

To cite:
DOI: 10.5281/zenodo.7538880

Hard- and soft-constrained thermoacoustic neural networks

Defne E. Ozan

Department of Aeronautics, Imperial College London, SW7 2AZ, London, UK

Luca Magri^{*}

Department of Aeronautics, Imperial College London, SW7 2AZ, London, UK and

The Alan Turing Institute, NW1 2DB, London, UK

Abstract

In thermoacoustic systems, if the heat release is sufficiently in phase with the acoustic pressure, self-excited oscillations can arise. These oscillations are known as thermoacoustic oscillations, which can have detrimental consequences to gas turbines and rocket engines. A typical nonlinear regime of the thermoacoustic dynamics is a limit cycle, which is characterised by a periodic orbit in the phase space. In this work, we develop and analyse physics-aware neural networks to learn periodic solutions of thermoacoustic systems from data. First, in addition to a data-driven loss, a physical residual penalises solutions that violate the conservation of mass, momentum, and energy. Second, periodicity is imposed by introducing periodic activation functions and a trainable angular frequency in the neural networks. Third, acoustic eigenfunctions are employed as spatial modes, while a jump discontinuity in velocity at the flame location is captured by additional discontinuous modes. We test the algorithm on synthetic data generated from a nonlinear time-delayed model of a Rijke tube, as well as a higher-fidelity model with a kinematic flame. We find that (i) physical constraints significantly improve the predictions from noisy or sparse data, (ii) periodic activation functions outperform conventional activation functions in terms of extrapolation capability and convergence rate, and (iii) spatial features such as boundary conditions and discontinuities can be hard-coded in the neural network with an a-priori selected spatial modes. This work opens up possibilities for the prediction of nonlinear thermoacoustics by combining physical knowledge and data.

I. INTRODUCTION

In combustors, the unsteady heat release and pressure can couple in a positive feedback loop. When the heat release is sufficiently in phase with the pressure, self-excited oscillations with finite amplitudes, known as thermoacoustic oscillations, can arise [1]. Because these oscillations can have detrimental effects on the system's structure and performance, their prediction and control are active areas of research, for example, in gas turbines [2], hydrogen-based gas turbines [3], and rocket engines [4].

In gas turbines, the acoustics can be modelled as linear because the mean-flow has a small Mach number [e.g., 4]. Thermoacoustic systems contain nonlinearities in the heat

* l.magri@imperial.ac.uk

release law, which, when coupled with the acoustics, can generate intricate behaviours via bifurcations and nonlinear self-excited oscillations [5]. A typical nonlinear regime that has often been experimentally evidenced in industrial combustors is a limit cycle [e.g., 6-9]. A limit cycle is characterised by a periodic orbit in the phase space. While limit cycles have a fixed period, they are not necessarily harmonic, i.e., their spectrum can contain peaks at multiple frequencies. Because of their practical implications, the prediction and control of thermoacoustic limit cycles has been extensively studied with models based on first principles. First-principles models are expressed as partial differential equations, which are typically time-delayed and nonlinear. In the time domain, a direct approach for the computation of limit cycles is the brute-force time-integration of the governing equations following a discretisation scheme. High-fidelity models such as Large-eddy simulations that model the acoustics and the flame simultaneously on fine grids provide highly accurate solutions, but they are computationally expensive [3]. To reduce the computational cost of brute-force time integration, continuation and shooting methods find limit cycles as exact periodic solutions of the equations and great accuracy, but they may be computationally expensive and require iterative methods [10]. Low-fidelity models reduce the computational effort at the expense of accuracy while obtaining models that can be used for stability, bifurcation analysis, and parametric studies. Generally, these approaches combine a linear acoustic solver with a heat release law. In this direction, limit cycle behaviour has been investigated by employing network models based on the travelling-wave [11-13], Galerkin decomposition of pressure and velocity using only a finite number of modes [14, 15], and a higher-fidelity numerical discretisation of the PDEs [16]. In the frequency domain, flame models are defined via a flame transfer function (FTF), or a flame describing function (FDF). FTF cannot predict the limit cycle amplitude since it is a linear function of only the input frequency and not the amplitude. In contrast, FDFs are defined as transfer functions parameterized with the input amplitude, thus, the zero growth rate mode can be identified and consequently the existence and amplitude of the limit cycle amplitude can be predicted [9]. This approach provides computationally cheap estimates, however, the FDF operates only with single input and single output frequencies. FDF has been extended to multiple frequencies [17, 18], albeit these remain approximate methods.

Data-driven methods have emerged as computationally cheap and flexible approaches. Methods such as Fast Fourier Transform (FFT), symbolic time series analysis (STSA) and

hidden Markov modelling (HMM) have been adopted for early detection and classification of thermoacoustic instabilities from time-series of thermoacoustic systems [19]. FFT is a robust and established method, however is not suitable for real-time use, as it requires long time series; STSA is recommended for time efficiency, while HMM is preferred for higher accuracy [19]. Neural networks have been developed for heat release law or flame response inference in thermoacoustic applications [20-22]. Dhadphale *et al.* [23] proposed a Neural ODE based method, which learns the system dynamics and provides a precursor for thermoacoustic instabilities from experimental measurements of pressure and heat release fluctuations. The downside of data-driven models is that they are often black-box models, which disregard the prior knowledge that the practitioner has on the system. Furthermore, the acquired data may also be corrupted with noise or may be scarce as it may be expensive to obtain.

In this paper, we propose a physics-aware data-driven method, which combine first-principles and data-driven approaches to leverage on their strengths, whilst reducing their weaknesses. By incorporating physical information, data-driven models can provide solutions that do not violate the physical laws or prior knowledge. In feedforward neural networks, physics-information is often introduced as a soft-constraint acting upon the loss [24]. This approach has been popularised by Raissi *et al.* [25] with physics-informed neural networks (PINNs) in prototypical flows. In thermoacoustics, Novoa and Magri [26] combined an ensemble Kalman filter and neural networks for real-time data assimilation, in which a physics-based low-fidelity model provides a prior distribution on the solution.

In this work, we develop thermoacoustic neural networks for learning thermoacoustic oscillations from data. The objectives of this paper are three-fold: (i) develop neural networks architectures, in which the thermoacoustic physics is both hard and soft constrained in the architecture; (ii) apply and compare the neural networks to infer nonlinear oscillations from data; and (iii) infer the heat release rate from the trained model. For this purpose, we enforce constraints in both time and space domains. First, we impose the periodicity of the underlying solution via periodic activation functions. Second, following a separation of variables of the neural network in the spirit of Galerkin decomposition, we enforce spatial structures using a physically-motivated choice of spatial modes. In addition, the training can be informed by a physics residual that penalises solutions that violate conservation of mass, momentum, and energy. This physical loss term helps regularising in case of overfitting.

This paper is organised as follows. In Section II, we provide a background on thermo-

coustic systems and feedforward neural networks. In Section III, we develop neural networks that are constrained based on the physics of thermoacoustics. In Sections IV and V, we apply the developed thermoacoustic neural networks on the data from a Rijke tube and data from a higher-fidelity model.

II. BACKGROUND ON THERMOACOUSTICS AND FEEDFORWARD NEURAL NETWORKS

A. Thermoacoustic system

We consider a thermoacoustic system composed of a straight duct of length \tilde{L} with a compact heat source located at $\tilde{x} = \tilde{x}_f$, where \sim denotes a dimensional variable. We make the following assumptions about the system: 1) the acoustics are one-dimensional, i.e., the tube is sufficiently longer than its diameter for the cut-on frequency to be high enough for longitudinal acoustics only to propagate; 2) the heat release from the flame acts as a pointwise monopole source of sound (compact assumption); 3) there is no entropy wave convection; 4) boundary conditions are ideal; and 5) effects of viscosity and heat conduction are negligible.

The dynamics are governed by the dimensional equations derived from mass, momentum, and energy conservation. Modelling the heat source as a compact source results in two duct segments related by jump conditions that are enforced at the heat source location. For brevity, the suffices 1 and 2 will denote conditions before and after the flame, i.e., $\tilde{x} = \tilde{x}_{f,1}$ and $\tilde{x} = \tilde{x}_{f,2}$, respectively. The jump conditions are found from mass, momentum, and energy fluxes across the flame with the ideal gas law [e.g., 27]. The governing equations and the jump conditions are linearised by assuming that the flow variables can be expressed as infinitesimal perturbations on top of a mean-flow, i.e., $(\tilde{\cdot}) = (\tilde{\cdot}) + (\tilde{\cdot})'$, where \sim denotes the steady mean-flow variable and $'$ denotes the unsteady perturbations. Under the low Mach number assumption, the acoustics are governed by [e.g., 28]

$$\frac{\partial \tilde{\rho}'}{\partial \tilde{t}} + \tilde{\tilde{u}} \frac{\partial \tilde{\rho}'}{\partial \tilde{x}} + \tilde{\tilde{\rho}} \frac{\partial \tilde{u}'}{\partial \tilde{x}} = 0, \quad (1a)$$

$$\tilde{\tilde{\rho}} \frac{\partial \tilde{u}'}{\partial \tilde{t}} + \tilde{\tilde{u}} \frac{\partial \tilde{u}'}{\partial \tilde{x}} + \frac{\partial \tilde{p}'}{\partial \tilde{x}} = 0, \quad (1b)$$

$$\frac{\partial \tilde{p}'}{\partial \tilde{t}} + \tilde{\tilde{u}} \frac{\partial \tilde{p}'}{\partial \tilde{x}} + \gamma \tilde{\tilde{\rho}} \frac{\partial \tilde{u}'}{\partial \tilde{x}} = (\gamma - 1) \frac{\tilde{\dot{q}'}}{\tilde{A}} \delta(\tilde{x} - \tilde{x}_f), \quad (1c)$$

where $\tilde{\rho}$ is the density; \tilde{u} is the velocity; \tilde{p} is the pressure; \tilde{q} is the heat release rate; \tilde{A} is the cross-sectional area of the duct; γ is the heat capacity ratio; and δ is the Dirac delta distribution. By eliminating the mean density and temperature in the downstream region using mass conservation, $[\tilde{A}\tilde{\rho}\tilde{u}]_1^2 = 0$, the jump conditions across the flame are [11, 27]

$$[\tilde{A}\tilde{p}']_1^2 + \tilde{A}_1\tilde{u}_1[\tilde{u}]_1^2\tilde{\rho}'_1 + \tilde{A}_1\tilde{\rho}_1\tilde{u}_1[\tilde{u}']_1^2 + \tilde{A}_1\tilde{\rho}_1\tilde{u}'_1[\tilde{u}]_1^2 - [\tilde{A}]_1^2\tilde{p}'_1 = 0, \quad (2a)$$

$$\frac{\gamma}{(\gamma-1)} \left[\tilde{A}\tilde{p}'\tilde{u} + \tilde{A}\tilde{\rho}\tilde{u}' \right]_1^2 + \frac{1}{2}\tilde{A}_1\tilde{u}_1[\tilde{u}^2]_1^2\tilde{\rho}'_1 + \tilde{A}_1\tilde{u}_1 \left(\left(\frac{1}{2}\tilde{u}_2^2 - \frac{3}{2}\tilde{u}_1^2 \right) \tilde{u}'_1 + \tilde{u}_1\tilde{u}_2\tilde{u}'_2 \right) = \tilde{q}. \quad (2b)$$

In Sections IV and V, the governing equations are employed with different simplifications to generate the synthetic data for our study. We will impose the governing equations in their PDE form (1) as soft-constraints during the training of our neural networks. This approach will be discussed in more detail in Section III C. Furthermore, we will leverage the energy jump condition (2b) in order to algebraically infer the time series of the heat release rate, \tilde{q} from the trained model.

B. Feedforward neural networks

Given measured data $\mathcal{D} = \{x_k, t_k, p'_{meas}(x_k, t_k), u'_{meas}(x_k, t_k)\}_{k=1}^N$, the goal is to learn the underlying functions $p'(x, t)$ and $u'(x, t)$. A feedforward neural network (FNN) defines a nonlinear mapping between an input and an output vector. Let \mathbf{y} be an input vector, a multi-layer feedforward neural network $\mathbf{f} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_L}$ is defined as

$$\mathbf{f}(\mathbf{y}) = \mathbf{W}^{(L)}\phi^{(L-1)}(\mathbf{W}^{(L-1)} \dots \phi^{(1)}(\mathbf{W}^{(1)}\mathbf{y} + \mathbf{b}^{(1)}) \dots + \mathbf{b}^{(L-1)}) + \mathbf{b}^{(L)}. \quad (3)$$

In this paper, we will use the rectified linear unit (ReLU), $\phi(z) = \max\{0, z\}$, and the hyperbolic tangent, $\phi(z) = \tanh(z)$ as activation functions. The multi-layered composition of these operations enables neural networks to approximate nonlinear functions in a given range [29]. The FNN that models solutions from the Rijke tube takes the spatial and time coordinates, x and t , as inputs to provide the pressure and velocity fluctuations, $p'(x, t)$ and $u'(x, t)$, i.e., $\mathbf{f} : (x, t) \rightarrow (\hat{p}', \hat{u}')$. This is a standard architecture that is fully data-driven, i.e., no prior knowledge is embedded in the network (Figure 1). The network's weights and biases are optimized with respect to a loss function \mathcal{L}

$$\boldsymbol{\chi}^* = \arg \min_{\boldsymbol{\chi}} \mathcal{L}(\boldsymbol{\chi}). \quad (4)$$

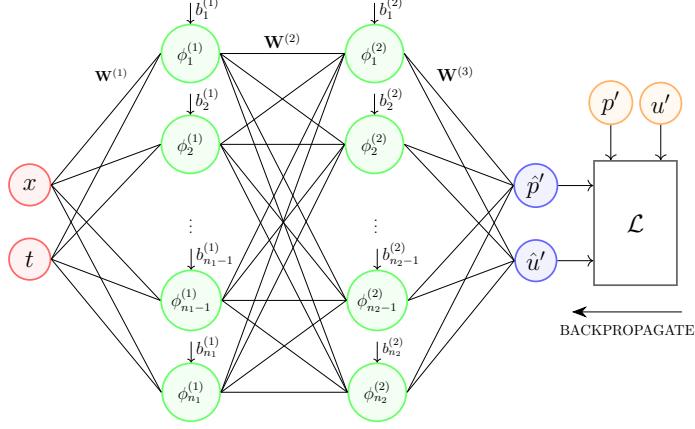


FIG. 1. Standard feedforward neural network architecture with no prior knowledge embedded in the architecture. The outputs are the acoustic pressure and velocity as functions of time and space.

