

Prediction of Acoustic Fields using a Lattice-Boltzmann Method and Deep Learning

Mario Rüttgers^{1,2,3}[0000-0003-3917-8407],
Seong-Ryong Koh²[0000-0001-7020-7267], Jenia Jitsev²[0000-0002-1221-7851],
Wolfgang Schröder^{1,3}[0000-0002-3472-1813], and
Andreas Lintermann^{2,3}[0000-0003-3321-6599]

¹ Institute of Aerodynamics and Chair of Fluid Mechanics, RWTH Aachen University, Wüllnerstraße 5a, 52062 Aachen, Germany

² Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Wilhelm-Johnen-Straße, 52425 Jülich, Germany

³ Jülich Aachen Research Alliance Center for Simulation and Data Science, Seffenter Weg 23, 52074 Aachen, Germany
m.ruettgers@aia.rwth-aachen.de

Abstract. Using traditional computational fluid dynamics and aeroacoustics methods, the accurate simulation of aeroacoustic sources requires high compute resources to resolve all necessary physical phenomena. In contrast, once trained, artificial neural networks such as deep encoder-decoder convolutional networks allow to predict aeroacoustics at lower cost and, depending on the quality of the employed network, also at high accuracy. The architecture for such a neural network is developed to predict the sound pressure level in a 2D square domain. It is trained by numerical results from up to 20,000 GPU-based lattice-Boltzmann simulations that include randomly distributed rectangular and circular objects, and monopole sources. Types of boundary conditions, the monopole locations, and cell distances for objects and monopoles serve as input to the network. Parameters are studied to tune the predictions and to increase their accuracy. The complexity of the setup is successively increased along three cases and the impact of the number of feature maps, the type of loss function, and the number of training data on the prediction accuracy is investigated. An optimal choice of the parameters leads to network-predicted results that are in good agreement with the simulated findings. This is corroborated by negligible differences of the sound pressure level between the simulated and the network-predicted results along characteristic lines and by small mean errors.

Keywords: Deep convolutional neural networks · Aeroacoustic predictions · lattice-Boltzmann method.

1 Introduction

State-of-the-art machine learning (ML), e.g., deep learning (DL) techniques that require very large datasets for successful training, can greatly benefit from high-

performance computing (HPC) simulations. Such simulations can be used to generate lots of training data. They come with the flexibility to obtain datasets corresponding to various task setting parameterizations, which can be used to train ML models. In contrast, obtaining data from experiments can be costly, less flexible, and sometimes even impossible. Trained ML models are capable of performing different forms of predictions on variables of interest if novel input is provided. Their knowledge is based on observations of phenomena acquired from the training on simulated data. Such data-driven models are often used as surrogate models to accelerate predictions compared to classical computationally demanding simulators, given the accuracy provided is sufficient.

Especially in the field of computational fluid dynamics (CFD), DL models trained on simulated data are capable of accelerating the prediction of flow fields. Conventional flow solvers need time to reach solutions at which the impact of initial conditions vanishes. Then, they can be used to compute, e.g., averaged results of the flow. In this case, the period of averaging needs to be bridged before the results can be analyzed. To overcome this issue, methods to accelerate the prediction of steady flow fields using convolutional neural networks (CNNs) are studied [7, 3]. In [7], the flow over simplified vehicle bodies is predicted with CNNs. The corresponding surrogate model is considerably faster than traditional flow solvers. In [3], CNNs are successfully applied to predict flow fields around airfoils with varying angles of attack and REYNOLDS numbers. Lee and You [16] predict the unsteady flow over a circular cylinder using DL methods. They reveal large-scale vortex dynamics to be well predictable by their models. In [17], CNNs to predict unsteady three-dimensional turbulent flows are investigated. The CNNs correctly learn to transport and integrate wave number information contained in feature maps. Additionally, a method that can optimize the number of feature maps is proposed. Unsteady flow and force coefficients are the main focus of the investigations in [22], in which a data-driven method using a CNN for model reduction of the Navier-Stokes equations is presented. In [27], a generative adversarial network (GAN) to forecast movements of typhoons is used and satellite images along with velocity information from numerical simulations are incorporated. This allows for 6-hour predictions of typhoons with an averaged error $< 95.6km$. Unlike numerical predictions on HPC systems, the GAN-based method takes only seconds. Bode et al. [4] propose a physics-informed GAN and successfully model flow on subgrid scales in turbulent reactive flows.

To improve quality and robustness of DL models, training is frequently performed on very large data sets obtained from simulations run on HPC systems. In aerodynamic problems, small-scale structures and/or fluid mechanics based perturbations can strongly influence the acoustic field although they might contain only a small amount of total energy. In many engineering applications, modeling flow-induced sound requires interdisciplinary knowledge about fluid mechanics, acoustics, and applied mathematics. Furthermore, the numerical analysis demands high-resolution numerical simulations to accurately determine the various flow phenomena, e.g., turbulent shear layers [24], fluid-structure interactions [6], and combustion processes [29], that determine the acoustic field. The sheer quan-

tity and often high dimensionality of the parameters describing such flow fields complicate post-processing of the simulated data. This poses a challenge to derive new control models and to make progress in design optimizations [13, 33]. The turn-around time between prototyping and manufacturing depends on the complexity of fundamental physical mechanisms. A recent effort to enhance the efficiency of design development employs an ML framework to predict acoustic fields of a variety of fan nozzle and jet configurations [21]. Although the concept has not yet been realized, this ML-based approach illustrates a prospective possibility to reduce design cycle times of new engine configurations.

The main objective of the present study is the prediction of acoustic fields via a robust ML model based on a deep encoder-decoder CNN. The CNN is trained by acoustic fields containing noise sources surrounded by multiple objects. The numerical results are obtained from simulations using a lattice-Boltzmann (LB) method. They include the simulation of wave propagation, reflection, and scattering due to the interaction with sound-hard surfaces.

In the following, the numerical methods to predict room aeroacoustics with CNNs are described in Sec. 2. Subsequently, results from the sound fields predicted by CNNs are presented and juxtaposed to results of LB simulations in Sec. 3. Finally, a summary is given, conclusions are drawn, and an outlook is presented in Sec. 4.

2 Numerical methods

To generate training data for the CNN, aeroacoustic simulations are run with an LB method on two-dimensional rectangular meshes. The LB method is described in Sec. 2.1, followed by a presentation of the geometrical setup, and the computational meshes in Sec. 2.2. Section 2.3 explains the imposed boundary and initial conditions. Section 2.4 describes how the acoustic fields are analyzed. Finally, the network architecture for the prediction of aeroacoustic fields is presented in Sec. 2.5.

2.1 Lattice-Boltzmann method

To compute the aeroacoustic pressure field, an LB method is employed. The governing equation is the Boltzmann equation with the simplified right-hand side (RHS) Bhatnagar-Gross-Krook (BGK) collision term [2]

$$\frac{\partial f}{\partial t} + \xi_k \frac{\partial f}{\partial x_k} = -\frac{1}{\tau}(f - f^{eq}). \quad (1)$$

The particle probability density functions (PPDFs) $f = f(\vec{x}, \vec{\xi}, t)$ describe the probability to find a particle of a fluid around a location \vec{x} with a particle velocity $\vec{\xi}$ at time t [1, 8]. The left-hand side (LHS) of Eq. (1) describes the evolution

Table 1. Physical quantities of the setup and the non-dimensional viscosity ν .

