Solving the population balance equation for non-inertial particles dynamics using probability density function and neural networks: Application to a sooting flame

Andréa Seltz, Pascale Domingo, Luc Vervisch

Solving the population balance equation for non-inertial particles dynamics using probability density function and neural networks: Application to a sooting flame

Andrea Seltz,[1] Pascale Domingo,[1] and Luc Vervisch[1, a)]

CORIA - CNRS, Normandie Université, INSA de Rouen, Technopole du Madrillet, BP 8, 76801 Saint-Etienne-du-Rouvray, France

Numerical modeling of non-inertial particles dynamics is usually addressed by solving a population balance equation (PBE). In addition to space and time, a discretisation is required also in the particle-size space, covering a large range of variation controlled by strongly nonlinear phenomena. A novel approach is presented in which a hybrid stochastic/fixed-sectional method solving the PBE is used to train a combination of an artificial neural network (ANN) with a convolutional neural network (CNN) and recurrent long short-term memory artificial neural layers (LSTM). The hybrid stochastic/fixed-sectional method decomposes the problem into the total number density and the probability density function (PDF) of sizes, allowing for an accurate treatment of surface growth/loss. After solving for the transport of species and temperature, the input of the ANN is composed of the thermochemical parameters controling the particle physics and of the increment in time. The input of the CNN is the shape of the particle size distribution (PSD) discretised in sections of size. From these inputs, in a flow simulation the ANN-CNN returns the PSD shape for the subsequent time step or a source term for the Eulerian transport of the particle size density. The method is evaluated in a canonical laminar premixed sooting flame of the literature and for a given level of accuracy (i.e., a given discretisation of the size space), a significant computing cost reduction is achieved (6 times faster compared to a sectional method with 10 sections and 30 times faster for 100 sections).

[a)]Electronic mail: vervisch@coria.fr

I. INTRODUCTION

Laminar or turbulent flows carrying non-inertial particles are at the core of many chemical and mechanical engineering processes. On the numerical modeling side, the solving of a population balance equation (PBE) for simulating the time evolution of the particle size distribution (PSD) is usually the starting point to tackle particle dynamics.[1–4]

Three major difficulties arise when dealing with population balance equations. The first one lies in the physical modeling of nucleation, surface growth/loss, agglomeration/coagulation and sometimes breakage of the particles. Depending on the problem addressed, these phenomena can involve quite complex thermophysical properties of the solid and liquid or/and gaseous phases at play.

The second difficulty results from the large range of sizes covered by the particles, making the discretisation in size space uneasy to predict both the distribution of size and its moments accurately.

The third difficulty comes from the numerical solving of the particle size distribution itself. The physics behind the terms of the PSD equation can be strongly non-linear, with a PSD time evolution prone to spurious oscillations and divergence. For instance, the growth/loss of particle surface takes the form of a non-linear convective effect of the PSD in size space (i.e., surface growth/loss transport the PSD towards larger/smaller sizes without modification of its shape). Thereby, because of growth/loss, it may be challenging to secure stability of the solution keeping a minimum level of spurious diffusion of the PSD in size space.[5] Similarly, properly predicting the moments of the particle size distribution in an agglomeration/coagulation process may require a specific numerical treatment.[6]

A large variety of methods for simulating the time evolution of PSD have been discussed in the literature and applied with success, including advanced numerical strategies proposed recently to address the issues mentioned above.[7–13] However for many of these approaches, securing accuracy in the PSD shape prediction goes with a large computing effort, jeopardising the systematic application of the most precise methods to three-dimensional and unsteady simulations for virtual prototyping of real systems. Advanced methods for solving the PSD have been applied to canonical problems featuring low or moderate Reynolds numbers studied with direct numerical simulation (DNS),[14–16] while simplified approaches are usually developed to deal with complex three-dimensional flows, specifically in the context of particulate emissions from burners and com-

bustion systems.[17–19]

To progress in this direction, this paper reports an attempt in which the effort made in grid size resolution and precision of PSD solving, serves to train a set of neural networks for solving the population balance equation. Neural networks are subsequently coupled with the flow solution, thus allowing for a significant reduction of the computing cost, still preserving the accuracy provided by the numerical method generating the training database.

The PBE takes the form of a balance equation for $n(v^*;\underline{x},t)$, the number of particles per unit of flow volume and per unit size (or particles number density), where the characteristic particle size is denoted v^*. The generic form of functionals used in flow physics was first studied by Hopf[20,21] in the 50's. Then in the 70's, O'Brien and his team popularised the concept of probability density functions (PDF) for turbulent scalar mixing studies.[22–24] Recently, Bouaniche et al.[5] have discussed how solving the PBE can benefit from the knowledge on PDF as discussed by O'Brien, when he pioneered theoretical and computational approaches in line with PDF for scalars (temperature, mass fractions, etc.). Indeed, $n(v^*;\underline{x},t)$ may be decomposed into $N_T(\underline{x},t)$, the total number of particles per unit of volume and $\overline{P}(v^*;\underline{x},t)$, the PDF of particle size

$$n(v^*;\underline{x},t) = N_T(\underline{x},t)\overline{P}(v^*;\underline{x},t)\,. \tag{1}$$

Then, instead of solving for $n(v^*;\underline{x},t)$, it is preferred to solve for both $N_T(\underline{x},t)$ and $\overline{P}(v^*;\underline{x},t)$, to ease the treatment of the non-linear surface growth/loss term with a Monte Carlo solution for the PDF.

The training of the neural networks is performed along these lines with the recently proposed hybrid stochastic-sectional approach.[5] This hybrid technique combines Monte Carlo and fixed-sectional methods, to minimise the discretisation errors when solving the surface growth/loss term of the population balance equation, and to compute agglomeration using a sectional formalism. It relies on a fixed number of stochastic particles and sections, with a numerical algorithm minimising errors, even for a moderate number of stochastic particles. This method was previously validated against canonical PSD evolutions for which analytical solution exist.[5] It was also applied to a sooting flame, in order to further analyse the relation between the mobility diameter, measured in the experiments, and the equivalent sphere diameter, introduced in the modeling. The influence of the fractal particle shape on the simulated particle size distribution was also explored.[25]

The introduction of artificial neural networks in reactive flow simulation is a rapidly growing field, to reduce both computing cost and memory requirements. The optimisation of networks

has received attention in chemistry tabulation[26–32] and also to develop data driven modeling[33–40] or again to analyse experimental measurements[34,41] and to propose digital-twins for process control.[42] Works have been specifically devoted to the application of neural networks to study and to model the dynamics of particles in flows and their size distributions. In this context, the improvement of physical models driving the PBE through ANN have been discussed,[43] control particle size strategies using ANN have been implemented,[44] particle size distributions have been approximated with ANN,[45] measurements of particle size distributions have been coupled with ANN,[46] memory effects were introduced in networks with feedback connections to progress in Gaussian moment methods[47] and dynamic modeling of component formation was coupled with a special class of ANN (regularisation networks), in order to predict the solid-liquid equilibrium in complex multi-phase flows.[48]

Here, a combination of artificial neural networks (ANN) and convolutional neural networks (CNN)[49–51] is discussed. These ANN and CNN are both augmented with one additional recurrent long short-term memory artificial neural layer (LSTM).[52–54] The balance equations for thermochemistry are solved in a regular manner. Then on one hand, the ANN part deals with the thermochemical parameters of the PSD evolution (chemical species mass fractions, temperature, etc.), i.e., the inputs of the physical models driving particle nucleation, surface growth/loss and agglomeration/coagulation. On the other hand, the shape of the PSD is analysed by the CNN, which is specialised for image segmentations tasks, thus well adapted to predict a distribution. To better capture the path followed by the solution of the non-linear problem, the LSTM allows for storing information from previous time-steps in order to facilitate the optimization of the ANN-CNN. Once ANN and CNN concatenated, the vector of thermochemical parameters, the particle size distribution and the increment in time δt, serve as input to the neural networks, which returns the PSD at time t+δt, thus saving a large amount of the computing time usually devoted to solving for the homogeneous carrier-phase part of the population balance equation (i.e., nucleation, surface growth/loss and agglomeration/coagulation of the particles). In practice, the number density of particles is discretised over a set of sections, each section is transported by the turbulent flow solving for convection in the eulerian context, while the neural networks solve for the rest of the particle physics.

In the present work, a canonical laminar one-dimensional rich sooting premixed ethylene flame is considered for training the neural networks to predict carbon particulate emissions. Both a priori and a posteriori tests are reported. Well-established physical models are applied, the objective

being to demonstrate the potential of the approach under a given physical/chemical modeling framework.

The population balance equation formalism is recalled in the next section along with the main lines of the hybrid stochastic fixed-sectional method for solving the PBE, before summarising the various physical models introduced in this work for sooting flames. In the subsequent section, the machine learning procedure and its training over the laminar premixed sooting flame is discussed. Finally, the results of a posteriori tests are presented.

II. POPULATION BALANCE EQUATION MODELING

Let us denote $n(v;\underline{x},t)$ the number of particles of characteristic size v (in terms of volume or mass, v is a continuous independent variable), per unit of flow volume and per unit of characteristic size, of an aerosol submitted to simultaneous nucleation, surface variation and agglomeration. $n(v;\underline{x},t)$ follows a population balance equation, an integro-partial-differential equation of the hyperbolic type:[4]

$$\frac{\partial n(v;\underline{x},t)}{\partial t} + \underline{u}\cdot\nabla n(v;\underline{x},t) = -\frac{\partial}{\partial v}\left[G(v;\underline{x},t)n(v;\underline{x},t)\right] + \dot{A}(v;\underline{x},t)\,, \tag{2}$$

with $\underline{u}$ the velocity vector. $G(v;\underline{x},t) > 0$ is the surface growth rate or $G(v;\underline{x},t) < 0$ the surface loss rate. This growth/loss rate may depend on both v, the particle size, and $(\underline{x},t)$, the position and time, because the surface kinetics is usually controlled by the local gaseous/liquid environment. The RHS contains a source term $\dot{A}(v;\underline{x},t)$, which is composed of $\dot{h}(v) > 0$ (respectively $\dot{h}(v) < 0$) the nucleation (respectively disappearance) rate and an integral which accounts for agglomeration following the continuous counterpart of the so-called Smoluchowski equation, with $\beta(v,v^+)$ the collision kernel for two particles of volume v and v^+, hence

$$\begin{aligned}\dot{A}(v;\underline{x},t) = \dot{h}(v;\underline{x},t) &+ \frac{1}{2}\int_0^v \beta(v-v^+,v^+)n(v-v^+;\underline{x},t)n(v^+;\underline{x},t)dv^+ \\ &- n(v;\underline{x},t)\int_0^\infty \beta(v,v^+)n(v^+;\underline{x},t)dv^+\,.\end{aligned} \tag{3}$$

From Eq. (2), the total number of particles per unit of volume

$$N_T(\underline{x},t) = \int_0^\infty n(v;\underline{x},t)dv\,, \tag{4}$$

evolves according to

$$\frac{\partial N_T(\underline{x},t)}{\partial t} + \underline{u}\cdot\nabla N_T(\underline{x},t) = \int_0^\infty \dot{A}(v;\underline{x},t)dv\,. \tag{5}$$

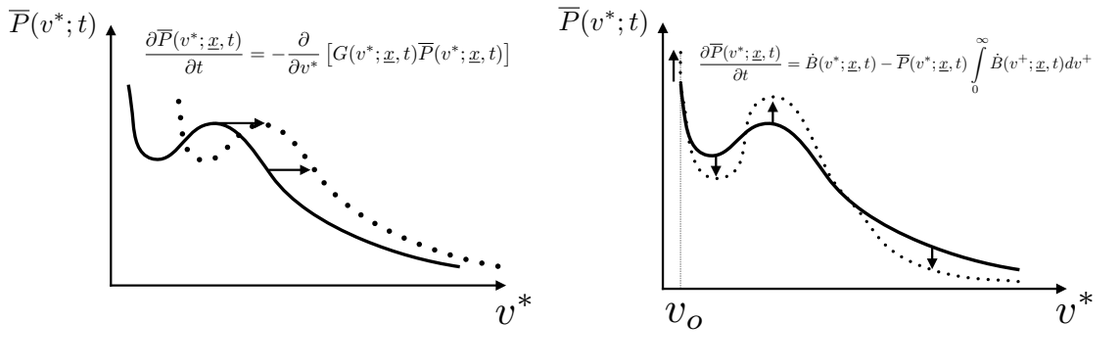
A. Hybrid stochastic/sectional PBE solving with PDF

From the relation $n(v^*;\underline{x},t) = N_T(\underline{x},t)\overline{P}(v^*;\underline{x},t)$, between the particle size density, the total particle density and $\overline{P}(v^*;\underline{x},t)$ the PDF of size, using Eqs. (2) and (5), the balance equation for $\overline{P}(v^*;\underline{x},t)$ is derived, where v^* denotes the sample space (or the stochastic variable) associated to the physical particle size v,

$$\frac{\partial \overline{P}(v^*;\underline{x},t)}{\partial t} + \underline{u}(\underline{x},t)\cdot\nabla\overline{P}(v^*;\underline{x},t) = \overbrace{-\frac{\partial}{\partial v^*}\left[G(v^*;\underline{x},t)\overline{P}(v^*;\underline{x},t)\right]}^{(i)} + \underbrace{\dot{B}(v^*;\underline{x},t) - \overline{P}(v^*;\underline{x},t)\int_0^\infty \dot{B}(v^+;\underline{x},t)dv^+}_{(ii)}\,. \tag{6}$$

The term (i) is the convection of the PDF in size space according to surface growth or loss (Fig. 1(a) illustrates the PDF evolution with $G > 0$). The term (ii) is a source/sink whose amplitude is driven by nucleation and agglomeration of the particles according to the rate given by $\dot{B}(v^*;\underline{x},t) = \dot{A}(v^*;\underline{x},t)/N_T(\underline{x},t)$ (Fig. 1(b)). This second term is decomposed in two parts: a local decrease or increase of the probability according to the sign of $\dot{B}(v^*;\underline{x},t)$ and a redistribution, proportional to the probability of finding size v^*, of the sum of $\dot{B}(v^*;\underline{x},t)$ over all sizes (last term in Eq. (6)), so that this RHS vanishes when integrated over all particle sizes, thereby preserving the normalisation of the PDF.

