows the network to work on different types of body shapes. However, the problem of using a single neural network for various types of boundary conditions, such as reflective and nonreflective acoustic conditions, is still an open question.

Consequently, the present work proposes a unified generic framework based on a deep autoregressive convolutional neural network to tackle both bounded and unbounded acoustic problems in quiescent flows, with potential scattering by inner objects. The work focuses on two mains aspects: 1) the ability of the trained neural network to perform in various physics regimes dependent on the choice of boundary conditions (some of which differ significantly from the ones present in the chosen database), and 2) the influence on the choices of hyperparameters employed to train an accurate and stable model for long-term autoregressive predictions.

First, the methodology regarding the choice of neural network architecture and its training are presented in Sec. II, Then, Sec. III describes the unique dataset employed for training the neural network and its validation. In Sec. V, several test cases representative of unbounded problems are evaluated using the present framework, and they are compared with the LBM reference. In particular, Secs. V.C and V.D explore two additional cases representative of typical industrial applications, namely, a NACA airfoil and an engine nacelle. Ducted configurations as well as a completely closed domain are also studied in Sec. VI. Section VII studies the influence of the main hyperparameters of the learned model. Finally, Sec. VIII discusses the computational cost of the presented method with respect to traditional numerical computational fluid dynamics (CFD) methods.

II. Deep Learning Methodology

A. Modeling Dynamical Systems with a Learned Surrogate

The objective of this work is to efficiently approximate the complete space–time evolution of a dynamical system with a surrogate model trained on high-fidelity data. Here, the system of interest is the propagation and scattering of acoustic waves with hard-reflecting obstacles in quiescent flows, which can be described by the following Cauchy problem for the acoustic density ρ wave equation:

$$\begin{cases} \dfrac{\partial^2 \rho(\boldsymbol{x}, t)}{\partial t^2} + c_0^2 \Delta \rho(\boldsymbol{x}, t) = f(\boldsymbol{x}, t), & \boldsymbol{x} \in \mathcal{D}, t \in [0, T] \\ \mathcal{B}(\rho(\boldsymbol{x}, t)) = 0, & \boldsymbol{x} \in \partial \mathcal{D} \\ \rho(\boldsymbol{x}, 0) = \mathcal{I}_0(\boldsymbol{x}), & \end{cases} \tag{1}$$

where ρ is defined in the domain $\mathcal{D}$ with boundaries $\partial\mathcal{D}$. $\mathcal{B}$ is the boundary operator that enforces the boundary conditions (here, $\mathcal{B} = 0$ expresses the absence of sources along the boundaries). Typically, for hard-reflecting walls, it simply reads $\nabla\rho(\boldsymbol{x} \cdot \boldsymbol{n})|_{\partial\mathcal{D}} = 0$. Note that c_0 is the speed of sound, f is a source term, and $\mathcal{I}_0$ is the initial condition expressed as a volume distribution.

For an arbitrary boundary condition operator $\mathcal{B}$, Eq. (1) is usually resolved numerically by discretizing the state variable in space and time, and subsequently integrating the time evolution from t to $t + \Delta t$ using an iterative solver, where Δt is the time step between to iterations.

The goal of the present study is to recast this numerical integration as an optimization problem in order to learn the space–time evolution of the acoustic density. Discretizing the state variable in space and time, let $\rho_i \in \mathbb{R}^{d_1 \times d_2}$ be the solution of Eq. (1) at time step i on a uniformly spaced discretization of $\mathcal{D}$ defined by a grid with d_i nodes in each direction.

The learned PDE operator G, defined by its trainable parameters θ, must fulfill the following time-invariant equation:

$$\rho_{i+1} = G(X_{i+1}, \theta) \tag{2}$$

where $X_{i+1} = \{\rho_i, \rho_{i-1}, \ldots, \rho_{i-k}\}$ is the input of the surrogate model, corresponding to the $k+1$ previous time states.

The operator G is obtained by solving the following optimization problem:

$$\arg\min_{\theta} \mathcal{L}(G(X_{i+1}, \theta), \rho_{i+1}) \tag{3}$$

where $\mathcal{L}$ is a metric of the distance between the approximated prediction from the learned operator $\hat{\rho}_{i+1} = G(X_{i+1}, \theta)$ and the actual solution ρ_{i+1}. Here, ρ_{i+1} is obtained through a high-fidelity numerical simulation database as described in Sec. III, but it could be obtained by other means: analytical solutions (if existing), experimental observation, or a blending of the previous.

Such a space–time operator can be learned through a highly nonlinear neural network regressor. For high-dimensional databases, such as the ones encountered in computational aeroacoustics or computational fluid dynamics, the use of convolutional networks is an efficient solution to simultaneously learn the spatial and time integration, as shown in earlier works [13,14,16]. The main advantage with respect to traditional numerical schemes is twofold: in comparison with explicit schemes, such as Euler or Runge–Kutta integrators, the learned surrogate is not constrained by the classical time-stepping stability constraints [12,20]. Furthermore, it does not require any matrix inversion as implicit integrators do.

B. Autoregressive Spatiotemporal Prediction of Wave Propagation

Once trained, the resulting neural network G is employed in an autoregressive strategy to predict the complete spatiotemporal evolution of acoustic fields. Although the network is only trained to predict one single time step ahead from the input data, this autoregressive strategy is used as a time integrator to propagate signals up to an arbitrary time horizon.

Formally, the acoustic density at time $t = i\Delta t$ can be calculated as

$$\rho_i = \underbrace{G \circ G \circ \ldots \circ G}_{i-k\text{times}}(X_{k+1}) \tag{4}$$

where $X_{k+1} = \{\rho_k, \rho_{k-1}, \ldots, \rho_0\}$ contains the initial condition. In practice, the last predicted frame ($i+1$ in Fig. 1) is added to a new series of inputs (composed of frames at $i-k+1, \ldots,$ and $i+1$). In Fig. 1, the input is composed of k consecutive frames $t = i-k$, $t = i-k+1, \ldots, t = i$ of acoustic density and of a Boolean mask representing the position of the obstacles. The output is the next frame at $t = i+1$ with a resolution of $N \times N$ voxels. Convolution kernels have a size of $k \times k$, and each rectangular block is proportional to the number of convolutional filters per layer ($c\#$). ReLU activations are used, except at the last layer, where an identity mapping is used to obtain both positive and negative outputs. Skip connections are employed to facilitate the training of the network.

In practice, $k = 1$; and this choice will be justified in Sec. VII. Note that in previous works [12], four input snapshots ($k = 3$) were employed because it was seen that varying this parameter had an insignificant influence on the overall performance of the neural network. However, the effects of k have to be studied again in this work due to a fundamental change in the nature of the data: whereas in Ref. [12] all the simulations conserved the acoustic energy over time, this conservation no longer holds in this work because nonreflecting boundary conditions are potentially employed (see Sec. III for further details regarding the dataset).

C. Neural Network: U-Net Convolutional Network

A U-net convolutional neural network [21] is employed as shown in Fig. 1. It follows an encoder-decoder layout characterized by an U-shape in the successive convolution and downsampling/upsampling operations. Such an architecture is capable of treating large structured image-like inputs with few trainable parameters by sliding small local trainable convolutional filters around the input (here, 3×3 convolutional filters are used). In comparison with classical fully connected networks, convolutional networks naturally encode neighborhood information between pixels, which is key when resolving partial differential equations. This can be related to the stencils used to calculate the spatial gradients in a discrete grid.

Furthermore, chained convolutions are combined with downsampling operations to increase the receptive field [22] in order to capture long-range spatial information. The U-net architecture combines convolutions with pairs of down- and upsampling operations to treat the input in a multiscale way. In this case, maximum pooling operation along with bilinear upsampling is employed. The multiscale treatment separately processes the different scales of the problem, specializing the different filters to focus either on local or global flow features. The U-net network was found to converge more quickly than the multiscale model used in previous works [12]. This can be attributed to the presence of skip connections that facilitate the backward flow of gradients during the optimization of the network. Moreover, U-net networks have better inference time performance as compared with the multiscale net of Ref. [12], which further accelerates the acoustic predictions.

Leaky nonlinear rectifying linear units (ReLUs) are employed after each convolution operation to create a nonlinear mapping between learned features. For a fixed set of boundary conditions, the modeled wave equation [Eq. (1)] is linear. However, when dealing with changing boundary conditions (different reflecting conditions), such linearity no longer holds: the solution cannot be superposed for two separated domains with reflecting obstacles to obtain the solution for the superposed geometry. Nonetheless, the possibility of using a linear mapping by replacing all nonlinear activation by identity functions in the network has been explored. However, such an approach has been found to not yield accurate results in the neighborhood of reflecting obstacles. This suggests

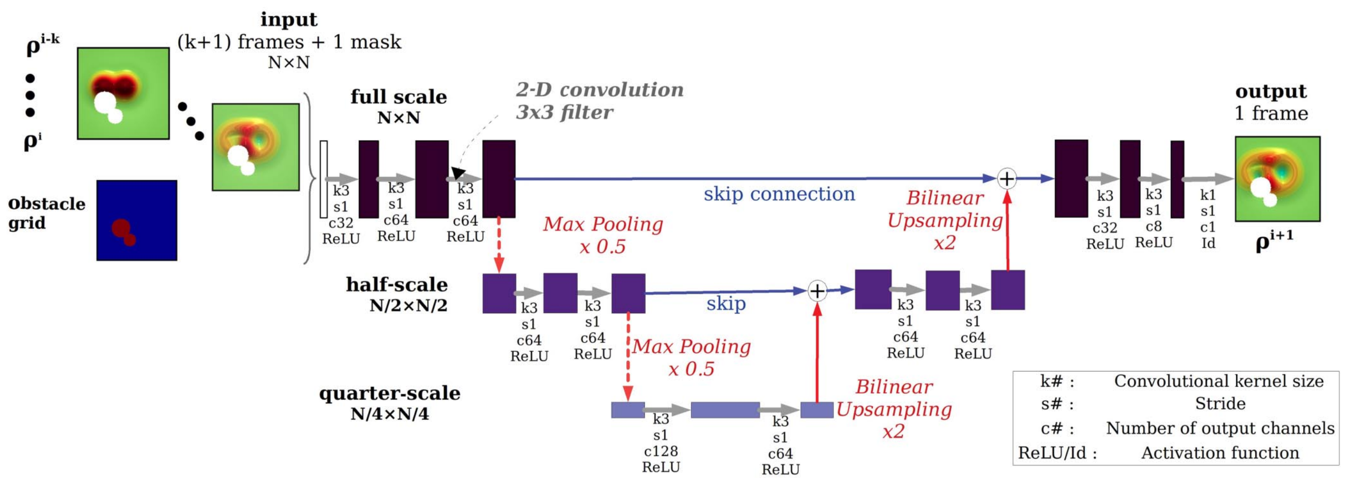

Fig. 1 Schematics of the U-net multiscale CNN [21] with three scales of convolutions at full, half-, and quarter-resolutions.

that the usage of the nonlinearity is key to building a physical surrogate that treats complex boundary conditions.

The input to the neural network corresponds to the previously defined X_{i+1} vector concatenated with an additional obstacle grid g indicating the presence of the reflecting obstacles with a Boolean encoding (ones assigned to obstacle pixels and zeros to fluid ones). Thus, the positions of the boundary conditions are explicitly given as inputs to the neural network, as employed in other works using convolutional neural networks in fluid-mechanics-related problems [7,23,24].

The time step Δt at which the prediction $t \to t + \Delta t$ is performed is fixed by the temporal spacing between the input and output; thus, in order to change its value, a new training is required.

III. Dataset: Propagating Gaussian Pulses in Unbounded Domain with Hard Wall Obstacles

As in Ref. [13], the numerical code based on the lattice-Boltzmann method is used to generate the data for the optimization of the network on the training dataset. Such a dataset, employed to feed inputs into the neural network and compare its outputs with supervised references, is described in the following section. Particular attention is paid to the validation of the LBM code on the wave scattering by objects to guarantee the accuracy of the training database.

A. Numerical Setup

The dataset consists of 120 (100 for training and 20 for validation) 2-D simulations of time-propagating acoustic waves on square domains with nonreflecting boundary conditions (dubbed the "unbounded" domain) and varying random obstacles with hard-reflecting walls (cylinders and rectangles), which are placed inside the numerical domain, as sketched in Fig. 2. The computational domain has a size of $D \times D$. Similar to the dataset used in Ref. [13], the acoustic sources are composed of a random number of Gaussian pulses, which are used as initial conditions for the fluctuating density field, namely,

$$\rho(\boldsymbol{x}, 0) = \sum_{p=1}^{N_p} \varepsilon_p \exp\left(-\frac{\log 2}{b_p^2} \|\boldsymbol{x} - \boldsymbol{x}_p\|\right) \tag{5}$$

where the number of initial pulses N_p is sampled from an uniform distribution $\mathcal{U}[1, 5]$; ε is the pulse amplitude, which is here fixed to $\varepsilon_p = 10^{-3}$; b_p is the pulse half-width; and $\boldsymbol{x}_p$ is the pulse center coordinate such that $\boldsymbol{x}_p = \mathcal{U}[0.2D, 0.8D]^2$.