mesh	$ \Delta\tilde{x} [m] $	$ \Delta\tilde{t} [s] $	$ \tilde{\omega} [Hz] $	$ \tilde{\nu} [m^2/s] $	ν
\mathcal{M}_c	0.2	$3.4 \cdot 10^{-4}$	58.8	$1.551717 \cdot 10^{-5}$	$1.318959 \cdot 10^{-7}$
\mathcal{M}_f	0.1	$1.7 \cdot 10^{-4}$	117.6	$1.551717 \cdot 10^{-5}$	$2.637920 \cdot 10^{-7}$

of fluid particles in space and time, while the RHS describes the collision of particles. The collision process is governed by the relaxation parameter $1/\tau$ with relaxation time τ to reach the Maxwellian equilibrium state f^{eq} . The discretized form of Eq. (1) yields the lattice-BGK equation

$$f_k(\vec{x} + \xi_k \Delta t, t + \Delta t) = f_k(\vec{x}, t) - \frac{1}{\tau} (f_k(\vec{x}, t) - f_k^{eq}(\vec{x}, t)). \quad (2)$$

The quantity Δt is the time increment and τ is a function of the kinematic viscosity ν and the speed of sound c_s , i.e.,

$$\tau = \frac{\nu + \Delta t c_s^2 / 2}{c_s^2}. \quad (3)$$

In the LB context, the spatial and temporal spacing are set to $\Delta x = \Delta t = 1.0$ such that $c_s = 1/\sqrt{3}$. Table 1 exemplarily lists the LB viscosity for two meshes \mathcal{M}_c and \mathcal{M}_f with different resolutions. Note that these values are derived in Sec. 2.3. The LB viscosity is an artificial parameter simply influencing the time step, i.e., how much physical time \tilde{t} is covered by a single Δt in the simulation. Using the viscosities listed in Tab. 1 would lead to extremely small time steps. For this reason and in order to conduct numerically stable simulations, ν is set to a feasible value according to [28]. The indices k in Eq. (2) depend on the discretization scheme and represent the different directions of the PPDFs. In this work, the two-dimensional discretization scheme with 9 PPDFs, i.e., the D2Q9 model [25] is used. The discretized equilibrium PPDF is given by

$$f_k^{eq} = w_k \rho \left(1 + \frac{\xi_k \vec{u}}{c_s^2} + \frac{(\xi_k \vec{u})^2}{2c_s^4} - \frac{\vec{u}^2}{2c_s^2} \right), \quad (4)$$

where the quantities w_k are weighting factors for the D2Q9 scheme given by $4/9$ for $k \in \{0\}$, $1/9$ for $k \in \{1, \dots, 4\}$, and $1/36$ for $k \in \{5, \dots, 8\}$, and \vec{u} is the fluid velocity. The macroscopic variables can be obtained from the moments of the PPDFs, i.e., the density $\rho = \sum_k f_k$. The pressure can be computed using the ideal gas law by $p = c_s^2 \rho = (1/3)\rho$.

The LB method has been chosen for several reasons [18]: (i) the computations can be performed efficiently in parallel, (ii) it is straightforward to parallelize the code, (iii) boundary conditions can easily be applied in contrast to, e.g., cut-cell methods, and (iv) there is no need to solve a pressure Poisson-equation for quasi-incompressible flow as the pressure and hence the acoustic field is an explicit result of the lattice-BGK algorithm. Furthermore, the LB method can be applied for low to high KNUDSEN numbers Kn . In the continuum limit, i.e. for small Kn , the Navier-Stokes and Euler equations can directly be derived from the Boltzmann equation and the BGK model [8].

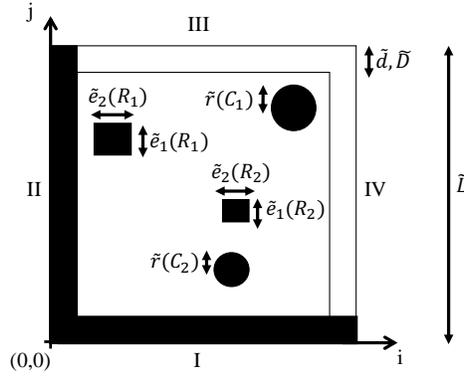


Fig. 1. Computational domain.

2.2 Geometrical setup and computational meshes

The computational domain has a square shape containing randomly distributed objects. In physical space, denoted in the following by $\langle \cdot \rangle$, the domain has an edge length of $\tilde{L} = 25.6m$. Throughout this study, the number of objects varies depending on the complexity of a computation. The domain of the most complex case is shown in Fig. 1. It has two rectangular objects R_1 and R_2 and two circular objects C_1 and C_2 . Their size is a function of the characteristic length $\tilde{C} = \tilde{L}/16$, i.e., R_1 and R_2 have edge lengths $\tilde{e}_1(R_1), \tilde{e}_2(R_1), \tilde{e}_1(R_2), \tilde{e}_2(R_2) \in [\tilde{C}, 2\tilde{C}]$, and C_1 and C_2 have radii $\tilde{r}(C_1), \tilde{r}(C_2) \in [\tilde{C}, 2\tilde{C}]$. All objects have a minimum distance of $d = \tilde{C}$ from the domain boundaries and may overlap.

Two-dimensional uniformly refined meshes \mathcal{M}_f and \mathcal{M}_c with two distinct resolutions are generated in Cartesian coordinates. In the fine mesh \mathcal{M}_f each cell has an edge length of $\Delta\tilde{x}_f = (1/16)\tilde{C} = 0.1m$ resulting in 256×256 cells. The coarse mesh \mathcal{M}_c has a cell length of $\Delta\tilde{x}_c = (1/8)\tilde{C} = 0.2m$ and a total of 128×128 cells.

2.3 Boundary and initial conditions

Two types of boundary conditions are imposed at the four domain boundaries according to [11], i.e., non-reflecting (NRBCs) and wall boundary conditions (WBCs) are prescribed. As shown for boundaries III and IV in Fig. 1, the NRBCs have a buffer layer thickness of $\tilde{D} = \tilde{C}$ to ensure a complete dissipation of acoustic waves and to avoid reflective phenomena at the domain boundaries. In the buffer layer, an absorption term [11]

$$F_{ad} = -\sigma(f_k^{eq}(\vec{x}, t) - f_a), \quad (5)$$

with weighting factor f_a and $\sigma = \sigma_m(\tilde{\delta}/\tilde{D})^2$ is added to Eq. (2). The quantity $\tilde{\delta}$ is the distance to the buffer layer and σ_m is a constant specified as 0.1.

The WBCs are characterized by a no-slip behavior, where the PPDFs are reflectively bounced back. They are imposed as a layer with thickness $\tilde{D} = \tilde{C}$ as shown for boundaries I and II in Fig. 1, i.e., the computational domain is reduced by this thickness. In computations with WBC, a maximum number of three domain boundaries is specified as WBC in a random process. To prevent strong overlaps of acoustic waves, which may cause numerical instabilities, at least at one domain boundary an NRBC is imposed.

The acoustic fields, which are exploited to train the CNN model, are configured by a simple source S defined by a sinusoidal function given by

$$S(t) = A \cdot \sin(2\pi\omega t), \quad (6)$$

with a frequency $\omega = 0.02 \cdot (1/\Delta t)$ and the amplitude $A = 0.1 \cdot \rho_\infty$ and $\rho_\infty = 1.0$ in the LB context. A set of the training data is generated by the computational domains with a noise source restricted by a geometry, i.e., the minimum distance \tilde{C} between the noise source and the sound-hard objects satisfies the condition $L < 2\tilde{C}$ where L is a distance between monopoles and domain boundaries. With $\omega = 1/T$, this yields a non-dimensionalized harmonic period of $T = 50\Delta t$. One wavelength λ is computed from $\lambda = u_w/\omega$, with $u_w = \Delta x/\Delta t$ being the velocity with which information is transported in the LB context. This results in $\lambda = 50\Delta x$ for computations in this study, if not stated otherwise.