In the PDF studies devoted to scalar transport and micro-mixing by O'Brien[22] and others,[55–57] the statistical behavior of conserved quantities was examined, thus in cases where $\dot{B}(v^*;\underline{x},t) = 0$ in Eq. (6). Considering physical particles, which can be created by nucleation or can disappear by agglomeration, breakage or oxidation, this source becomes non-zero and it requires a specific treatment when the PDF is decomposed into a set of stochastic particles for simulating its time evolution. Here, when training the neural networks, the agglomeration source terms in $\dot{B}(v^*;\underline{x},t)$ are computed following the approach by Kumar and D. Ramkrishna,[58] combined with a Monte Carlo procedure to properly reassign the stochastic particles sizes following the physical models used for agglomeration/coagulation.[5] The roundoff error between the exact continuous PSD and



(a) Surface growth (b) Nucleation and agglomeration

FIG. 1. Sketch of the PDF evolution according to Eq. (6). (a): Surface growth. (b): Nucleation and agglomeration.

the distribution reconstructed from the PDF discretised over the stochastic particles is corrected by transporting residuals with a fixed-sectional 2-point method.[59] All the details concerning this numerical method for solving the PBE using the PDF concept and its careful validation against analytical solutions may be found in Bouaniche et al.[5] and are not repeated here for sake of brevity.

During the PDF solving, the number of particles per unit of flow volume within a section of size may be written

$$N_i(\underline{x},t) = \int_{I_{v_i}} n(v^*;\underline{x},t)dv^* = N_T(\underline{x},t)\int_{I_{v_i}} \overline{P}(v^*;\underline{x},t)dv^*\,, \tag{7}$$

where the interval $I_{v_i} \equiv [v_i^{inf}, v_i^{sup}]$ defines the i-th fixed-section of size. M sections are considered for training the CNN. The training is performed from a database of correlated $N_i(t)$ and $N_i(t+\delta t)$ (for $i = 1,\cdots,M$) applying surface growth/loss, nucleation and agglomeration of the particles, i.e., only terms (i) and (ii) of Eq. (6). Transport in physical space is not included in the neural networks as it will be handled separately by the flow solver. From the relations (2) and (7), the M scalar balance equations for $N_i(\underline{x},t)$ that need to be considered together with other usual aerotheremochemistry equations may be written

$$\frac{\partial N_i(\underline{x},t)}{\partial t} + \underline{u}\cdot\nabla N_i(\underline{x},t) = \int_{I_{v_i}} \left(-\frac{\partial}{\partial v^*}\left[G(v^*;\underline{x},t)n(v^*;x,t)\right] + \dot{A}(v^*;x)\right) dv^* = \dot{\Omega}_i(\underline{x},t)\,. \tag{8}$$

It is therefore $\dot{\Omega}_i(\underline{x},t)$ that will be computed from the ANN-CNN as discussed thereafter.

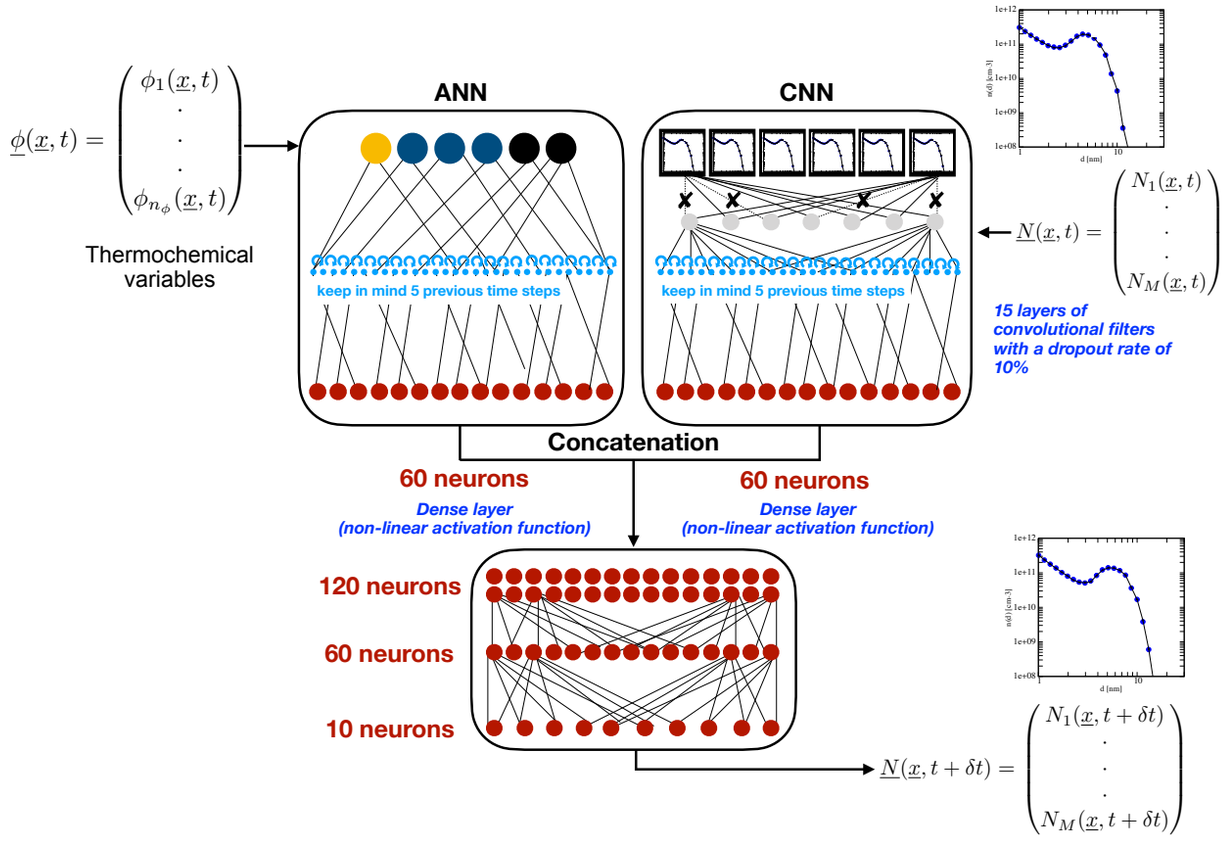


FIG. 2. Neural networks architecture. ANN inputs: Thermochemical variables and δt, time step. CNN input: Particle size distribution at time t. Concatenated networks output: Particle size distribution at time $t+\delta t$. Information of the 5 previous time-steps are stored in the LSTM.

B. Neural networks for PBE solving

The problem is decomposed into the thermochemical quantities (species concentrations, temperature, etc.), collected in a vector $\underline{\phi}(\underline{x},t)$ of dimension n_ϕ controlling $G(v^*;\underline{x},t)$, $\dot{A}(v^*;\underline{x},t)$ and $\dot{B}(v^*;\underline{x},t)$ in Eqs. (5) and (6), and the particle size distribution, expressed in terms of $N_i(\underline{x},t)$ for the i-th section of size, organised into a vector $\underline{N}(\underline{x},t)$ of dimension M, the number of sections of size considered. The neural networks are trained to solve for nucleation/breakage, surface growth/loss and agglomeration/coagulation so that for an input $(\underline{\phi}(\underline{x},t),\underline{N}(\underline{x},t),\delta t)$, the output is $\underline{N}(\underline{x},t+\delta t)$.

The architecture of the numerical model, summarised in Fig. 2, is a multi-headed neural network which takes the form of a directed graph of layers. It is a combination of an artificial neural network tracking the evolution of the thermochemical vector $\underline{\phi}(\underline{x},t)$, while a convolutional neural network is used for the particle number density $\underline{N}(\underline{x},t)$. Both of these neural networks include

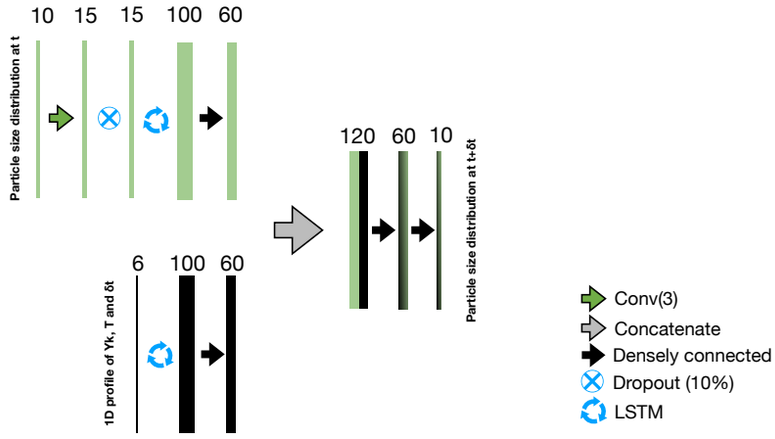


FIG. 3. Structure of the ANN-CNN with LSTM for PBE solving.

feedback connections,[53,54] to account for the time history of the signals.

Neural networks allow for modeling strongly non-linear phenomena by organising in layers a large number of linear functions of the form,

$$O_i^k = f\left[w_o^k + \sum_{j=1}^{n} w_{ij}^k x_j^\ell\right], \tag{9}$$

where O_i^k is the output of the i-th neuron of the k-th layer, x_j^ℓ is the input from the j-th neuron of the ℓ-th layer, w_{ij}^k is the weight attributed to that input, w_o^k is the biais and f is the activation function. The w_{ij}^k are optimised to match a given set of outputs for a given set of inputs. In the CNN, additional layers are exploring the input data (mostly filtering and max pooling) to extract features associated to the control parameters. Then, the neural networks are trained to recognise these features, which are associated to output values.[49,50,60]

The evolutions of the thermochemical parameters $\underline{\phi}(\underline{x},t)$ (solved separately from neural networks) and of δt, which may vary from iteration to iteration, both enter an ANN featuring a recurrent long short-term memory neural layer (LSTM)[52] composed of 100 neurons, to which a 60-neuron layer is connected. The feedback connections of this neural network are calibrated over a duration characterised by 5 iterations in time. During training, each LSTM cell (100 neurons) uses the solution at 't' as a training label (or target value), whereas the solutions at the 5 previous iterations serve as training data. Increasing this number of stored iterations was found not to improve accuracy significantly, while below 4 iterations stored, accuracy cannot be secured. As usually done in machine learning based approaches, every component of $\underline{\phi}(\underline{x},t)$ is normalised by

its maximum level seen in the training database, these maximum must be stored to subsequently use the networks.

Because of the large range covered by the particle sizes during their evolution, before entering the CNN devoted to the PSD analysis, its signal is normalised by $\langle \underline{N} \rangle$, the mean computed over all sections for all positions and times of the training database,

$$N_i^\star(\underline{x},t) = \frac{N_i(\underline{x},t) - \langle \underline{N} \rangle}{\sigma_N} \,. \tag{10}$$

$\langle \underline{N} \rangle$ and $\sigma_N = \left\langle [N_i(\underline{x},t) - \langle \underline{N} \rangle]^2 \right\rangle$ computed from the training database, are stored to reconstruct $N_i(\underline{x},t) = \sigma_N N_i^\star(\underline{x},t) + \langle \underline{N} \rangle$ during the simulation from the dimensionless CNN output.

The normalised input $N_i^\star(\underline{x},t)$ goes through 15 layers of convolutional filters with a dropout rate of 10% (i.e., 10% of the data do not proceed to the next step). To account for the time history of the PSD, the 100 neurons LSTM layer used in the CNN is connected to a 60-neuron dense layer (a dense layer is driven by the same formulas as the linear layers, but the end result is passed through a non-linear activation function). As mentioned above, the time history is collected over 5 iterations.