Two types of reflecting obstacles are used: cylinders and rectangles. For each simulation, the set of cylinders is defined as follows:

$$\mathcal{C} = \{\boldsymbol{x} \in \mathcal{D} \,\|\, \|\boldsymbol{x} - \boldsymbol{x}_c\|^2 < r_c^2\},\ c \in [0, N_c]$$
$$\text{with } N_c = \mathcal{U}[0, 3],\ r_c = \mathcal{U}[0.08D, 0.15D],\ \boldsymbol{x}_c = \mathcal{U}[0.2D, 0.8D]^2 \tag{6}$$

where N_c is the number of cylinder, and r_c is their radius with respect to their center coordinates $\boldsymbol{x}_c$.

Similarly, the set of rectangles is defined as follows:

$$\mathcal{R} = \{\boldsymbol{x} = (x, y) \in \mathcal{D} \,\|\, |x - x_r| < w_r \cap |y - y_r| < h_r\},\ r \in [0, N_r]$$
$$\text{with } N_r = \mathcal{U}[0, 3],\ w_r = \mathcal{U}[0.01D, 0.3D],\ h_r = \mathcal{U}[0.01D, 0.3D],$$
$$x_r = \mathcal{U}[0.2D, 0.8D],\ y_r = \mathcal{U}[0.2D, 0.8D] \tag{7}$$

where N_r is the number of rectangles; h_r is their height; and w_r is the width with respect to their left bottom point, defined by coordinates $\boldsymbol{x}_r = (x_r, y_r)$.

Boundary conditions at the obstacle walls are modeled with bounce-back conditions. This type of boundary is not generally fitted for inclined boundaries, such as the cylinder wall. However, in the absence of mean flow, if the wavelength of acoustic waves is sufficiently large with respect to the cell size Δx, then the bounce-back "staircase" will be compact and have no effect on the accuracy of the numerical results. This is demonstrated in Sec. III.B, where the LBM is validated for wave scattering by a cylinder. Nonreflecting boundary conditions are modeled using a perfectly matched layer that prescribes a damping term in the governing equations in order to attenuate outgoing acoustic modes [25]. The output fields from the LBM simulations are cropped to only feed the physical domain to the neural network, with each direction being discretized with$N_{\text{nodes}} = 200$.

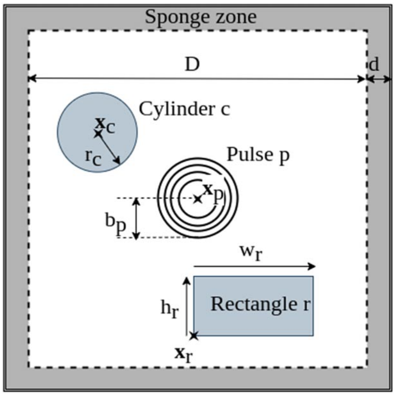

Fig. 2 Schematic of a typical dataset simulation.

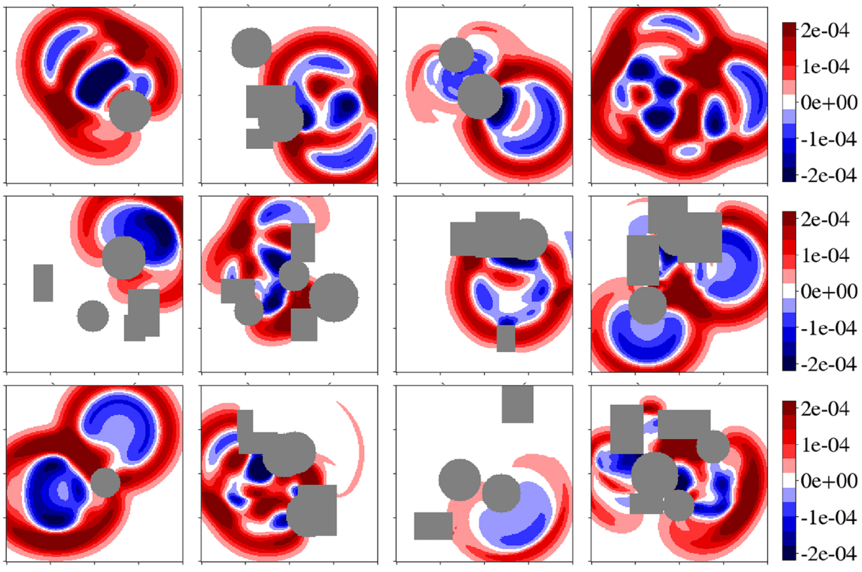

Fig. 3 Snapshots of acoustic density for different dataset simulations at a fixed time ($\tau = 0.35$).

This dataset aims at studying the capacity of the neural network to predict the scattering of acoustic waves by such hard-wall reflecting obstacles and their subsequent propagation into the far field. Furthermore, complex interactions between acoustic waves and reflecting walls are also simulated because the random arrangement of obstacles may act in some cases as a waveguide, rearranging the signals into modelike shapes. Figure 3 shows an example for different simulations in the dataset. Initial conditions are composed of Gaussian pulses of acoustic density located at random positions. Obstacles (cylinders and rectangles) are randomly located in the computational domain. Because the obstacles may overlap, this creates complex geometries where the acoustic signals reflect and diffract.

The results are reported using a normalized time based on the size of the domain D, which is defined as

$$\tau_D = tc_0/D \tag{8}$$

A grid spacing of $\Delta x = D/(N_{\text{nodes}} - 1) = 0.5025$ is employed. Notice that in the LBM, the value of the time step Δt is proportional to the grid spacing, which is $\Delta t = c_s/c_0 \Delta x$ here. This acts as a Courant–Friedrichs–Lewy condition, depending on the sound speed in lattice units c_s, which is determined by the lattice choice, which is equal to $(c_s)_{lb} = 1/\sqrt{3} \simeq 0.57$ here in the 2-D nine-velocity lattice-Boltzmann lattice [26].

The time τ_D corresponds to the propagation time of an acoustic wave from one boundary to the other. Each training simulation is stopped at $\tau_D^{\text{train}} = 1.44$ (corresponding to 500 LBM iterations), leaving sufficient time for all signals to leave the computational domain. The effect of such a parameter will be studied in Sec. VII. Density fields are recorded at time steps that are multiples of $\Delta\tau = 0.0087$ (i.e., every three LBM iterations). The latter time step is the one used by the convolutional neural network (CNN) in order to demonstrate the CNN capability to replicate high-fidelity codes while mitigating their CFL constraint. In practice, the computed LBM fields are packed into groups of $k + 1$ inputs and one target frame for the training phase, as detailed in Sec. II.B. The choice of k will be justified in Sec. VII.

B. Validation of the Dataset

The LBM code is validated with respect to the scattering of a Gaussian pulse by a cylinder. A schematic is shown in Fig. 4a. A rigid cylinder of radius R is located at $(x, y) = (0, 0)$ inside a domain of size $30R \times 30R$, plus an additional $2R$ of sponge zones at each boundary. A density Gaussian pulse of half-width of $b/R = 0.4$ is located at $(x, y) = (8R, 0)$, and a probe is located at $(x, y) = (0, 10R)$. The LBM employs a recursive and regularized BGK (BGK stands for Bhathagar, Gross and Krook [27]) collision model (with the LBM) [28] to maintain code stability with low numerical dissipation. Two cases are compared: the first one with the pulse half-width is discretized by 12 lattice points (901 lattice points in the domain excluding sponge zones), and the second one is discretized with 18 points per half-width (1351 lattice points in the domain, excluding sponge zones).

The results are compared to the analytical solution derived by Tam [29]. For a pulse of half-width b and a pulse parameter of $\alpha = (\log 2)/b^2$ located at (x_s, y_s), the total unknown potential can be divided in two components related, respectively, to the incident (subscript i) and reflected waves (subscript r):

$$\phi(x, y, t) = \phi_i(x, y, t) + \phi_r(x, y, t) \tag{9}$$

The total pressure field is found as follows:

$$\begin{aligned} p(x, y, t) &= -\frac{\partial \phi}{\partial t} = -\frac{\partial}{\partial t}(\phi_i + \phi_r) \\ &= Re\left\{\int_0^\infty (A_i(x, y, \omega) + A_r(x, y, \omega))e^{-i\omega t}\,\mathrm{d}\omega\right\} \end{aligned} \tag{10}$$

where $A_i(x, y, \omega)$ is given by

$$A_i(x, y, \omega) = \frac{1}{2\alpha} e^{-\frac{\omega^2}{(4\alpha)}} J_0(\omega r_s) \tag{11}$$

and $r_s = \sqrt{(x - x_s)^2 + (y - y_s)^2}$. J_k represents the Bessel function of first kind of order k. $A_r(x, y, \omega)$ is given in polar coordinates (r, θ):

$$A_r(x, y, \omega) = \sum_{k=0}^{\infty} C_k(\omega) H_k^{(1)}(r\omega)\cos(k\theta) \tag{12}$$

where

$$\begin{aligned} C_k(\omega) &= \left(\frac{\omega}{2b} e^{-\frac{\omega^2}{(4\alpha)}}\right) \frac{\varepsilon_k}{\pi\omega H_k^{(1)'}(r_0\omega)} \\ &\quad \times \int_0^\pi J_1(\omega r_{s0}) \frac{r_0 - x_s\cos(\theta) - y_s\sin(\theta)}{r_{s0}} \cos(k\omega)\,\mathrm{d}\theta \end{aligned} \tag{13}$$

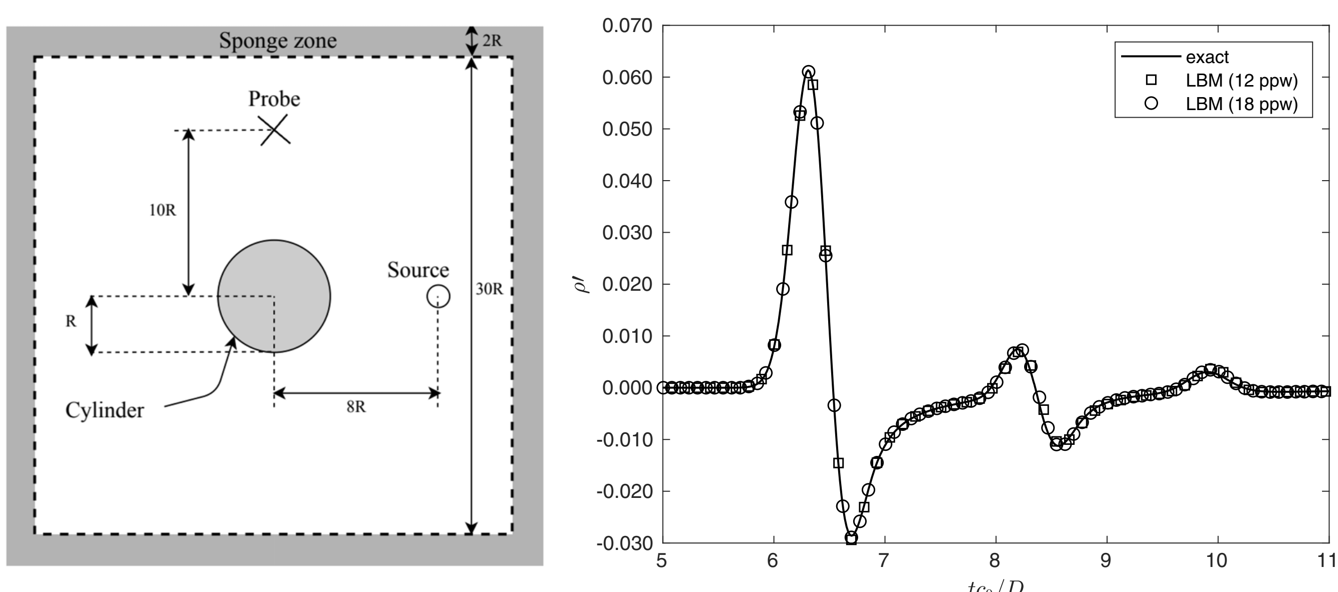

Fig. 4 Configuration used for the validation case (Fig. 4a): a cylinder of radius R located at the domain center scatters a Gaussian pulse. Measurements are performed at probe location. Fluctuating density ρ' at probe location $(x, y) = (0, 10R)$ for dimensionless time $\tau = tc_0/D$ for different initial pulse spatial resolution $b/\Delta x$ (indicated as point-per-wavelength or ppw in the legend) and reference analytical solution (Fig. 4b).

with $\varepsilon_0 = 1$ and $\varepsilon_k = 2$ for $k \neq 0$. Also, r_{s0} is defined as

$$r_{s0} = \sqrt{r_0^2 - 2r_0 x_s \cos\theta - 2r_0 y_s \sin\theta + x_s^2 + y_s^2} \tag{14}$$

and $H_k^{(1)'}$ is the derivative of the Hankel function of the first kind of order k.

The pressure solution obtained in Eq. (10) can be related to the fluctuating density field through the linearized equation of state: $p = c_0^2 \rho$.

Figure 4b shows the time evolution of the density at the probe location for two different resolutions of the LBM lattice. A resolution of 12 points per half-width is sufficient to capture all the physics of the problem accurately. The first amplitude peak represents the direct propagation of the initial Gaussian pulse, whereas the other two peaks at $tc_0/D = 8$ and 10 arise from the pulse scattered by the cylinder wall. Therefore, the pulses in the dataset are resolved with $b/\Delta = 12$ points per initial half-width.

C. Testing Data

The autoregressive methodology is tested in a series of unbounded and bounded cases to demonstrate the ability of the framework to predict acoustic propagation for several configurations with a quiescent mean flow. The objective is to demonstrate that a single neural network, trained on a dataset such as the one described in Sec. III.A, can perform accurately on all the test scenarios without the need to build a tailored database for each case. The interest of such approach is to avoid the repetition of the costly training process for each new configuration, as well as to demonstrate that the employed neural network is able to perform well on new cases not seen during the training process.