The relationship between ω in LB context and the frequency $\tilde{\omega}$ in physical space is obtained by inserting

$$\Delta \tilde{t} = \Delta \tilde{x} \frac{c_s}{\tilde{c}_s}, \quad (7)$$

with the physical speed of sound $\tilde{c}_s = 340m/s$ at reference temperature $T_\infty = 298.15K$ into the equation for the frequency $\tilde{\omega} = 0.02(1/\Delta \tilde{t})$. The relationship between ν in the LB context and the kinematic viscosity $\tilde{\nu}$ in physical space is given by

$$\nu = \tilde{\nu} \cdot \frac{\Delta \tilde{t}}{(\Delta \tilde{x})^2} \quad \text{with} \quad \tilde{\nu} = \frac{1}{\tilde{\rho}_\infty} \cdot \frac{K_1 \cdot T_\infty^{\frac{3}{2}}}{T_\infty + K_2}, \quad (8)$$

The latter equation is Sutherland's law [32] with $\tilde{\rho}_\infty = 1.184kg/m^3$, $K_1 = 1.458 \cdot 10^{-6}kg/(ms \cdot K^{1/2})$, and $K_2 = 110.4K$. Table 1 lists all necessary variables in their dimensional and non-dimensional form for \mathcal{M}_f and \mathcal{M}_c .

2.4 Evaluation of acoustic fields

The acoustic fields are determined by a set of the computational domains which include at least one noise source and randomized solid surfaces. For fluid cells at location (i, j) , $i, j \in \{1, \dots, m\}$, the sound pressure level SPL is defined by

$$SPL(i, j) = 20 \log_{10}(p'_{\text{rms}}(i, j)), \quad (9)$$

where the maximum number of mesh points m is $m = 128$ for the coarse grid and $m = 256$ for the fine grid configurations. The root-mean-square (rms) values

of pressure fluctuations p' are calculated by

$$p'_{\text{rms}}(i, j) = \sqrt{\frac{\sum_{n=1}^N (p_n(i, j) - p_{\text{avg}}(i, j))^2}{N}}, \quad (10)$$

where $p_{\text{avg}}(i, j)$ is the mean pressure averaged over the time period N , and $p_n(i, j)$ is the instantaneous pressure resulting from the simulation at a time step n within that period. Simulations are carried out for 3,000 time steps. The averaging period $N = 2,000$ starts after 1,000 time steps when the acoustic field is fully developed.

2.5 Machine learning techniques

An encoder-decoder CNN is trained to predict the *SPL* in a supervised manner using results of the aforementioned LB simulations. The CNN is fed with four types of input data:

- (i) types of boundary condition;
- (ii) location of monopoles;
- (iii) cell distances for objects;
- (iv) cell distances for monopoles.

To correctly predict aeroacoustic fields, the CNN needs to learn the impact of the various boundary conditions and the location of monopoles on the acoustic field. Therefore, considering inputs (i) and (ii), cells at location (i, j) are assigned segmentation values

$$\mathcal{T}(i, j) = \begin{cases} 0, & \text{empty or NRBC cell} \\ \frac{1}{2}, & \text{WBC or object cell} \\ 1, & \text{monopole cell} \end{cases}. \quad (11)$$

A sensitivity analysis of the input data has been performed before the training. This analysis revealed that solely using boundary parameters leads to poor predictions of the network, i.e., it is not effective for CNNs learning from flow simulations. This is in line with findings in [7]. Since acoustic signals propagate with a certain wavelength and amplitude at a certain sound speed, distances are also important parameters for learning. For this purpose, inputs (iii) and (iv) are provided to the CNN in the form of distance functions Φ_o for objects and Φ_m for monopoles. Such an approach has previously been used for CNNs to predict steady-state flow fields [7, 3]. The distance functions are defined by

$$\Phi_o(\mathbf{x}) = \begin{cases} d(\mathbf{x}, \partial\Omega) & \mathbf{x} \notin \Omega \\ 0 & \mathbf{x} \in \{\partial\Omega, \Omega\} \end{cases} \quad \text{and} \quad \Phi_m(\mathbf{x}) = d(\mathbf{x}, M), \quad (12)$$

i.e., for each cell \mathbf{x} with location (i, j) in a domain the minimal distances $d(\mathbf{x}, \partial\Omega)$ and $d(\mathbf{x}, M)$ to the boundary $\partial\Omega$ of an object Ω and to a monopole M are determined. Obviously, it is $\Phi_o(\mathbf{x}) = \Phi_m(\mathbf{x}) = 0$ on the boundary and exactly at the monopole source. For Φ_o , an assignment of negative distances for cells inside of an object, as it is usually used by signed-distance functions, turned out to have a negative impact on predictions, which is why $\forall(\mathbf{x} \in \Omega : \Phi_o = 0)$. The distances are computed by the fast marching method [30] and are normalized by \tilde{L} . Learning from distances like inputs (iii) and (iv) alone results in mispredictions near domain boundaries. A combination of all presented types of inputs has been found to favorably affect predictions.

In the following, the CNN used for predicting the *SPL* fields is referred to as acoustics field predictor (AFP). The corresponding network architecture is shown in Fig. 2 for a case that uses arrays with the size of \mathcal{M}_f as inputs. Inputs (i) and (ii) are combined to one array. Together with fields (iii) and (iv) they are stacked to form channels of the input. It should be noted that physical quantities such as the pressure distribution are a solution of the acoustic fields computation and constitute the ground truth. They are not known a priori and hence cannot be used for training.

The architecture is inspired by distinct architectures that employ long skip connections between encoder and decoder layers [26, 19, 34], like for instance U-net architectures, which have been successfully used for medical image segmentation [26]. Skip connections between encoding and decoding paths allow the re-use and fusion of features on different scales. To preserve information from features on all scales, the activity of each encoder layer is directly fed to the corresponding decoder layer via long skip connections. These connections are chosen to have residual form, adding the activity of encoder layers to the output of decoder layers. This setup is similar to [19], however, different from the original U-net architecture, where long skip connections have dense form and concatenate layers on the same scale. As depicted in Fig. 2, the residual long skip connections perform identity mapping by adding source encoder layer outputs to target decoder layer outputs [9, 19]. This kind of connectivity allows for direct gradient flow from higher to lower layers across all hierarchy stages during the backward pass, which prevents common issues with vanishing gradients in deep architectures. In contrast to dense long skip connections, residual skip connections lead to smaller numbers of activations to be handled in the decoding path during the forward and backward passes. As a consequence, they decreased memory consumption and are more efficient and faster in training without sacrificing prediction accuracy. Short skip residual connections are also used in so called convolutional residual blocks (Conv-Blocks). Here, convolutional layers, batch normalization (BN), and rectified linear unit (ReLU) activation functions are employed. BN acts as a regularizer, shifting activity of the layers to zero mean, unit variance. This leads to faster and more reliable network convergence [10]. The number of feature maps (FMs) is a multiple of a given factor Y . The output of the first convolutional layer is added to the input of the last ReLU activation, see Fig. 2, which defines residual short skip connections in Conv-Blocks.

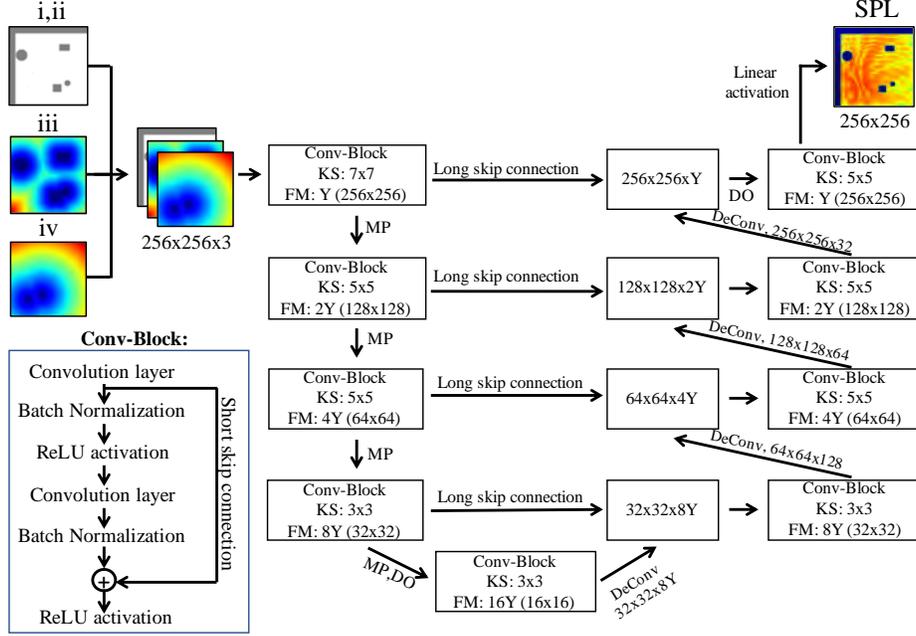


Fig. 2. Network architecture of the AFP including size and number of feature maps (FMs) as a multiple of Y , kernel size (KS), 2×2 maximum pooling layers (MP), dropout layers (DO), convolutional blocks, and deconvolutional layers.