The 60 neurons of each branch (thermochemistry and particle sizes) are then concatenated (Fig. 2), leading to a 120-neuron layer, in which the concatenation operation is channel-wise. Finally, two dense connected layers return the prediction $\underline{N}^\star(\underline{x},t+\delta t)$, which is then transformed back into $\underline{N}(\underline{x},t+\delta t)$ applying Eq. (10). Hence, the ANN-CNN is cast so that

$$N_i(\underline{x},t+\delta t) = \mathscr{F}_i \left[\underline{\phi}(\underline{x},t), \underline{N}(\underline{x},t), \delta t \right] \,. \tag{11}$$

Two options exist to implement such ANN-CNN in a flow solver. If a fractional step method is used with splitting[61] of the fluid mechanics operators contributing to the evolution of the $N_i(\underline{x},t)$, flow transport is first applied leading to the intermediate values $N_i(\underline{x},t^\star)$, which enter the ANN-CNN to return $N_i(\underline{x},t+\delta t)$. In the second option of a fully explicit and direct integration of the RHS of the $N_i(\underline{x},t)$ balance equations, the source $\dot{\Omega}_i(\underline{x},t)$ in Eq. (8) may be expressed as

$$\dot{\Omega}_i(\underline{x},t) = \frac{\mathscr{F}_i[\underline{\phi}(\underline{x},t), \underline{N}(\underline{x},t), \delta t] - N_i(\underline{x},t)}{\delta t} \,. \tag{12}$$

With both options, the time-step values seen in the flow solver must be contained within the range of time steps considered during the training of the neural networks. (Indeed, the highly non-linear interpolations by neural networks are prone to divergence when inputs values are not strictly within their training bounds.)

The ANN-CNN is built using the Keras and TensorFlow Python library with GPU support (`www.tensorflow.org`).

C. Soot physical modeling

The objective of this work is to isolate and study the potential of data driven approaches for solving the PBE, well established hypothesis and physical modeling are therefore introduced to generate the training database, so that the results obtained on a canonical problem will be similar to those previously reported in the literature. The novelty is the drastic reduction in computing time provided by the method, without loss in accuracy.

The soot particles are modeled as spherical and in the hybrid stochastic/fixed-sectional method, each stochastic particle discretising $\overline{P}(v^*;\underline{x},t)$, the PDF of sizes, carries information on a characteristic volume v^k, with $k = 1,\cdots,N_P$, where N_P is the total number of stochastic particles considered.

Nucleation is assumed to be driven by the collision of two pyrenes ($C_{16}H_{10}$) molecules,[62]

$$\dot{H}(\underline{x},t) = 0.5\beta_{\text{py}}N_{\text{py}}^2\,, \tag{13}$$

with

$$N_{\text{py}} = [C_{16}H_{10}]\mathscr{N}_A\,, \tag{14}$$

the volume number of pyrene molecules and where $\mathscr{N}_A$ is the Avogadro constant.

The surface rate of change G is simplified in condensation of pyrene molecules on soot particles (G_{Cond}), C_2H_2 (acetylene) addition by the so-called HACA mechanism and surface oxidation by O_2 and OH ($G_{\text{HACA,Oxi}}$),[62] i.e.,

$$G(v^k,t) = G_{\text{Cond}}(v^k,t) + G_{\text{HACA,Oxi}}(v^k,t) \tag{15}$$

is applied to control the surface growth/loss of each stochastic particle v^k ($dv^k(t)/dt = G(v^k,t)$). The condensation source term reads:

$$G_{\text{Cond}}(v^k,t) = m_{\text{py}}\dot{H}_{\text{Cond}}(v^k,t)/N_i(t)\,, \tag{16}$$

(for $v^k \in I_{v_i}$), with m_{py} the mass of one pyrene molecule and

$$\dot{H}_{\text{Cond}}(v^k,t) = \beta_{v^k,\text{py}}N_i(t)N_{\text{py}}\,. \tag{17}$$

The surface reaction source term is cast in:

$$G_{\mathrm{HACA,Oxi}}(v^k;t) = (\dot{\omega}_{\mathrm{C_2H_2}} + \dot{\omega}_{\mathrm{O_2}} + \dot{\omega}_{\mathrm{OH}})/N_i(t)\,, \tag{18}$$

with the chemical sources,

$$\dot{\omega}_{\mathrm{C_2H_2}} = 2W_{\mathrm{C}} k_{\mathrm{C_2H_2}} [R_{v^k}][\mathrm{C_2H_2}]\,, \tag{19}$$

$$\dot{\omega}_{\mathrm{O_2}} = -2W_{\mathrm{C}} k_{\mathrm{O_2}} [R_{v^k}][\mathrm{O_2}]\,, \tag{20}$$

$$\dot{\omega}_{\mathrm{OH}} = -W_{\mathrm{C}} k_{\mathrm{OH}} [S_{v^k}][\mathrm{OH}]\,, \tag{21}$$

where W_{C} is the molar mass of Carbon, and $k_{\mathrm{C_2H_2}}$, $k_{\mathrm{O_2}}$, k_{OH} are calculated as in previous works.[62] $[S_{v^k}]$ and $[R_{v^k}]$ read[62,63]

$$[S_{v^k}] + [R_{v^k}] = s^k \chi\, \alpha_{\mathrm{HACA}}(v^k) N_i(t)/\mathscr{N}_A\,, \tag{22}$$

s^k is the particle surface and χ is the number of sites per unit surface of soot. Mature particles are expected to provide a lower proportion of active sites per unit of surface and this is accounted for using[25,62]

$$\alpha_{\mathrm{HACA}}(v^k) = \tanh\left(\frac{a}{\log(\rho_{\mathrm{soot}} \times v^k/(W_{\mathrm{C}}/\mathscr{N}_A)} + b\right)\,, \tag{23}$$

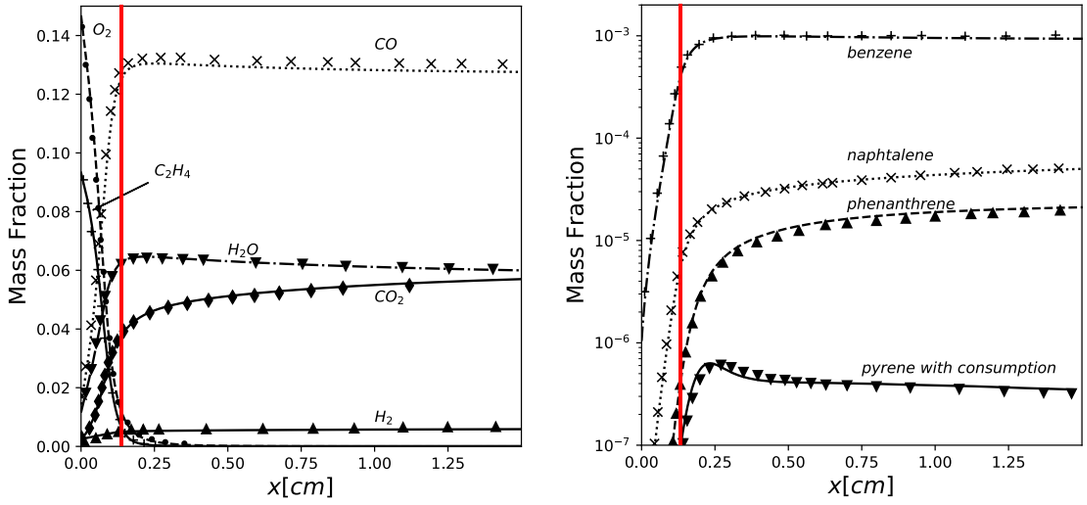
with $a = 12.65 - 0.00563T$ and $b = -1.38 + 0.00068T$.

The collision rates entering the Smoluchowski equation expressing the agglomeration, are calculated depending on the Knudsen number.[64,65] Collision frequencies are calculated in the same manner for collisions between pyrene molecules (nucleation) or between pyrene molecules and soot particles (condensation). The numerical implementation of the discretisation of the Smoluchowski equation[58] follows the now standard method by Eberle et al.[66] to account for the source term $\dot{B}_i$ (Eq. (6)) in each I_{v_i} section considered.

III. ANN-CNN TRAINING AND VERIFICATION

A. One-dimensional sooting flame

A one-dimensional fuel rich (equivalence ratio of 2.07) laminar premixed ethylene-argon-oxygen flame studied both experimentally and numerically in the literature[67], is first considered to evaluate the proposed approach. The fresh gases flowing at a temperature of 300 K stabilise a flame propagating at the velocity of 8.26 cm/s. In mole fractions, the composition of the injected mixture is $X_{\mathrm{C_2H_4}} = 0.133$, $X_{\mathrm{O_2}} = 0.193$ and $X_{\mathrm{Ar}} = 0.674$.



(a) Major species (b) PAH

FIG. 4. (a) Major species mass fractions and (b) PAH across the sooting one-dimensional flame. Lines: present simulation. Symbols: reference simulation.[67] Vertical red line: position of the maximum heat release rate.

The complex molecular transport and the detailed chemical scheme by Appel et al.[62] (101 species and 544 elementary reactions) is used to simulate the gaseous part of the problem (thermochemical parameters that will enter the ANN) with the specialised CANTERA reacting flow solver.[68] The length of the one-dimensional computational domain is 0.02 m and the mesh is composed of 2906 points, it is refined at the reaction zone down to a mesh cell of 0.607μm. To minimise error compensation between chemistry and heat transfer, as usually done in the numerical study of one-dimensional sooting flame, the measured temperature profile is imposed.[67] This reference one-dimensional flame of the literature was already simulated using the hybrid stochastic/fixed-sectional approach solving for the PDF of size, the comparisons of the results against measurements and previous simulations may be found in Bouaniche et al.[25]

The PBE one-dimensional simulations are performed in a moving reference frame, so that the derivative versus the Lagrangian residence time $\tau(x)$ relates to the position through the flame

$$\tau(x) = \int_0^x \frac{1}{u(x^+)} dx^+ , \tag{24}$$

where $u(x)$ results from the flame simulation in physical space.

During its training, the artificial neural network tracks the evolution of T, the temperature, and of the mass fractions of C_2H_2, the fuel, of O_2, the oxidizer, of OH, an intermediate radical, and of

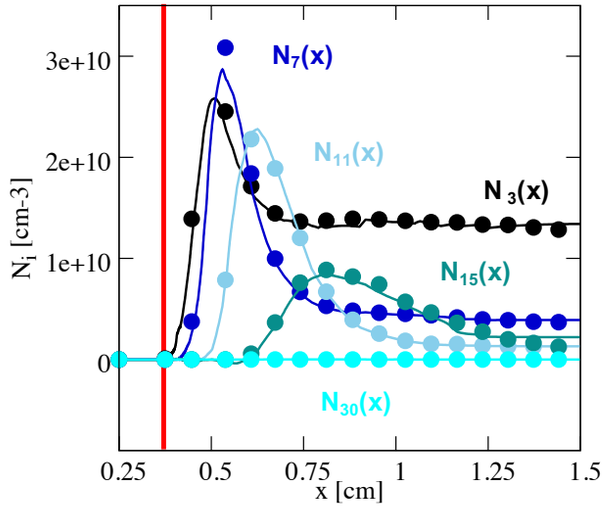


FIG. 5. Distribution across the one-dimensional premixed flame of particle number density for representative sections of the training database with $M = 30$. Red line: position of the maximum heat release rate. Section mid-diameter $0.5(v_i^{inf} + v_i^{sup})$, N_3: 1.15 nm. N_7: 1.97 nm. N_{11}: 3.3 nm. N_{15}: 5.82 nm. N_{30}: 44.22 nm.

C_6H_6, the pyrene, therefore $\underline{\phi} = (T, Y_{C_2H_2}, Y_{O_2}, Y_{OH}, Y_{C_6H_6})$. These species were selected because of their dual impact on both flame and soot chemistry. C_6H_6 controls the modeling the particle surface growth due to condensation. The temperature, the mass fractions of C_2H_2, O_2 and OH allow for quantifying the overall progress of combustion, to characterise the gaseous environment in which the soot particles are traveling. Adding more chemical species in the input vector of the ANN was found not to significantly improve the predictions by the ANN-CNN.

The consumption of pyrene by nucleation is accounted for by coupling the solving of this species with the PBE simulation (i.e., a sink term is added to the balance equation for the pyrene mass fraction). Figure 4 shows the comparison against major and PAH (polycyclic aromatic hydrocarbon) species obtained in this work and previous results reported in the literature.[67] The chemical one-dimensional flame structure obtained follows expected results and can be used for training the neural networks.