When no physics-constraint is imposed, the data-driven loss is quantified by the mean-squared error (MSE) between the predictions and the data

$$\mathcal{L}_{DD} = \frac{1}{N} \sum_{k=1}^N \frac{1}{2} (p'_{meas}(x_k, t_k) - \hat{p}'(x_k, t_k))^2 + (u'_{meas}(x_k, t_k) - \hat{u}'(x_k, t_k))^2, \quad (5)$$

where $\hat{p}'(x_k, t_k)) = f_1(x_k, t_k; \chi)$ and $\hat{u}'(x_k, t_k)) = f_2(x_k, t_k; \chi)$ are the outputs of the neural network.

III. ACOUSTIC NEURAL NETWORKS

The dynamics of thermoacoustic oscillations are dominated by unstable eigenfunctions [e.g., 28], which are periodic oscillations in time. We propose neural networks that can naturally infer periodic dynamics, which are the basis functions of nonlinear thermoacoustic behaviours. The proposed networks embed the prior knowledge through the activation functions (Section III A), through the architecture (hard constraint, Section III B), and through a penalization loss function in the training (soft constraint, Section III C).

A. Unconstrained networks with periodic activations

Given enough capacity by means of number of neurons and layers, FNNs can fit the data within the training range. This can be considered as an interpolation problem, by which

we mean that the network predicts on data points that is within the bounds of the training input range. When the prediction is performed for scenarios outside the training input range, we call this an extrapolation problem. Thermoacoustics are dominated by acoustic eigenfunctions that are periodic in space and time. Therefore, we propose using periodic activations, e.g. $\sin(z)$, in the thermoacoustic FNNs. The physics of the system, namely the periodic nature of the solutions, is hard-coded in the network itself via the choice of activation function, which provides an inductive bias on the function space of the network and greatly improves extrapolation [30] [31]. The weights of a layer l with the sine activation are initialised from a uniform distribution in the range $[-\sqrt{3/\text{fan}_{\text{in}}}, \sqrt{3/\text{fan}_{\text{in}}}]$, where fan_{in} is the number of neurons in the layer $n^{(l-1)}$ [30].

B. Hard-constrained networks

FNNs are flexible and useful tools for function approximation, however, they can suffer from under- or overfitting, especially in the case of scarce or noisy data. This requires a careful hyperparameter tuning, which becomes computationally expensive in proportion with the size of network. Even then, the network may not be able to capture some features of the solution because of missing data. In order to counteract these shortcomings, we exploit the physical knowledge about the spatiotemporal basis of the thermoacoustic solution . Assuming a zero Mach number, the dynamics are governed by the non-dimensional equations [e.g., 32],

$$\bar{\rho} \frac{\partial u'}{\partial t} + \frac{\partial p'}{\partial x} = 0, \quad (6a)$$

$$\frac{\partial p'}{\partial t} + \frac{\partial u'}{\partial x} + \zeta p = \dot{q}' \delta(x - x_f), \quad (6b)$$

where the density can be modelled as

$$\bar{\rho} = \begin{cases} \bar{\rho}_1 & 0 \leq x \leq x_f, \\ \bar{\rho}_2 & x_f \leq x \leq 1, \end{cases} \quad (7)$$

and ζ represents the frequency-dependent acoustic damping. The dimensional variables have been scaled as, $x = \tilde{x}/\tilde{L}$; $t = \tilde{t}\tilde{c}/\tilde{L}$, where \tilde{c} is the mean speed of sound; $u' = \tilde{u}'/\tilde{c}$; $\rho = \tilde{\rho}'/\tilde{\rho}$; $p = \tilde{p}'/(\tilde{\rho}\tilde{c}^2)$; $\dot{q}' = \tilde{q}'(\gamma - 1)/(\tilde{\rho}\tilde{c}^3)$. The boundary conditions are fully-reflective, i.e., $p'(x = 0) = p'(x = 1) = 0$. A physically-justified method to solve the set of PDEs (6)

is to project the equations onto the acoustic eigenfunctions Magri and Juniper [32],

$$p'(x, t) = \sum_{j=1}^{N_g} \begin{cases} -\mu_j(t) \sin(\omega_j \sqrt{\bar{\rho}_1} x), & 0 \leq x \leq x_f, \\ -\mu_j(t) \left(\frac{\sin \gamma_j}{\sin \beta_j} \right) \sin(\omega_j \sqrt{\bar{\rho}_2} (1-x)), & x_f \leq x \leq 1, \end{cases} \quad (8a)$$

$$u'(x, t) = \sum_{j=1}^{N_g} \begin{cases} \eta_j(t) \frac{1}{\sqrt{\bar{\rho}_1}} \cos(\omega_j \sqrt{\bar{\rho}_1} x), & 0 \leq x \leq x_f, \\ -\eta_j(t) \frac{1}{\sqrt{\bar{\rho}_2}} \left(\frac{\sin \gamma_j}{\sin \beta_j} \right) \cos(\omega_j \sqrt{\bar{\rho}_2} (1-x)), & x_f \leq x \leq 1, \end{cases} \quad (8b)$$

where N_g is the number of modes retained in the approximation, and

$$\gamma_j = \omega_j \sqrt{\bar{\rho}_1} x_f, \quad \beta_j = \omega_j \sqrt{\bar{\rho}_2} (1 - x_f).$$

The acoustic angular frequencies ω_j are the solutions of the dispersion relationship,

$$\sin \beta_j \cos \gamma_j + \cos \beta_j \sin \gamma_j \sqrt{\frac{\bar{\rho}_1}{\bar{\rho}_2}} = 0. \quad (9)$$

In the limit that there is no jump in mean-flow density, i.e., $\bar{\rho}_1 = \bar{\rho}_2$, $\omega_j = j\pi$. This function decomposition is referred to as the Galerkin method.

1. Galerkin neural networks

We propose a network structure inspired by the Galerkin decomposition of the system. The chosen Galerkin modes (8) are a known nonlinear transformation of the spatial coordinate x , which is introduced in the network. We will refer to this network architecture as the *Galerkin network (GalNN)*. The Galerkin network is composed of a part that transforms x into the *a-priori* known Galerkin modes, and a second part that is a FNN that takes t as an input and provides the unknown Galerkin amplitudes, $\mu(t)$ and $\eta(t)$, as outputs. The pressure and velocity are determined from the estimated Galerkin amplitudes using (8). This architecture is shown in Figure 2(a). The time evolution of the Galerkin amplitudes can be obtained as an intermediary step. This network automatically satisfies the acoustic boundary conditions.

Whereas the acoustic pressure is continuous across the flame, the velocity fluctuations undergo a discontinuity due to the dilation from the heat-release rate. When such a discontinuity is approximated with continuous mode shapes, such as the Galerkin modes we are using, the Gibbs phenomenon manifests itself as high-frequency oscillations around the

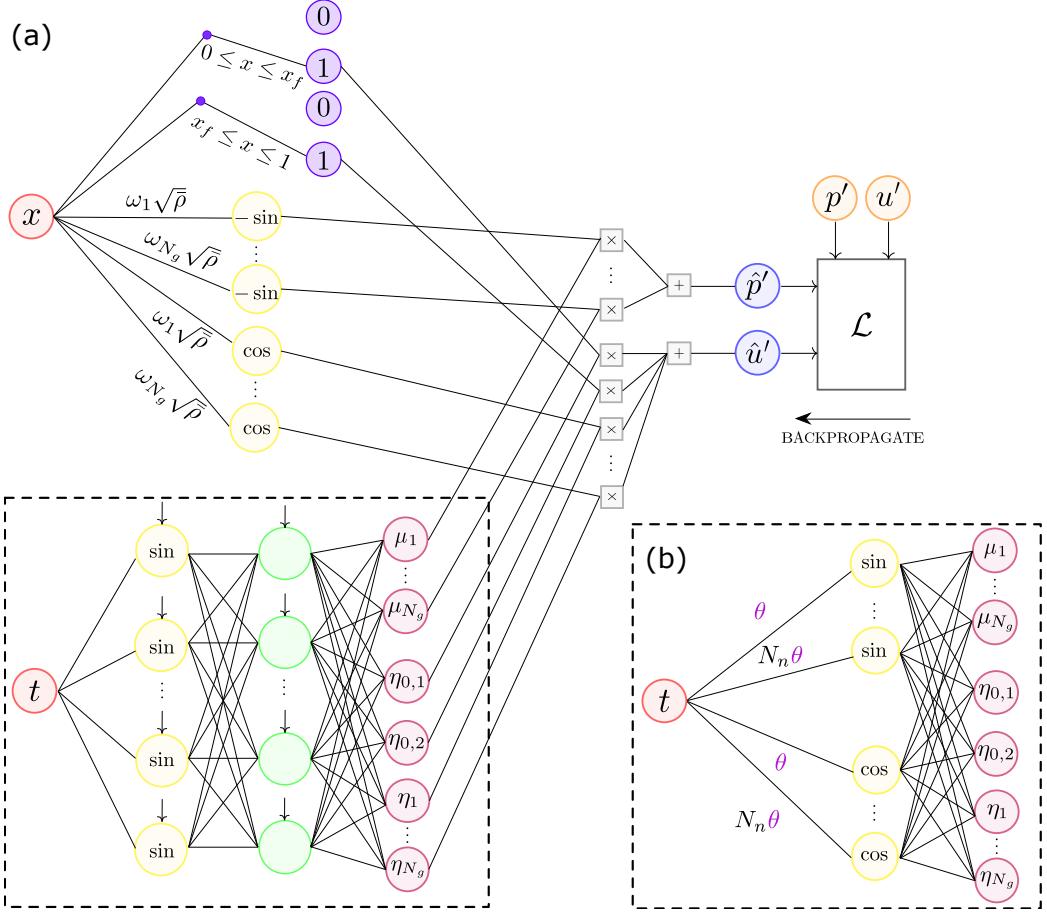


FIG. 2. (a) Galerkin neural network architecture (*GalNN*) separates the feed-forward neural network into spatial and temporal branches, where the physics of thermoacoustics are constrained in a hard fashion by a physical spatial basis, the acoustic eigenfunctions. Discontinuity jump modes are illustrated as switches depending on the upstream or downstream location, these modes will be used for higher-fidelity data. (b) Temporal branch of the Galerkin neural network is tailored for periodic bases such that the fundamental angular frequency is enforced. It is reconstructed so that the weights are integer multiples of only one trainable fundamental angular frequency.

discontinuity, which cause an unphysical behaviour. To physically capture the discontinuity and eliminate unphysical oscillations in the predictions, we propose to add two new spatial

modes in the velocity modes

$$\Psi_{0,1}(x) = \begin{cases} 1, & 0 \leq x \leq x_f, \\ 0, & x_f < x \leq 1, \end{cases} \quad (10a)$$

$$\Psi_{0,2}(x) = \begin{cases} 0, & 0 \leq x < x_f, \\ 1, & x_f \leq x \leq 1. \end{cases} \quad (10b)$$

These modes are multiplied by independent coefficients, $\eta_{0,1}$ and $\eta_{0,2}$, which allows them to capture a jump discontinuity at the flame location.

2. Periodic Galerkin networks

To tailor the Galerkin neural network to limit-cycle solution, we enforce periodic activations (Section III A). For simplicity, if we consider a single-layer periodically activated GalNN, then the outputs of the network are given as a sum of sinusoids, where the hidden layer weights represent the angular frequencies and the biases represent the phases, i.e., $\sin(W^{(1)}t + b^{(1)})$. For a signal given as a sum of sinusoids, its angular frequency is the greatest common divisor of the angular frequencies of its components. For a limit cycle, the greatest common divisor is its fundamental angular frequency, and the other frequencies are its harmonics. This applies to a periodically activated neural network as well. However, the weights of the neural network are initialised randomly and then trained on a finite amount of data, so it is numerically unlikely for the learned weights to be exact integer multiples of a fundamental frequency. To overcome this numerical challenge, we add a further constraint on the GalNN to guarantee periodic behaviour with the fundamental frequency of the limit cycle. We create a hidden layer that takes time as input and outputs $[\sin(m\theta), \cos(m\theta)]$, $m = 1, 2, \dots, M$, such that the weights in the periodic activations are integer multiples of one trainable variable, θ . The variable θ is initialised to the non-dimensional angular frequency of the acoustic system, π . Hence, upon training, θ corresponds to the angular frequency of the limit cycle, and the number of harmonics, M can be regarded as a hyperparameter. Since we have both sine and cosine activations for the same angular frequency, the phase information is automatically captured, hence we do not require a bias term. The temporal branch of the GalNN tailored for limit cycles is shown in Figure 2 (b).

TABLE I. Thermoacoustic neural networks. All of the networks below can be equipped with a soft constraint that penalises non-physical solutions in the loss function (see Sections III C). When the soft constraint is present, the abbreviation will be prefixed with PI (physics-informed). For example, a Galerkin network with the soft constraint will be abbreviated as PI-GalNN.