Sketches of the different domain configurations and initial pulse locations are shown in Fig. 5. Gray hashed zones represent non-reflecting boundary conditions, whereas black zones represent reflecting obstacles: a) free-field propagation; b) scattering by a cylinder; c) scattering by a NACA0020 airfoil; d) propagation inside a nacelle and scattering into far field; e) in-duct wave propagation for

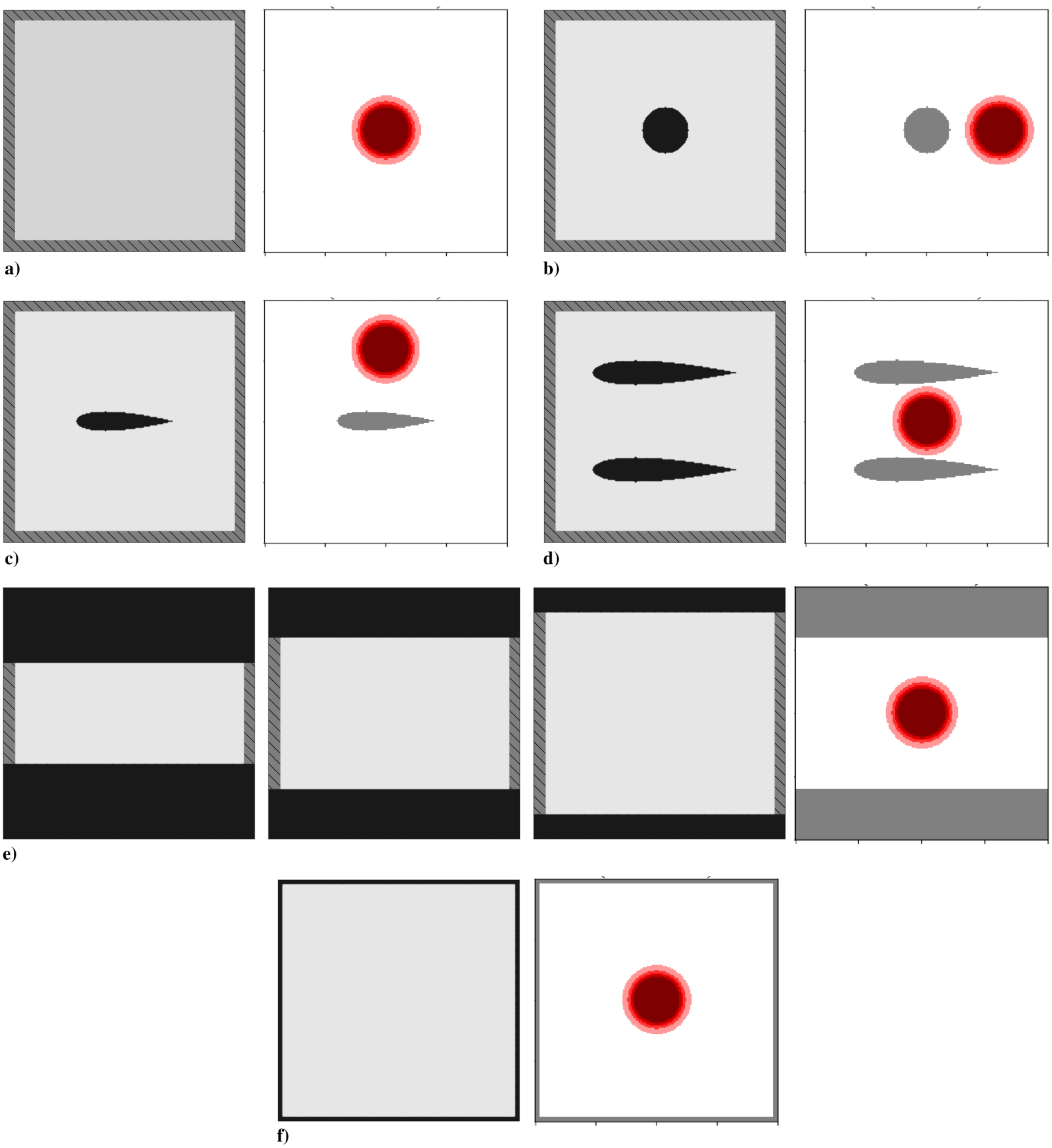

Fig. 5 Schematic view of test geometries and density-field initial condition.

different diameter-to-domain length ratios (from left to right) of $h/D = 0.4$, 0.6, and 0.8; and f) closed-box propagation. All the domains are kept at the size of $D \times D$. First, the network is tested on a series of open bounded cases (i.e., nonreflecting boundary conditions are employed in the four boundaries of the domain), shown in Figs. 5a–5d, respectively. Then, bounded configurations are also studied, where at least two opposite domain boundaries are set as reflective conditions, thus testing the ability of the network to propagate acoustics in ductlike shapes. The lower part of Fig. 5 shows the four tested bounded configurations, shown in Fig. 5e for the ductlike geometry and Fig. 5f for the closed-box domains.

IV. Training and Evaluation Methodology

The methods used to train the neural network in practice are presented, providing the most important hyperparameters of the training in detail. Some emphasis is also put on the repeatability of the neural network, which is inherently stochastic. Such a study is necessary in order to obtain more robust results when dealing with autoregressive predictions, as demonstrated in Ref. [30].

A. Data Normalization

Neural networks benefit from scaling the inputs so that the mean and the standard deviation of their inputs are close to zero and one, respectively [31], in order to improve the stability of the neural network training. However, scaling all the inputs with their statistical moments generates a loss of physical information for the neural network, most importantly in time-dependent problems, because the network may no longer capture the relative changes of amplitude between different time steps, accounting for phenomena such as diffusion or, in the present case, energy decay (e.g., the acoustic signals leave the computational domain after some time). Thus, two types of input normalization are tested in this work in order to evaluate their performance when performing long-term autoregressive predictions.

Global Normalization

The first normalization, used in a U-net network dubbed GlobUnet (Globally-normalized U-net), consists of a global scaling to normalize all inputs by $1/\rho_0$, where ρ_0 is the amplitude of one Gaussian pulse at $t = 0$ such that

$$\tilde{\rho}_i = \frac{\rho_i}{\rho_0} \tag{15}$$

This scaling intends to preserve the original amplitude scaling of the dataset, which may be important to account for time-decaying phenomena. The disadvantage of this method is that some of the network inputs when the acoustic signal has mostly left the computational domain may have a small amplitude as compared with the initial condition. Thus, the network may struggle when processing such low-energy inputs, and it may overfit to high-energy signals.

Local Normalization

The second normalization, denoted as LocUnet (Locally-normalized U-net), employs an input-to-input scaling. In particular, each input is normalized by $\sigma(\rho_{i-k})$, where σ is the standard deviation of first frame for each input (composed of consecutive spatiotemporal frames of density, as explained in Sec. II):

$$\tilde{\rho}_i = \frac{\rho_i}{\sigma(\rho_{i-k})} \tag{16}$$

Because k consecutive frames are used at each input, the network is presented with information about amplitude decay between all the different snapshots, although more locally than for the global normalization. The advantage of such a strategy is that it removes the low-signal energy problem of the global normalization. It may, however, amplify spurious noise (e.g., numerical reflections at the boundaries) with low amplitude in the original dataset scale. Both strategies are compared for all test cases in Secs. V and VI.

B. Training the Neural Network and Repeatability

The supervised training of the neural network G is performed over the dataset composed of N_s pairs of input–target samples $(X_{i+1}, \rho_{i+1}))_{\leq i \leq N_s}$. Equation (3) is solved by defining the loss function $\mathcal{L}_2$ used to train the neural network, which is computed at the output of G, and reads

$$\mathcal{L}_2 = \frac{1}{N} \sum_{n=1}^{N} \| \hat{\rho}_{i+1}^n, \rho_{i+1}^n \|_2^2 \tag{17}$$

where the summation is done over the total number N of data samples of prediction–reference pairs. The loss minimizes the L2-norm mean square error (MSE) of the prediction $\hat{\rho}_{i+1}^n$ with respect to the reference field ρ_{i+1}^n.

In practice, the training of the neural network is performed by optimizing its parameters by using a variant of the classical minibatch stochastic gradient descent algorithm, which is the Adam optimizer [32], with an initial learning rate of 4×10^{-4} and a learning rate schedule that decreases its value by 15% each time the network validation loss reaches a plateau for more than 10 optimizer passes on the full dataset (called epochs). Note that convolutional networks are translation invariant by construction [33]. However, they are not invariant by rotation. Thus, data augmentation is performed during the training of the network: input–target pairs are randomly flipped in angles of 90, 180, and 270 deg. Furthermore, the 120 database simulations are split into training and validation sets using a 100:20 ratio.

To study the robustness of the training regarding the choice of the training–validation split, a fivefold cross-validation study is performed; i.e., five independent trainings are performed, and the holdout validation data are different for each run. Furthermore, the initial values of the neural network are changed from run to run by varying the random number seed. Repeating the neural network training with slightly different initial conditions allows the collection of statistics during the training and testing phases, enabling us to quantify the variability of the method. In fact, a similar neural network as the present one was found in Ref. [30] to be extremely sensitive to the training conditions (hardware, code optimization, etc.), which results in a large variability in the predictions during the autoregressive prediction phase. By collecting statistics during the training and evaluation phase, it is possible to evaluate the significance of the possible improvements to hyperparameter changes (e.g., influence of input normalization).

The PyTorch open-sourced framework [34] is employed to perform the training operations. The total training time in a NVIDIA Tesla V100 Graphics Processing Unit (GPU) takes 24 wall-clock hours. Table 1 summarizes the parameters employed to train both cases. The resulting training and validation curves are presented in Fig. 6 for both GlobUnet (left, in red) and LocUnet (right, in green). Both training and validation metrics are plotted for both configurations. Because the validation error remains very similar to the training error in all cases, no early stopping was applied; and the training is stopped after the convergence of the loss curve after about 300,000 optimizer steps with a batch size of 32. After an extensive hyperparameter search, the optimum choice of input frames k is found to depend on the employed normalization. Thus, $k = 1$ and $k = 3$ are found optimal for LocUnet and GlobUnet, respectively. A discussion on this choice follows in Sec. VII. In Secs. V and VI, the results for both baselines are compared.

C. Evaluation Metrics

Besides the loss function [Eq. (17)] employed for training the network, some additional metrics are used to assess the accuracy of

Table 1 Summary of the neural network baseline found for each normalization

Name	No. of simulations (training/validation)	Input normalization	k	τ_D^{train}	No. of runs
GlobUnet	120 (100/20)	ρ_0	3	1.44	5
LocUnet	120 (100/20)	$\sigma(\rho_{i-k})$	1	1.44	5

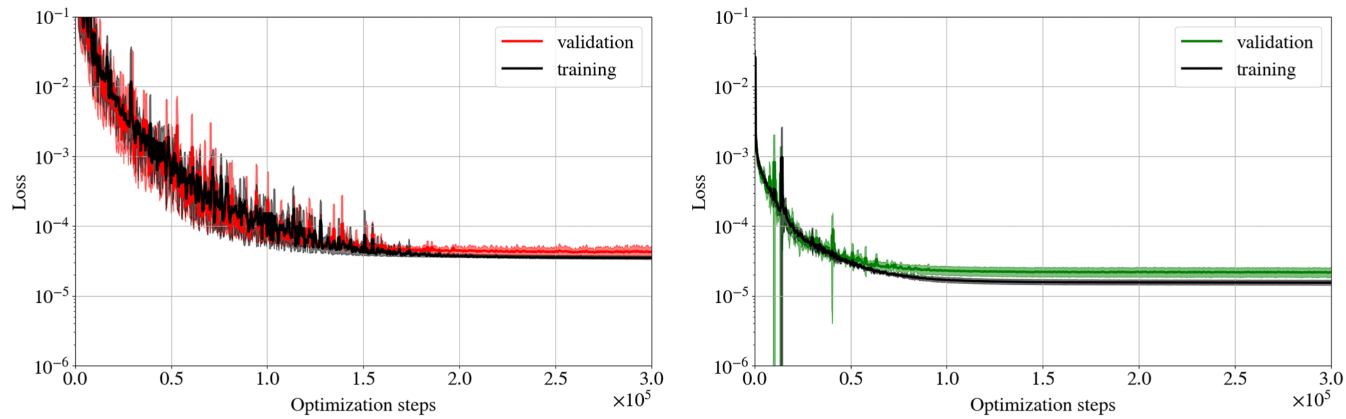

Fig. 6 Evolution of training and validation errors during neural network training for GlobUnet (left) and LocUnet (right). Error curves averaged for five trainings per normalization. Standard deviation depicted with a shadow zone.

the predictions. The gradient difference loss (GDL) error [35] is employed to measure the error performed on the norm of the spatial gradients and can be calculated as follows:

$$\begin{aligned}\mathcal{L}_2(\|\nabla\hat{\rho},\nabla\rho\|) = \frac{1}{N}\sum_{n=1}^{N} \|\{\hat{\rho}_{i+1}^n(x,y)-\hat{\rho}_{i+1}^n(x-\Delta x,y)\} \\ -\{\rho_{i+1}^n(x,y)-\rho_{i+1}^n(x-\Delta x,y)\}\|_2^2 + \|\{\hat{\rho}_{i+1}^n(x,y) \\ -\hat{\rho}_{i+1}^n(x,y-\Delta y)\}-\{\rho_{i+1}^n(x,y)-\rho_{i+1}^n(x,y-\Delta y)\}\|_2^2\end{aligned} \tag{18}$$

A backward finite difference scheme is employed to calculate such an error. Note that in previous works [36], this metric was added to the loss function in combination with the mean-square-error loss [Eq. (17)] in order to constrain the network to directly learn the spatial variations of the data. However, such a strategy was found to degrade the results significantly in the present work, due to the necessity to treat the spatial derivatives at obstacle boundaries. This results in noisy signals for the spatial gradients that lead to overfitting. Future works should deal with a careful treatment of spatial gradients near the solid boundaries.