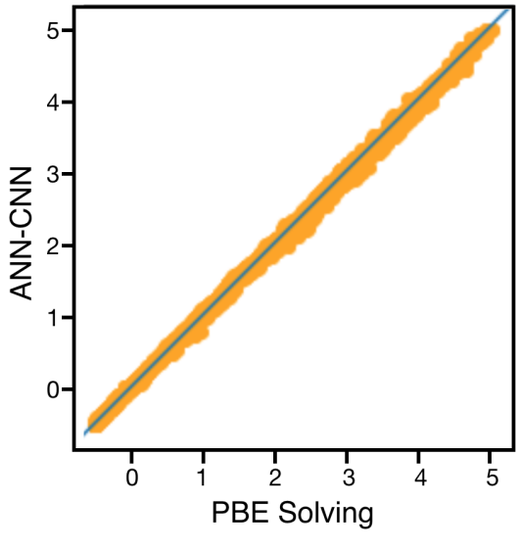


FIG. 6. Scatter plot (orange) of normalised particle number densities as predicted by the ANN-CNN versus the values from the PBE solution with the hybrid stochastic-fixed sectional method used for training.

B. Training and testing of the ANN-CNN

The soot number density $N_i(x)$ (Eq. (7)) for this steady flow problem is discretised over a number M of sections, $I_{v_i} \equiv [v_i^{inf}, v_i^{sup}]$, defined from a geometric grid $v_i^{inf} = v_o F_s^i$, with $F_s = 1.5$ and $v_o = m_o \times \rho_{soot}$ for $m_o = 2m_{\text{py}}$, representative of the mass of nascent soot particles from the collision of two pyrene molecules, corresponding to a diameter of 0.88 nm. The larger particle diameter to be considered is 51 nm. The ANN-CNN prediction is evaluated for $M = 10$ and 30 sections. The training is performed over a solution database obtained with the hybrid stochastic/fixed-sectional method using $N_P = 1000$ stochastic particles and a fixed number of 30 sections. The 10 sections ANN-CNN is trained interpolating the 30 sections database over 10 sections. The impact of using a smaller number of sections in the ANN-CNN simulation than for the solution providing the training database is then evaluated, as a strategy to reduce the computing time.

Figure 5 shows the distributions through the one-dimensional flame, i.e., from fresh to burnt gases, of number densities of the training database for representative sections. The simulations starts with all the sections set at zero ($N_i(x) = 0, \forall i$), then particles nucleate and undergo surface growths. Some number densities of particles, as for N_3 (diameter of 1.15 nm), reach a peak value before decreasing to a plateau, these particles will be those found in the burnt gases. For other characteristic sizes, such as 3.3 nm (N_{11} in Fig. 5), N_i is produced to then vanish in burnt

gases, because of thermophysical conditions favouring both agglomeration and oxidation of these particles.

The size of the database is 2906 (number of mesh points) $\times[M$ (number of sections) + 6 (nu10mber of thermochemical quantities & time step)], for example 104 616 data for $M = 30$ sections. To train the neural networks, 80% of the points through the flame are randomly selected. For this database, the time steps entering the ANN are within the interval $[2.5 \cdot 10^{-6}\text{s}, 7.5 \cdot 10^{-5}\text{s}]$ with a mean value centred at $1.0 \cdot 10^{-5}\text{s}$. This time-step distribution results from Eq. (24), thereby from the velocity profile through the flame and the non-uniform distribution of the mesh points through the reaction zone.

An Adam optimizer and a loss function based on mean absolute percentage error are used.[69] The 20% remaining data serve to test the ANN-CNN accuracy. The particle number densities, as predicted by the networks versus those of the hybrid stochastic/fixed-sectional method, are displayed in Figure 6. This result is obtained after 1500 epochs (number of times that the learning algorithm worked through the entire training dataset). The solution of the system converges with a maximum local error on the PSD of less than 10% and a mean error of less than 1%, which is sufficient to accurately predict the PSD in a posteriori tests performed, in which the error accumulate.

C. A posteriori tests

To test the method on conditions similar to those of its implementation in a flow solver, the balance equation for the $N_i(x)$ should now be solved. The equation for the $N_i(x)$ may be transformed into the moving frame for the steady one-dimensional flame

$$u\frac{dN_i(x)}{dx} = \frac{dN_i(x(\tau))}{d\tau} = \dot{\Omega}_i(\underline{x}(\tau))\,. \tag{25}$$

The derivative versus time, $dN_i(x(\tau))/d\tau$, is discretised using a third-order minimal storage Runge-Kutta scheme,[70] with the RHS $\dot{\Omega}_i(\underline{x}(\tau))$ of every sub-step of the integration reconstructed from the ANN-CNN (Eq. (11)) using Eq. (12).

For a number of sections of $M = 30$, Fig. 7 shows the comparison between the PSD predicted with the hybrid stochastic/fixed-sectional method to solve for the population balance equation and the results obtained with Eq. (25). The reference simulation of this flame by direct Monte Carlo solutions of non-inertial particles dynamics, which aims at performing a direct numerical

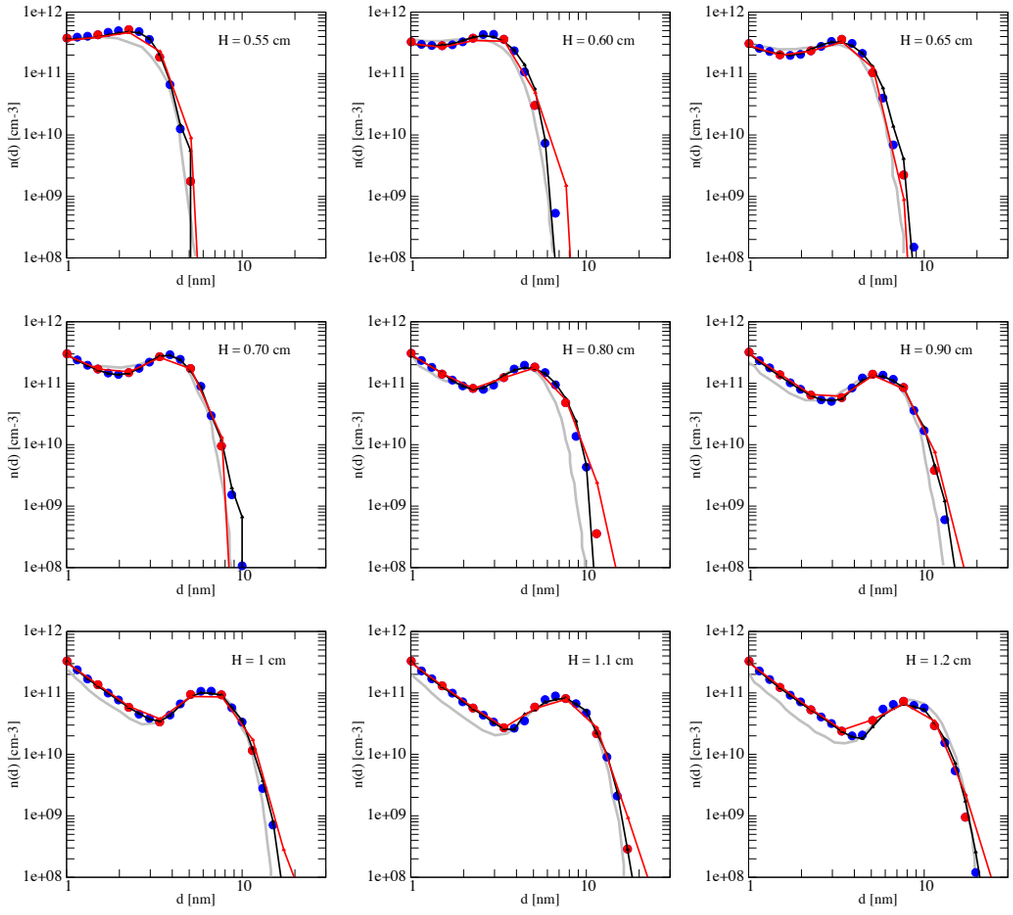


FIG. 7. Particle sizes distribution at different heights H through the one-dimensional premixed flame. Grey line: Reference simulation by Zhao et al.[67] Symbols: Training simulation with 1000 Monte-Carlo particles.[25] Solid line: ANN-CNN. Blue: $M = 30$ sections. Red: $M = 10$ sections.

simulation of the elementary physical phenomena acting on the particles,[67] is also given in these figures. As usually done for soot, in these plots the PSD is cast in $n(\mathrm{d}) = dN(\log(\mathrm{d}))/d\log(\mathrm{d})$, with d the particle diameter in nm. Figure 7 also suggests that the ANN-CNN operating with $M =$ 10 sections, but trained by interpolation over 30 sections and 1000 stochastic particles, provides results with an accuracy similar to the one obtained with the ANN-CNN operating with $M = 30$ sections. This confirms the potential of the approach relying on the training over a large database, to then reduce the dimensionality of the problem solved and thus minimise the computing cost.

The total number density $N_T = \int_{d_o}^{\infty} n(d) d\log(d)$ is compared against measurements in Fig. 8 for a diameter of the primary particle $d_o = 3$ nm (see[25] for a discussion on d_o). As expected from the results for the PSD, the ANN-CNN reproduces the evolution of $N_T(x)$ across the flame, as

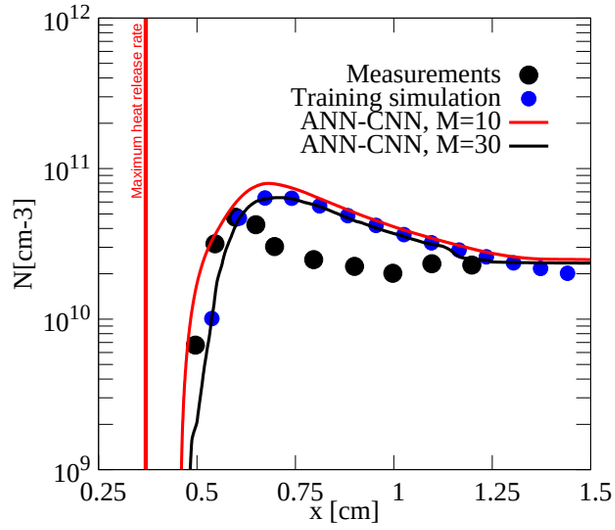


FIG. 8. Number density for particles above 3 nm. Black circle: Measurements.[67] Symbols: Training simulation with 1000 Monte-Carlo particles.[25] Solid lines: ANN-CNN, Black: $M = 30$ sections, Red: $M = 10$ sections.

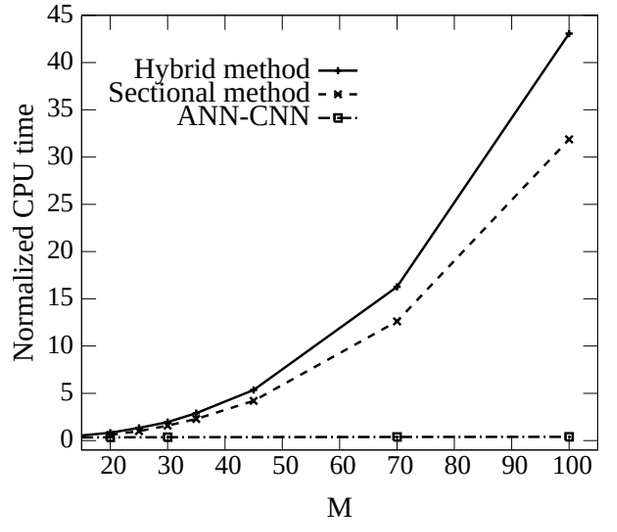


FIG. 9. Normalised computing cost versus the number of sections. Dashed line: Sectional two-point method.[59] Solid line: Hybrid stochastic/fixed-sectional method. Dashed-dotted line and square: ANN-CNN.

previously obtained solving the PBE with the hybrid method.

This operation is repeated for various numbers of sections. Figure 9 shows the corresponding computing cost normalised by the cost of a reference simulation with a two-point sectional method[59] with 25 sections. Here, the training and the a posteriori simulations are performed with

the same number of sections. The computing effort with the ANN-CNN varies only with the number of sections transported, i.e., the number of balance equations solved (Eq. (8)), in which the interactions between the sections is cast in a source term provided by the neural networks. Solving for the PBE (Eq. (2)) is much more demanding in terms of computing when the number of sections increases, because of the direct simulation of the nucleation, of the agglomeration, of the surface growth and of the oxidation processes. Thereby, a significant computing cost reduction is observed with the neural networks, up to more than 30 for 100 sections. Such reduction should allow for coupling detailed soot modeling with the flow solution over complex three-dimensional geometries.

IV. CONCLUSION

A strategy has been discussed to train a combination of artificial neural networks (ANN) and convolution neural networks (CNN) to solve for the population balance equation (PBE) of flowing non-inertial particles. The numerical framework to build the training database relies on a hybrid stochastic-fixed sectional approach, based on the probability density function (PDF) concept[22] in order to benefit from an accurate treatment of the surface growth/loss of the particles.

Applied to a canonical sooting flame, the ANN deals with the thermochemical inputs, driving the physics of the carbon particulate emissions, while the CNN processes the particle size distribution (PSD). Both networks are augmented with one additional long short-term memory artificial neural layer, before being concatenated to return the new PSD for a given increment in time, which is then used to construct a source term entering the eulerian transport equation for particle number densities of a few sections of sizes. For a given level of complexity of the thermo-physical modeling of the particles, which is present in the source terms through the neural networks, the computing cost for accurately transporting these sections stays very moderate compared to the usual cost of PBE solving. Therefore results confirm the potential of the approach, both in terms of accuracy and cost reduction.