A combination of long and short skip connections leads to faster convergence and stronger loss reduction [5]. In the encoder path, downscaling is performed by 2×2 maximum pooling layers (MP). To further avoid overfitting, yet another regularization method, dropout (DO) [31] is used during training, with a DO probability of $P = 0.5$. The final layer is fully connected with a linear activation function, which is frequently used for regression outputs [15]. Weights and biases are initialized from a truncated normal distribution centered around the origin with a standard deviation of $\sigma_{std} = \sqrt{2/f}$, where f is the number of connections at a layer [9]. They are updated by an adaptive moments (ADAM) optimizer [12]. The ADAM optimizer adjusts the learning rate (LR) by considering an exponentially decaying average of gradients computed in previous update steps. The initial learning rate is set to $LR = 0.001$. The batch size BS represents the number of training data passed to the network in a single training iteration. In Sec. 3 it will be shown that in this context a batch size of $BS = 5$ achieves the best results. Therefore, it is used throughout this study, if not stated otherwise. The ground truth GT distribution SPL_{GT} is obtained from

$$SPL_{GT} = \frac{SPL - SPL_{mean}}{SPL_{std}}, \quad (13)$$

where SPL_{mean} and SPL_{std} are the mean and the standard deviation of the complete training dataset of the a priori simulations. The predictions need to be

denormalized before the SPL can be analyzed.

Data augmentation is used to increase training data diversity and to encourage learning of useful invariances. Therefore, the coordinate axes i and j are transposed randomly. Furthermore, for inputs (i) and (ii), the segmentation values $\mathcal{Y}(i, j)$ are changed to augmented inputs $\mathcal{Y}_{augm}(i, j)$ according to

$$\mathcal{Y}_{augm}(i, j) = \begin{cases} \mathcal{Y}(i, j) & i = j \\ \mathcal{Y}(j, i) & i \neq j. \end{cases} \quad (14)$$

The total loss L_{tot} between simulated (superscript “*sim*”) and predicted (superscript “*pred*”) SPL values is defined by

$$L_{tot} = L_{MSE} + 2 \cdot \underbrace{(L_{GDL}^I + L_{GDL}^{II} + L_{GDL}^{III} + L_{GDL}^{IV})}_{L_{GDL}}, \quad (15)$$

which is a combination of the mean squared error MSE

$$L_{MSE} = \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m (SPL(i, j)^{sim} - SPL(i, j)^{pred})^2 \quad (16)$$

with $1 \leq i, j \leq m$ and a gradient difference loss L_{GDL} . Gradient losses GDL in i - and j -directions are considered by L_{GDL_I} and $L_{GDL_{II}}$, and diagonal gradients by $L_{GDL_{III}}$ and $L_{GDL_{IV}}$.

Three types of gradient losses are addressed in this work. The four directions indicated by roman numbers I – IV in Eq. (15) are defined by introducing integer variables k and l , i.e., the four directions are denoted by $I : (k = 1, l = 1)$, $II : (k = 2, l = 2)$, $III : (k = 1, l = 2)$, and $IV : (k = 2, l = 1)$. In the first type, L_{GDL_A} , the difference between two neighboring cells is considered, inspired by the gradient loss in the work of Mathieu et al. [20]

$$L_{GDL_A} = \frac{1}{(m-1)(m - \text{mod}(p, 2))} \cdot \sum_i \sum_j [SPL_{i+s, j+t}^{sim} - SPL_{i, j}^{sim} - SPL_{i+s, j+t}^{pred} + SPL_{i, j}^{pred}]^2. \quad (17)$$

In Eq. (17) the gradient losses of four neighboring points are defined by the notations $p = \text{mod}(k, 2) + \text{mod}(l, 2)$, $s = 1 - \text{mod}(k+1, 2) \cdot \text{mod}(l+1, 2)$, and $t = (-1)^{k+1} \cdot \text{mod}(p, 2) + 1 - s$. The gradient loss terms of the first type have a 1st-order accuracy in terms of a forward difference (FD) formulation [23]. To integrate radial propagation of a point source into the loss function, central difference (CD) schemes are added. The gradient loss L_{GDL_B} uses a 2nd-order accurate CD formulation that incorporates two neighboring cells. The 2nd-order accurate gradient loss terms in a two-dimensional domain read

$$L_{GDL_B} = \frac{1}{4 \text{mod}(k+l+1, 2) + 8 \text{mod}(k+l, 2)(m-2)(m-2 \cdot \text{mod}(p, 2))} \cdot \sum_i \sum_j [SPL_{i+s, j+t}^{sim} - SPL_{i-s, j-t}^{sim} - SPL_{i+s, j+t}^{pred} + SPL_{i-s, j-t}^{pred}]^2 \quad (18)$$

The third type of gradient loss, L_{GDL_C} , is formulated with a 4th-order accurate CD scheme and includes four neighboring cells, i.e., two cells in each direction

$$L_{GDL_C} = \frac{1}{144 \bmod(k+l+1, 2) + 32 \bmod(k+l, 2)(m-4)(m-4 \cdot \bmod(p, 2))} \cdot \sum_i \sum_j [-SPL_{i+2s, j+2t}^{sim} + 8SPL_{i+s, j+t}^{sim} - 8SPL_{i-s, j-t}^{sim} + SPL_{i-2s, j-2t}^{sim} + SPL_{i+2s, j+2t}^{pred} - 8SPL_{i+s, j+t}^{pred} + 8SPL_{i-s, j-t}^{pred} - SPL_{i-2s, j-2t}^{pred}]^2. \quad (19)$$

The cell-wise prediction accuracy is evaluated by the absolute error

$$\Xi(i, j) = \frac{|SPL^{pred}(i, j) - SPL^{sim}(i, j)|}{|SPL_{max}^{sim} - SPL_{min}^{sim}|}. \quad (20)$$

between SPL^{pred} and SPL^{sim} with $SPL_{max}^{sim} = \max(SPL^{sim})$ and $SPL_{min}^{sim} = \min(SPL^{sim})$. From the Ξ distribution of each simulation a mean absolute error

$$\Gamma = \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m \Xi(i, j) \quad (21)$$

is calculated to evaluate the prediction quality.

3 Results

In the following, findings of a grid convergence study are discussed in Sec. 3.1. Results of network-predicted acoustic fields are presented for three cases 1–3 in Secs. 3.2, 3.3, and 3.4. The complexity of the cases is continuously increased. The acoustic simulations are conducted on multiple graphics processing units (GPUs). At average, a solution on \mathcal{M}_f is obtained in $\approx 120s$ on a single GPU. Up to ten GPUs are employed to accelerate the process. Once trained, the network predictions take only a fraction of a second on a single modern GPU and only a few seconds on any low end computer such as a laptop. For all computations the GPU partition of the JURECA system [14], Forschungszentrum Jülich, is employed. Each GPU node is equipped with two NVIDIA K80 GPUs.