Additionally, another metric is employed: namely, the acoustic density energy, which is related to the spatially integrated root mean square of the density signal:

$$E_\rho(t) = \sqrt{\int_\Omega \frac{\rho^2}{2}\,\mathrm{d}\Omega(t)} \tag{19}$$

Such an energy represents a portion of the total acoustic energy, with the other part being linked to the fluctuating velocity. Such a fluctuating velocity could be calculated by integrating in time the linearized momentum equation (possible with two or more time steps of fluctuating density or pressure, i.e., $k > 0$).

Finally, in order to obtain a representative value for a full rollout of the prediction in time, three additional time-integrated metrics are employed, namely, the integrated mean square error (IMSE) over time, which is defined as

$$\mathcal{E}(\hat{\rho},\rho) = \frac{1}{T}\int_0^T \mathcal{L}_2(\hat{\rho},\rho)(t)\,\mathrm{d}t \tag{20}$$

the integrated gradient difference loss (IGDL)

$$\mathcal{E}(\|\nabla\hat{\rho},\nabla\rho\|) = \frac{1}{T}\int_0^T \mathcal{L}_2(\|\nabla\hat{\rho},\nabla\rho\|)(t)\,\mathrm{d}t \tag{21}$$

and the integrated energy error (IEE)

$$\mathcal{E}(E_{\hat{\rho}},E_\rho) = \frac{1}{T}\int_0^T \|E_{\hat{\rho}}-E_\rho\|_2^2(t)\,\mathrm{d}t \tag{22}$$

Such metrics are employed in Sec. VII to study the influence of different network hyperparameters in the prediction error.

V. Density-Field Prediction in Unbounded Test Cases

This section presents the results for the first four tests (test cases a–d), defined in Sec. V. All available networks (five runs per normalization strategy, with 10 runs in total) are evaluated for the different test cases presented next, which are representative of typical benchmarks in computational acoustics with nonreflecting boundary conditions.

A. Free-Field Propagation

The initial condition for the open bounded case (Fig. 5a) corresponds to a Gaussian pulse of half-width of $b/D = 0.06$, which is located in the center of the domain. The neural network is provided with the first two frames ($k = 1$) of the simulation, which are issued from a LBM simulation with the same boundary and initial conditions; and the autoregressive method subsequently unrolls the complete prediction until the nondimensional time $\tau = 1.7$ is reached.

Snapshots at several times are shown in Fig. 7 for the two studied normalization strategies and compared with the reference LBM. The results show that all neural networks perform well during all the phases of propagation, even when the pulse interacts with the nonreflective boundary. More quantitative results are shown in Fig. 8, where a slice at constant $y/D = 0.5$ is extracted. For each normalization strategy, both the mean and standard deviation (shaded area) are depicted. The reference LBM solution is plotted with white circles. The LocUnet solutions seem to be less prone to variability at later times of the propagation, with a lower deviation in the error than GlobUnet. This can be confirmed by inspecting the three quantities described in Sec. IV.C, namely, the evolution of the mean square error $\mathcal{L}_2$ (relative to the initial pulse amplitude ρ_0), the evolution of the gradient difference loss $\mathcal{L}_2(\|\nabla\hat{\rho},\nabla\rho\|)$ (relative to the norm of the spatial gradients at t=0) $\|\nabla\rho_0\|$, and the evolution of the acoustic pressure energy over time E_ρ. Figure 9 shows the evolution of these metrics over time, depicting the mean and standard deviation (error bars) for the five different runs per normalization. Three regimes in the error behavior are observed: one with constant energy levels, where the pulse has not yet reached the boundary ($\tau < 0.5$); the time of evacuation of the acoustic signal through the nonreflecting boundary conditions ($0.5 < \tau < 0.7$); and finally the time where only a fraction of the initial signal energy remains in the domain ($\tau > 0.7$). In the first regime, both normalization approaches show similar trends, even though the LocUnet strategy has a small advantage in terms of the mean squared error and the spatial gradient error. Then, both networks stabilize in very similar error levels (at $\tau = 0.7$). After the pulse has completely left the computational domain, after $\tau = 0.8$, the GlobUnet prediction starts to drift in terms of energy levels, resulting in a significant increase of error accumulation and variability within runs. It even leads to divergence of the predictions

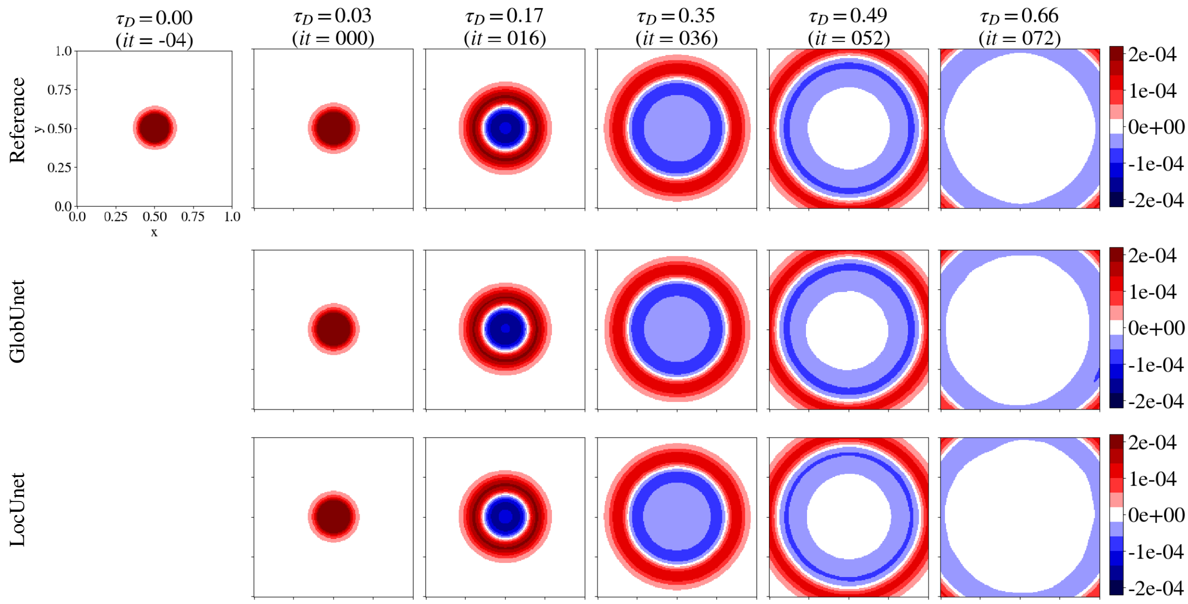

Fig. 7 Acoustic density prediction for test case a for propagation of a Gaussian pulse in free field.

in some runs. On the other hand, the use of the local normalization mitigates such an error accumulation over time, even at long times of simulations when almost no signal is still in the domain (see $\tau = 1.73$ in Fig. 8). Furthermore, it manages to reach a very similar energy level across all the runs, indicating a stable rollout in time, even though a small offset in energy levels is observed.

These observations suggest that employing the local normalization is key in obtaining a stable autoregressive prediction capability, with low variability between runs. This first test highlights one of the key challenges of the application of data-driven methods to problems that switch from energy-preserving to energy-decaying dynamics, eventually reaching low residual levels in the domain and increased noise-to-signal ratios. The normalization of the model inputs plays an important role in avoiding overfitting to only one of the regimes and learning the actual wave operator that remains identical in all cases (i.e., the propagation of waves in free field).

B. Scattering by a Cylinder

Next, as in the validation case of the dataset in Sec. III.B, the scattering of a Gaussian pulse by a cylinder is studied (test case b). The computational domain is sketched in Fig. 5b: the initial condition is an initial Gaussian pulse of half-width of $b/D = 0.06$, which is located at a distance $x/D = 0.25$ from the cylinder of radius $R/D = 0.075$, located in the center of the domain; whereas the four boundaries are left as nonreflecting conditions.

The results for such a test case are shown in Fig. 10. The density fields of both the reference LBM simulation and the neural network

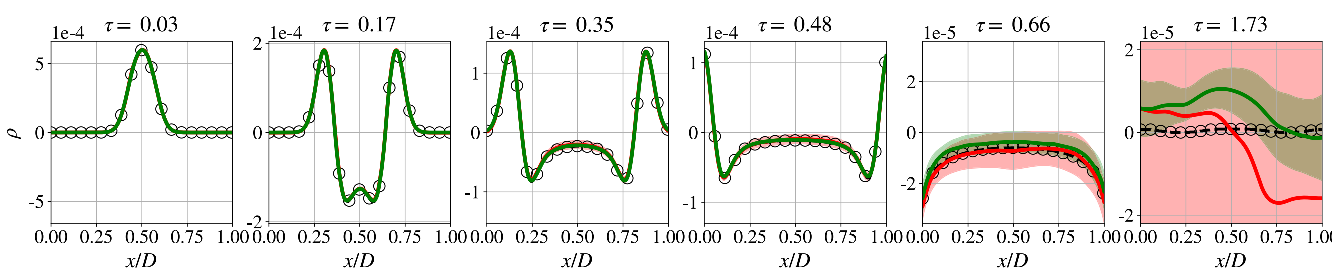

Fig. 8 Slices of density fields at $y/D = 0.5$ for test case a for different times and networks: LBM reference (circles), GlobUnet runs (red), and LocUnet runs (green). Scale changes between time steps for improved visualization.

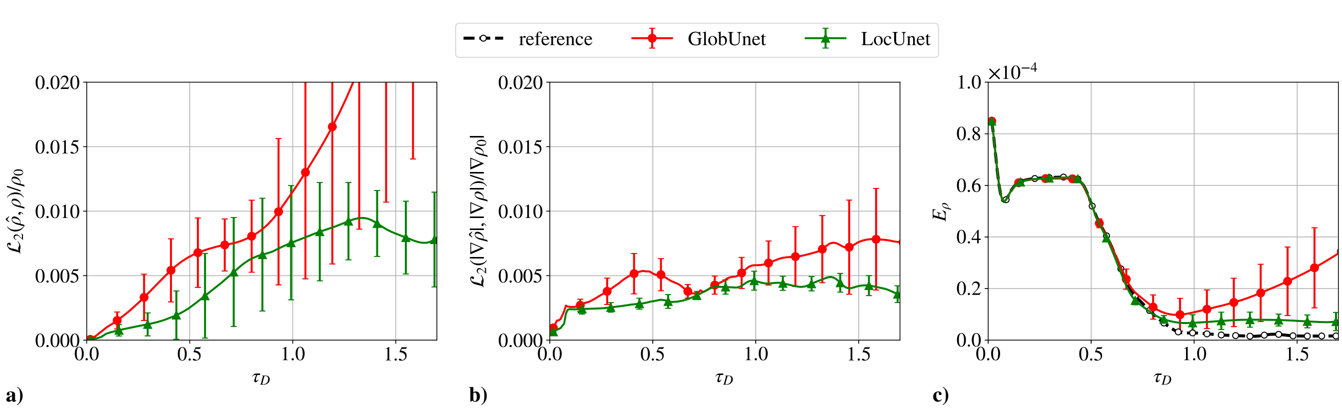

Fig. 9 Evolution of mean square error (left), gradient-difference error (center), and acoustic density energy (right) for test case a (free-field propagation).