Even though this procedure using neural networks seems promising for solving population balance equations, some points still need to be addressed. The major one is the choice of the training database to deal with three-dimensional flows. In the present work, the training, the testing of the neural networks and the subsequent simulation were performed on the very same flow configuration. As for any neural network based modeling, it is fully of practical interest only when the

training is performed on a generic problem, preferably aside form the targeted flow simulation.[28,71] This was recently achieved for detailed chemistry reduction and pre-integration with ANN using a stochastic micro-mixing problem solving for the PDF of the thermochemical variables, to train the networks without performing any flow simulation.[32] Coupling this approach with the present PBE solving with ANN-CNN would be an interesting route to follow in future work, in order to generalise the method to the application of three-dimensional turbulent combustion systems.

ACKNOWLEDGMENT

This paper in which stochastic methods are combined with neural networks was prepared to honour the memory of Professor Edward E. O'Brien, who pioneered and popularised the PDF concept for the numerical simulation of complex turbulent flows.

The PhD of the first author is funded by ANRT (Agence Nationale de la Recherche et de la Technology), SAFRAN-AE under the CIFRE No 1643/2016. Computing time has been provided by CRIANN (Centre Régional Informatique et d'Applications Numériques de Normandie).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon request. The code of the Hybrid Stochastic/Sectional method used for training is available at https://www.coria-cfd.fr/index.php/File:Nps-master.zip.

REFERENCES

[1]D. Ramkrishna, *Population balances. Theory and applications to particulate systems in engineering* (Academic Press San Diego, 2000) p. pp. 356.

[2]D. L. Marchisio and R. O. Fox, *Computational Models for Polydisperse Particulate and Multiphase Systems* (Cambridge University Press, 2013) p. 508.

[3]R. O. Fox, *Computational models for turbulent reacting flows* (Cambridge University Press, 2003).

[4]J. Solsvik and H. A. Jakobsen, "The foundation of the population balance equation: A review," Journal of Dispersion Science and Technology **36**, 510–520 (2015).

[5]A. Bouaniche, L. Vervisch, and P. Domingo, "A hybrid stochastic/fixed-sectional method for solving the population balance equation," Chem. Eng. Sci. **209**, 115198 (2019).

[6]A. Liu and S. Rigopoulos, "A conservative method for numerical solution of the population balance equation, and application to soot formation," Combust. Flame **205**, 506–521 (2019).

[7]D. L. Marchisio and R. O. Fox, eds., *Multiphase reacting flows: Modelling and simulation*, CISM International Center for Mechanical Sciences, Vol. 492 (Springer-Verlag, Wien, 2007) p. 269.

[8]F. Sewerin and S. Rigopoulos, "An explicit adaptive grid approach for the numerical solution of the population balance equation," Chemical Engineering Science **168**, 250 – 270 (2017).

[9]R. Irizarry, "Fast Monte Carlo methodology for multivariate particulate systems-I: Point ensemble Monte Carlo," Chemical Engineering Science **63**, 95–110 (2008).

[10]R. Irizarry, "Fast Monte Carlo methodology for multivariate particulate systems-II: τ-PEMC," Chemical Engineering Science **63**, 111–121 (2008).

[11]M. Singh, H. Y. Ismail, T. Matsoukas, A. B. Albadarin, and G. Walker, "Mass-based finite volume scheme for aggregation, growth and nucleation population balance equation," Proc. R. Soc. A **475**, 20190552 (2019).

[12]S. Salenbauch, C. Hasse, M. Vanni, and D. L. Marchisio, "A numerically robust method of moments with number density function reconstruction and its application to soot formation, growth and oxidation," J. Aerosol Sci. , 34–49 (2019).

[13]A. Boje, J. Akroyd, S. Sutcliffe, and M. Kraft, "Study of industrial titania synthesis using a hybrid particle-number and detailed particle model," Chemical Engineering Science **219**, 115615 (2020).

[14]F. Bisetti, A. Attili, and H. Pitsch, "Advancing predictive models for particulate formation in turbulent flames via massively parallel direct numerical simulations," Phil. Trans. R. Soc. A **372**, 20130324 (2014).

[15]A. Liu, C. E. Garcia, F. Sewerin, B. A. O. Williams, and S. Rigopoulos, "Population balance modelling and laser diagnostic validation of soot particle evolution in laminar ethylene diffusion flames," Combust. Flame **222**, 384–400 (2020).

[16]L. Cifuentes, J. Sellmann, I. Wlokas, and A. Kempf, "Direct numerical simulations of nanoparticle formation in premixed and non-premixed flame-vortex interactions," Phys. Fluids **32**, 093605 (2020).

[17]J. Boulanger, W. S. Neill, F. Liu, and G. J. Smallwood, "An improved phenomenological soot formation submodel for three-dimensional Diesel engine simulations: Extension to agglomeration of particles into clusters," J. Eng. Gas Turbines Power **130**, 062808 (2008).

[18]C. G. Moniruzzaman and F. Yu, "A 0D aircraft engine emission model with detailed chemistry and soot microphysics," Combust. Flame **159**, 1670–1686 (2012).

[19]B. Franzelli, A. Vié, and N. Darabiha, "A three-equation model for the prediction of soot emissions in LES of gas turbines," Proc. Combust. Inst. **37**, 5411–5419 (2019).

[20]E. Hopf, "Statistical hydromechanics and functional calculus," J. Rational Mech. Anal. **1**, 87–123 (1952).

[21]W. Kollmann, *Navier-Stokes Turbulence, Theory and Analysis* (Springer Nature Switzerland, 2019) p. 725.

[22]C. Dopazo and E. O'Brien, "Functional formulation of nonisothermal turbulent reactive flows," Phys. Fluids **17**, 1968–1975 (1974).

[23]E. E. O'Brien, "The probability density function (pdf) approach to reacting turbulent flows," in *Turbulent Reacting Flows* (Academic Press London, 1980) p. 185.

[24]R. E. Meyers and E. E. O'Brien, "The joint PDF of a scalar and its gradient at a point in a turbulent fluid," Combust. Sci. Tech. **26**, 123–134 (1981).

[25]A. Bouaniche, J. Yon, P. Domingo, and L. Vervisch, "Analysis of the soot particle size distribution in a laminar premixed flame: A hybrid stochastic/fixed-sectional approach," Flow Turbulence and Combust. **104**, 753–775 (2020).

[26]F. Christo, A. Masri, E. Nebot, and S. Pope, "An integrated PDF/neural network approach for simulating turbulent reacting systems," Symp. (Int.) Combust. **26**, 43–48 (1996).

[27]J. A. Blasco, N. Fueyo, C. Dopazo, and J. Ballester, "Modelling the temporal evolution of a reduced combustion chemical system with artificial neural network," Combust. Flame **113**, 38–52 (1998).

[28]L. L. Franke, A. K. Chatzopoulos, and S. Rigopoulos, "Tabulation of combustion chemistry via artificial neural networks (ANNs): Methodology and application to LES-PDF simulation of sydney flame L," Combust. Flame **185**, 245–260 (2017).

[29]F. R. R. Padilha and A. L. D. Bortoli, "Solutions for a laminar jet diffusion flame of methly formate using a skeletal mechanism obtained by applying ANNs," Journal of Mathematical Chemistry **57**, 2229–2247 (2019).

[30]O. Owoyele, P. Kundu, M. M. Ameen, T. Echekki, and S. Som, "Application of deep artificial neural networks to multi-dimensional flamelet libraries and spray flames," International Journal of Engine Research **21**, 151–168 (2020).

[31]R. Ranade, S. Alqahtani, A. Farooq, and T. Echekki, "An ann based hybrid chemistry framework for complex fuels," Fuel **241**, 625–636 (2019).

[32]K. Wan, C. Barnaud, L. Vervisch, and P. Domingo, "Chemistry reduction using machine learning trained from non-premixed micro-mixing modeling: Application to DNS of a syngas turbulent oxy-flame with side-wall effects," Combust. Flame **220**, 119–129 (2020).

[33]L. V. Lorang, B. Podvin, and P. L. Quéré, "Application of compact neural network for drag reduction in a turbulent channel flow at low Reynolds numbers," Phys. Fluids **20**, 045104 (2008).

[34]R. Ranade and T. Echekki, "A framework for data-based turbulent combustion closure: A posteriori validation," Combustion and flame **210**, 279–291 (2019).

[35]Z. Nikolaou, C. Chrysostomou, L. Vervisch, and R. S. Cant, "Progress variable variance and filtered rate modelling using convolutional neural networks and flamelet methods," Flow Turbulence Combust. **103**, 485–501 (2019).

[36]C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinsot, "Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates," Combust. Flame **203**, 255–264 (2019).

[37]A. Seltz, P. Domingo, L. Vervisch, and Z. M. Nikolaou, "Direct mapping from les resolved scales to filtered-flame generated manifolds using convolutional neural networks," Combust. Flame **210**, 71–82 (2019).

[38]X. Zhu, Z. Cai, J. Wu, Y. Cheng, and Q. Huang, "Convolutional neural network based combustion mode classification for condition monitoring in the supersonic combustor," Acta Astronautica **159**, 349–357 (2019).

[39]B. Liu, J. Tang, H. Huang, and X.-Y. Lu, "Deep learning methods for super-resolution reconstruction of turbulent flows," Phys. Fluids **32**, 025105 (2020).

[40]Z. M. Nikolaou, C. Chrysostomou, Y. Minamoto, and L. Vervisch, "Evaluation of a neural network-based closure for the unresolved stresses in turbulent premixed v-flames," Flow Turbulence and Combust. (in press).

[41]K. Wan, S. Hartl, L. Vervisch, P. Domingo, R. Barlow, and C. Hasse, "Combustion regime identification from machine learning trained by Raman/Rayleigh line measurements," Combust. Flame **219**, 168–274 (2020).

[42]G. Aversano, A. Bellemans, Z. Li, A. Coussement, O. Gicquel, and A. Parente, "Application of reduced-order models based on PCA & Kriging for the development of digital twins of reacting flow applications," Computers & Chemical Engineering **121**, 422–441 (2019).

[43]"Population balance model-based hybrid neural network for a pharmaceutical milling process," Journal of Pharmaceutical Innovation **5**, 161–168 (2010).

[44]P. Georgieva and S. F. de Azavedo, "Neural network-based control strategies applied to a fed-batch crystallization process," International Journal of Chemical and Molecular Engineering **1** (2007).

[45]G. Cogoni and P.-J. Frawley, "Particle size distribution reconstruction using a finite number of its moments through artificial neural networks: A practical application," Crystal Growth & Design **15**, 239–246 (2015).

[46]J. Schäfer, P. Schmitt, M. W. Hlawitschka, and H.-J. Bart, "Measuring particle size distribution in multiphase flows using a convolutional neural network," Chemie Ingenieur Technik **91**, 1688–1695 (2019).

[47]S. H. Bryngelson, A. Charalampopoulos, T. P. Sapsis, and T. Colonius, "A gaussian moment method and its augmentation via lstm recurrent neural networks for the statistics of cavitating bubble populations," International Journal of Multiphase Flow **127**, 103262 (2020).

[48]"Application of population balance theory for dynamic modeling of methane and ethane hydrate formation processes," Energy & Fuels **32**, 8131–8144 (2018).

[49]Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**, 436–444 (2015).

[50]J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks **61**, 85–117 (2015).

[51]H. H. Aghdam and E. J. Heravi, *Guide to Convolutional Neural Networks* (Springer, Switzerland Cham, 2017) p. 282.

[52]S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation **9**, 1735–1780 (1997).

[53]X.-S. Zhang, "Feedback neural networks," in *Neural Networks in Optimization. Nonconvex Optimization and its Application*, Vol. 46 (Springer, Boston, MA, 2000) pp. 137–175.

[54]S. Herzog, C. Tetzlaff, and F. Wörgötter, "Evolving artificial neural networks with feedback," Neural Networks **123**, 153–162 (2020).

[55]S. B. Pope, "Pdf method for turbulent reacting flows," Prog. Energy Combust. Sci. **11**, 119–195 (1985).

[56]W. Kollmann, "The pdf approach to turbulent flow," Theor. and Comp. Fluid Dynamics **1**, 349–285 (1990).

[57]D. C. Haworth, "Progress in probability density function methods for turbulent reacting flows," Prog. Energy Combust. Sci. **36**, 168–259 (2010).

[58]S. Kumar and D. Ramkrishna, "On the solution of population balance equations by discretization – I: A fixed pivot technique," Chemical Engineering Science **51**, 1311 – 1332 (1996).

[59]S. Park and S. Rogak, "A novel fixed-sectional model for the formation and growth of aerosol agglomerates," Journal of Aerosol Science **35**, 1385 – 1404 (2004).

[60]R. Wason, "Deep learning: Evolution and expansion," Cognitive Systems Research **52**, 701–708 (2018).

[61]J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics* (Springer, 1996) p. 364.