3.1 Grid convergence study

A grid convergence study is conducted in a free-field domain containing only a single monopole at the center and no walls. The impact of doubling the number of cells used to resolve one wavelength λ on the SPL accuracy is investigated. Therefore, the wavelength resolutions at a distance of up to 4 wavelengths in radial direction from the source, which corresponds to the maximum appearing

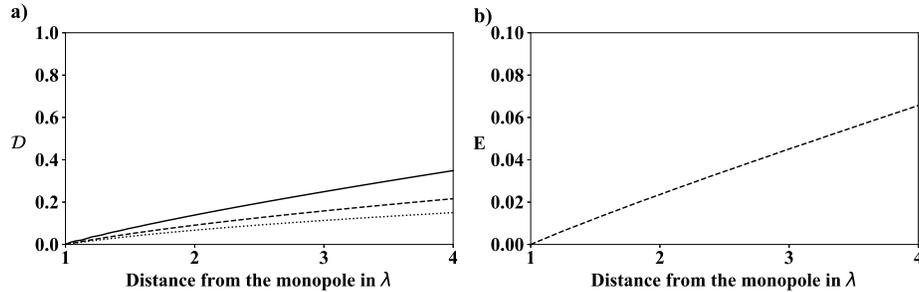


Fig. 3. a) $\mathcal{D} = (SPL - SPL_{max})/SPL_{max}$ at a distance from up to 4λ in radial direction from a monopole placed in the center of a free field. Three resolutions for one wavelength are juxtaposed: $\mathcal{D}(\lambda = 100\Delta x)$ \cdots , $\mathcal{D}(\lambda = 50\Delta x)$ $---$, $\mathcal{D}(\lambda = 25\Delta x)$ $---$. b) Error E between $\mathcal{D}(\lambda = 50\Delta x)$ and $\mathcal{D}(\lambda = 100\Delta x)$.

distance considered in the subsequently discussed cases 1-3, is analyzed. In order to obtain results in a farfield from the center for $\lambda = 100\Delta x$, the domain is extended to $1,024 \times 1,024$ cells. Figure 3a) shows the divergence $\mathcal{D} = (SPL - SPL_{max})/SPL_{max}$ from the maximum SPL value SPL_{max} , which appears at a distance of one wavelength from the monopole location, for $\lambda = 25\Delta x$, $50\Delta x$, and $100\Delta x$. From this figure, it is evident, that the divergence increases with increasing distance from the monopole. Furthermore, Fig. 3b) shows the error for $\lambda = 50\Delta x$ compared to $\lambda = 100\Delta x$, i.e., $E = \mathcal{D}(\lambda = 50\Delta x) - \mathcal{D}(\lambda = 100\Delta x)$. Throughout this work a wavelength of $\lambda = 50\Delta x$ is used, which covers distances up to 2λ in cases 1-2, and up to 4λ in case 3. At distances 2λ and 4λ , errors of $E = 0.0239$ and $E = 0.0657$ are obtained. It should be noted that using $\lambda = 100\Delta x$ would massively increase the computational effort and hence, as the corresponding error is acceptable, meshes with $\lambda = 50\Delta x$ are employed in all cases.

3.2 Case 1: Simple setup and parameter study

The domain in case 1 contains one monopole M_1 at the center $(8C, 8C)$ and one randomly positioned circular object C_1 . Each computational domain consists of 128×128 cells in the two dimensions. The acoustic solutions of 3,000 simulations are split into 2,600 training data, 200 validation data, and 200 test data. Three sub-cases 1A, 1B, and 1C listed in Tab. 2 are configured by one noise source and one solid object. In case 1A, the number of FMs is investigated by varying the factor Y as shown in Fig. 2. Variations of $Y = 8$, $Y = 16$ and $Y = 32$ lead to 517,867, 2,066,001 and 8,253,089 trainable parameters. It is evident from comparing Figs. 4b), 4c), and 4d) with the simulation results in Fig. 4a) that $Y = 32$ qualitatively reproduces the simulation best. For $Y = 8$, the AFP completely fails to generate a physically meaningful SPL field. In case of $Y = 16$, acoustic waves distant from the object are reproduced well, but superpositions of acoustic waves in the vicinity of the object are too strong, see

Table 2. Simulation configurations defined by objects, the number of noise sources (no. noise) and simulations (no. sim) generated by randomized distributions of objects. The number of feature maps (FMs) is defined by Y . The gradient losses GDL are calculated by FD, 2nd-order-accurate, and 4th-order-accurate CD schemes. The quantities BS and Γ are the batch size and the mean acoustic error.

case	object(s)	no. noise	no. sim	Y	GDL method	BS	Γ
1A	C_1	1	3,000	8	FD	5	0.17506
	C_1	1	3,000	16	FD	5	0.03312
	C_1	1	3,000	32	FD	5	0.00887
1B	C_1	1	3,000	32	FD	5	0.00887
	C_1	1	3,000	32	2nd order CD	5	0.00671
	C_1	1	3,000	32	4th order CD	5	0.00222
1C	C_1	1	3,000	32	2nd order CD	5	0.00671
	C_1	1	3,000	32	2nd order CD	10	0.00626
	C_1	1	3,000	32	2nd order CD	20	0.00413
2	R_1, C_1	1	3,000	32	2nd order CD	5	0.00359
	R_1, C_1	1	6,000	32	2nd order CD	5	0.00280
3	R_1, R_2, C_1, C_2	2	6,000	32	2nd order CD	5	0.02581
	R_1, R_2, C_1, C_2	2	10,000	32	2nd order CD	5	0.02268
	R_1, R_2, C_1, C_2	2	20,000	32	2nd order CD	5	0.01937

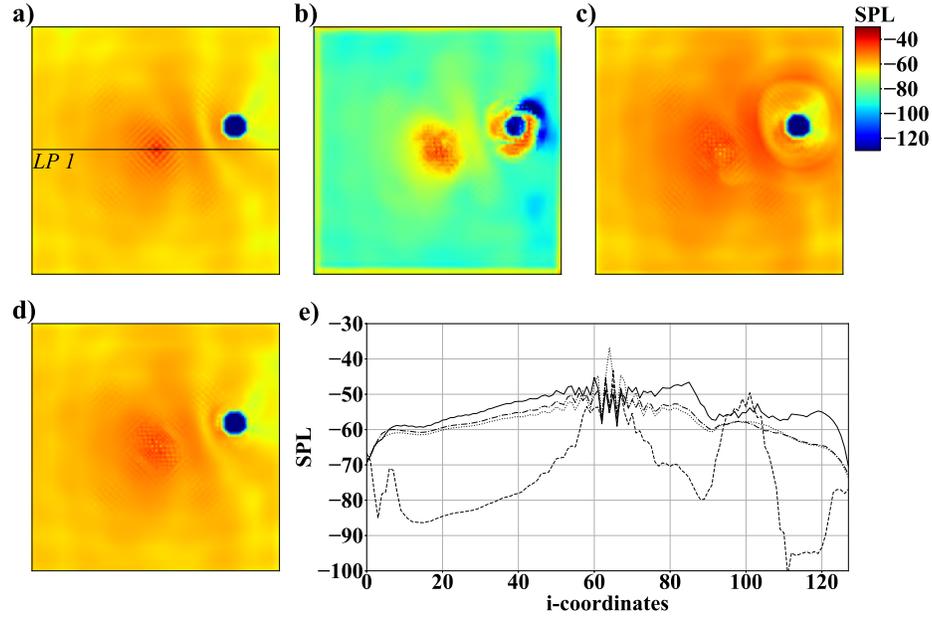


Fig. 4. Example of SPL fields of case 1A: a) simulation result, b) network prediction with $Y = 8$, c) $Y = 16$, and d) $Y = 32$; e) SPL distribution at $j = 64$ along $LP1$: simulation result \cdots , network prediction with $Y = 8$ $---$, $Y = 16$ $—$, and $Y = 32$ $- \cdot -$.