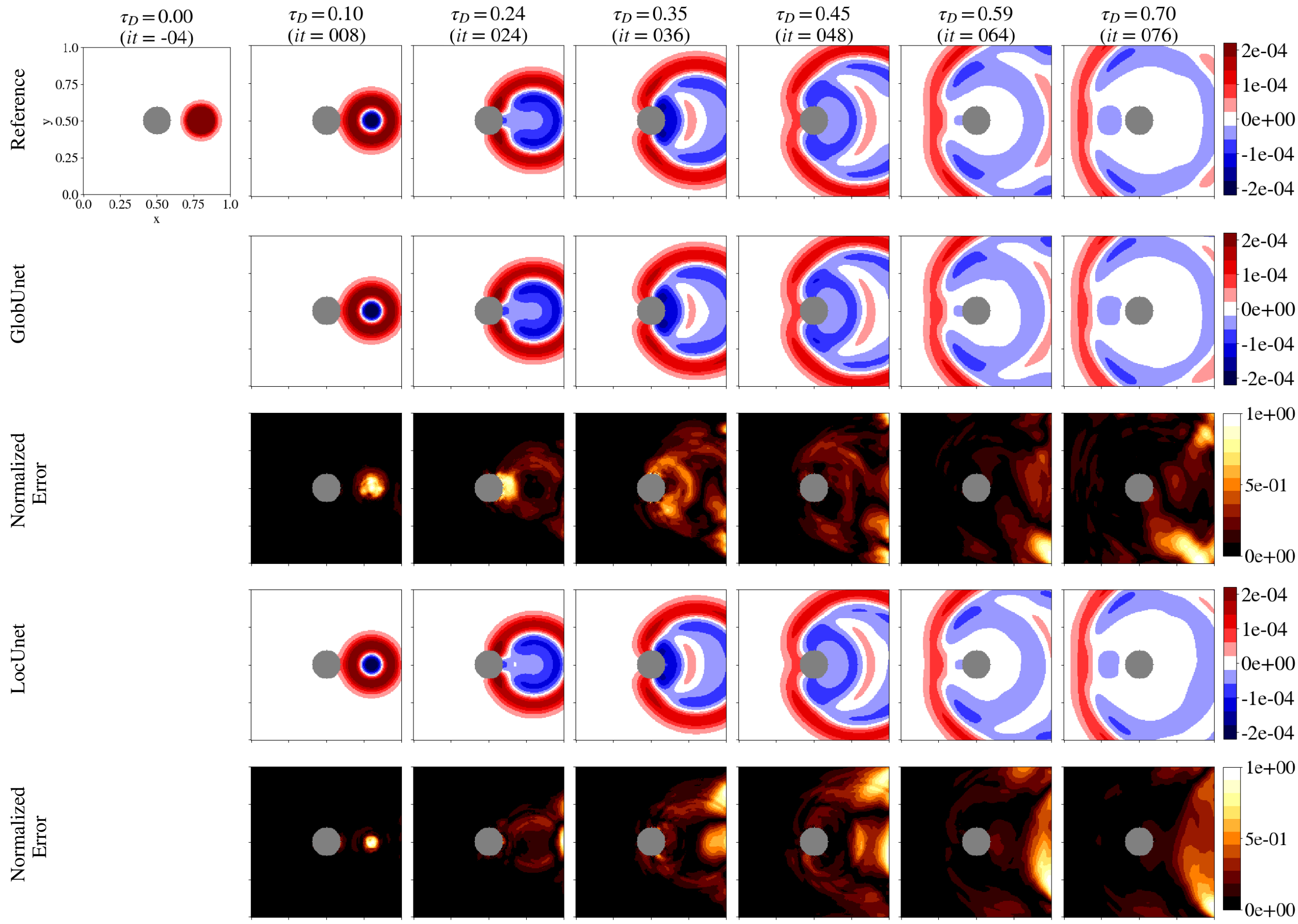

Fig. 10 Acoustic density prediction for test case b with cylindrical obstacles for different times as compared with reference LBM target (first row) and normalized spatial errors.

predictions are compared, along with the local error normalized by the L_∞ norm of the error, in order to show where the error is concentrated. The challenge for the neural network is to correctly predict the scattering of the pulse by the cylinder, as well as to correctly model the nonreflecting boundary conditions, as was already discussed for the free-field propagation case. Slices of the density fields at two positions of $y/D = 0.5$ and $y/D = 0.25$ are shown in Fig. 11, as well as the evolution of the global metrics in Fig. 12. For all four tested networks, a similar behavior is observed as in the previous test: the error grows initially next to the cylinder wall, whereas at later times, the regions of high error are concentrated near the boundaries. The LocUnet again achieves a lower error than GlobUnet with similar behavior as the one observed for the free-field propagation (see Sec. V.A), which is typically for times after $\tau = 1$.

Both strategies exhibit a concentration of error by the rightmost domain boundary at $x/D = 1$ at times around $\tau = 0.7$, when the wave reflected by the cylinder reaches that boundary. Nonetheless, the LocUnet manages to closely follow the signal oscillations even for such long-time horizons, and GlobUnet only manages to capture the high-amplitude signals at early simulation times; however, it fails for the lower-amplitude peaks after $\tau \simeq 1.0$. This suggests that the global scaling forces the network to focus on high-amplitude signals, whereas lower-amplitude ones have been neglected during the training process. Thus, it demonstrates the superiority of the local normalization, which can adapt to varying signal amplitudes and capture the energy-decaying dynamics accurately.

C. Scattering by a NACA0020 Airfoil

The third test case corresponds to the scattering of a Gaussian pulse wave with a NACA0020 airfoil. The computational domain is sketched in Fig. 5c. The airfoil leading edge is located at coordinate $(x, y) = (0.3D, 0.5D)$, whereas the chord and thickness are $0.4D$ and $0.08D$, respectively. The initial acoustic source is a Gaussian pulse located at $(x, y) = (0.5D, 0.8D)$, above the airfoil trailing edge. This

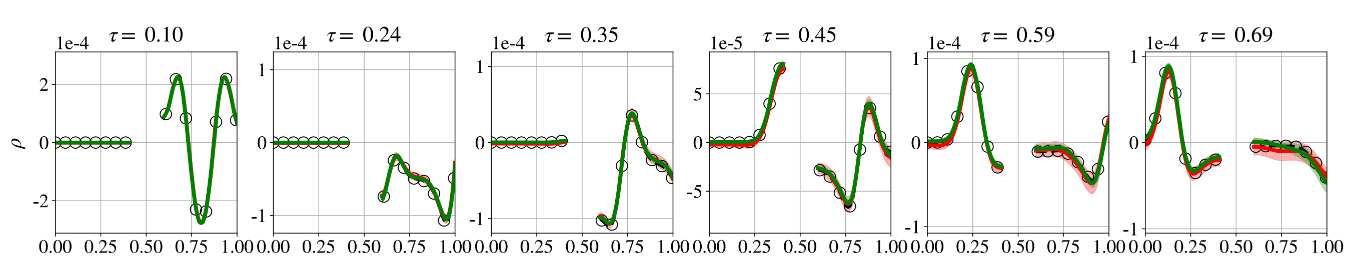

Fig. 11 Slice of density fields at different times at $y/D = 0.5$ for networks: LBM reference (circles), GlobUnet (red), and LocUnet (green) for test case b.

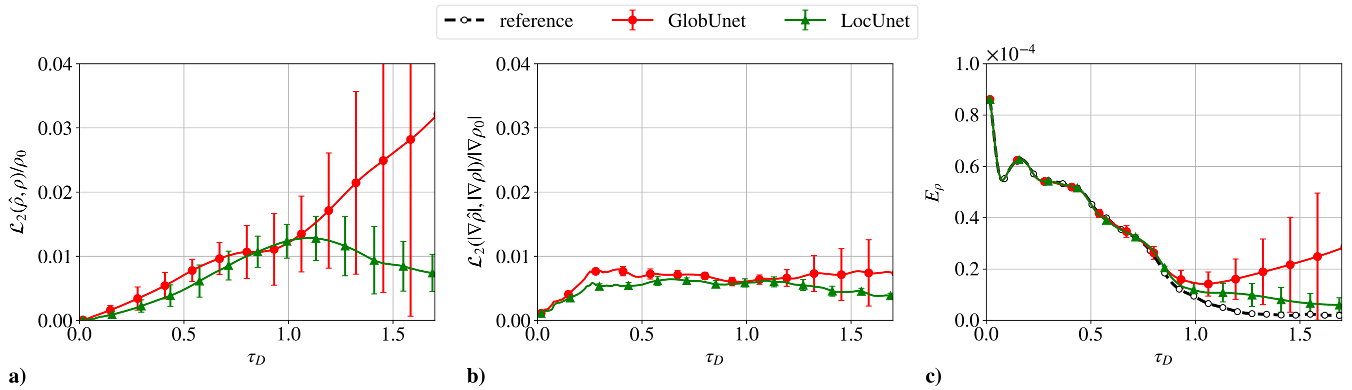

Fig. 12 Evolution of mean square error (left), gradient-difference error (center), and acoustic density energy (right) for test case b (scattering with cylinder).

is the first test case where the network encounters a geometry that was not present in the database. The purpose is to study the ability of the network to predict the strong scattering typical of sharp edges, as in the airfoil trailing edge.

The results in Figs. 13 and 14 show that for low times of propagation ($\tau < 0.4$), the error concentrates in the trailing-edge region: as seen, for example, in the slices for $\tau = 0.1$ and $\tau = 0.31$, both GlobUnet and LocUnet do not closely fit the density levels at the trailing edge ($x/D = 0.7$), indicating the difficulty of the network to handle sharp edges, which it was not trained for. Nonetheless, the local error performed by LocUnet is smaller than the GlobUnet one, which has a large variability in this region. The leading edge is, on the contrary, well predicted. This offset in density creates some dispersion phenomena visible in the error fields of Fig. 13: the backscattered wave from the trailing edge is not in phase with the reference simulation. Interestingly, the prediction error near the sharp trailing edge has a pattern similar to a dipolar source term. A better understanding of this localized error and possible corrections are left for future work.

D. Scattering by a Two-Dimensional Nacelle

To demonstrate the capabilities of the presented framework in an industrial-relevant case, a simplified duct-nacelle configuration is tested, as shown in Fig. 5d. Each side of the nacelle is modeled as a symmetric NACA airfoil at a zero angle of attack with respect to the horizontal direction. Both airfoil leading edges are located at

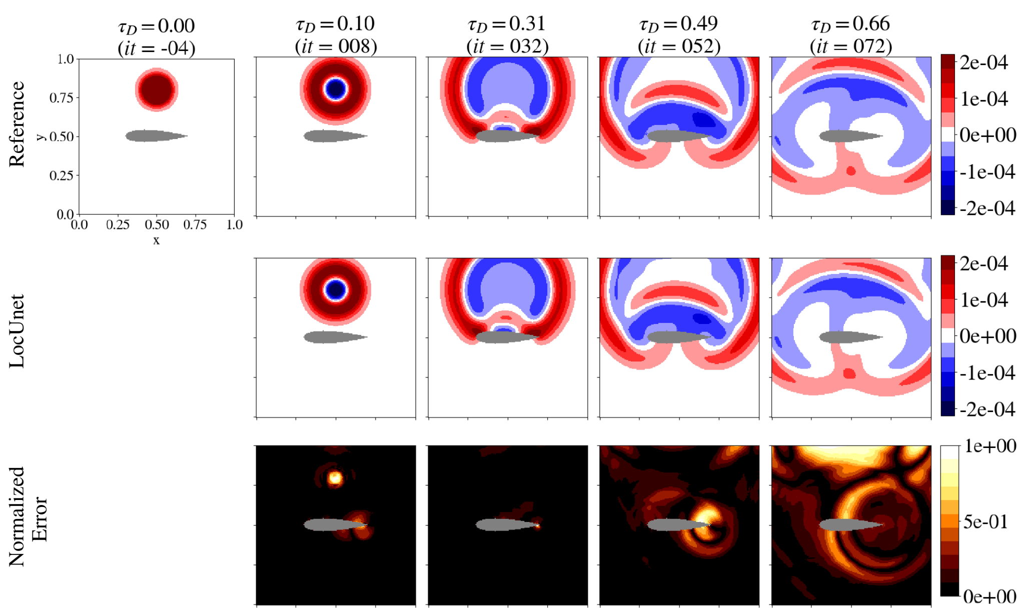

Fig. 13 Acoustic density prediction for test case c (scattering by a NACA0020 profile).

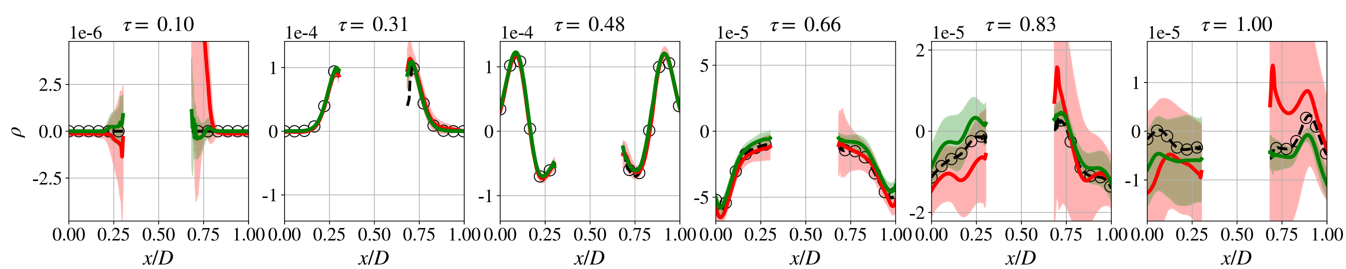

Fig. 14 Slices of density fields at $y/D = 0.5$ for test case c (NACA airfoil) for different times and networks: LBM reference (circles), GlobUnet (red), and LocUnet (green).