[62]J. Appel, H. Bockhorn, and M. Frenklach, "Kinetic modeling of soot formation with detailed chemistry and physics: laminar premixed flames of C2 hydrocarbons," Combust. Flame **121**, 122 – 136 (2000).

[63]D. Aubagnac-Karkar, J.-B. Michel, O. Colin, P. E. Vervisch-Kljakic, and N. Darabiha, "Sectional soot model coupled to tabulated chemistry for Diesel RANS simulations," Combust. Flame **162**, 3081 – 3099 (2015).

[64]A. Kazakov and M. Frenklach, "Dynamic modeling of soot particle coagulation and aggregation: Implementation with the method of moments and application to high-pressure laminar premixed flames," Combust. Flame **114**, 484 – 501 (1998).

[65]G. Blanquart and H. Pitsch, "A joint volume-surface-hydrogen multi-variate model for soot formation," in *Combustion Generated Fine Carbonaceous Particles*, edited by H. Bockhorn, A. D'Anna, A. F. Sarofim, and H. Wang (KIT Scientific Publisher, Karlsruhe, 2009) pp. 437–463.

[66]C. Eberle, P. Gerlinger, and M. Aigner, "A sectional PAH model with reversible PAH chemistry for CFD soot simulations," Combust. Flame **179**, 63 – 73 (2017).

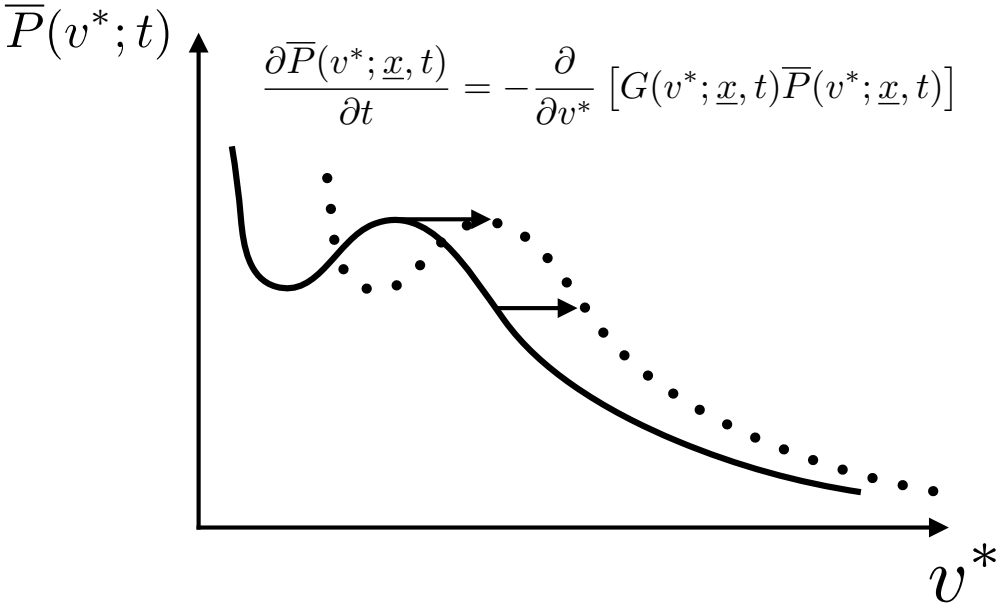
[67]B. Zhao, Z. Yang, M. V. Johnston, H. Wang, A. S. Wexler, M. Balthasar, and M. Kraft, "Measurement and numerical simulation of soot particle size distribution functions in a laminar premixed ethylene-oxygen-argon flame," Combust. Flame **133**, 173 – 188 (2003).

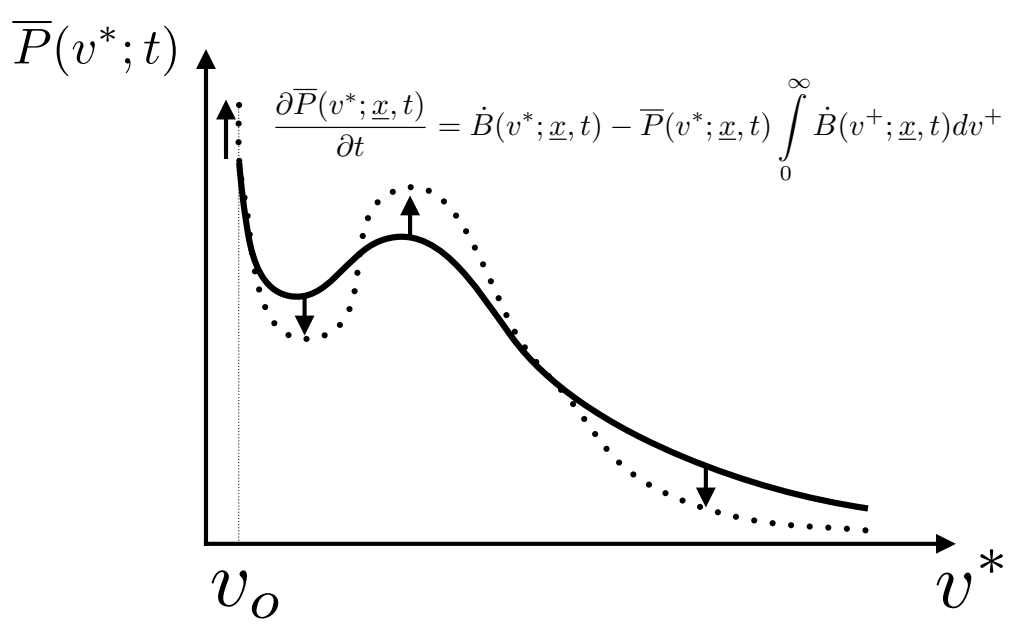
[68]D. G. Goodwin, H. K. Moffat, and R. L. Speth, "Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes," (2017), version 2.3.0.

[69]D. P. Kingma and J. L. Ba, "ADAM: A method for stochastic optimization," https://arxiv.org/pdf/1412.6980 (2017).

[70]A. A. Wray, "Minimal storage time-advancement schemes for spectral methods," Tech. Rep. (Center for turbulence research Report, Stanford University, 1990).

[71]A. K. Chatzopoulos and S. Rigopoulos, "A chemistry tabulation approach via Rate Controlled Constrained Equilibrium (RCCE) and Artificial Neural Networks (ANNs), with application to turbulent non-premixed $CH_4/H_2/N_2$ flames," Proceedings of the Combustion Institute **34**, 1465–1473 (2013).


$\overline{P}(v^*;t)$
$\frac{\partial \overline{P}(v^*;\underline{x},t)}{\partial t} = -\frac{\partial}{\partial v^*}\left[G(v^*;\underline{x},t)\overline{P}(v^*;\underline{x},t)\right]$
v^*


$\overline{P}(v^*;t)$
$\frac{\partial \overline{P}(v^*;\underline{x},t)}{\partial t} = \dot{B}(v^*;\underline{x},t) - \overline{P}(v^*;\underline{x},t) \int_0^\infty \dot{B}(v^+;\underline{x},t)dv^+$
v_o
v^*

$$\underline{\phi}(\underline{x},t) = \begin{pmatrix} \phi_1(\underline{x},t) \\ \cdot \\ \cdot \\ \cdot \\ \phi_{n_\phi}(\underline{x},t) \end{pmatrix}$$

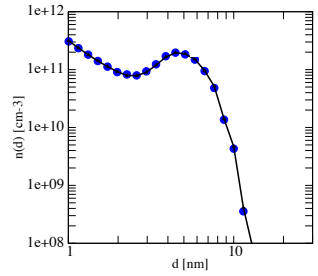
Thermochemical variables

ANN

CNN

keep in mind 5 previous time steps

keep in mind 5 previous time steps



$$\underline{N}(\underline{x},t) = \begin{pmatrix} N_1(\underline{x},t) \\ \cdot \\ \cdot \\ \cdot \\ N_M(\underline{x},t) \end{pmatrix}$$

15 layers of convolutional filters with a dropout rate of 10%

Concatenation

60 neurons

Dense layer (non-linear activation function)

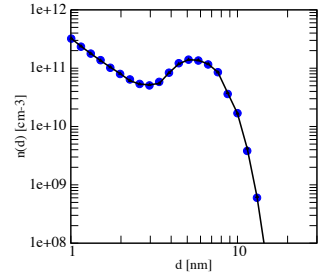
60 neurons

Dense layer (non-linear activation function)