Network	Abbreviation	Constraint	Separation of time and space	Can extrapolate in time	Section
Feedforward neural network	FNN	No	No	No	II B
Periodically-activated feedforward neural network	P-FNN	No	No	Yes - via activation function	III A
Galerkin neural network	GalNN	Hard	Yes	Yes - can be further tailored to guarantee fundamental frequency	III B 1, III B 2

C. Soft Constraints

Prior knowledge can be embedded as a penalization term in the loss function to optimize during training [e.g., 24, 25]. In thermoacoustics, neural network predictions should fulfill the acoustic conservation laws, therefore, we penalize solutions that violate momentum and energy equations in the loss function

$$\mathcal{L} = \lambda_{DD}\mathcal{L}_{DD} + \lambda_M\mathcal{L}_M + \lambda_E\mathcal{L}_E, \quad (11)$$

where \mathcal{L}_{DD} is the data-driven loss (5), and \mathcal{L}_M and \mathcal{L}_E are the residuals from the momentum (1b) and energy (1c) equations, respectively. The regularization factors λ_{DD} , λ_M , and λ_E are hyperparameters. The residuals are found by evaluating the non-dimensional form of the governing PDEs (1) and (6) depending on the low-fidelity physical model of the thermoacoustic system. Expressing the momentum equation (1b) compactly as $F_M(\partial p'/\partial x, \partial u'/\partial x, \partial u'/\partial t) = 0$ and the energy equation (1c) as $F_E(p', \partial p'/\partial x, \partial u'/\partial t, \partial u'/\partial x, \dot{q}') =$

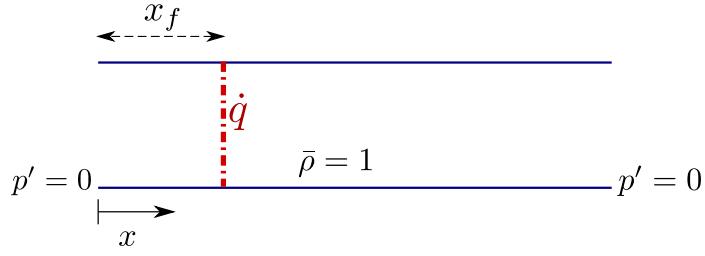


FIG. 3. Schematic of Rijke tube.

0, yield the physics-informed losses

$$\mathcal{L}_M = \frac{1}{N + N_s} \sum_{k=1}^{N+N_s} F_M \left(\frac{\partial \hat{p}'(x_k, t_k)}{\partial x}, \frac{\partial \hat{u}'(x_k, t_k)}{\partial x}, \frac{\partial \hat{u}'(x_k, t_k)}{\partial t} \right)^2, \quad (12a)$$

$$\mathcal{L}_E = \frac{1}{N + N_s} \sum_{k=1}^{N+N_s} F_E \left(\hat{p}'(x_k, t_k), \frac{\partial \hat{p}'(x_k, t_k)}{\partial x}, \frac{\partial \hat{u}'(x_k, t_k)}{\partial t}, \frac{\partial \hat{u}'(x_k, t_k)}{\partial x}, \hat{q}'(t_k) \right)^2, \quad (12b)$$

where N is the number of training data points, i.e., $\{x_k, t_k\}_{k=1}^N$, and N_s is the number of uniformly sampled points over the whole training domain. As the physical loss is not data-driven, we can evaluate it at any location in time and space, thus enabling better interpolation between the given data points, especially in the case of sparse measurements. Obtaining \mathcal{L}_M and \mathcal{L}_E requires the evaluation of the physics equations using the network. For this purpose, we employ automatic differentiation of the output variables of the network with respect to the input variables to obtain the gradients in the governing equations. For the implementation of the automatic differentiation of the network, `GradientTape` functionality from TensorFlow [33] is used.

The thermoacoustic neural networks are summarised in Table I

IV. RESULTS ON THE RIJKE TUBE

A. Rijke tube

The Rijke tube (Figure 3) is a prototypical setup, which captures the qualitative nonlinear dynamics and bifurcations observed in real-life applications [14]. In this model, we further assume that the mean flow has a zero Mach number with uniform density. We solve the associated PDEs (6) using the Galerkin decomposition (8), with $\bar{\rho}_1 = \bar{\rho}_2$, which implies that

$\omega_j = j\pi$. By substituting the pressure and velocity variables in (6) with their Galerkin decompositions in (8) and projecting the dynamics onto the Galerkin modes, the dynamics of the Galerkin variables η_j and μ_j are described by a $2N_g$ -dimensional system of ordinary differential equations

$$\dot{\eta}_j - \mu_j j\pi = 0, \quad (13a)$$

$$\dot{\mu}_j + \eta_j j\pi + \zeta_j \mu_j + 2\dot{q}' \sin(j\pi x_f) = 0. \quad (13b)$$

The modal damping is given by $\zeta_j = c_1 j^2 + c_2 j^{1/2}$ [34], and the heat release rate is described by a modified King's law $\dot{q}' = \beta \left(\sqrt{|1 + u'(x_f, t - \tau)|} - 1 \right)$, where β and τ are the heat release strength and the flame time delay, respectively [35]. The time-delayed problem is transformed to an initial value problem via an advection function with the dummy variable v so that it can be solved with a standard time-marching scheme [36]

$$\frac{\partial v}{\partial t} + \frac{1}{\tau} \frac{\partial v}{\partial X} = 0, \quad 0 \leq X \leq 1, \quad v(X = 0, t) = u'(x_f, t). \quad (14)$$

The PDE (14) is discretised using a Chebyshev spectral method [37]. When calculating the heat-release term in the physics-informed loss, the time-delayed velocity at the flame can be predicted from the network as $\hat{u}'(x_f, t_k - \tau)$ (for the GalNN $\hat{u}'(x_f, t_k - \tau) = \sum_{j=1}^{N_g} \hat{\eta}_j(t_k - \tau) \cos(j\pi x_f)$, i.e. by evaluating the network at $(x = x_f, t = t_k - \tau)$). Therefore, since the model takes time as an input for both FNN and GalNN architectures, there is no need to account for the time delay with a separate dummy variable. Also, note that the evaluation of the heat-release law requires the system parameters x_f , β , and τ to be known. For the FNN, the implementation details concerning the calculation of heat release and modal damping are provided in the Appendix A. For the GalNN, as Galerkin amplitudes are available, the physical residuals of the Rijke tube can be directly computed from the ODEs (13).

B. Periodically-activated feedforward neural networks

We simulate the Rijke tube system (6) with the parameters; $N_g = 10$, $N_c = 10$, $x_f = 0.2$, $\beta = 5.7$, $\tau = 0.2$, $c_1 = 0.1$, $c_2 = 0.06$. After a transient period, this system settles onto a limit cycle with a period of ≈ 1.927 time units. Discarding the transient, a training dataset is generated from the pressure and velocity timeseries sampled at the non-dimensional frequency of 1 kHz. Before training, the inputs are standardized to have a zero mean and unit

TABLE II. Summary of neural network model properties trained on the Rijke tube data.

	ReLU FNN	tanh FNN	sin FNN	sin-ReLU FNN	GalNN	Per. GalNN
# hidden layers	5	5	3	2	2	1
Activations	ReLU	tanh	sine	sine-ReLU	sine	harmonics
# neurons	96	96	32	64	16	40
Learning rate	0.001	0.001	0.001	0.001	0.001	0.001
Optimiser				Adam		
Batch size				64		

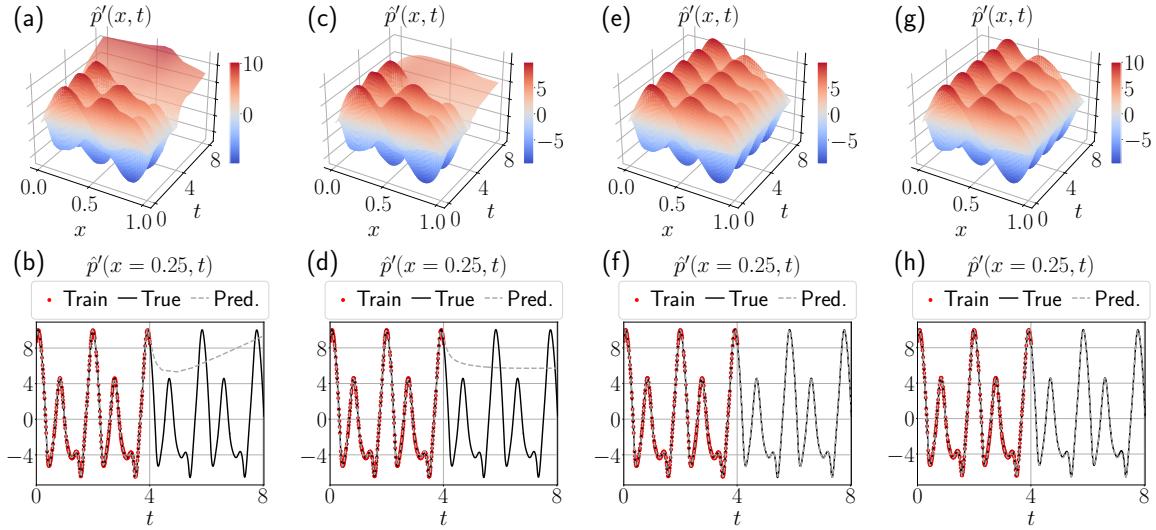


FIG. 4. Predictions of acoustic pressure fluctuations by feedforward neural networks with different activations; (a-b) ReLU, (c-d) tanh, (e-f) sine in all hidden layers, (g-h) sine-ReLU (sine in the first layer, ReLU in the rest of the hidden layers). Top row shows the predictions in both time and space, bottom row shows the predictions in time at a fixed location in the tube along with the training data points and the ground truth.

variance. Figures 4(a,b) and (c,d) show the results of FNNs with ReLU and tanh activation functions when trained on this dataset. The network hyperparameters (number of layers, number of neurons, learning rate) have been tuned via a grid search, in which we chose the architectures that resulted in the smallest validation error. The network properties are provided in Table II. The ReLU network is initialised with the method of He *et al.* [38], and the tanh network with the method of Glorot and Bengio [39]. As illustrated in Figures 4

(a,b) and (c,d), FNNs with conventional activation functions fail at extrapolating periodic functions in time.

Figure 4(e,f) shows the extrapolation capability of an FNN equipped with sine activation. The activation function is modified to include a hyperparameter a such that it becomes

$$\phi(z) = \frac{1}{a} \sin(az). \quad (15)$$

This hyperparameter is used to fine tune the frequency content of the learned functions. For this dataset, it is determined as $a = 10$ with a grid search, the effect of varying a is shown in more detail in the Appendix B. Physically, since the weights are initialised as $\sim Uniform(-1.22, 1.22)$ (Section III A), multiplying this by $a = 10$, gives a good initial guess on the order of magnitude of the angular frequency for the training. Figure 4(g,h) shows the case for the sine-ReLU configuration, which can also extrapolate in time. Deeper architectures are required when training with purely ReLU or tanh networks compared to sine or sine-ReLU networks. For example, in this case we used two layers of sine and sine-ReLU vs. five layers of ReLU and tanh, and even then the ultimate training and validation losses of sine and sine-ReLU networks were one order of magnitude smaller. To conclude, using periodic activations produces more physical neural network models for thermoacoustics than the conventional activations, which can also extrapolate in time, whilst speeding up training and hyperparameter search.

C. Learning from partial state measurements

Realistically, full state measurements are difficult to perform in thermoacoustics, i.e., only pressure measurements are taken with microphones. To emulate such a scenario, we perform the training using only pressure data, discarding the velocity data. Therefore, the data-driven loss becomes

$$\mathcal{L}_{DD} = \frac{1}{N} \sum_{k=1}^N (p'(x_k, t_k) - \hat{p}'(x_k, t_k))^2. \quad (16)$$

Our aim is to learn a model that can also output velocity information without ever being exposed to measurements from it during training. Figure 5 shows the predictions of the sine feedforward and Galerkin neural networks trained with and without physics-information. When the loss is computed solely from the data, only the pressure can be learned. On the

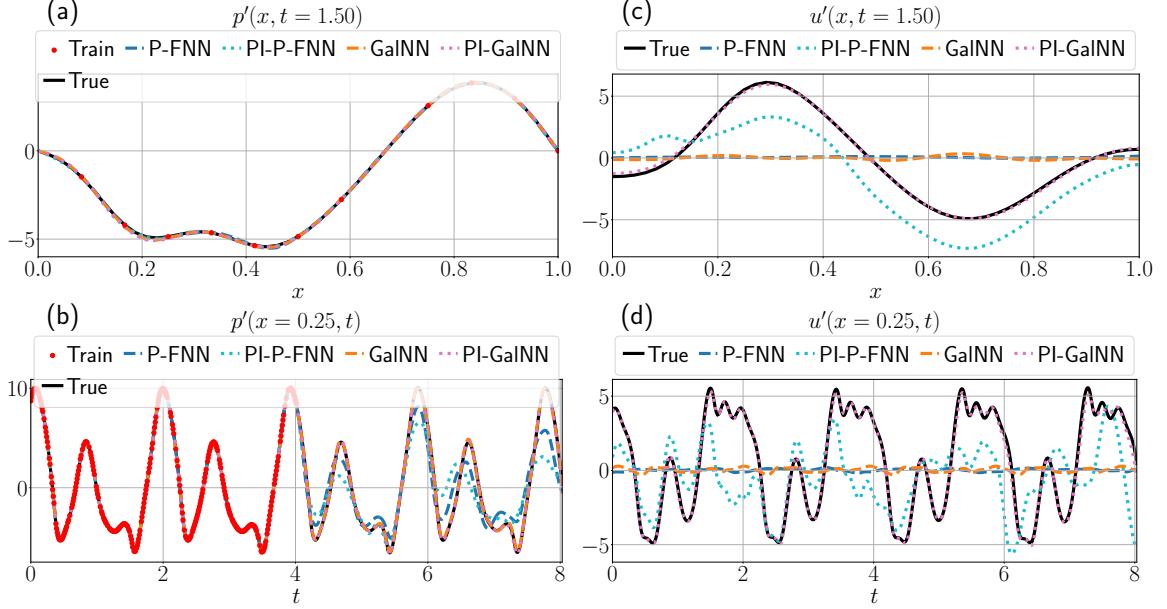


FIG. 5. Predictions of acoustic pressure (a-b) and velocity (c-d) fluctuations from only pressure observations when the training is performed with and without physics-based loss using the periodically activated feedforward neural network and Galerkin neural network. Top row shows pressure and velocity shapes along the tube at a fixed time instance, bottom row shows pressure and velocity timeseries at a fixed point in the tube.

other hand, velocity predictions are made possible by enforcing the physics via the physics-informed loss term. For the purely data-driven networks, the output weights related to velocity are not updated, hence the velocity predictions are made with the initial weight values which are small. The physics-informed forward neural network captures the trend of the velocity state, however there is a negative offset from the actual values. On the other hand, the physics-informed Galerkin network can accurately reconstruct the velocity.