Fig. 4c). The *SPL* distribution shown in Fig. 4e) along the characteristic line *LP1*, see Fig. 4a), substantiates these findings. The valley between M_1 and C_1 , and the decrease of the *SPL* value in the shadow of C_1 are only captured well for $Y = 32$. Furthermore, the CNN has problems capturing fluctuations at the center of M_1 as non-physical *SPL* values are found at isolated locations close to the object. The mean error Γ listed in Tab. 2 shows $Y = 32$ to have the lowest deviation among the three computations. The training time to reach a convergence of the loss function increased from approximately one hour for $Y = 8$ up to two and four hours for $Y = 16$ and $Y = 32$.

To overcome inaccurate predictions close to monopoles, the nature of a noise source is incorporated into the loss function of the AFP. A simple FD gradient loss does not consider that monopoles are point sources spreading waves into all directions. In case 1B, two variations of losses are investigated that are based on the CD formulations provided in Sec. 2.5. From Fig. 5 it is obvious that thereby non-physical *SPL* values vanish near objects. Furthermore, Fig. 5c) shows an improvement of the *SPL* distribution at the center and surroundings of M_1 predicted by a 2nd-order accurate CD gradient loss. In contrast, using a 4th-order accurate CD formulation lowers the accuracy of the predictions near M_1 , see Fig. 5d). It is, however, evident from Tab. 2 that a slightly lower Γ is achieved than using a 2nd-order formulation. This is due to the 4th-order accurate CD gradient loss computations reproducing simulations slightly better at locations distant from monopoles and objects, see Fig. 5e). *SPL* fluctuations at the center of M_1 are by far closer to the ground truth using the 2nd-order accurate formulation. Since this study focuses on the prediction of complex acoustic fields with multiple noise sources, the advantages of the 2nd-order accurate formulation are considered more valuable, i.e., in the following this type of loss is employed.

The impact of *BS* is investigated in Fig. 6. Figure 6e) plots the *SPL* distribution along line *LP3*, see Fig. 6a). Although predictions with $BS = 10$ and $BS = 20$ show a slight decrease of Γ , see Tab. 2, several shortcomings are recognizable in predicted *SPL* fields. Figures 6c) and e) show that with $BS = 10$ non-physical fluctuations near the objects are introduced. These fluctuations are also present for $BS = 20$ and are superimposed by inaccuracies appearing in the vicinity of M_1 and at the domain boundaries, i.e., $BS = 5$ delivers the best results.

3.3 Case 2: Influence of the number of training data

In case 2, the number of training, validation, and test data is analyzed. Compared to case 1, the complexity is increased by adding a rectangular object R_1 to the domain. The training, validation, and test data are composed of 2,600, 200 and 200 simulations for a total of 3,000, and of 5,200, 400, and 400 for a total of 6,000 simulations. The setups for these cases are summarized in Table 2. Figure 7 compares the results of an LB simulation qualitatively and quantitatively along line *LP4*, see Fig. 7a), with predictions generated by using 3,000 and 6,000 simulations for learning. When the amount of data is increased, non-physical fluctuations disappear in regions, where sound waves propagate towards

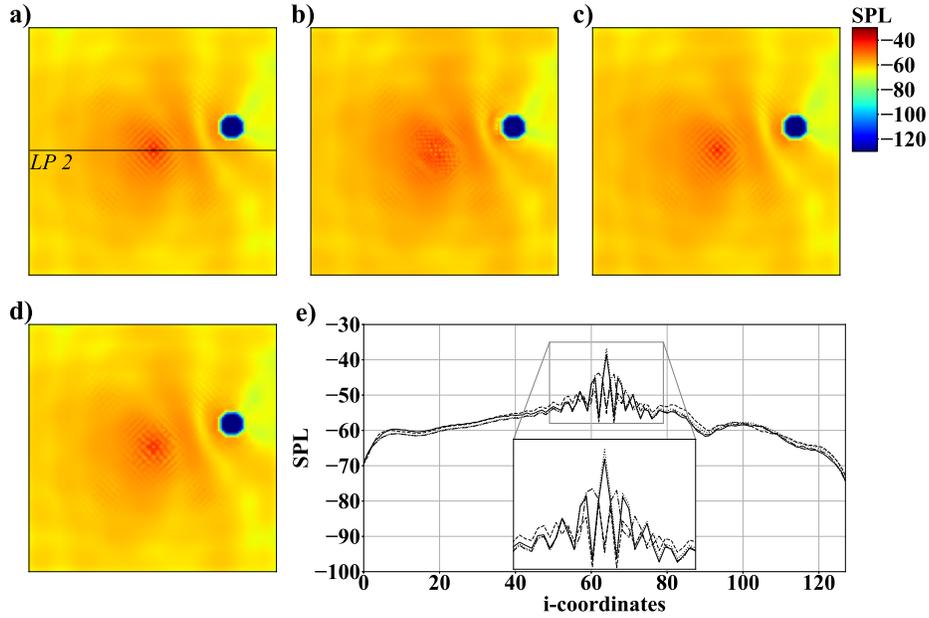


Fig. 5. Example of SPL fields of case 1B: a) simulation result, b) network prediction with FD, c) a 2nd-order accurate CD, and d) a 4th-order accurate CD gradient loss; e) SPL at $j = 64$ along $LP2$: simulation result \cdots , network prediction with FD $---$, 2nd-order accurate CD $—$, and 4th-order accurate CD gradient losses $- \cdot -$.

the surface of R_1 . Furthermore, the predictions of the acoustic field in the vicinity of C_1 improve from 3,000 to 6,000 training datasets.

3.4 Case 3: Complex setup and impact of increasing training data

Case 3 ties on to the findings from the previous cases to predict SPL fields in a domain containing objects C_1 , C_2 , R_1 , and R_2 , see Fig. 1, on \mathcal{M}_f . From \mathcal{M}_c to \mathcal{M}_f the number of trainable parameters increases from 8,253,089 to 8,256,225. NRBC and WBC boundary conditions are imposed randomly at the domain boundaries. Two monopoles M_1 and M_2 are placed inside of the domain. M_1 is located at $(5C, 5C)$ and M_2 is positioned randomly. For the training, validation, and testing of the AFP, a total number of 20,000 simulations is used. Results of computations with different simulation inputs are compared to the ground truth in Fig. 8. Note that the WBC is imposed at domain boundary IV, however, the complete thickness \tilde{D} is not visualized in the figure. The first case uses 6,000 simulations with a distribution of 5,200, 400, and 400 for training, validation, and testing. The second case employs 10,000 simulations with a distribution of 8,800, 600, and 600 for training, validation, and testing. The last case employs all 20,000 simulations with a distribution of 18,000, 1,000, and 1,000 for training, validation, and testing. For reference, the different setups and the corresponding