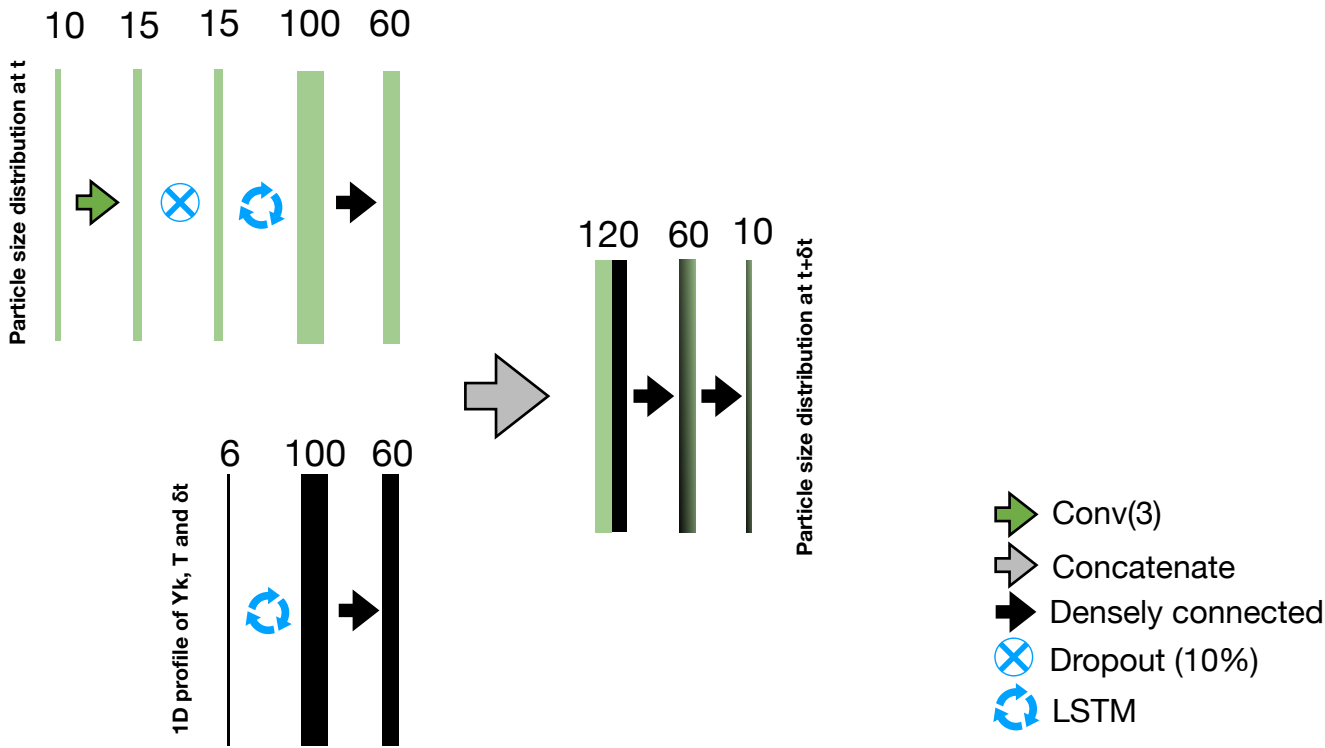
120 neurons

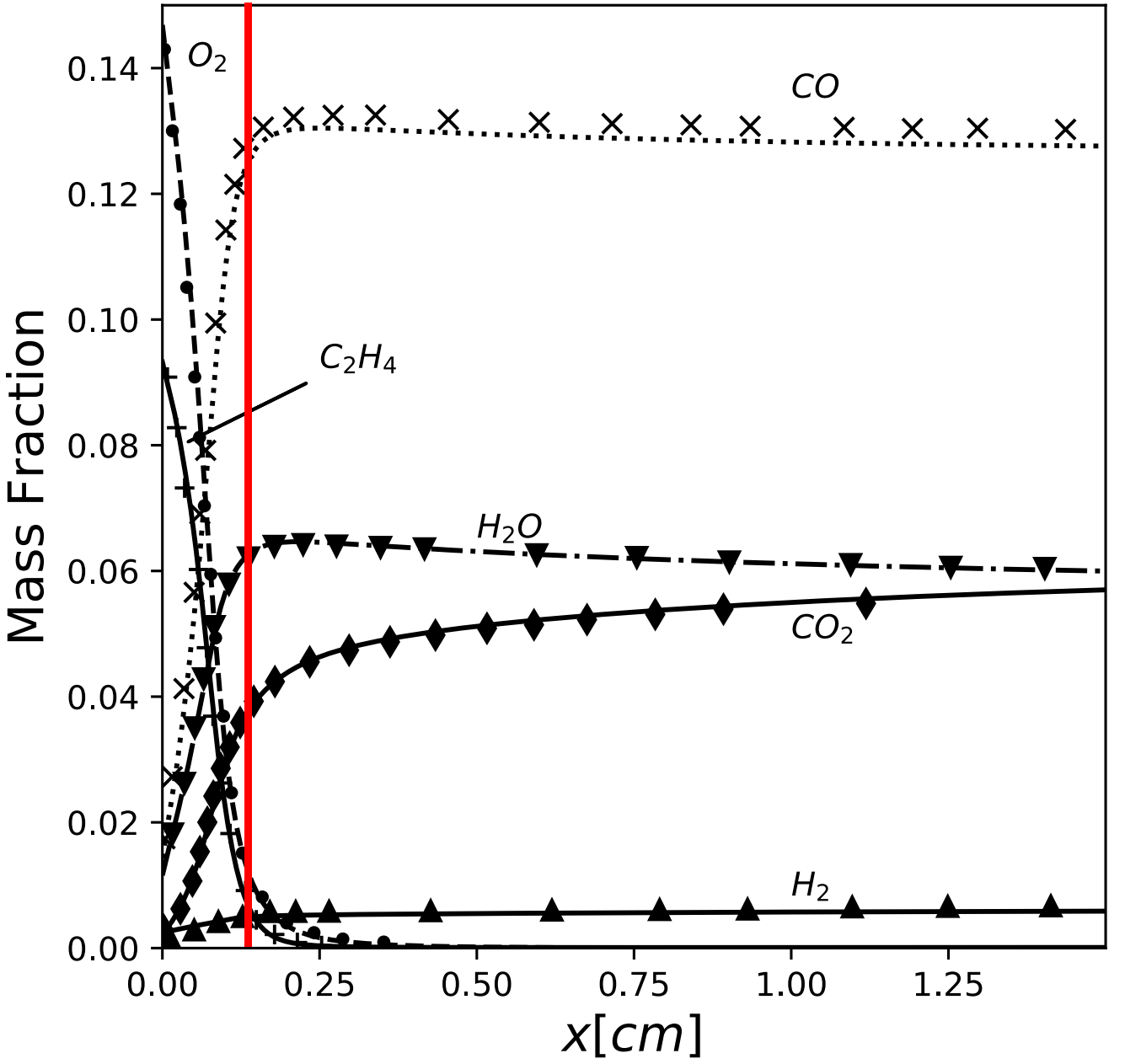
60 neurons

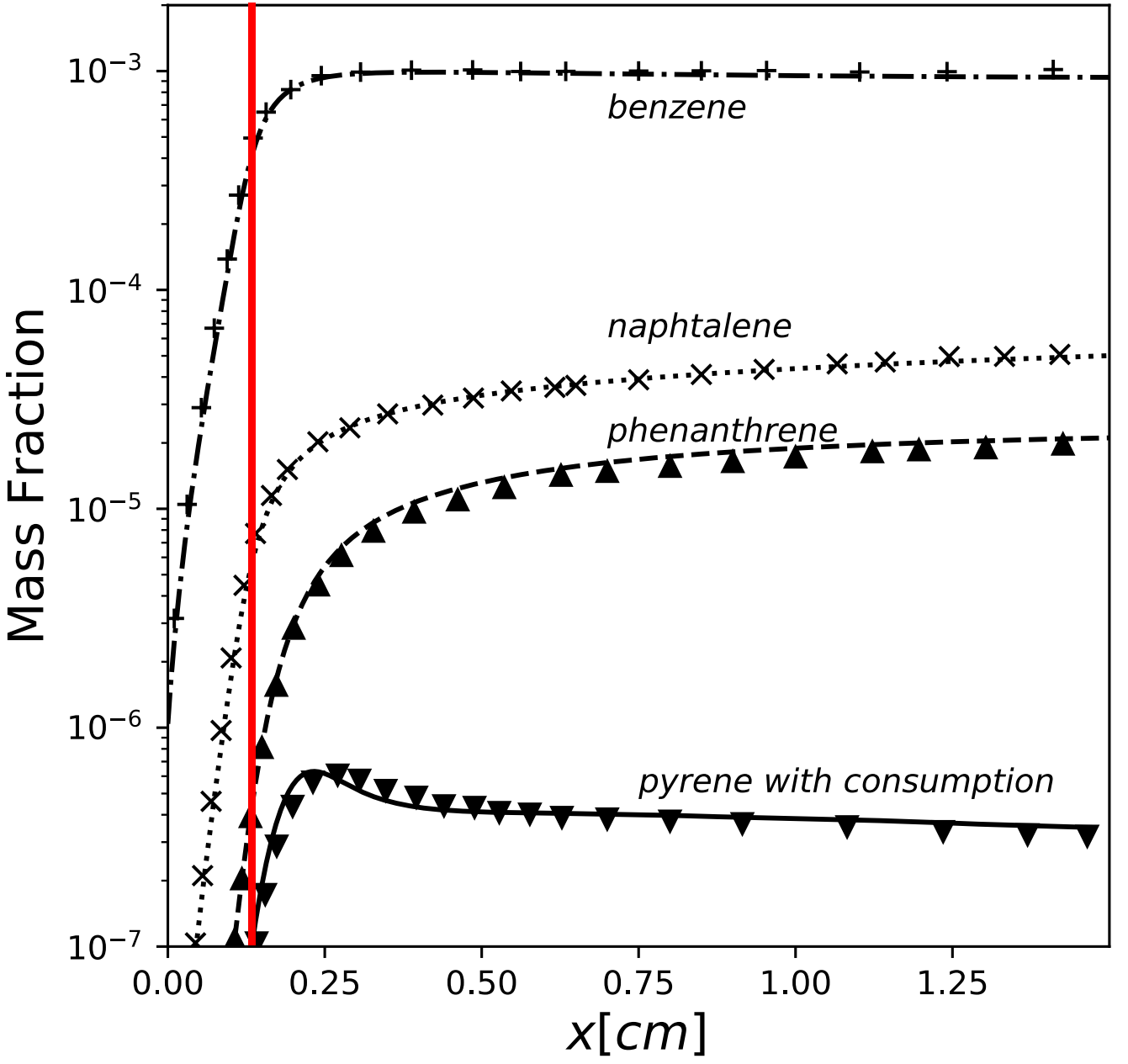
10 neurons

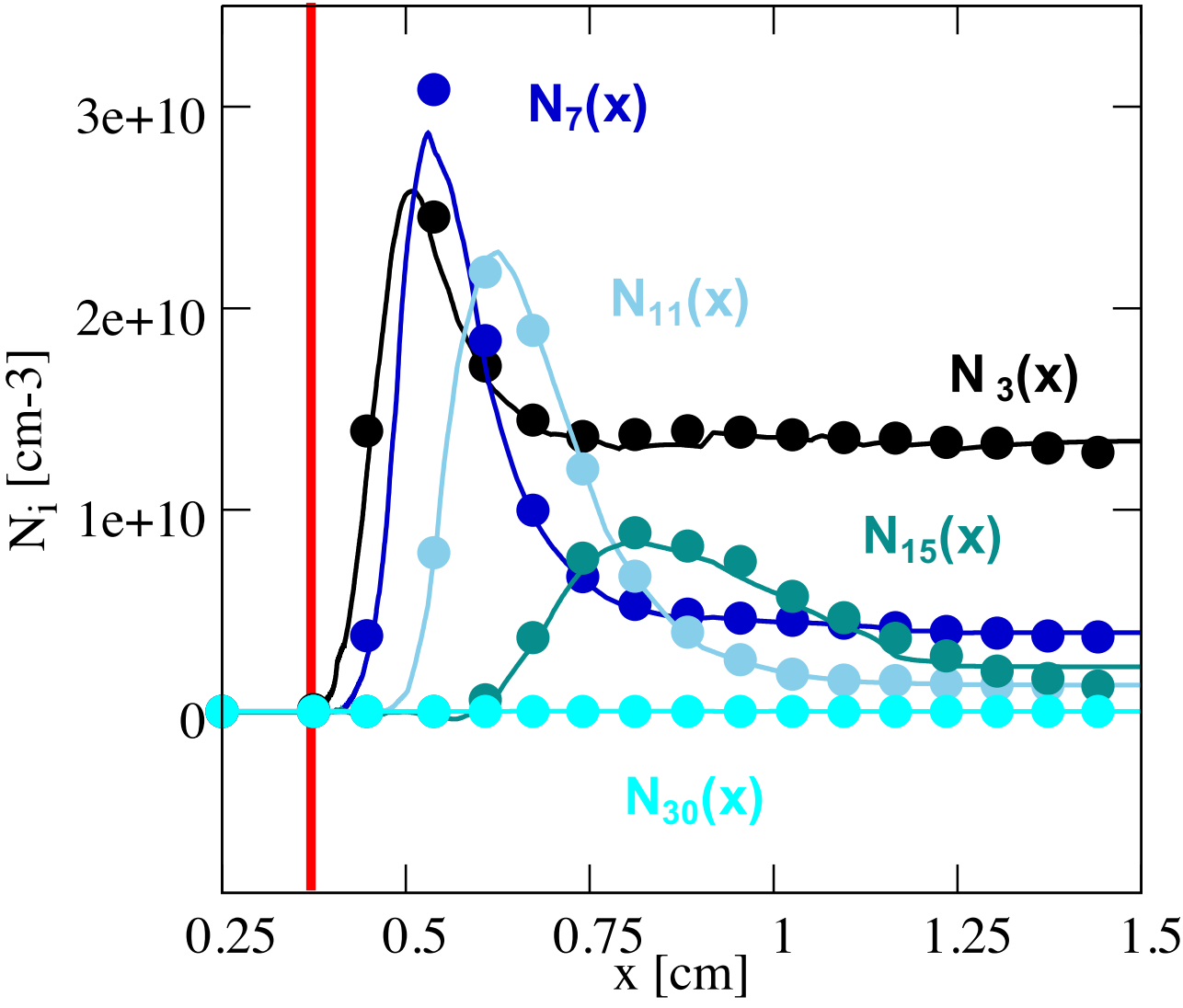


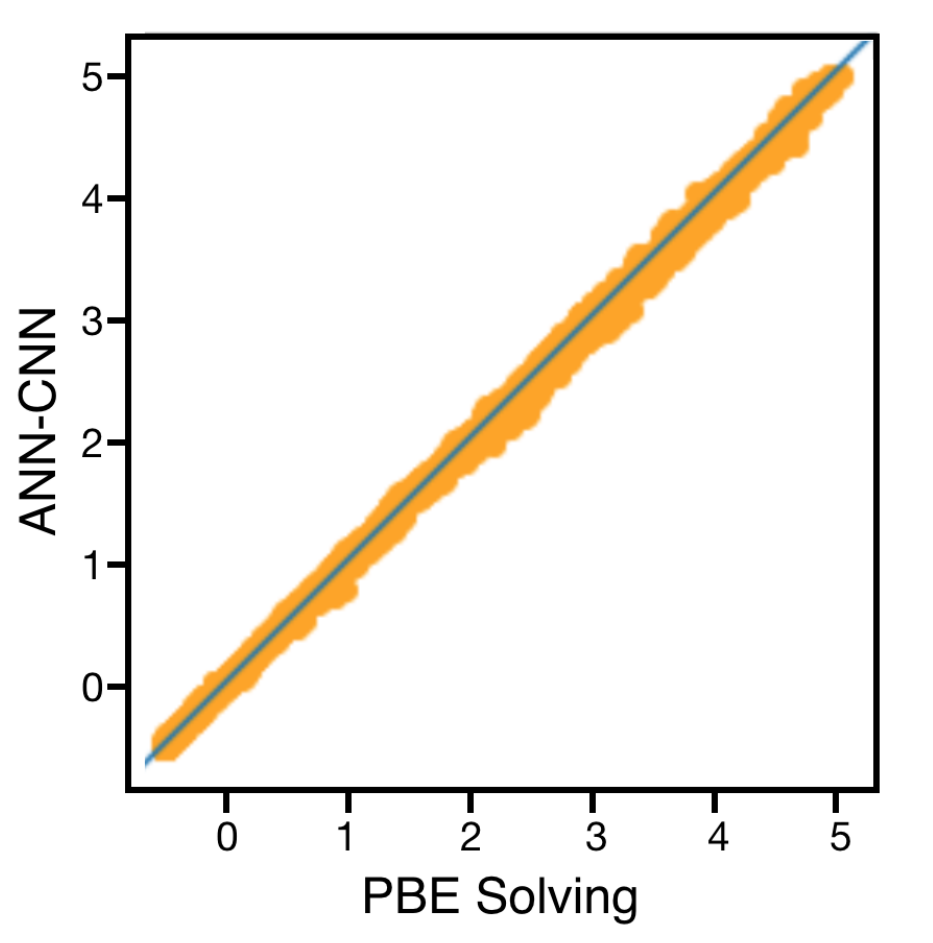
$$\underline{N}(\underline{x},t+\delta t) = \begin{pmatrix} N_1(\underline{x},t+\delta t) \\ \cdot \\ \cdot \\ \cdot \\ N_M(\underline{x},t+\delta t) \end{pmatrix}$$