D. Long-term extrapolation

We showed that networks equipped with periodic activations make accurate predictions on data points chosen in a time range right after the training. Here, we analyse how the learned models extrapolate in the long term.

1. Periodic solutions

The ideal model of a periodic solution should capture the periodicity that the system exhibits for all times. Network predictions over a long time period are plotted in Figure IV D (a) and (b), for sine FNN and GalNN, respectively. Over a long time range, we observe a beating-like phenomenon. We notice a similar occurrence with the trained weights of the first layer of the networks. In the frequency spectra of the Galerkin variables, the peaks are at the harmonics of the angular frequency of the limit cycle, which we determined as $\theta^* = 3.2605$ by inspection of the frequency spectrum and autocorrelation of the timeseries. After the training, this is captured by the weights in the first hidden layer. Recall that after the first sine activation [15] layer, we obtain $\sin(aW_1^{(1)}x + aW_2^{(1)}t + ab^{(1)})$ for the sine feedforward network, and $\sin(aW^{(1)}t + ab^{(1)})$ for the Galerkin network. The optimised weights (after multiplying with the hyperparameter a of the sine activation [15]) are placed close to the harmonics of the angular frequency of the limit cycle. Figure 7 shows the (sorted) weights of the sine feedforward network and a Galerkin network. The harmonics of the angular frequency of the limit cycle are shown in dashed horizontal lines. Although these weights are close to the harmonics of the frequency of the limit cycle, they are not exact integer multiples of each other. Even the ones clustered around one frequency slightly differ from each other. This can be observed in the power spectral density (PSD) of the predicted timeseries as well, shown in Figure IV D (d) and (e), where the predicted frequencies are clustered around the harmonics, but do not exactly match with the peak.

The network is not aware what the period of the predicted function. We overcome this issue by using the GalNN for this specific purpose as described in Section III B 2. The long term predictions are shown in Figure 6 (c) and the power spectral density of the predicted signal in Figure 6 (f). (For the periodic Galerkin network, one can also directly compute the PSD of the Galerkin variables from the weights of the output layer. The ℓ_2 -norm of the weights that multiply the sine and cosine activations at the same angular frequency gives the contribution of that frequency in the spectrum.) The estimated θ^* by the periodic Galerkin network is 3.2615 and the period of the predicted signal is $T^* = 1.9264$, for the true signal these are read from the PSD as $\theta^* = 3.2605$ and $T^* = 1.9271$. With this constrained structure, we eliminate the beating behaviour without using more training data.

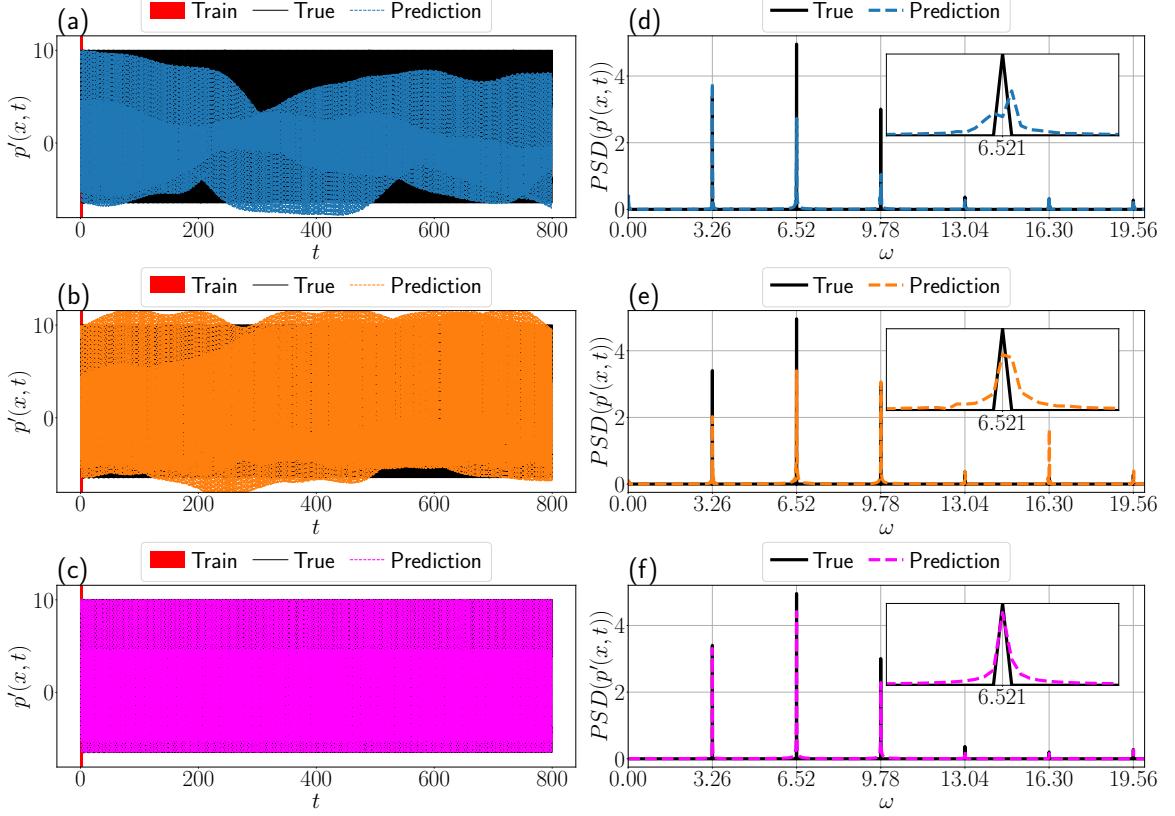


FIG. 6. Long term prediction of the pressure time series with (a) P-FNN, (b) GalNN, and (c) periodic GalNN. (The range of the training data is the first 4 time units shown inside the red segment.) Power spectral density of true and predicted time-series with (d) P-FNN, (e) GalNN, and (f) periodic GalNN.

2. Quasiperiodic and chaotic oscillations

In addition to limit cycles, thermoacoustic systems can also exhibit quasiperiodic and chaotic oscillations through bifurcations [e.g., 5]. With $x_f = 0.2$ and $\tau = 0.2$ and increasing β , the system takes a Ruelle-Takens-Newhouse route to chaos; at $\beta = 0.1$, it bifurcates from a fixed point solution to a limit cycle, at $\beta = 5.8$ from a limit cycle to a quasiperiodic attractor, and at $\beta = 6.5$ from a quasiperiodic attractor to chaos [36]. We apply our method on quasiperiodic data from $\beta = 6$ regime and chaotic data from $\beta = 7$ regime. For the quasiperiodic system, we train a Galerkin network with 2 sine layers of 96 neurons on data from the 80 time units long time series following a hyperparameter tuning with a grid search.

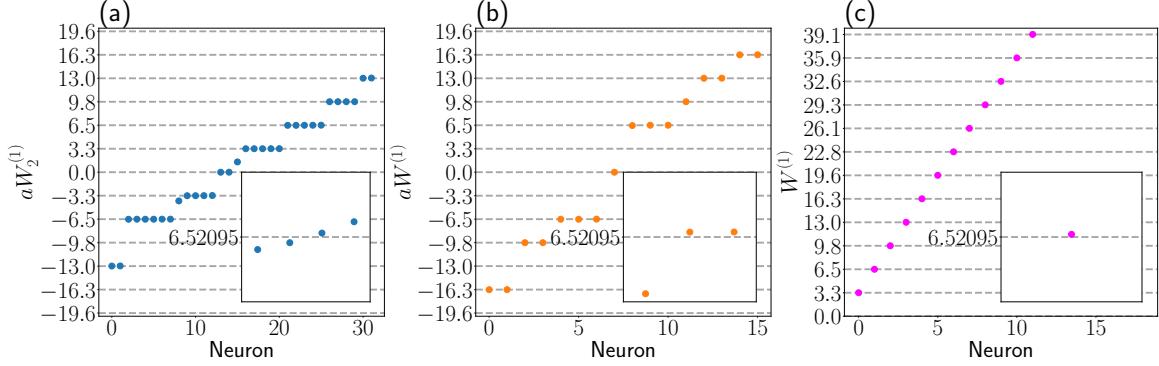


FIG. 7. Trained weights of the first hidden layers of (a) P-FNN, (b) GalNN, and (c) periodic GalNN. These weights are multiplied with the time input before periodic activation, i.e., $\sin(aW_1^{(1)}x + aW_2^{(1)}t + ab^{(1)})$ for P-FNN, $\sin(aW^{(1)}t + ab^{(1)})$ for GalNN, where a is a hyperparameter, and $\sin(W^{(1)}t)$ for periodic GalNN.

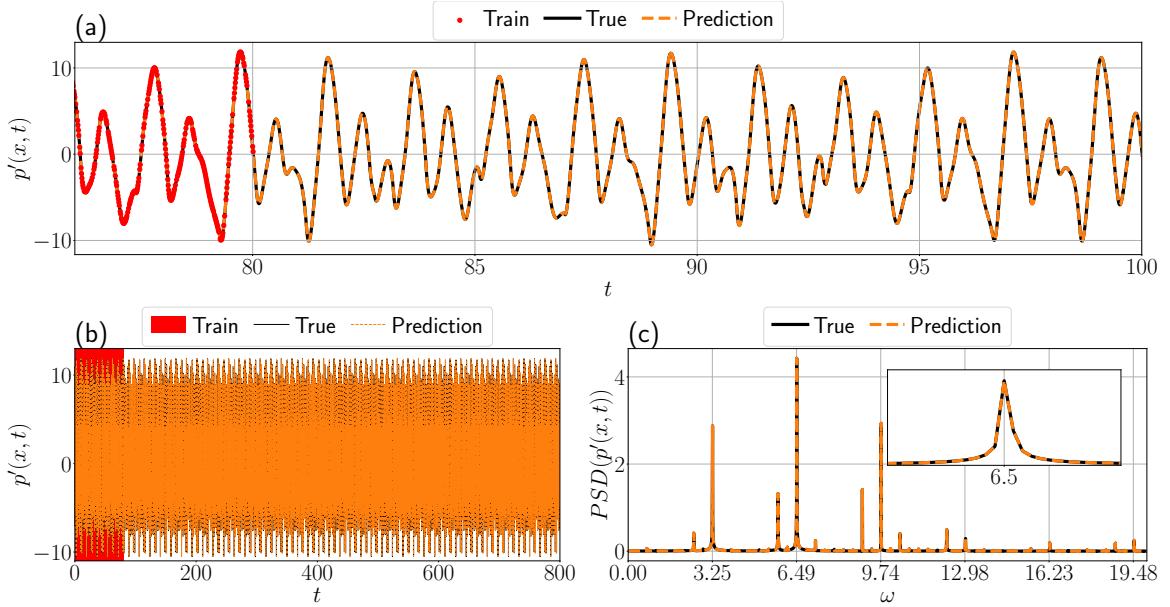


FIG. 8. Prediction of the acoustic pressure from a quasiperiodic system with GalNN, in (a) right after the training range, and in (b) long term. (The range of the training data is the first 80 time units shown inside the red segment.) Power spectral densities of ground truth and predicted time-series are shown in (c).

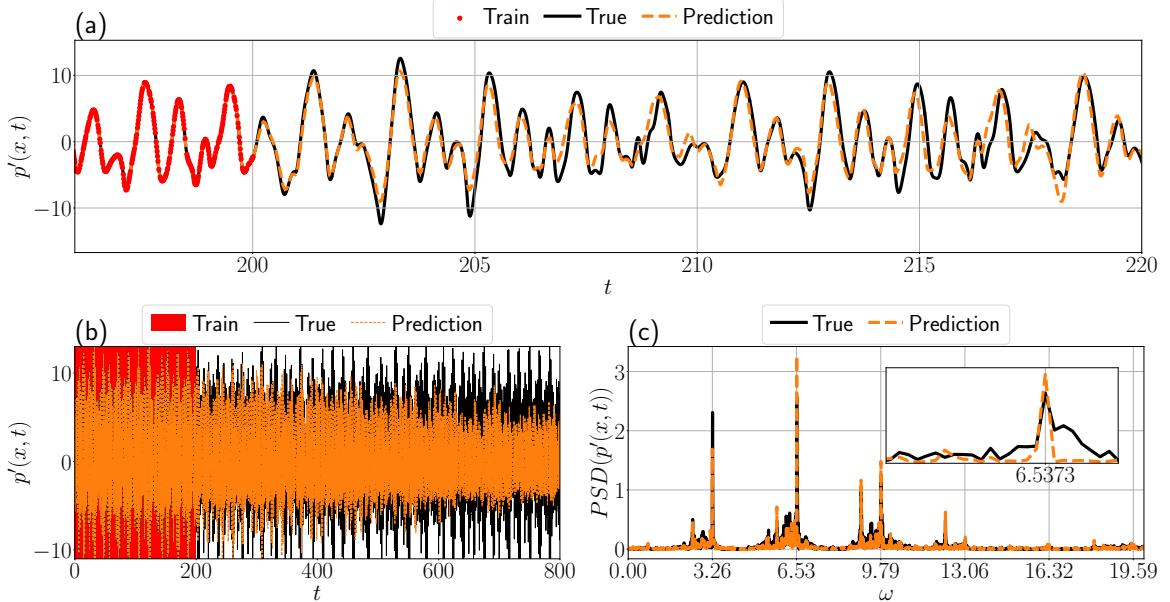


FIG. 9. Prediction of the acoustic pressure from a chaotic system with GalNN, in (a) right after the training range, and in (b) long term. (The range of the training data is the first 200 time units shown inside the red segment.) Power spectral densities of true and predicted time-series are shown in (c).