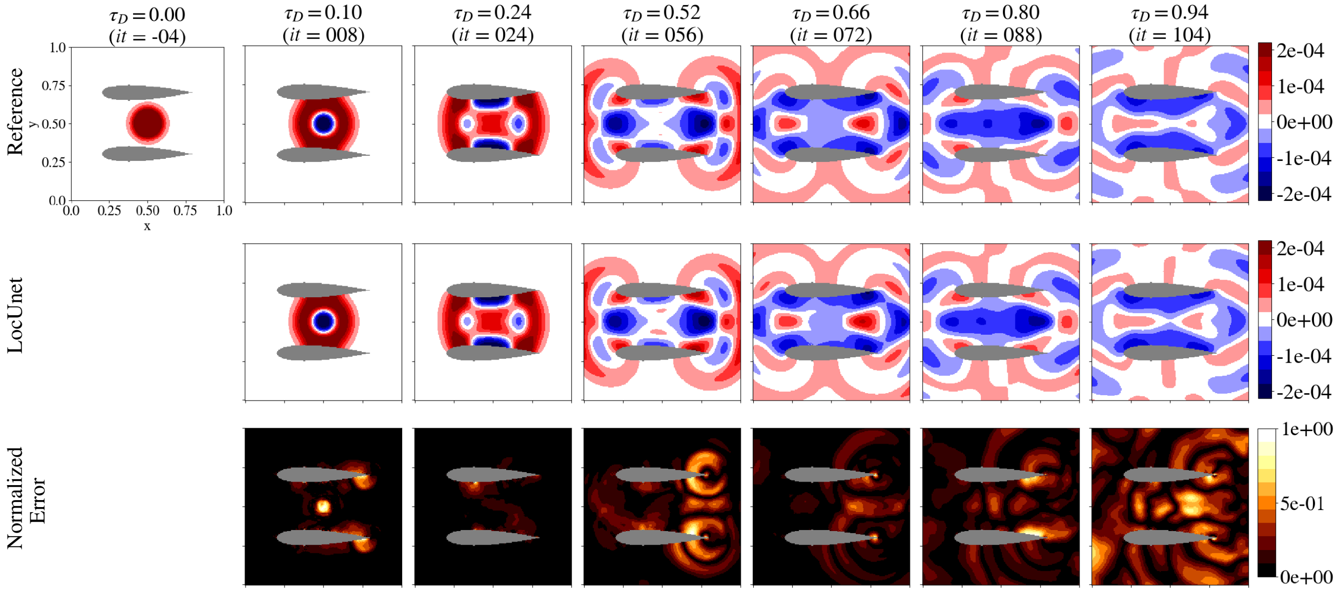

Fig. 15 Acoustic density prediction for test case d with two NACA0020 airfoils forming a nacelle.

coordinates of $(x, y) = (0.2D, 0.3D)$ and $(x, y) = (0.2D, 0.7D)$, whereas the chord and thickness are set at $0.6D$ and $0.12D$, respectively. A Gaussian pulse source is initially located inside the duct at $(x, y) = (0.5D, 0.5D)$, and the neural network is unrolled to propagate the wave through the duct and into the far field.

This test case contains most of the physics studied in this work: wave reflection, duct acoustics, and scattering of waves at both ends of the duct. The results shown in Fig. 15 reveal the complex patterns that appear in such a case: axial-traveling waves propagate both upstream and downstream of the duct, whereas transverse waves remain trapped inside the duct, forcing the appearance of ductlike modes. Both GlobUnet and LocUnet manage to predict the density patterns appearing in such a scenario accurately. A slice along the axis of the nacelle ($y/D = 0.5$) (Fig. 16) shows the overall fitting of the LBM reference, even after the trapped modes become established after a few iterations ($\tau > 0.24$). The evolution of the acoustic energy shown in Fig. 17 stresses the presence of these trapped, duct-like modes through the periodic oscillation of energy and fluxes. Both neural networks show signs of phase shifts because they do not follow the oscillatory patterns after $\tau > 0.5$. However, LocUnet manages to follow the global trend of the oscillating energy, whereas the GlobUnet diverges faster from the reference simulation.

As seen in Sec. V.C, the scattering by both trailing edges creates a local error accumulation, which creates a phase shift in the subsequent propagation. This is more noticeable for GlobUnet, because the GDL error peaks at some discrete times of $\tau = 0.4$, 0.6, and 1, which correspond to the times at which the formed acoustic modes scatter symmetrically at both trailing edges. The error created at such

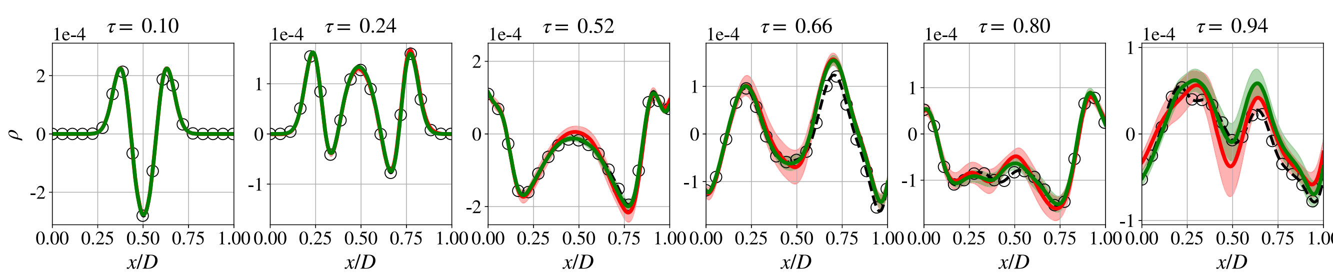

Fig. 16 Slices of density fields at $y/D = 0.5$ for test case d (nacelle) for different times and networks: LBM reference (circles), GlobUnet (red), and LocUnet (green).

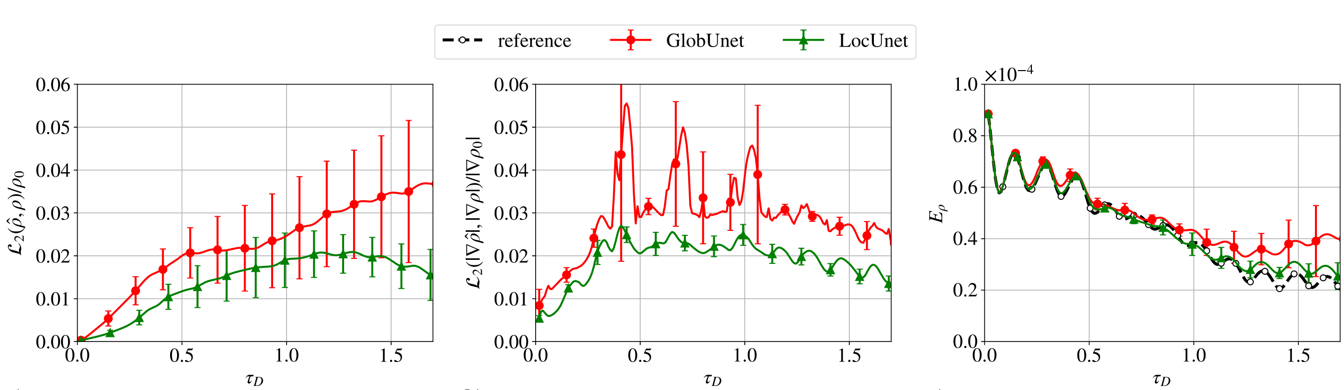

Fig. 17 Evolution of mean square error (left), gradient-difference error (center), and acoustic density energy (right) for test case d (nacelle).

moments then propagates and pollutes the rest of the time predictions. Thus, this test highlights that the studied method is able to accurately predict complex acoustic propagation cases that differ significantly from the training database. It also shows two of the main difficulties encountered by the surrogate model: the scattering with sharp edges, and the difficulty to predict in-duct propagation because of the presence of trapped modes. The scattering seems to be the cause of the in-duct propagation discrepancy because it generates dispersion errors.

VI. Acoustic Prediction in Bounded Domain

A. In-Duct Propagation

To complete the observations made in the previous case regarding the difficulty of neural networks to tackle the prediction of in-duct wave propagation, a fifth test case is presented, where the propagation of a Gaussian pulse inside a two-dimensional duct of constant diameter is considered. It allows the study of the long-time propagation by the neural network of trapped modes without taking into account difficulties due to sharp edges. Three configurations are studied, as shown in Fig. 5e. The size of the domain is kept constant at $D \times D$, whereas three duct diameters h are investigated: $h/D = 0.4$, 0.6, and 0.8.

Figure 18 compares the results for LocUnet and the LBM reference. The initial bouncing on the duct walls is seen to be well predicted ($\tau = 0.38$). However, once the wave fronts simultaneously reach the four edges of the simulation, where the nonreflecting condition coincides with the duct wall, the error accumulates quickly in that region. This behavior creates some spurious backreflected waves that pollute the entire domain. This mismatch creates a dispersion error that can be observed in the energy evolution for the three studied cases (Fig. 19). In the three cases, the LocUnet method demonstrates its superiority because it manages to track the energy evolution much closer than the globally scaled network. However, the phase dispersion is also visible in both cases, which is an effect of the error created by the edges of the simulation. These can be attributed to the fact that such a type of junction between nonreflecting and wall boundary conditions is never seen during training. Thus, it is hard for the network to infer the behavior at such points. Here, it is hypothesized that even if the duct walls are of infinite length in the axial direction, the network has always learned on finite-length obstacles. Thus, it may be possible that it sees the duct end as an open end, thus creating some reflected waves because of the wave scattering at a duct opening. To mitigate this behavior, the dataset could be augmented with some examples of infinite length ducts.

Furthermore, the analysis of the energy evolution over time in Fig. 19 reveals that the networks reduce their accuracy when the duct aspect ratio is decreased. The smaller the h/D becomes, the smaller the wavelength λ of the associated acoustic mode is, thus increasing the frequency ω of the acoustic signal through the dispersion relation

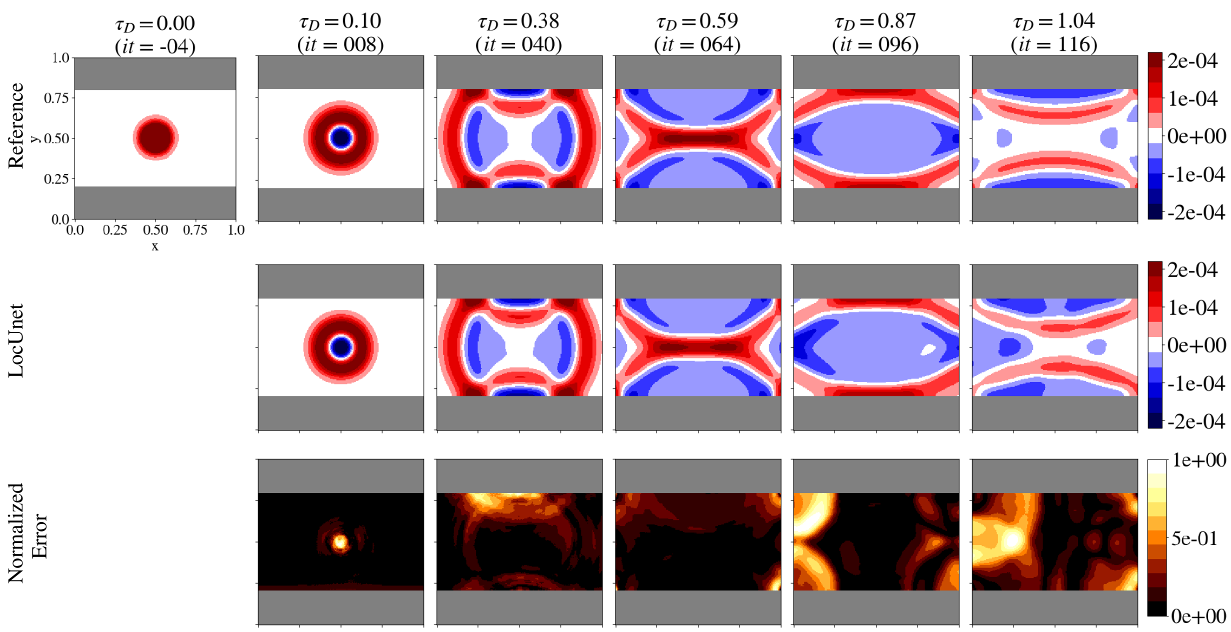

Fig. 18 Acoustic density prediction for test case e for in-duct wave propagation with $h/D = 0.6$.

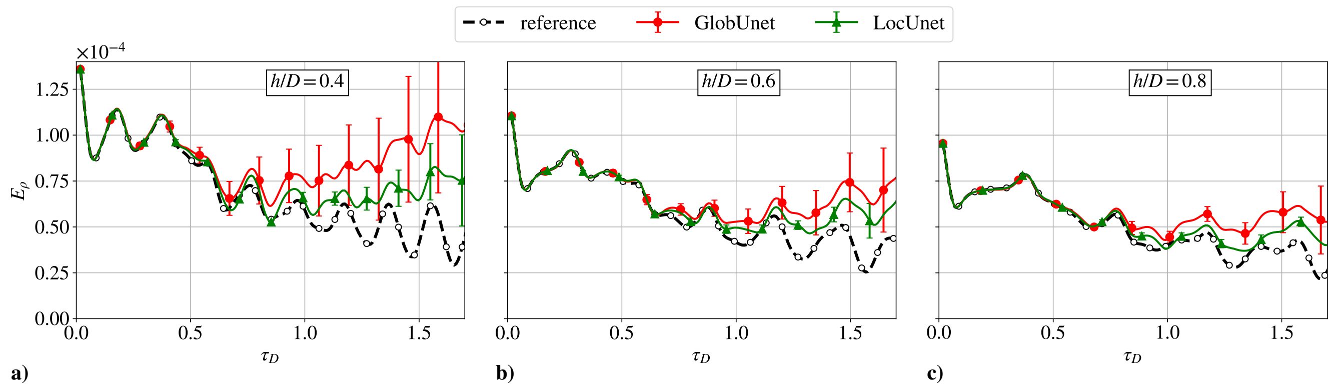

Fig. 19 Evolution of acoustic density energy for test case e: a) $h/D = 0.4$, b) $h/D = 0.6$, and c) $h/D = 0.8$ (in-duct propagation).

$\omega = 2\pi c_0/\lambda$. Thus, such results suggest that the trained networks are able to predict low-frequency signals more accurately than high-frequency ones.