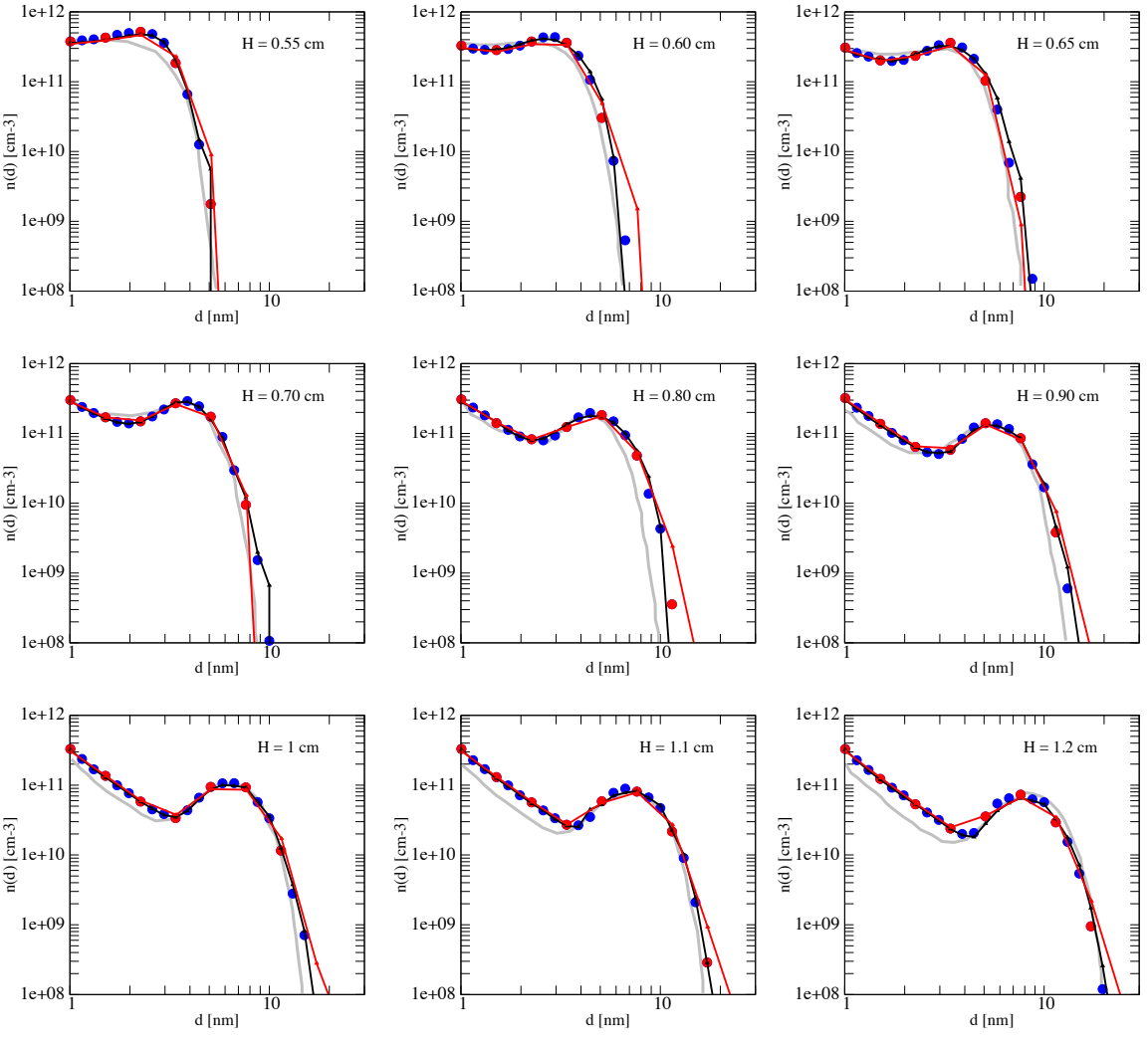

10
15
15
100
60
Particle size distribution at t
6
100
60
1D profile of Yk, T and δt
120
60
10
Particle size distribution at t+δt
Conv(3)
Concatenate
Densely connected
Dropout (10%)
LSTM

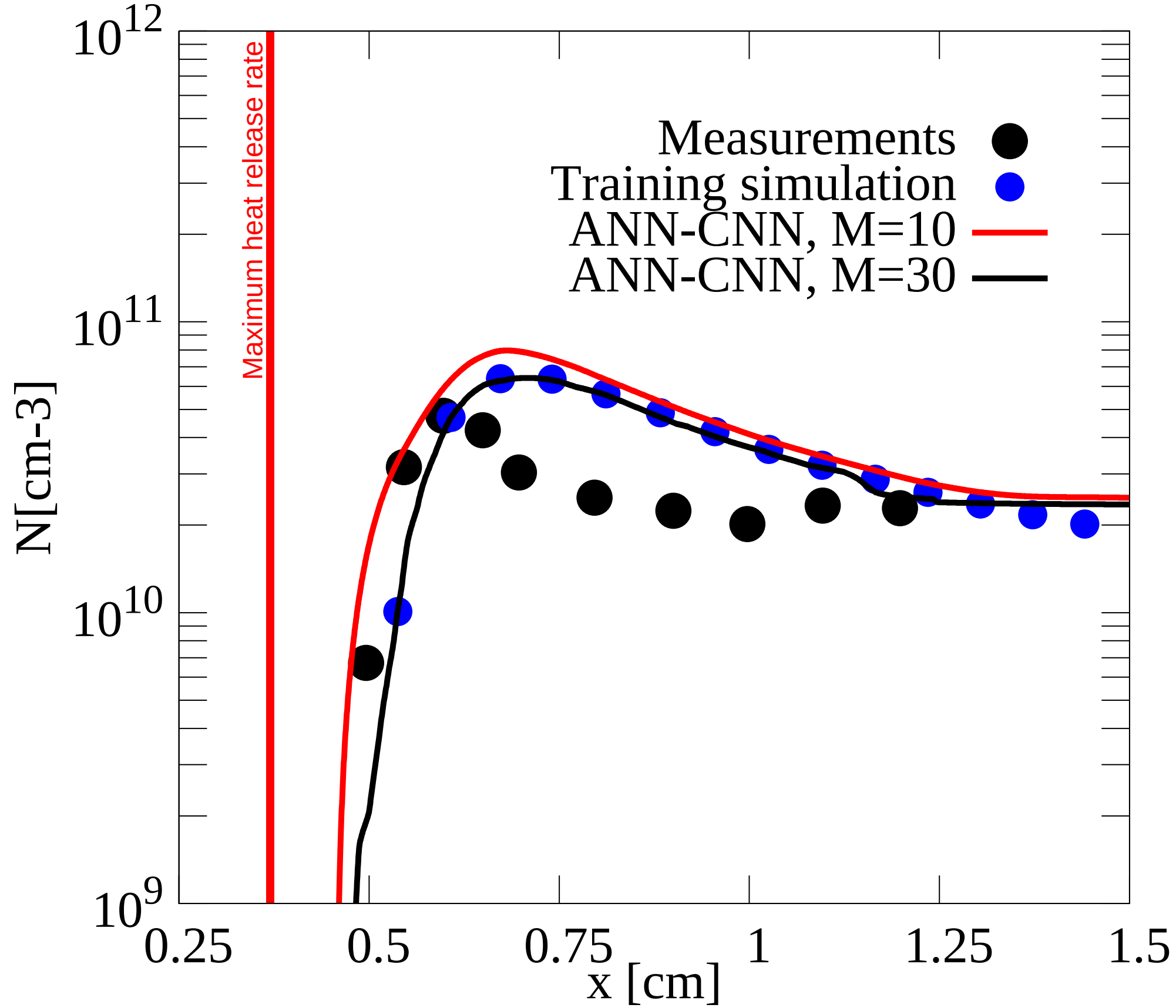

Mass Fraction
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.00
0.25
0.50
0.75
1.00
1.25
x[cm]
O2
CO
C2H4
H2O
CO2
H2

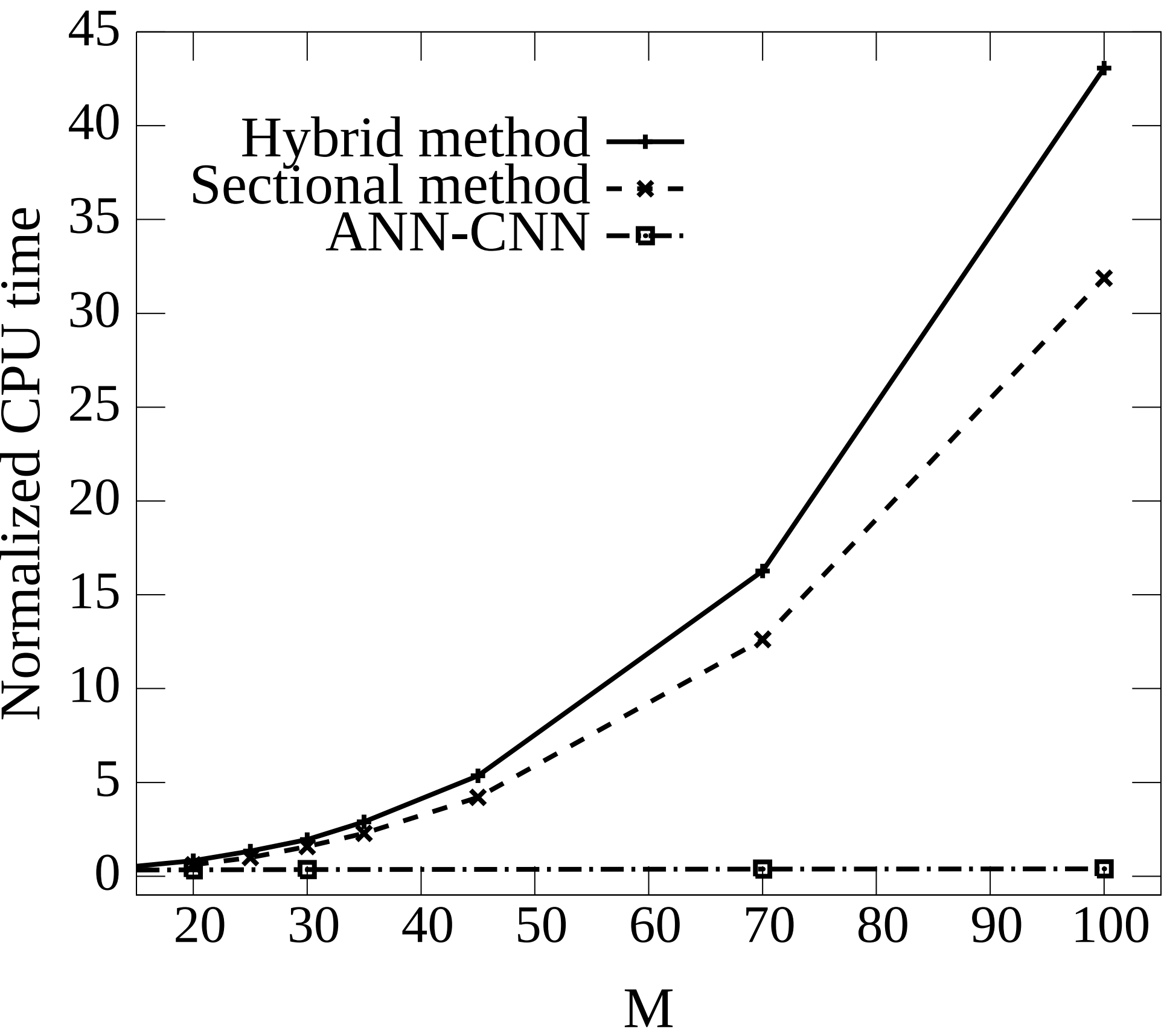

Mass Fraction
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
benzene
naphtalene
phenanthrene
pyrene with consumption
0.00
0.25
0.50
0.75
1.00
1.25
$x[cm]$


N7(x)
N11(x)
N 3(x)
N15(x)
N30(x)
3e+10
2e+10
1e+10
0
Ni [cm-3]
0.25
0.5
0.75
1
1.25
1.5
x [cm]


ANN-CNN
5
4
3
2
1
0
0
1
2
3
4
5
PBE Solving


H = 0.55 cm
H = 0.60 cm
H = 0.65 cm
H = 0.70 cm
H = 0.80 cm
H = 0.90 cm
H = 1 cm
H = 1.1 cm
H = 1.2 cm
n(d) [cm-3]
d [nm]
1e+12
1e+11
1e+10
1e+09
1e+08
1
10


Maximum heat release rate
Measurements
Training simulation
ANN-CNN, M=10
ANN-CNN, M=30
N[cm-3]
x [cm]
10^12
10^11
10^10
10^9
0.25
0.5
0.75
1
1.25
1.5


Hybrid method
Sectional method
ANN-CNN
Normalized CPU time
M
0
5
10
15
20
25
30
35
40
45
20
30
40
50
60
70
80
90
100