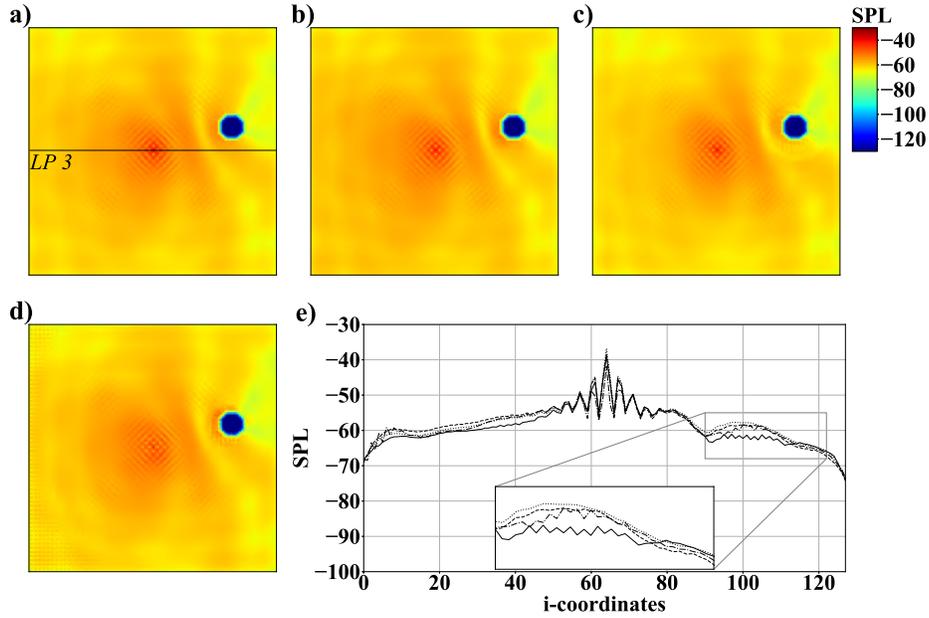


Fig. 6. Example for SPL fields of case 1C: a) simulation result, b) network prediction with $BS = 5$, c) $BS = 10$, and d) $BS = 20$; e) SPL distribution at $j = 64$ along line $LP3$: simulation result \cdots , network prediction with $BS = 5$ $---$, $BS = 10$ $—$, and $BS = 20$ $- \cdot -$.

results are listed in Tab. 2. Obviously, the error Γ decreases when the number of training data is increased. From Figs. 8c) and e) it is evident that the AFP trained with 8,800 datasets overpredicts the SPL near M_1 . In general, it can be stated that with an increasing complexity the SPL is more difficult to predict compared to cases 1 and 2. To be more specific, from case 1 to case 3 the error Γ increases by one order of magnitude, i.e., it is at $\Gamma = 0.01937$ in case 3. However, complex acoustic fields are reproduced. For a number of 18,000 simulation, training took 96 hours to reach a convergence of the loss function.

4 Summary, conclusions, and outlook

A deep learning method has been developed to predict the sound pressure level distribution in two-dimensional aeroacoustic setups including multiple randomly distributed rectangular and circular objects as hard reflective surfaces and monopoles as sound sources. The deep learning method is based on an encoder-decoder convolutional neural network, which has been trained with numerical simulations based on a lattice-Boltzmann method. To analyze the accuracy of the network predictions, various learning parameters have been tuned by successively increasing the complexity of the prediction cases and by analyzing different

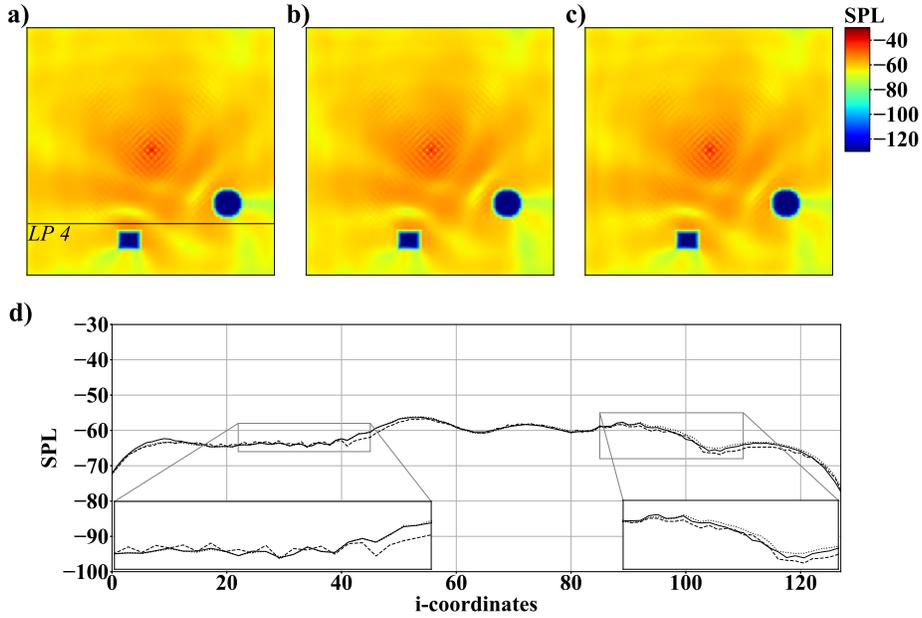


Fig. 7. Example of *SPL* fields of case 2: a) simulation result, b) network prediction with 3,000, and c) 6,000 simulations; *SPL* distribution at $j = 30$ along *LP4*: simulation result \cdots , network prediction with 3,000 $---$, and 6,000 simulations $—$.

loss functions. A network containing 8,256,225 trainable parameters, a combination of the mean-squared error loss and gradient loss formulated by a 2nd-order accurate central difference scheme, and a batch size of five positively influenced the predictions. A number of 18,000 datasets has been used to train the deep neural network. A mean absolute error of less than 2% shows the neural network being capable of accurately predicting the acoustic fields. The study has been complemented with a grid convergence study, which revealed that a resolution of 50 cells for a single wavelength is sufficient to yield accurate results.

At present, the method is spatially limited to two-dimensional cases. However, most engineering applications, e.g., design processes to find optimal layouts for low-noise turbojet engines, feature three-dimensional phenomena. Extending the presented deep learning method to learn from three-dimensional simulations will lead to accelerated predictions of three-dimensional aeroacoustic problems. Furthermore, realistic acoustic fields are frequently characterized by interactions of multiple noise sources with various frequencies and amplitudes. Therefore, it is necessary to extend the current setup to monopoles with multiple frequencies and amplitudes. Apart from increasing the domain's complexity, the level of generalization will be increased. The presented acoustic field predictor has been trained and tested on similar situations. Its capabilities to generalize will be enhanced by testing on situations that have not been part of the training process, e.g., training with four objects and testing with five. Instead of strictly separating different gradient losses, the impact of combining them in a single loss

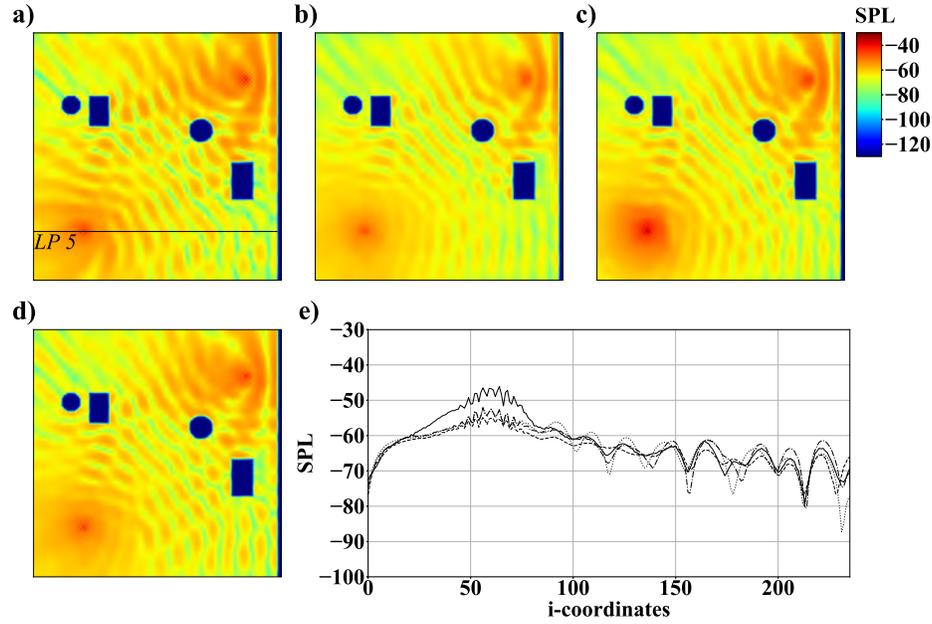


Fig. 8. Example of *SPL* fields of case 3: a) simulation result, b) network prediction with 6,000, c) 10,000, and d) 20,000 simulations; *SPL* distribution at $j = 80$ along *LP5*: simulation result \cdots , network prediction with 6,000 $-\cdot-$, 10,000 $-$, and 20,000 simulations $- - -$.

and employing individual weights will be analyzed. In addition, physics-informed losses that allow the network to comply with physical laws of aeroacoustics will be integrated. Furthermore, adversarial training will be investigated by adding a discriminator with an adversarial loss to the current architecture. Such GAN type architectures have the potential to help finding a suitable loss from the training data. It is also worth mentioning that the method presented in this study has the potential to support solving noise control problems. It remains to investigate if a dedicated acoustic field predictor that can quickly give feedback on the arrangement of multiple monopoles is capable of finding optimal acoustic setups. Therefore, the presented acoustic field predictor will be integrated into a reinforcement learning loop.