B. Closed Box Propagation

The last test case corresponds to the same type of boundary condition studied in Ref. [13], namely, a hard-reflecting wall on the four boundaries of the computational domain. The initial condition corresponds to a Gaussian pulse centered at $(x, y) = (0.5D, 0.5D)$. In practice, the neural network is given a geometry mask with one continuous layer of "obstacle" pixels around the computational domain.

The results in Fig. 20 show the evolution of the neural network predictions for the LocUnet network. The network manages to reproduce the highly symmetric patterns until times around $\tau = 0.6$. For later times, a slight phase shift (originating from the corners) unbalances the symmetry. As the energy content of this problem remains constant, it is a hard problem for networks that have only seen decaying problems in the training database. Nonetheless, the network still manages to keep an accurate level of symmetry, which demonstrates the capability of the training model to operate accurately outside the training distribution.

VII. Influence of Parameters

In the two previous sections, both normalization strategies have been trained with a different set of input frames ($k = 1$ and $k = 3$ for LocUnet and GlobUnet, respectively). In this section, the influence of such parameters on the neural network accuracy is explored. Additionally, the τ_D^{train} parameter (the duration of the dataset simulations) is included in this hyperparameter study because it was found to possibly play a role in stabilizing the long-term autoregressive predictions. Table 2 summarizes the different studied hyperparameters.

Five training runs are again performed for each pair of hyperparameters. To evaluate their overall effect on autoregressive performance, the integral quantities presented in Sec. IV.C (namely the IMSE, IGDL, and IEE) are calculated for each of the test cases (test cases a–f). The results are presented in Fig. 21 with the mean and standard deviation for each set of runs. The baseline (best overall performing strategy for each normalization) is hashed. Several conclusions can be extracted from such results:

A. Influence of Number of Inputs k

Decreasing k seems to benefit LocUnet across all metrics and test cases. Furthermore, the variability between runs also decreases with k, demonstrating the advantage of reducing the number of input frames. It is not clear why this reduction benefits the neural network. Recent works that studied this phenomenon reached opposite conclusions: Pfaff et al. [37] drew similar conclusions as in the current work. Using a single neural network that directly maps the spatio-temporal evolution $t \to t + \Delta t$, they claimed that $k > 1$ leads to overfitting. On the other hand, other works like those of Hasegawa

Table 2 Summary of the hyperparameter search

Parameter	Value
k	$\{1, 3\}$
τ_{train}	$\{D_1, D_2\} = \{0.87, 1.44\}$
Normalization	GlobUnet/LocUnet

et al. [19] found that increasing k benefits the long-term accuracy of the predictions. Different from the present work, they separated the problem into a dimensionality reduction network and a learned recurrent model that integrates the low-dimensional space. Furthermore, they focused on periodic flows such as the wake behind bluff bodies. Thus, this remains an open question that is left as future work.

B. GlobUnet Energy Drift and Energy Correction

The effect of k in GlobUnet is less clear: for the IMSE and IEE metrics, decreasing k does not improve the results in general; whereas it improves the IGDL. Such an erratic behavior confirms the observation made in Secs. V and VI: because GlobUnet focuses on high-energy signals, the energy and MSE metrics tend to diverge after reaching the high noise-to-signal ratio regime. However, the GDL improves, indicating that the main problem of GlobUnet is the drift in mean signal levels (energy). A possible remedy is to constrain the network to fulfill the acoustic energy conservation over time. In a previous work [12], the authors employed an energy-preserving correction in order to correct such an energy drift. However, it relied on the assumption that the drift was spatially uniform, which allowed the derivation of an analytical expression to correct the network a posteriori (i.e., after the training phase). Such a possibility was explored in the current work, but the hypothesis of uniform drift no longer holds for the trained networks (LocUnet or GlobUnet). The use of a GDL loss in Ref. [12] could be the cause for obtaining such a uniform drift; whereas in the present work, such a loss results in high training errors due to the overfitting of the noisy numerical spatial gradients close to the obstacles boundaries (e.g., sharp edges, such as the trailing edge of the NACA airfoil case).

C. Influence of Simulation Time τ_D^{train}

Two datasets with different simulations times have been tested: D_1 and D_2 with $\tau_D^{\text{train}} = 0.87$ and $\tau_D^{\text{train}} = 1.44$, respectively. Increasing the time of simulation seems to consistently reduce the error with the LocUnet normalization in terms of the IMSE, IGDL, and IEE. It also benefits the GlobUnet normalization consistently across the test cases. This suggests that the neural network performance improves with a dataset that captures the full range of the studied dynamics accurately. The choice of dataset becomes a key design parameter when using data-driven methods. Undersampling the problem dynamics could lead to the so-called distribution shift [38] that affects the generalization performance outside the training range.

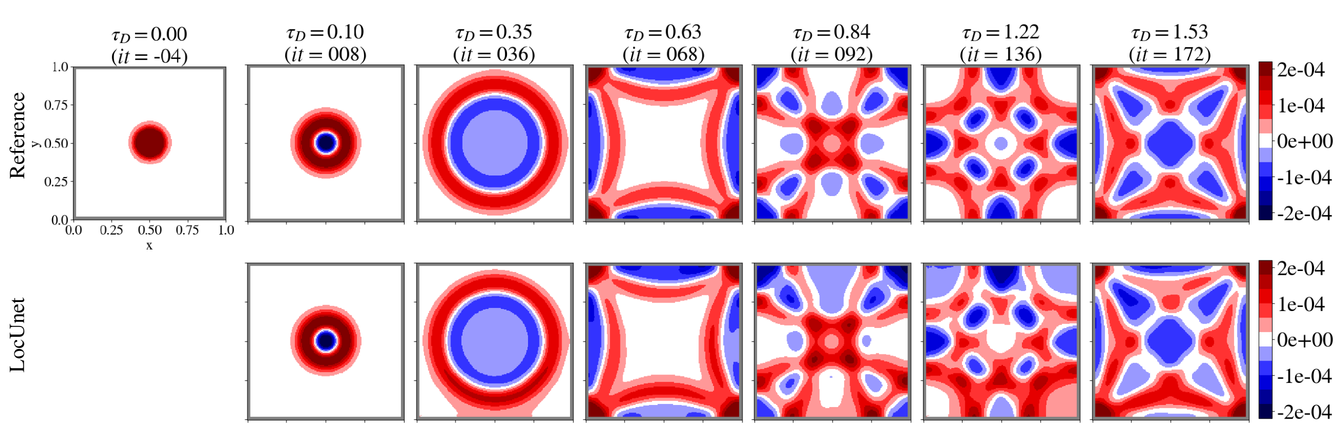

Fig. 20 Acoustic density prediction for test case f of closed-box wave propagation.

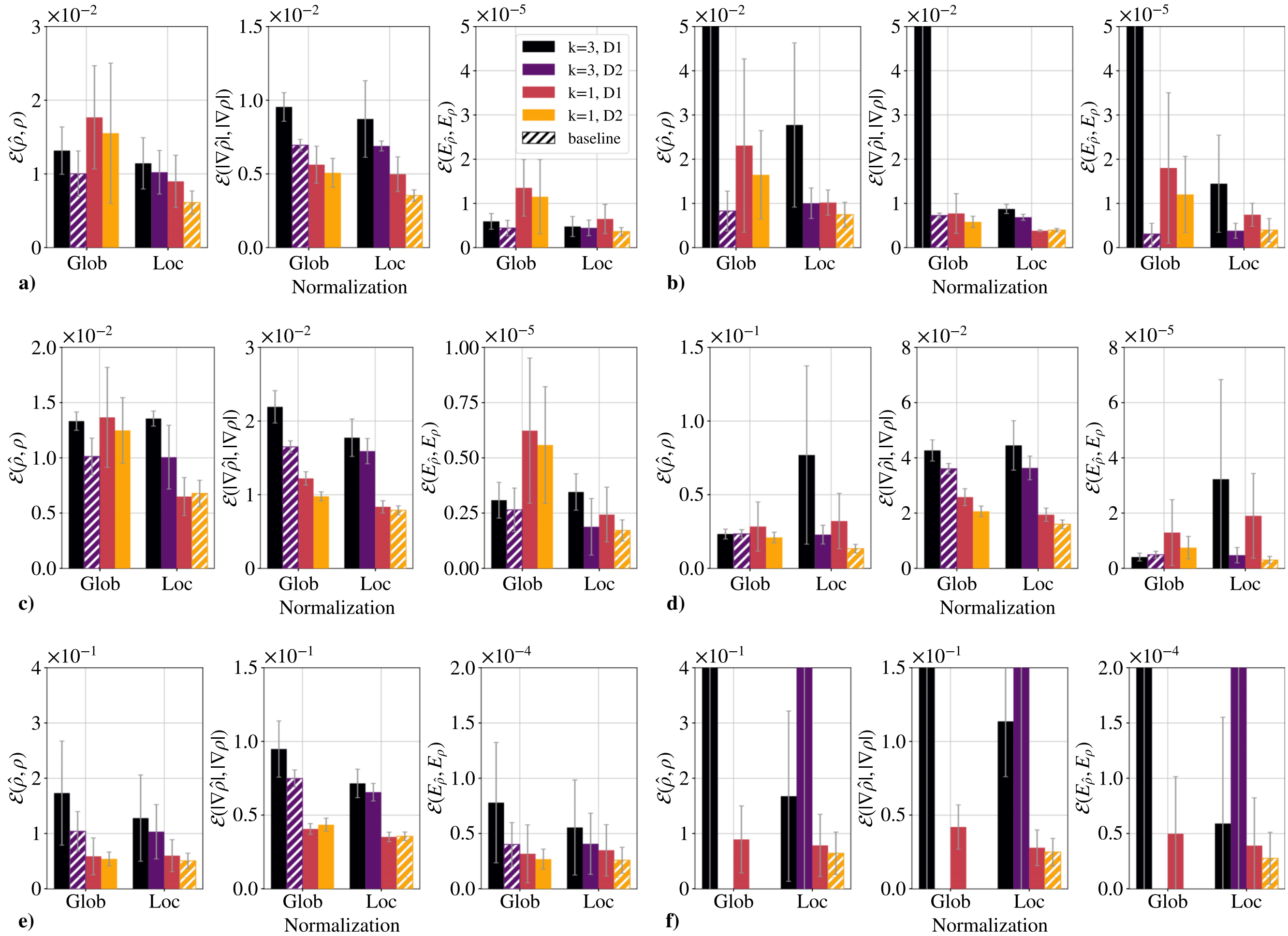

Fig. 21 Integral error measures: IMSE (left), IGDL (center), and IEE (right) for test cases a) open bound, b) cylinder, c) isolated airfoil, d) nacelle, e) ductlike with $h/D = 0.6$, and f) closed box. Employed baselines are hashed, corresponding to results depicted in Secs. V and VI.

VIII. Computational Performance

The computational costs associated with the presented method are presented next. Such costs can be divided into three main phases: the database generation; the training of the neural network; and the inference phase, where the trained network is employed in an autoregessive manner. In the latter phase, these costs are compared to the cost of performing a direct lattice-Boltzmann simulation.

The database and training costs are presented in Tables 3 and 4. Both phases correspond to an offline phase of the deep learning pipeline, which must be performed only once (in the case in which the hyperparameters, such as the dataset size or optimizer learning rate, are already optimal). It can be seen that the most costly phase remains the training, which must be performed with a hardware accelerator (GPU). Note that the cost of database generation remains limited due to the 2-D nature of data. In the case of three-dimensional simulations, this cost would increase significantly, along with the training cost.

Table 3 Computational cost of data generation

Hardware	Wall time per iteration, ms	Wall time per simulation, s	Wall time per database, s
CPU Intel Gold Xeon 6126	8.922	4.461	535

Table 4 Computational cost of data generation and training of a neural network

Hardware	Wall time per epoch, s	Wall time per training, days
GPU NVIDIA Tesla V100 32 GB	0.014293	1

For the inference phase, Table 5 presents a comparison similar to the one performed in Ref. [12]: the cost of one time-step prediction is compared between CPU and GPU hardware for the baseline serial LBM code (baseline 1) and the implemented U-net neural network; and the time to reach a fixed nondimensional time is $\tau = 2.88$. A second reference for a Message Passing Interface (MPI)-parallel run of the LBM code in a single-node 24-core machine is also presented (baseline 2). The acceleration factors with respect to both references are written in the two last columns. For a time step of the neural network of $\Delta t = \Delta_{\mathrm{LBM}}$, and using the same CPU hardware, the neural network is slower than the LBM reference (even more with respect to the parallel reference). However, as was already shown in Ref. [12], several strategies can be followed to speed up the computations. First, using a GPU accelerator manages a speedup of more than 12 times with respect to baseline 1 while still being slower than baseline 2. Further accelerations can be attained by processing several initial conditions treated by the neural network in parallel (i.e., batched simulations; 47 times for a batch size of 64). Also, modifying the time step of the neural network to $\Delta t = 3\Delta_{\mathrm{LBM}}$ gives a significant speedup with respective to baseline 2 (2.5 times). This strategy, which was already presented in Ref. [12], allows the relaxation of stability criteria on the time step (e.g., the CFL number) through learning the solution of the wave equation instead of using explicit numerical time steppers. When both strategies are combined, a large acceleration factor of 141 can be achieved with respect to the MPI-based simulation.