The predictions are shown in Figure 8 with the PSD. The Galerkin network can capture the quasiperiodic behaviour and the frequency spectrum correctly. For the chaotic system, we train a Galerkin network with 4 sine layers of 256 neurons on data from the 200 time units long time series. The predictions are shown in Figure 9 along with the PSD. To train on chaotic data, we required a longer time series and a deeper and wider network because the nonlinear dynamics is richer than the quasiperiodic solution. The network's prediction correctly captures the dominant frequencies and the amplitude of the true signal.

V. RESULTS ON HIGHER-FIDELITY DATA

A. Higher-fidelity data

In order to assess how well our GalNN approach generalises to higher-fidelity data, which is not generated by the Galerkin method, we employ a higher-fidelity model that includes the effects of the mean flow and has a kinematic flame model for the flame [11]. The solution of

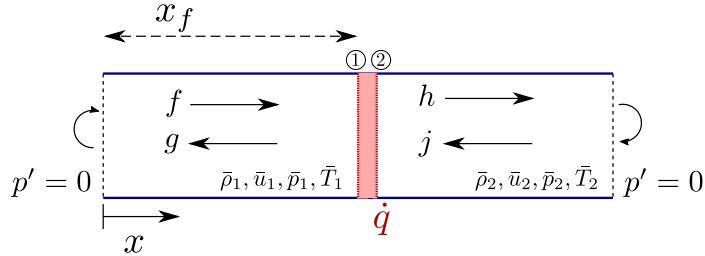


FIG. 10. Schematic of travelling wave model of Rijke tube.

the PDEs (1b) and (1c) can be obtained by a travelling wave approach in conjunction with the jump conditions (2a) and (2b) at the flame. The pressure and velocity fluctuations are expressed as functions of two acoustic travelling waves, which propagate up- and downstream of the tube (Figure 10). These waves are derived by applying the method of characteristics to the acoustic wave equation and defined as f and g , with propagation velocities $\tilde{c}_1 \pm \tilde{u}_1$ in the upstream region $\tilde{x} \leq \tilde{x}_f$; and h and j , with propagation velocities $\tilde{c}_2 \pm \tilde{u}_2$ in the downstream region $\tilde{x} \geq \tilde{x}_f$. The boundary conditions are open and fully reflective at both ends, i.e., $f(\tilde{t}) = -g(\tilde{t} - \tilde{\tau}_u)$ and $j(\tilde{t}) = -h(\tilde{t} - \tilde{\tau}_d)$, where $\tilde{\tau}_u$ and $\tilde{\tau}_d$ are the travelling times of the waves from the flame to the up- and downstream boundaries, respectively. The full set of equations that describe the dynamics of the waves are given by

$$\mathbf{X} \begin{bmatrix} g(\tilde{t}) \\ h(\tilde{t}) \end{bmatrix} = \mathbf{Y} \begin{bmatrix} g(\tilde{t} - \tilde{\tau}_u) \\ h(\tilde{t} - \tilde{\tau}_d) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\dot{q} - \dot{\bar{q}}}{A_1 \tilde{c}_1} \end{bmatrix}, \quad (17)$$

where the matrices \mathbf{X} and \mathbf{Y} are functions of the mean-flow variables and are obtained from the jump conditions (2a) and (2b). We non-dimensionalise the variables as, $x = \tilde{x}/\tilde{L}$; $t = \tilde{t}\tilde{c}_{ref}/\tilde{L}$; $u' = \tilde{u}'/\tilde{c}_{ref}$; $\rho = \tilde{\rho}'/\tilde{\rho}_{ref}$; $p = \tilde{p}'/(\tilde{\rho}_{ref}\tilde{c}_{ref}^2)$; $\dot{q}' = \tilde{q}'(\gamma - 1)/(\tilde{\rho}_{ref}\tilde{c}_{ref}^3)$ where $(\cdot)_{ref} = (\tilde{L}_u(\cdot)_1 + \tilde{L}_d(\cdot)_2)/\tilde{L}$ is chosen as the weighted average of the mean values of variable (\cdot) across the flame. We generate the data with the code implementation of the kinematic flame model from [27][40]. We simulate a system that results in limit cycle oscillations using the mean flow parameters for the inlet velocity $\tilde{u}_1 = 4 \text{ m/s}$; for the inlet temperature $\tilde{T}_1 = 293 \text{ K}$; for the inlet and outlet pressure $\tilde{P}_1 = \tilde{P}_2 = 101300 \text{ Pa}$; and for the heat release rate $\tilde{Q} = 2000 \text{ W}$. This is a low Mach number configuration with $\bar{M}_1 = 0.0117$ and $\bar{M}_2 = 0.0107$. The non-dimensional flame location is at $x = 0.61$. While creating the training and validation sets, we consider two types of validation; (i) validation of the interpolation capability, and (ii) validation of the extrapolation capability, i.e. when the data is sampled

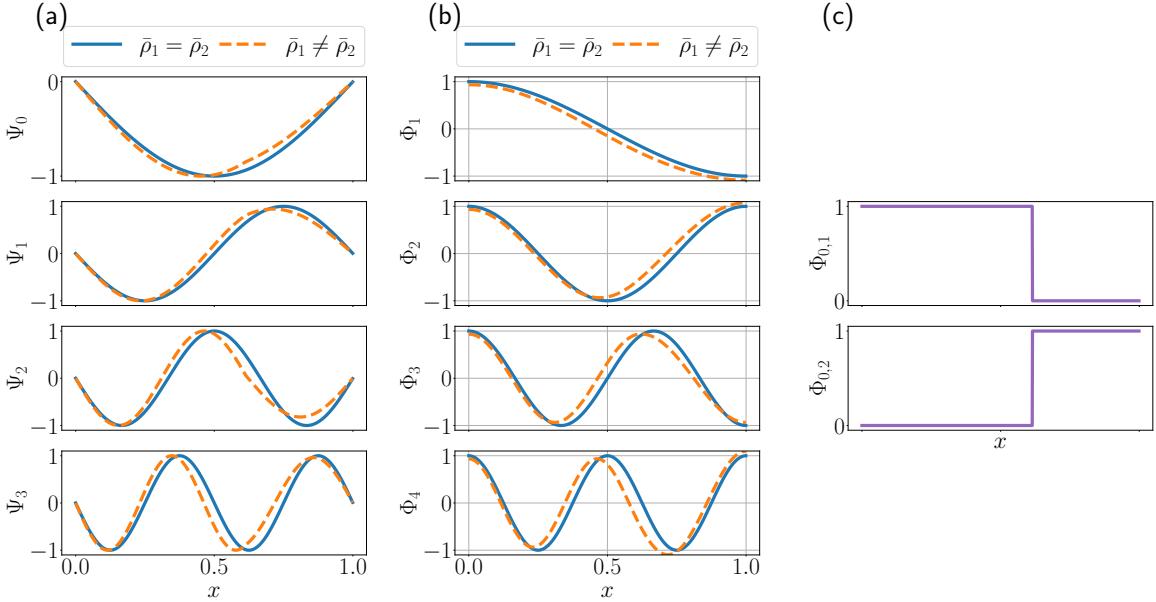


FIG. 11. Spatial modes, (a) pressure, (b) velocity, (c) velocity jump modes. Galerkin modes with $\omega_j = j\pi$, i.e., when $\bar{\rho}_1 = \bar{\rho}_2$, are shown in straight lines; Galerkin modes determined from the dispersion relationship when $\bar{\rho}_1 \neq \bar{\rho}_2$ are shown in dashed lines.

from outside the time range of training. A good validation of the interpolation loss prevents overfitting. Validation of extrapolation loss indicates how well the model generalises beyond the time range of training. After discarding a transient, we split the first 4 non-dimensional time units long time series (approximately two periods) of the limit cycle into training and validation of the interpolation sets, assigning a 80% to 20% ratio between them. The validation data for the extrapolation is then taken as the next 4 non-dimensional time units.

B. Comparison of networks performance for reconstruction

The higher-fidelity data takes into account the mean density jump at the heat source location. Therefore, the high-fidelity data includes a jump discontinuity in the velocity shape at the flame location, which was not the case in the Galerkin decomposed model of the Rijke tube. We analyse the performance of the different networks that we have built in Section III on the reconstruction of the higher-fidelity data. We compare three configurations of the feedforward neural network with; 1) ReLU activations in all hidden layers, 2) sine activation in all hidden layers, 3) sine activation in the first layer, ReLU activation in the rest

TABLE III. Summary of neural network model properties trained on higher-fidelity data. FNNs with different activations; ReLU, sine, sine-ReLU (sine in the first layer, ReLU in the rest). GalNNs with different choice of spatial basis; 1) with $\omega_j = j\pi$, i.e., when $\bar{\rho}_1 = \bar{\rho}_2$, 2) with ω_j from the real mean flow densities, i.e., when $\bar{\rho}_1 \neq \bar{\rho}_2$, 3) with the addition of the modes for jump discontinuity.

	ReLU FNN	sin FNN	sine-ReLU FNN	GalNN (1)	GalNN (2)	GalNN (3)
# hidden layers	6	3	4	1	1	1
# Galerkin modes	-	-	-	20	20	20
Activations	ReLU	sine	sine-ReLU	harmonics	harmonics	harmonics
# neurons	64	64	32	20	20	20
Learning rate	0.01	0.01	0.01	0.0004	0.0004	0.0004
Optimiser				Adam		
Batch size				32		

of the hidden layers, and three configurations of the periodic Galerkin neural network with one hidden trainable frequency layer; 1) with $\omega_j = j\pi$ as the acoustic angular frequencies as before, i.e., when $\bar{\rho}_1 = \bar{\rho}_2$, 2) with ω_j determined from the dispersion relationship using the true mean flow density values of the system, i.e., when $\bar{\rho}_1 \neq \bar{\rho}_2$, 3) the Galerkin modes chosen as 2) with the addition of the modes for jump discontinuity [10]. The first four pressure and velocity Galerkin modes, and velocity jump modes are illustrated in Figure [11].

For each configuration of activations, the hyperparameters of the architecture and training (number of layers, number of neurons, and learning rate) have been tuned with a grid search, in which we chose the set of hyperparameters that resulted in the lowest validation loss. These properties are provided in Table [III] for each network. The training and validation loss histories belonging to all models are shown in Figure [12]. The ReLU network has higher training and interpolation losses, cannot fit the data as well as the sine and sine-ReLU networks, and cannot extrapolate. For all the other networks (sine FNN, sine-ReLU FNN, GalNN (1), GalNN (2), GalNN (3)), the validation losses are close to the training loss. The periodic Galerkin network, GalNN (3) scores the lowest losses. The periodic Galerkin networks can have similar or better prediction performance as the other deeper feedforward neural networks even with one hidden layer following a physical choice of spatial basis. Figure [13] visualises the velocity predictions and their errors between the predictions and

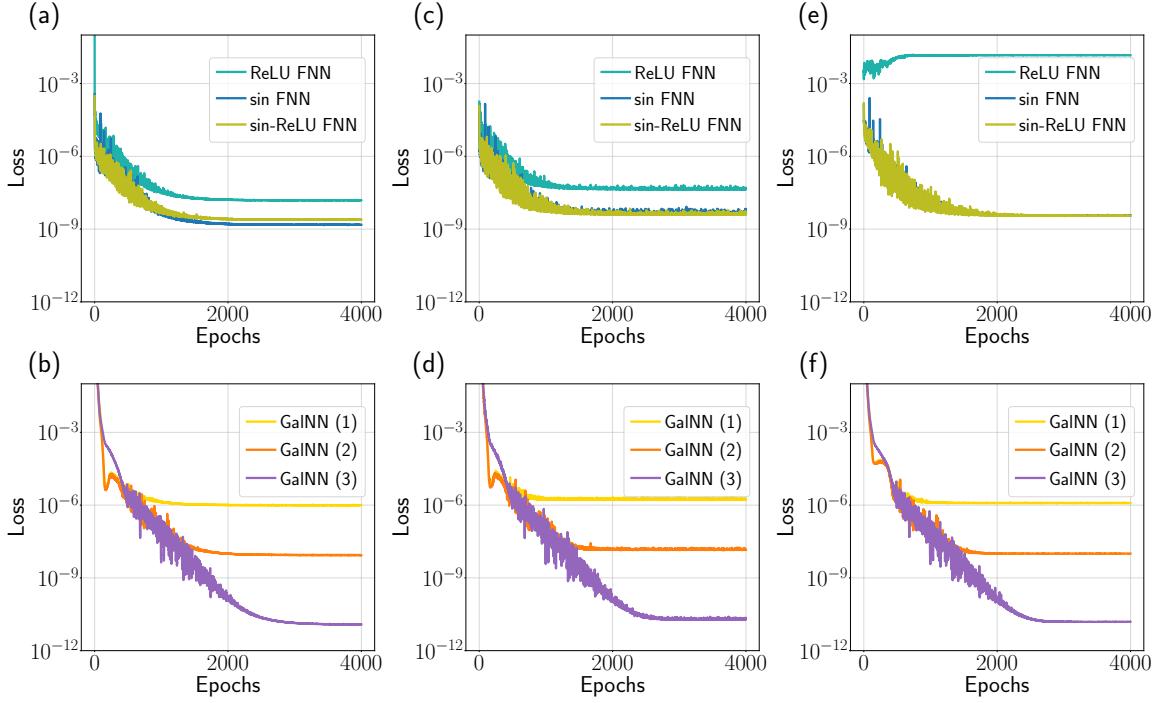


FIG. 12. (a-b) Training, (c-d) validation of interpolation, and (e-f) validation of extrapolation loss histories of different configurations of neural networks on the higher fidelity data. FNNs with different activations; ReLU, sine, sine-ReLU (sine in the first layer, ReLU in the rest). GalNNs with different choices of spatial basis; 1) with $\omega_j = j\pi$, i.e., when $\bar{\rho}_1 = \bar{\rho}_2$, 2) with ω_j from the real mean flow densities, i.e., when $\bar{\rho}_1 \neq \bar{\rho}_2$, 3) with the addition of the jump modes.