Acknowledgments. The authors gratefully acknowledge the computing time granted through the Jülich Aachen Research Alliance (JARA) on the supercomputer JURECA [14] at Forschungszentrum Jülich.

References

1. Benzi, R., Succi, S., Vergassola, M.: The lattice Boltzmann equation: theory and applications. *Phys. Rep.* **222**(3), 145–197 (1992). [https://doi.org/10.1016/0370-1573\(92\)90090-M](https://doi.org/10.1016/0370-1573(92)90090-M)

2. Bhatnagar, P.L., Gross, E.P., Krook, M.: A Model for Collision Processes in Gases. I. Small Amplitude Processes in Charged and Neutral One-Component Systems. *Phys. Rev.* **94**(3), 511–525 (1954). <https://doi.org/10.1103/PhysRev.94.511>
3. Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K., Kaushik, S.: Prediction of aerodynamic flow fields using convolutional neural networks. *Comput. Mech.* **64**(2), 525–545 (2019). <https://doi.org/10.1007/s00466-019-01740-0>
4. Bode, M., Gauding, M., Lian, Z., Denker, D., Davidovic, M., Kleinheinz, K., Jitsev, J., Pitsch, H.: Using Physics-Informed Super-Resolution Generative Adversarial Networks for Subgrid Modeling in Turbulent Reactive Flows (2019)
5. Drozdal, M., Vorontsov, E., Chartrand, G., Kadoury, S., Pal, C.: The Importance of Skip Connections in Biomedical Image Segmentation. In: *Deep Learning and Data Labeling for Medical Applications, DLMIA 2016, LABELS 2016*, Lect. Notes Comput. Sci., pp. 179–187. Springer, Cham, Athens, Greece (2016). https://doi.org/10.1007/978-3-319-46976-8_19
6. Ewert, R., Schröder, W.: On the simulation of trailing edge noise with a hybrid LES/APE method. *J. Sound Vib.* **270**, 509–524 (2004). <https://doi.org/10.1016/j.jsv.2003.09.047>
7. Guo, X., Li, W., Iorio, F.: Convolutional Neural Networks for Steady Flow Approximation. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pp. 481–490. ACM Press, New York, New York, USA (2016). <https://doi.org/10.1145/2939672.2939738>
8. Hänel, D.: *Molekulare Gasdynamik, Einführung in die kinetische Theorie der Gase und Lattice-Boltzmann-Methoden*. Springer-Verlag (2004)
9. He, K., Zhang, X., Ren, S., Sun, J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In: *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034. IEEE (2015). <https://doi.org/10.1109/ICCV.2015.123>
10. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: *ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pp. 448–456. W&CP, Lille, France (2015). <https://doi.org/10.5555/3045118.3045167>
11. Kam, E.W.S., So, R.M.C., Leung, R.C.K.: Lattice Boltzman Method Simulation of Aeroacoustics and Nonreflecting Boundary Conditions. *AIAA Journal* **45**(7), 1703–1712 (2007). <https://doi.org/10.2514/1.27632>
12. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization (2014)
13. Koh, S.R., Meinke, M., Schröder, W.: Numerical analysis of the impact of permeability on trailing-edge noise. *J. Sound Vib.* **421**, 348–376 (May 2018). <https://doi.org/10.1016/j.jsv.2018.02.017>
14. Krause, D., Thörnig, P.: JURECA: Modular supercomputer at Jülich Supercomputing Centre. *JLSRF* **4**, A132 (2018). <https://doi.org/10.17815/jlsrf-4-121-1>
15. Lathuilière, S., Mesejo, P., Alameda-Pineda, X., Horaud, R.: A comprehensive analysis of deep regression (2018)
16. Lee, S., You, D.: Data-driven prediction of unsteady flow over a circular cylinder using deep learning. *J. Fluid Mech.* **879**, 217–254 (2019). <https://doi.org/10.1017/jfm.2019.700>
17. Lee, S., You, D.: Mechanisms of a convolutional neural network for learning three-dimensional unsteady wake flow (2019)
18. Lintermann, A., Meinke, M., Schröder, W.: Zonal Flow Solver (ZFS): a highly efficient multi-physics simulation framework. *Int. J. Comput. Fluid Dyn.* pp. 1–28 (2020). <https://doi.org/10.1080/10618562.2020.1742328>

19. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: *Advances in neural information processing systems*. pp. 2802–2810 (2016)
20. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error (2015)
21. McKee, C., Harmanto, D., Whitbrook, A.: A conceptual framework for combining artificial neural networks with computational aeroacoustics for design development. *Proceedings of the International Conference on Industrial Engineering and Operations Management* (2018)
22. Miyanawala, T.P., Jaiman, R.K.: An efficient deep learning technique for the navier-stokes equations: Application to unsteady wake flow dynamics (2017)
23. Moin, P.: *Fundamentals of engineering numerical analysis*. Cambridge University Press/c 2001., London
24. Niemöller, A., Schlottke-Lakemper, M., Meinke, M., Schröder, W.: Dynamic load balancing for direct-coupled multiphysics simulations. *Comput. Fluids* **199**, 104437 (March 2020). <https://doi.org/10.1016/j.compfluid.2020.104437>
25. Qian, Y.H., D’Humières, D., Lallemand, P.: Lattice BGK Models for Navier-Stokes Equation. *EPL* **17**(6), 479–484 (1992). <https://doi.org/10.1209/0295-5075/17/6/001>
26. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015)
27. Rüttgers, M., Lee, S., Jeon, S., You, D.: Prediction of a typhoon track using a generative adversarial network and satellite images. *Sci. Rep.* **9** (2019). <https://doi.org/10.1038/s41598-019-42339-y>
28. Salomons, E.M., Lohman, W.J.A., Zhou, H.: Simulation of Sound Waves Using the Lattice Boltzmann Method for Fluid Flow: Benchmark Cases for Outdoor Sound Propagation. *PLOS ONE* **11**(1), e0147206 (2016). <https://doi.org/10.1371/journal.pone.0147206>
29. Schlimpert, S., Koh, S.R., Pausch, K., Meinke, M., Schröder, W.: Analysis of combustion noise of a turbulent premixed slot jet flame. *Combust. Flame* **175**, 292–306 (January 2017). <https://doi.org/10.1016/j.combustflame.2016.08.001>
30. Sethian, J.: *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge monographs on applied and computational mathematics, Cambridge University Press (1999)
31. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(56), 1929–1958 (2014)
32. Sutherland, W.: The viscosity of gases and molecular force. *Philos. Mag.* **36**(5), 507–531 (1893)
33. Zhou, B.Y., Koh, S.R., Gauger, N., Meinke, M., Schröder, W.: A discrete adjoint framework for trailing-edge noise minimization via porous material. *Comput. Fluids* **172**, 97–108 (August 2018). <https://doi.org/10.1016/j.compfluid.2018.06.017>
34. Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J.: Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging* (2019)