Table 5 Comparison of computational cost for reference LBM code and neural network tested on several hardware and with different hyperparameters[a]

Method	Time step; $\Delta t/\Delta_{\mathrm{LBM}}$	Batch size	Hardware	Wall time per iteration per batch size, ms	Wall time to $\tau_D = 2.88$, s	Acceleration factor (baseline 1)	Acceleration factor (baseline 2)
Baseline 1: serial LBM	1	1	CPU	19.9	19.9	1.0	——
Baseline 2: MPI-LBM 24 procs	1	1	CPU	1.40	1.40	——	1.0
U-net	1	1	CPU	25.8	25.8	0.77	0.05
U-net	1	1	GPU	1.64	1.64	12	0.85
U-net	3	1	GPU	1.64	0.546	36	2.5
U-net	1	8	GPU	0.205	0.204	97	6.8
U-net	1	64	GPU	0.0296	0.0296	672	47
U-net	3	64	GPU	0.0296	0.00988	2014	141

[a]For a batch size greater than one, batch size simultaneous predictions are performed.

IX. Conclusions

In this work, a method to deal with a variety of acoustic propagation cases in quiescent media with a deep learning surrogate model is presented. The neural networks are trained on a single database of high-fidelity LBM simulations that contain examples of acoustic wave propagation, reflection and scattering with obstacles, as well as free-field propagation. Multiple test cases demonstrate that a convolutional neural network trained on such a database, with some specific normalization (namely, local normalization by the input standard deviation), manages to closely reproduce the results of the LBM reference. Accurate results are obtained even on boundary conditions and scattering configurations not seen during the learning phase. The use of a locally normalized input significantly improves the results and reduces the long-term error accumulation of the autoregressive method. The different examples also highlight some of the challenges encountered by the network, which are also typical of traditional numerical solvers: the accurate treatment of nonreflective boundary conditions, the scattering of acoustic waves by sharp edges, and the prediction of duct modes in confined configurations. They also demonstrate that these highly efficient nonlinear networks may be capable of extrapolating to changes in the underlying statistical data distribution (e.g., energy-decaying versus energy-conserving flows).

Although neural network techniques for fluid dynamics and aeroacoustics are in their early phases of development, studying their capabilities in more complex cases remains a crucial step for their applicability to real-world problems. The insights gained in this work suggest that, besides the studied problem of treating the temporal integration accuracy [14], care should be taken when modeling problems with complex boundary conditions. Thus, techniques from the CFD community could be employed to improve the capabilities of such data-driven methods (e.g., nonreflective conditions, etc.).

Acknowledgments

This work was partly supported by the French “Programme d’Investissements D’avenir”: ANR-17-EURE-0005. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada through the Discovery Grant program (RGPIN-2019-1512). M. Bauerheim is also supported by the French Direction Générale de l’Armement through the Agence de l’Innovation de Défense (AID) POLA3 project. The authors wish to thank Wagner Gonçalves Pinto from the National Higher French Institute of Aeronautics and Space for fruitful discussions on the topic. The simulations were carried out using the Calcul en Midi-Pyrénées - CALMIP - supercomputing facilities (project P20035) located in Toulouse, France as well as the Compute Canada clusters (project kfy-564-ae).

References

[1] Lele, S. K., and Nichols, J. W., “A Second Golden Age of Aeroacoustics?” *Philosophical Transactions of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences*, Vol. 372, No. 2022, 2014, pp. 1–18.
https://doi.org/10.1098/rsta.2013.0321

[2] Moreau, S., “The Third Golden Age of Aeroacoustics,” *Physics of Fluids*, Vol. 34, No. 3, 2022.
https://doi.org/10.1063/5.0084060

[3] Lilley, G. M., “On the Noise from Jets. Noise Mechanism,” AGARD-CP-131, Neuilly-Sur-Seine, France, 1974, pp. 13.1–13.12.

[4] Ffowcs Williams, J. E., and Hawkings, D. L., “Sound Generation by Turbulence and Surfaces in Arbitrary Motion,” *Philosophical Transactions of the Royal Society of London, Series A: Mathematical, Physical and Engineering Sciences*, Vol. 264, No. 1151, 1969, pp. 321–342.
https://doi.org/10.1098/rsta.1969.0031

[5] Posson, H., and Peake, N., “The Acoustic Analogy in an Annular Duct with Swirling Mean Flow,” *Journal of Fluid Mechanics*, Vol. 726, July 2013, pp. 439–475.
https://doi.org/10.1017/jfm.2013.210

[6] Singh, A. P., Medida, S., and Duraisamy, K., “Machine-Learning-Augmented Predictive Modeling of Turbulent Separated Flows over Airfoils,” *AIAA Journal*, Vol. 55, No. 7, 2017, pp. 2215–2227.
https://doi.org/10.2514/1.J055595

[7] Thuerey, N., Weißenow, K., Prantl, L., and Hu, X., “Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows,” *AIAA Journal*, Vol. 58, No. 1, 2020, pp. 25–36.
https://doi.org/10.2514/1.J058291

[8] Keane, A. J., and Voutchkov, I. I., “Surrogate Approaches for Aerodynamic Section Performance Modeling,” *AIAA Journal*, Vol. 58, No. 1, 2020, pp. 16–24.
https://doi.org/10.2514/1.J058687

[9] Sekar, V., Zhang, M., Shu, C., and Khoo, B. C., “Inverse Design of Airfoil Using a Deep Convolutional Neural Network,” *AIAA Journal*, Vol. 57, No. 3, 2019, pp. 993–1003.
https://doi.org/10.2514/1.J057894

[10] De Paula, N. C., Marques, F. D., and Silva, W. A., “Volterra Kernels Assessment via Time-Delay Neural Networks for Nonlinear Unsteady Aerodynamic Loading Identification,” *AIAA Journal*, Vol. 57, No. 4, 2019, pp. 1725–1735.
https://doi.org/10.2514/1.J057229

[11] Liu, Z., Ning, F., Zhai, Q., Ding, H., and Wei, J., “A Novel Reduced-Order Model for Predicting Compressible Cavity Flows,” *Journal of Aircraft*, Vol. 59, No. 1, Jan. 2022, pp. 58–72.
https://doi.org/10.2514/1.C036298

[12] Alguacil, A., Bauerheim, M., Jacob, M. C., and Moreau, S., “Predicting the Propagation of Acoustic Waves Using Deep Convolutional Neural Networks,” *Journal of Sound and Vibration*, Vol. 512, Nov. 2021, Paper 116285.
https://doi.org/10.1016/j.jsv.2021.116285

[13] Alguacil, A., Bauerheim, M., Jacob, M. C., and Moreau, S., “Predicting the Propagation of Acoustic Waves Using Deep Convolutional Neural Networks,” *AIAA Aviation Forum*, AIAA Paper 2020-2513, 2020.
https://doi.org/10.2514/6.2020-2513

[14] Geneva, N., and Zabaras, N., “Modeling the Dynamics of PDE Systems with Physics-Constrained Deep Auto-Regressive Networks,” *Journal of Computational Physics*, Vol. 403, Feb. 2020, pp. 1–13, Paper 109056.
https://doi.org/10.1016/j.jcp.2019.109056

[15] Rüttgers, M., Koh, S. R., Jitsev, J., Schröder, W., and Lintermann, A., “Prediction of Acoustic Fields Using a Lattice-Boltzmann Method and Deep Learning,” *International Conference on High Performance Computing 2020*, Vol. 12321, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Science and Business Media, Cham, Switzerland, 2020, pp. 81–101.
https://doi.org/10.1007/978-3-030-59851-8_6

[16] Lino, M., Cantwell, C., Fotiadis, S., Pignatelli, E., and Bharath, A., "Simulating Surface Wave Dynamics with Convolutional Networks," Preprint, submitted 1 Dec. 2020, https://arxiv.org/abs/2012.00718.

[17] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L., "Physics Informed Machine Learning," *Nature Reviews Physics*, Vol. 3, May 2021, pp. 422–440. https://doi.org/10.1038/s42254-021-00314-5

[18] Moseley, B., Markham, A., and Nissen-Meyer, T., "Solving the Wave Equation with Physics-Informed Deep Learning," Preprint, submitted 21 June 2020, https://arxiv.org/abs/2006.11894v1.

[19] Hasegawa, K., Fukami, K., Murata, T., and Fukagata, K., "Machine-Learning-Based Reduced-Order Modeling for Unsteady Flows Around Bluff Bodies of Various Shapes," *Theoretical and Computational Fluid Dynamics*, Vol. 34, No. 4, 2020, pp. 367–383. https://doi.org/10.1007/s00162-020-00528-w

[20] Liu, Y., Nathan Kutz, J., and Brunton, S. L., "Hierarchical Deep Learning of Multiscale Differential Equation Time-Steppers," Preprint, submitted 22 Aug. 2020, https://arxiv.org/abs/2008.09768.

[21] Ronneberger, O., Fischer, P., and Brox, T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*, edited by N. Navab, J. Hornegger, W. Wells, and A. Frangi, Vol. 9351, Lecture Notes in Computer Science, Springer, Cham, Nov. 2015, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

[22] Luo, W., Li, Y., Urtasun, R., and Zemel, R. S., "Understanding the Effective Receptive Field in Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems 29*, edited by D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, Barcelona, Spain, 2016, pp. 4898–4906, https://proceedings.neurips.cc/paper/2016/hash/c8067ad1937f728f51288b3eb986afaa-Abstract.html [retrieved 16 May 2022].

[23] Ajuria Illarramendi, E., Bauerheim, M., and Cuenot, B., "Performance and Accuracy Assessments of an Incompressible Fluid Solver Coupled with a Deep Convolutional Neural Network," *Data-Centric Engineering*, Vol. 3, No. 8, 2022, Paper 2. https://doi.org/10.1017/dce.2022.2

[24] Ajuria Illarramendi, E., Alguacil, A., Bauerheim, M., Misdariis, A., Cuenot, B., and Benazera, E., "Towards an Hybrid Computational Strategy Based on Deep Learning for Incompressible Flows," *AIAA Aviation Forum*, AIAA Paper 2020-3058, 2020. https://doi.org/10.2514/6.2020-3058

[25] Brogi, F., Malaspinas, O., Chopard, B., and Bonadonna, C., "Hermite Regularization of the Lattice Boltzmann Method for Open Source Computational Aeroacoustics," *Journal of the Acoustical Society of America*, Vol. 142, No. 4, 2017, pp. 2332–2345. https://doi.org/10.1121/1.5006900

[26] Marié, S., Ricot, D., and Sagaut, P., "Comparison Between Lattice Boltzmann Method and Navier-Stokes High Order Schemes for Computational Aeroacoustics," *Journal of Computational Physics*, Vol. 228, Oct. 2008, pp. 1056–1070. https://doi.org/10.1016/j.jcp.2008.10.021

[27] Bhatnagar, P. L., Gross, E. P., and Krook, M., "A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems," *Physical Review*, Vol. 94, No. 3, 1954, pp. 511–525. https://doi.org/10.1109/ITCC.2003.1197559

[28] Malaspinas, O., "Increasing Stability and Accuracy of the Lattice Boltzmann Scheme: Recursivity and Regularization," Preprint, submitted 26 May 2015, https://arxiv.org/abs/1505.06900.

[29] Tam, C. K., *Computational Aeroacoustics: A Wave Number Approach*, Cambridge Aerospace Series, Cambridge Univ. Press, Cambridge, MA, 2012, pp. 295–297. https://doi.org/10.1017/CBO9780511802065

[30] Pinto, W. G., Alguacil, A., and Bauerheim, M., "On the Reproducibility of Fully Convolutional Neural Networks for Modeling Time–Space-Evolving Physical Systems," *Data-Centric Engineering*, Vol. 3, Cambridge Univ. Press, Cambridge, England, U.K., 2022, Paper 05482.

[31] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, Cambridge, MA, 2016.

[32] Bengio, Y., and LeCun, Y., "Adam: A Method for Stochastic Optimization," *3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, May 2015, pp. 1–15. https://doi.org/10.48550/arXiv.1412.6980

[33] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D., "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, Vol. 1, No. 4, 1989, pp. 541–551. https://doi.org/10.1162/neco.1989.1.4.541

[34] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., Devito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, Vancouver, BC, Canada, Dec. 2019, pp. 8024–8035.

[35] Bengio, Y., and LeCun, Y., "Deep Multi-Scale Video Prediction Beyond Mean Square Error," *4th International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2016, pp. 1–14. https://doi.org/10.48550/arXiv.1511.05440

[36] Alguacil, A., Bauerheim, M., Jacob, M. C., and Moreau, S., "A Generic Deep Learning Framework for Propagation and Scattering of Acoustic Waves in Quiescent Flows," *AIAA Aviation Forum*, AIAA Paper 2021-2239, 2021. https://doi.org/10.2514/6.2021-2239

[37] Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and Battaglia, P. W., "Learning Mesh-Based Simulation with Graph Networks," *9th International Conference on Learning Representations (ICLR)*, Virtual Event, Austria, May 2021, pp. 1–18. https://doi.org/10.48550/arXiv.2010.03409

[38] Wang, R., Maddix, D., Faloutsos, C., Com, F., Wang, Y., and Yu, R., "Bridging Physics-Based and Data-Driven Modeling for Learning Dynamical Systems," *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, edited by A. Jadbabaie, J. Lygeros, G. J. Pappas, P. Parrilo, B. Recht, C. J. Tomlin, and M. N. Zeilinger, Proceedings of Machine Learning Research, Vol. 144, PMLR, June 2021, pp. 385–398, https://proceedings.mlr.press/v144/wang21a.html.

C. Bailly
Associate Editor