the true data obtained with the networks. In this figure, we include only the sine-ReLU network among the FNNs as a representative, since it has the lowest validation error among the FNNs along with the sine network, and qualitatively, the predictions from these two networks do not differ from each other. The GalNN (1), which does not take into account that the mean flow density changes across flame, exhibits an error pattern that resembles an offset from the true solution, which indicates a biased model. This offset between the two sets of Galerkin modes can also be recognised in Figure 11(b). The prediction is also oscillatory and the error is especially high at the boundaries. The more physically motivated choice of GalNN (2), which takes into account the mean flow density change, eliminates these errors, however the discontinuity jump still can not be captured, which leads to high frequency oscillations in the prediction around the flame location. A similar phenomenon is

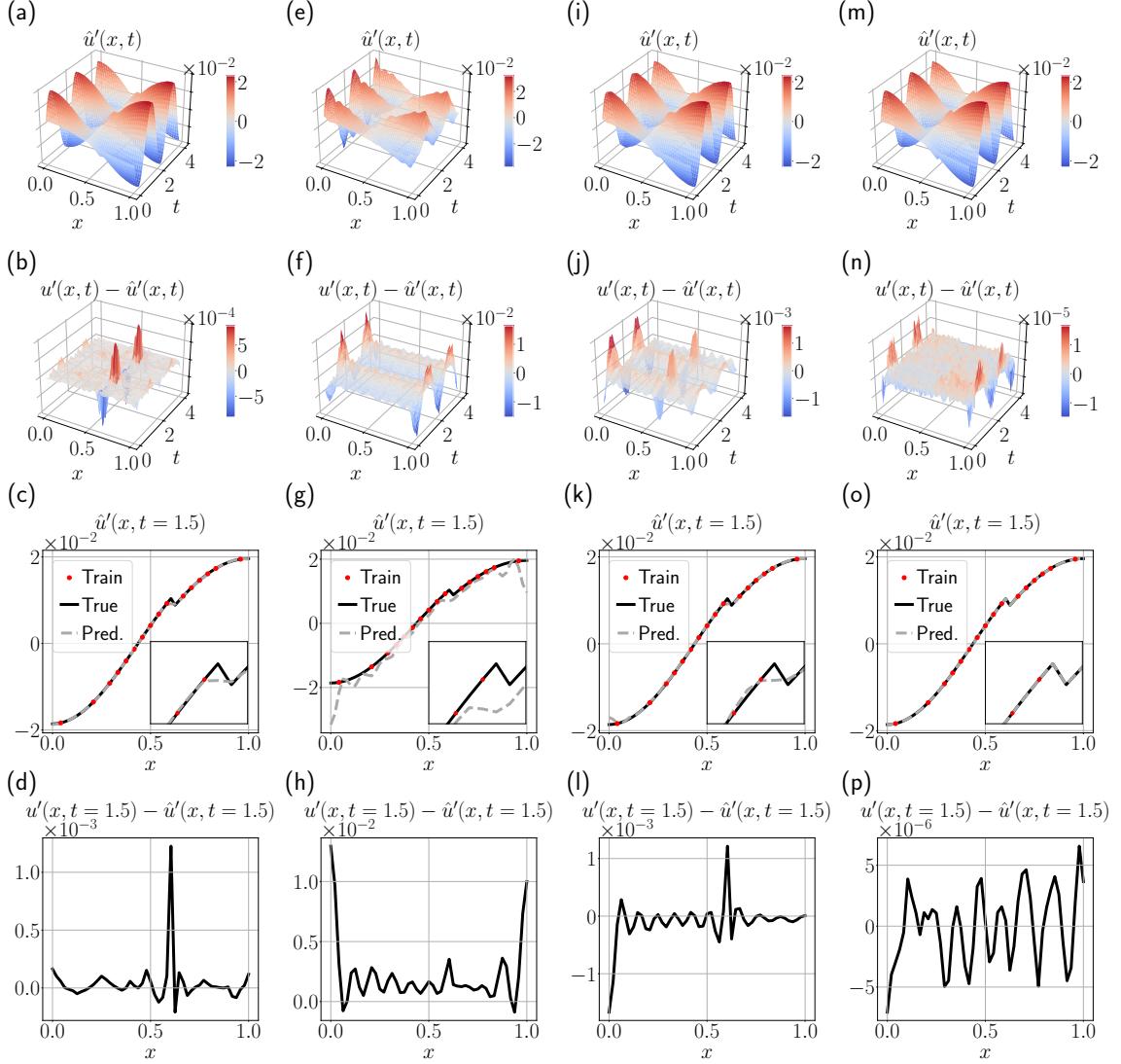


FIG. 13. Effect of choice of network architecture on the higher-fidelity data. From left to right, (a-d) sine-ReLU FNN (sine in the first layer, ReLU in the rest); (e-h) GalNN (1) $\omega_j = j\pi$; (i-l) GalNN (2) ω_j from the real mean flow density values; (m-p) GalNN (3) with the addition of the jump modes. From top to bottom, velocity prediction in time and space, error between ground truth and prediction in time and space, velocity prediction at a fixed time instance, error at a fixed time instance.

also observed for the sine-ReLU network, where the error is the highest at the flame location. This problem is overcome by the addition of the jump modes in GalNN (3), which results in the lowest training and validation errors overall. This network finds the non-dimensional

dominant angular frequency as $\theta^* = 3.3164$ and thus, the period as $T^* = 1.8946$. From the power spectral density of the true pressure signal, we read $\theta^* = 3.3166$ and $T^* = 1.8945$. The periodic Galerkin network successfully learns the correct period.

C. Robustness to sparsity and noise

We select GalNN (3) from the previous section because of its superior performance in comparison to the other architectures and investigate the regularisation methods that can be used in order to prevent overfitting in the case of corrupted or scarce data. We will compare a physics-based regularisation, i.e., the physics loss, with ℓ_1 - and ℓ_2 -norm regularisations, which are commonly used regularisations that act on the network weights as $\|W\|_1$ and $\|W\|_2$, and are added as an additional loss term. While ℓ_2 -norm prevents overfitting by reducing the magnitudes of the weights, ℓ_1 -norm induces sparsity. We inject zero mean Gaussian noise on the non-dimensional training data with a standard deviation equal to 10% of the standard deviation of the true solution function over the whole domain. Using the described regularisation methods, we train networks with 10 Galerkin modes on this new dataset this time given only on a coarse grid. The comparison of the obtained predictions when using 1) no regularisation, 2) ℓ_2 -norm, 3) ℓ_1 -norm, and 4) physics-based regularisation is shown in Figure 14. (Note that for this specific figure, the axes are at different orders of magnitude.) We have trained multiple networks with varying values of regularisation coefficients and show only the results for those with the lowest validation losses. Without regularisation, the model can fit the training data perfectly, however the interpolation points show spikes, which is a clear demonstration of overfitting. This case of overfitting emerges when there is a discrepancy between the number of sensors and the number of Galerkin modes. For the given data, the non-dimensional spacing in the x -domain, Δx , is equal to 0.1667. Nyquist-Shannon sampling theorem dictates the highest wavenumber that we can sample with this spacing, $k_{Nyquist} = \frac{\pi}{\Delta x} = 18.8496$. The wavenumbers of the Galerkin modes are determined a priori from the dispersion relationship. The highest number of Galerkin modes is then given as $N_g^* = \arg \max_j \{j \mid \omega_j \sqrt{\rho_1} \leq k_{Nyquist}, \omega_j \sqrt{\rho_2} \leq k_{Nyquist}\}$. For this system, we find $\{\omega_5 \sqrt{\rho_1}, \omega_5 \sqrt{\rho_2}\} = \{16.7795, 13.7531\}$ and $\{\omega_6 \sqrt{\rho_1}, \omega_6 \sqrt{\rho_2}\} = \{20.2816, 16.6235\}$, and thus N_g^* is 5 to be able to sample from both up- and downstream regions. (A similar argument can be made in the time domain to determine the number of neurons of the train-

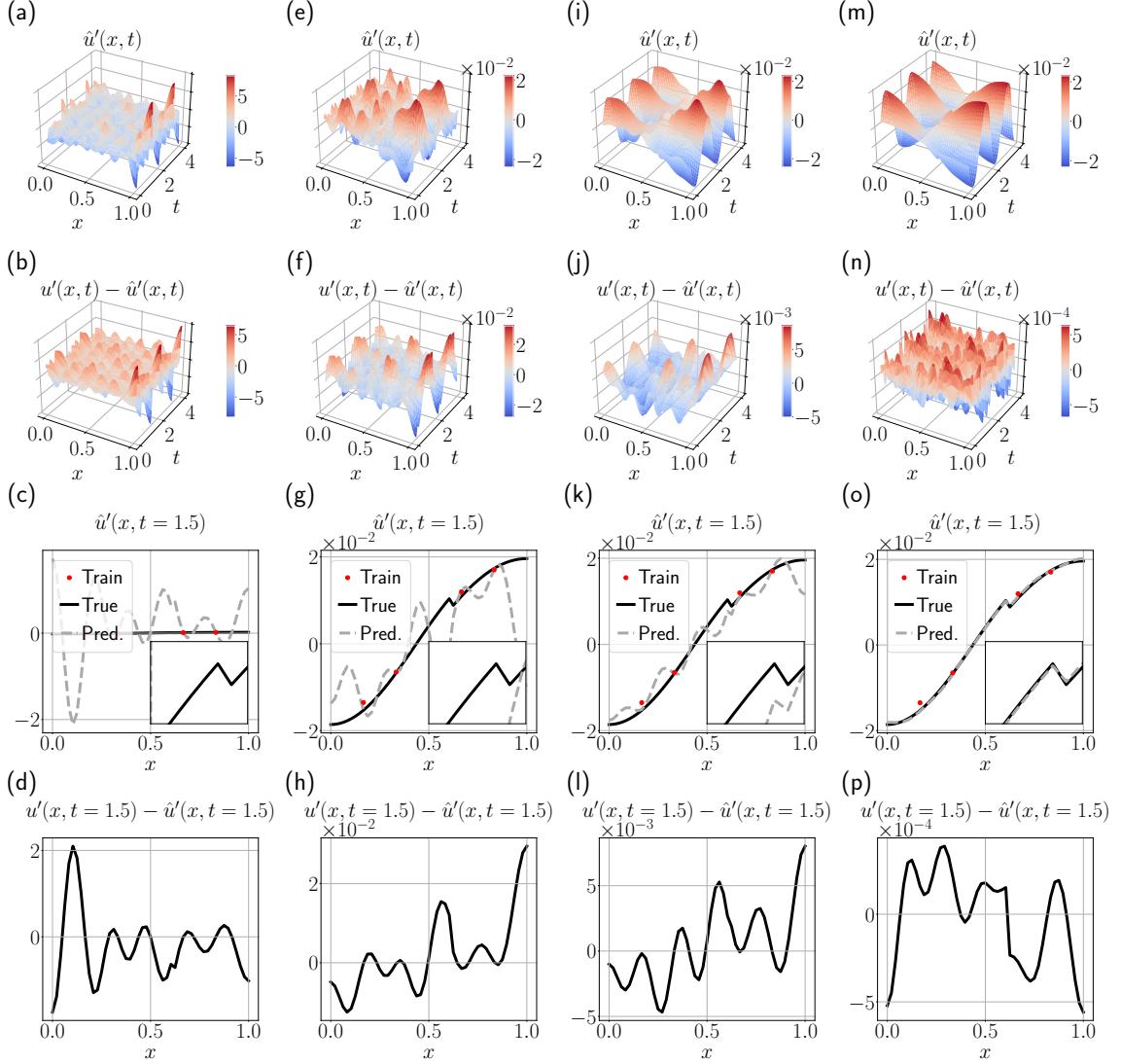


FIG. 14. Effect of regularisation on GalNN on the higher-fidelity data. From left to right, (a-d) no regularisation, (e-h) ℓ_2 -norm, (i-l) ℓ_1 -norm, and (m-p) physics-based regularisations. From top to bottom, velocity prediction in time and space, error between ground truth and prediction in time and space, velocity prediction at a fixed time instance, error at a fixed time instance.

able angular frequency layer.) The physics information significantly eliminates overfitting and outperforms ℓ_1 - and ℓ_2 -norm regularisations. The ℓ_2 -norm regularisation does not work as well because the weights are not selectively regularised, i.e., dominant modes/frequencies will be regularised in the same way as non-dominant ones. The ℓ_1 -norm regularisation performs better since modes corresponding to higher wavenumbers are discarded because of

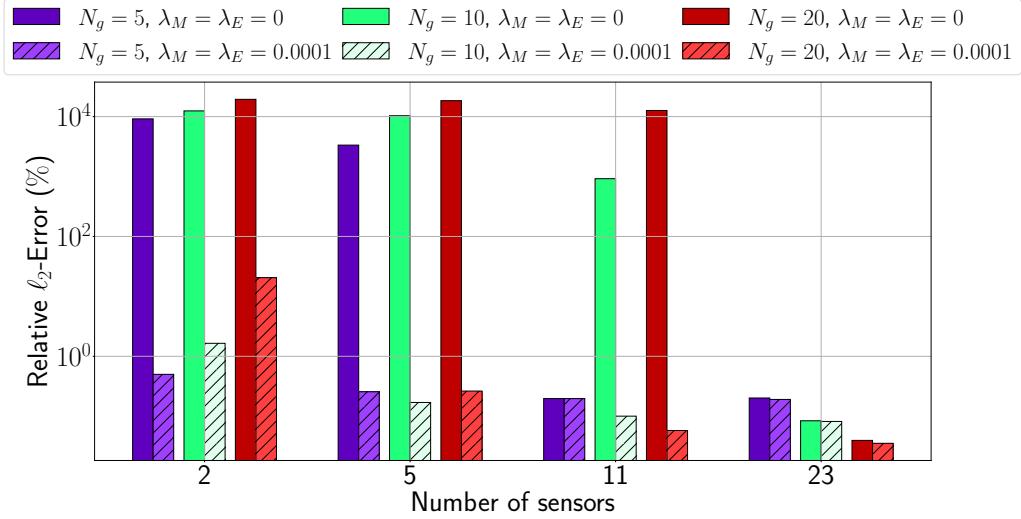


FIG. 15. Prediction performance of GalNNs with different number of modes when trained on data acquired from different number of sensor measurements with and without physics-based loss. (The weighting of the data-driven loss is fixed to $\lambda_D = 1$.)

ℓ_1 -norm promotes sparsity and in the physical basis, the energy is contained in the low frequency modes. We also found that even in the case where the number of Galerkin modes is suitable, ℓ_2 -norm performed worse than physics-based regularisation in terms of handling noisy data.

We conduct a quantitative study to investigate the robustness of the prediction performance to sparsity and noise. First, we vary the number of sensors along the tube. Figure 15 shows the relative ℓ_2 -error achieved by Galerkin networks with different number of modes when trained on noise-free data given on spatial grids of 2, 5, 11, and 23 points (for each of pressure and velocity). (Note that the number of Galerkin modes does not include the additional two jump modes for velocity.) The relative ℓ_2 -error is calculated from the prediction on the fine grid of 49 points over the training range by dividing the ℓ_2 -norm of the difference between ground truth and prediction by the ℓ_2 -norm of the ground truth. Since the method can capture the periodicity well, this error reflects the performance over all time. The networks are trained with and without physics-information in the loss function. We fix the weighting of the data-driven loss to $\lambda_D = 1$, and the physics loss to $\lambda_M = \lambda_E = 0.0001$ for physics-informed and to 0 otherwise. As discussed above, high number of Galerkin modes leads to overfitting for coarse grids, when the Nyquist wavenumber is not enough to resolve

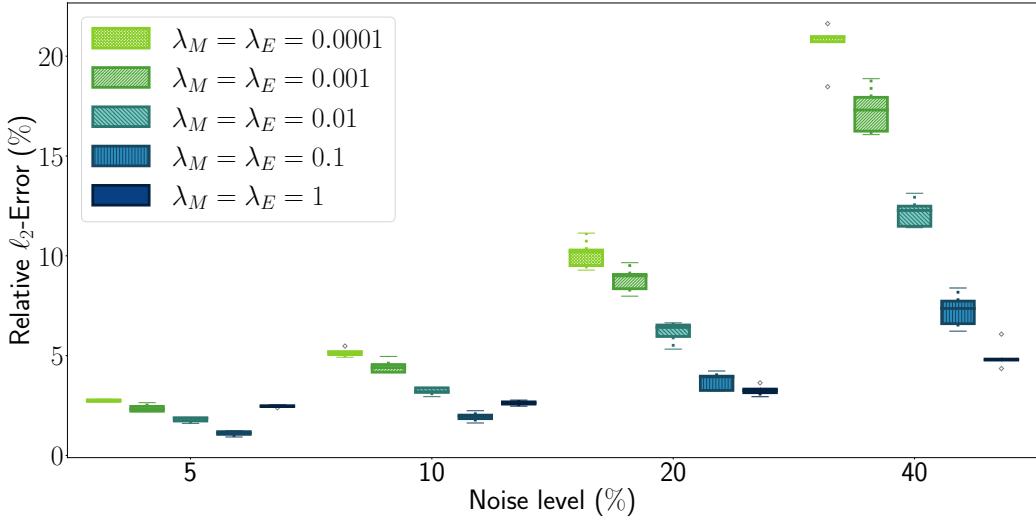


FIG. 16. Prediction performance of the GalNNs with different weightings of physics-based loss when trained on higher fidelity data corrupted with increasing levels of noise. (The weighting of the data-driven loss is fixed to $\lambda_D = 1$.)

the high wavenumbers, e.g., 5 sensors and 10 Galerkin modes. Furthermore, we observe high prediction error even when the Nyquist condition is satisfied, e.g. 5 sensors and 5 Galerkin modes. We found that this type of error is mostly concentrated at the boundaries of the velocity, as we have not provided any boundary data and therefore, the model overfits to the rest of the data, not generalising well over the boundaries. For pressure, this is not an issue, since the boundary conditions are Dirichlet and already encapsulated within the provided Galerkin basis. On the other hand, low number of Galerkin modes may not be enough to approximate the pressure and velocity shapes in fine detail, e.g., 11 sensors and 5 Galerkin modes. The addition of the physics-information shows a marked difference and helps overcoming these limitations.

Next, we choose the network with 10 modes and add increasing levels of noise to the data from 5 sensor measurements. We take 5 different realisations of noise and show the prediction relative ℓ_2 error as a box plot in Figure 16 for noise levels of 5, 10, 20, and 40 %. Noting that even in the noise-free case, the non-physics-informed network performs poorly, we focus on physics-informed training with varying weightings of the physics loss. As the noise level increases, the optimum weighting of the physics-information increases as well, since the data becomes less reliable. We obtain good predictions of pressure and velocity

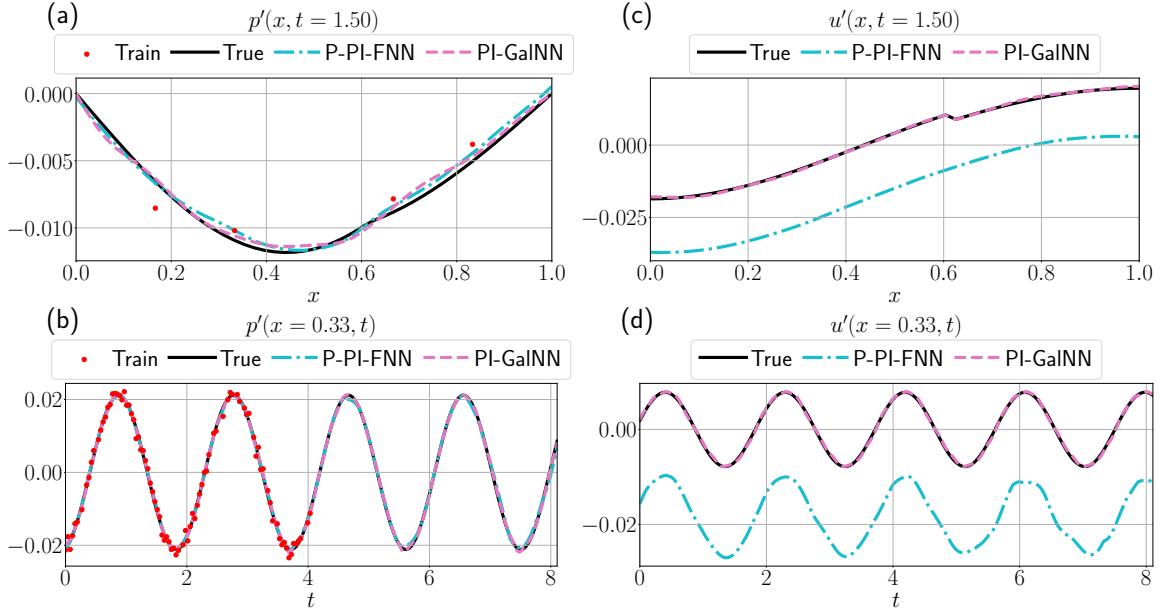


FIG. 17. Predictions of (a-b) acoustic pressure and (c-d) velocity fluctuations of higher-fidelity data from only pressure observations when the training is done with physics-based loss using the P-FNN and GalNN. Top row shows pressure and velocity shapes along the tube at a fixed time instance, bottom row shows pressure and velocity timeseries at a fixed point in the tube.

over the entire domain, even though the data is sparse and noisy. The network is also robust to different realisations of noise.

We conclude by demonstrating how the prediction performs in the absence of velocity data. For this purpose, we examine the case of the physics-informed periodic Galerkin network with 10 Galerkin modes trained on data from 5 sensor measurements and 10 % noise. The prediction results are compared with a two-layer physics-informed sine-ReLU feedforward neural network with 32 neurons in each layer in Figure 17. For the periodic GalNN, we report the following relative ℓ_2 -errors computed from predictions over a very fine spatial grid: 2.63 % for pressure and 2.75 % for velocity in the training range ($t = 0 - 4$ time units), and 2.84 % for pressure and 2.80 % for velocity in the extrapolation range ($t = 4 - 8$ time units). With the physics-informed periodic GalNN, we can reconstruct the velocity as well as pressure, even though we observe some overfitting to the noise. Recall that for this set of data, the physics-based loss is based only on the acoustic equations, hence we must rely on the data to recover the effect of the heat release law. So, increasing

the weighting of the physics-based loss could help with regularising further, however we must also be careful not to underfit the data. Furthermore, we achieved this performance with only two periods of the timeseries. Including more periods along with the already hard-constrained periodicity can also alleviate overfitting to noise. In comparison, for the sine-ReLU feedforward neural network, we report the following relative ℓ_2 -errors: 4.81 % for pressure and 9.77 % for velocity in the training range ($t = 0 - 4$ time units), and 164.22 % for pressure and 160.57 % for velocity in the extrapolation range ($t = 4 - 8$ time units). The FNN fits to the noisy data more compared to the GalNN, and can only recover the velocity with an offset, completely discarding the velocity jump.

D. Inferring heat release

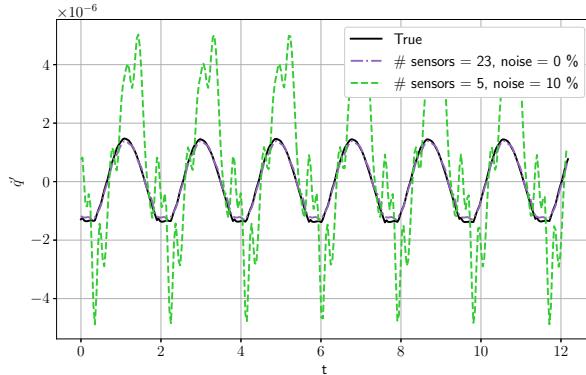


FIG. 18. Prediction of heat release from the trained networks.

Once the model is trained on pressure and velocity data, the heat release rate can be inferred from the energy jump condition (2b) as a consequence via an algebraic calculation. For this purpose, we evaluate the pressure and velocity fluctuations on either side of the flame using the trained model, i.e., $[\hat{p}'_1, \hat{u}'_1] = [\hat{p}'(x_f - h, t), \hat{u}'(x_f - h, t)]$ and $[\hat{p}'_2, \hat{u}'_2] = [\hat{p}'(x_f + h, t), \hat{u}'(x_f + h, t)]$. (The up-stream density fluctuation is determined from the pressure as $\hat{\rho}'_1 = \frac{\hat{p}'_1}{c_1^2}$.) The inferred heat release using the networks from the previous sections is shown in Figure 18. In the noise-free case, in which the data is given on a fine grid, there is a good match between the predicted time series and the time series from the simulation data. In the noisy case, in which the data is contaminated with 10 % noise and given on a

coarse grid of pressure and velocity measurements, the prediction accuracy of the network drops. Note that the order of magnitude of the non-dimensionalised heat release is much smaller than that of velocity or pressure, hence any errors in pressure or velocity prediction will have a larger impact on the heat release.

VI. CONCLUSION

In this work, we developed physics-constrained neural networks to learn nonlinear oscillations that occur in thermoacoustic systems, such as limit cycles, quasiperiodic and chaotic oscillations. Feedforward neural networks are flexible tools for function approximation, however they are also black-box models. These models also have a large hyperparameter space and require hyperparameter tuning, which can be computationally expensive. Physics-informed methods combine first-principles based knowledge with neural networks. Often, the physics information is provided to the network as a soft-constraint. We have demonstrated that by modifying the structure of the neural networks, i.e. hard-constraining the network, one can induce a bias towards more accurate models that can capture the system solutions better. As thermoacoustic systems are dominated by sinusoidal eigenfunctions, we promoted that a physically-motivated choice of sinusoidal activation functions can help tremendously, as they outperform conventional activation functions such as ReLU and tanh. Furthermore, for the limit cycle case, we implemented a layer that contains a trainable angular frequency and its harmonics in order to exactly retain the fundamental angular frequency of the given limit cycle solution. Inspired by Galerkin decomposition, we applied separation of variables on the neural network such that it can be examined in temporal and spatial branches, and forced the network to be thermoacoustic via physical choice of spatial basis. These spatial modes are chosen as the acoustic eigenfunctions of the thermoacoustic system and fulfill the boundary conditions. We showed the effect of the selection of these modes and found that a piece-wise formulation, which takes into account the jump in the mean flow density, works best. Furthermore, we added two additional step-like discontinuous modes to fully capture the jump discontinuity in the velocity at the flame location, which otherwise leads to the Gibbs phenomenon. For the cases of noisy and scarce measurements, we compared a physics-loss based regularisation with the conventional ℓ_1 and ℓ_2 -norm regularisations, and found that the physics-based regularization is superior for our application. The

physics-informed thermoacoustic neural network can reconstruct velocity from only pressure observables, which can be noisy and given on a coarse spatial grid. Finally, we showed that one can also infer the heat release from the trained model using physical jump conditions at the flame location. This work opens up possibilities to learn the nonlinear dynamics of thermoacoustics using physics-aware data-driven methods, as well as to develop more robust and explainable neural networks. Future research directions include testing the algorithm on experimental data.

ACKNOWLEDGMENTS

This research has received financial support from the ERC Starting Grant No. PhyCo 949388.

Appendix A: Calculation of modal damping and heat release terms in the physics-informed loss

1. Modal damping

The modal damping is defined a multiplication in the frequency domain and a convolution in the spatial domain with the pressure. Using an FNN, we predict the pressure at a given time in the spatial domain. In order to compute the contribution of the damping in the physics-informed loss, at each training step, we will transform the pressure predicted in the spatial domain to the frequency domain via Fast Fourier Transformation (FFT), compute the damping in the frequency domain, and then transform the result back to the spatial domain. When sampling from signals, the distinct Fourier frequencies are given as $\omega_m = m \frac{2\pi}{MT_s}$, $m = 0, 1, \dots, M/2$, where M is total number of samples, T_s is the sampling time, and MT_s gives the length of the sampled signal. During simulation, we considered N_g Galerkin modes. So, in our case, we set $M \geq 2N_g$. Notice that the spatial domain is $[0, 1]$, while the wavenumbers of the Galerkin modes have a resolution of π , which is only half-period in this domain. Consequently, the resolution of the Fourier frequencies must be set to $\frac{2\pi}{MT_s} = \pi$ and hence, $MT_s = 2$, which means that the FFT must be taken over a domain of $[0, 2]$, such that the samples are at $x_m = m \frac{2}{M}$, $m = 0, 1, \dots, M - 1$ and we have

one period of discrete samples. One could predict the pressure over the $[0, 2]$ domain, but this could lead to errors as there is no training data in this region. Instead, we first predict over the original $[0, 1]$ domain and then stack this prediction with its symmetric with respect $x = 1$, since we know that pressure is given as a sum of sines, which is an odd function. So, we have

$$\hat{p} = \left\{ \hat{p}(x_m, t_k) \mid x_m = m \frac{2}{M}, m = 0, 1, \dots, M - 1, M \geq 2N_g \right\}, \quad (\text{A1})$$

and its Fourier Transform $\mathcal{F}(\hat{p})$. Now, we take only the part of $\mathcal{F}(\hat{p})$ that corresponds to the positive frequencies and element-wise multiply it with the damping modes ζ ,

$$\xi_j = \mathcal{F}_j^+(\hat{p})\zeta_j, \quad j = 1, 2, \dots, N_g \quad (\text{A2})$$

gives convolution now including the negative frequencies. In the next step, we take the inverse Fourier transform. As a matter of fact, the training is done in batches, so the x locations we are interested in calculating the effect of damping at may not necessarily collide with the spatial grid that we have previously constructed. So, we will do the inverse Fourier transform for these locations in the training batch. Ultimately, the damping term is found as

$$\zeta p(x_k) = \frac{1}{2M} \sum_{j=1}^{N_g} \xi_j e^{ij\pi x_k} + \xi_j^* e^{-ij\pi x_k}, \quad (\text{A3})$$

where ξ^* denotes the complex conjugate. We use the contribution of damping in the residual of the energy equation (6b) when computing the physics-informed loss (12b).

2. Heat release

Although the heat-release term acts as a Dirac delta in the spatial domain, the solver implements Galerkin decomposition, which projects this term onto a truncated set of modes. Plugging back the Galerkin variables (8) in the left-hand side of the energy equation (6b), the remaining heat release term is equal to $\sum_{j=1}^{N_g} 2\dot{q}' \sin(j\pi x_f) \sin(j\pi x)$, which is the effective term in the simulations. As the number of Galerkin modes approaches to infinity, this approaches the Dirac delta. However, from a practical point of view, the number of modes in the simulation is finite, thus we use this expression to find the contribution of heat release in the residual of the energy equation (6b) when computing the physics-informed loss (12b).

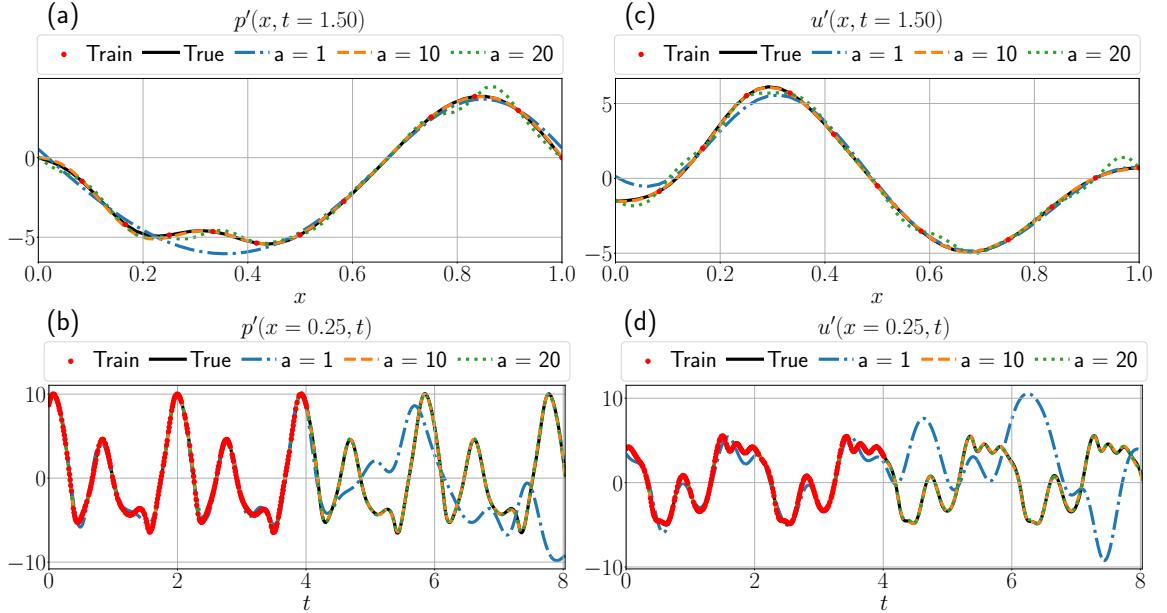


FIG. 19. Effect of hyperparameter a in the sine activation $\frac{1}{a} \sin(az)$

Appendix B: Effect of hyperparameter a in the sine activation

We observe that varying hyperparameter a in the sine activation formulated as $\frac{1}{a} \sin(az)$ affects the frequency of the learned functions. Hence, it is a hyperparameter that requires tuning. Figure 19 illustrates this effect for the Rijke tube data discussed in Section IV B for $a = 1, 10, 20$. Low a , $a = 1$, results in a low frequency model, whereas high, $a = 20$, results in a high frequency model, which can especially be observed in the pressure and velocity shapes in the spatial domain. We found the optimum value for this dataset as $a = 10$. Since the sampling frequency of the training data is high enough, we do not observe the effect of high a in the time domain. [30] reported similar findings in their studies when using the $1 + \sin^2(z)$ activation.

- [1] Rayleigh, The explanation of certain acoustical phenomena 1, *Nature* **18**, 319 (1878).
- [2] T. C. Lieuwen and V. Yang, *Combustion Instabilities In Gas Turbine Engines* (American Institute of Aeronautics and Astronautics, 2006).

- [3] T. Poinsot, Prediction and control of combustion instabilities in real engines, *Proceedings of the Combustion Institute* **36**, 1 (2017).
- [4] F. Culick and P. Kuentzmann, Unsteady motions in combustion chambers for propulsion systems, NATO RTO-AG-AVT-039, AGARDograph (2006).
- [5] L. Kabiraj, R. I. Sujith, and P. Wahi, Bifurcations of self-excited ducted laminar premixed flames, *Journal of Engineering for Gas Turbines and Power* **134**, 10.1115/1.4004402 (2012).
- [6] T. C. Lieuwen, Experimental investigation of limit-cycle oscillations in an unstable gas turbine combustor, *Journal of Propulsion and Power* **18**, 61 (2002).
- [7] K. I. Matveev and F. E. C. Culick, Limit-cycle properties of a rijke tube, *Technical Acoustics* **12**, 1 (2003).
- [8] J. Moeck, M. Bothien, S. Schimek, A. Lacarelle, and C. Paschereit, Subcritical thermoacoustic instabilities in a premixed combustor (American Institute of Aeronautics and Astronautics, 2008).
- [9] N. Noiray, D. Durox, T. Schuller, and S. Candel, A unified framework for nonlinear combustion instability analysis based on the flame describing function, *Journal of Fluid Mechanics* **615**, 139 (2008).
- [10] I. Waugh, S. Illingworth, and M. Juniper, Matrix-free continuation of limit cycles for bifurcation analysis of large thermoacoustic systems, *Journal of Computational Physics* **240**, 225 (2013).
- [11] A. P. Dowling, A kinematic model of a ducted flame, *Journal of Fluid Mechanics* **394**, 51 (1999).
- [12] A. P. Dowling and S. R. Stow, Acoustic analysis of gas turbine combustors, *Journal of Propulsion and Power* **19**, 751 (2003).
- [13] J. Li and A. S. Morgans, Time domain simulations of nonlinear thermoacoustic behaviour in a simple combustor using a wave-based approach, *Journal of Sound and Vibration* **346**, 345 (2015).
- [14] K. Balasubramanian and R. I. Sujith, Thermoacoustic instability in a rijke tube: Non-normality and nonlinearity, *Physics of Fluids* **20**, 044103 (2008).
- [15] M. P. Juniper, Triggering in the horizontal rijke tube: non-normality, transient growth and bypass transition, *Journal of Fluid Mechanics* **667**, 272 (2011).

- [16] T. Sayadi, V. L. Chenadec, P. J. Schmid, F. Richecoeur, and M. Massot, Thermoacoustic instability – a dynamical system and time domain analysis, *Journal of Fluid Mechanics* **753**, 448 (2014).
- [17] A. Orchini and M. P. Juniper, Flame double input describing function analysis, *Combustion and Flame* **171**, 87 (2016).
- [18] M. Haerlinger, M. Merk, and W. Polifke, Inclusion of higher harmonics in the flame describing function for predicting limit cycles of self-excited combustion instabilities, *Proceedings of the Combustion Institute* **37**, 5255 (2019).
- [19] C. Bhattacharya, J. O'Connor, and A. Ray, Data-driven detection and early prediction of thermoacoustic instability in a multi-nozzle combustor, *Combustion Science and Technology* **194**, 1481 (2022).
- [20] F. Selimefendigil and W. Polifke, A nonlinear frequency domain model for limit cycles in thermoacoustic systems with modal coupling, *International Journal of Spray and Combustion Dynamics* **3**, 303 (2011).
- [21] S. Jaensch and W. Polifke, Uncertainty encountered when modelling self-excited thermoacoustic oscillations with artificial neural networks, *International Journal of Spray and Combustion Dynamics* **9**, 367 (2017).
- [22] N. Tathawadekar, N. A. K. Doan, C. F. Silva, and N. Thuerey, Modeling of the nonlinear flame response of a bunsen-type flame via multi-layer perceptron, *Proceedings of the Combustion Institute* **38**, 6261 (2021).
- [23] J. M. Dhadphale, V. R. Unni, A. Saha, and R. I. Sujith, Neural ode to model and prognose thermoacoustic instability, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **32**, 013131 (2022).
- [24] I. E. Lagaris, A. Likas, and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE transactions on neural networks* **9**, 987 (1998).
- [25] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics* **378**, 686 (2019).
- [26] A. Novoa and L. Magri, Real-time thermoacoustic data assimilation, *Journal of Fluid Mechanics* **948**, 10.1017/jfm.2022.653 (2022).

- [27] J. G. Aguilar, *Sensitivity analysis and optimization in low order thermoacoustic models* (2019) doctoral thesis. University of Cambridge.
- [28] L. Magri, Adjoint methods as design tools in thermoacoustics, *Applied Mechanics Reviews* **71**, [10.1115/1.4042821](https://doi.org/10.1115/1.4042821) (2019).
- [29] K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, *Neural Networks* **2**, 359 (1989).
- [30] L. Ziyin, T. Hartwig, and M. Ueda, Neural networks fail to learn periodic functions and how to fix it (Curran Associates, Inc., 2020) pp. 1583–1594.
- [31] G. Raynaud, S. Houde, and F. P. Gosselin, Modalpinn: An extension of physics-informed neural networks with enforced truncated fourier decomposition for periodic flow reconstruction using a limited number of imperfect sensors, *Journal of Computational Physics* **464**, 111271 (2022)
- [32] L. Magri and M. P. Juniper, Global modes, receptivity, and sensitivity analysis of diffusion flames coupled with duct acoustics, *Journal of Fluid Mechanics* **752**, 237 (2014).
- [33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems* (2015), software available from tensorflow.org.
- [34] L. D. Landau and E. M. Lifschitz, *Fluid Mechanics*, 2nd ed. (Pergamon Press, 1987).
- [35] M. A. Heckl, Non-linear acoustic effects in the rijke tube, *Acustica* **72**, 63 (1990).
- [36] F. Huhn and L. Magri, Stability, sensitivity and optimisation of chaotic acoustic oscillations, *Journal of Fluid Mechanics* **882**, A24 (2020).
- [37] L. N. Trefethen, *Spectral Methods in MATLAB* (Society for Industrial and Applied Mathematics, 2000).
- [38] K. He, X. Zhang, S. Ren, and J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification (IEEE, 2015) pp. 1026–1034.
- [39] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks (PMLR, 2010) pp. 249–256.

[40] Code available at <https://github.com/xoeg/Flame-Transfer-Function-Tutorial>.