Application of Zone Method based Physics-Informed Neural Networks in Reheating Furnaces

Ujjal Kr Dutta
University College London
WC1E 6BT, UK

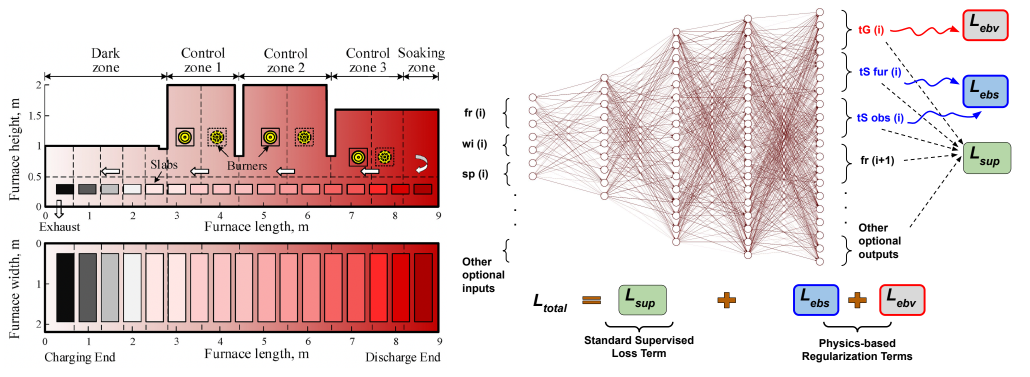
Aldo Lipani
University College London
WC1E 6BT, UK

Chuan Wang
Swerim AB
Box 812, SE-97125, Sweden

Yukun Hu
University College London
WC1E 6BT, UK
`yukun.hu@ucl.ac.uk`

Abstract

Foundation Industries (FIs) constitute glass, metals, cement, ceramics, bulk chemicals, paper, steel, etc. and provide crucial, foundational materials for a diverse set of economically relevant industries: automobiles, machinery, construction, household appliances, chemicals, etc. Reheating furnaces within the manufacturing chain of FIs are energy-intensive. Accurate and real-time prediction of underlying temperatures in reheating furnaces has the potential to reduce the overall heating time, thereby controlling the energy consumption for achieving the Net-Zero goals in FIs. In this paper, we cast this prediction as a regression task and explore neural networks due to their inherent capability of being effective and efficient, given adequate data. However, due to the infeasibility of achieving good-quality real data in scenarios like reheating furnaces, classical Hottel's zone method based computational model has been used to generate data for model training. To further enhance the Out-Of-Distribution generalization capability of the trained model, we propose a Physics-Informed Neural Network (PINN) by incorporating prior physical knowledge using a set of novel Energy-Balance regularizers.



(a) (b)

Figure 1: This figure is best viewed in color. Sub-figure (a) Illustration of a real-world furnace, and its subdivision as different zones. Image courtesy: [1]. A darker shade of red indicates a higher temperature. Under normal conditions, temperature increases towards the discharge end. Sub-figure (b) Illustration of incorporating zone method based regularization to train a Physics-Informed Neural Network (PINN).

1 Proposed Method

Introduction and Motivation: In this work, we tackle the key challenge of accurate and real-time temperature prediction in reheating furnaces, which are the energy-intensive bottlenecks common across the FIs. Available computational surrogate models based on Computational Fluid Dynamics (CFD) [2, 3], Discrete Element Method (DEM) [4], CFD-DEM hybrids [5], Two Fluid Models (TFM) [6], etc. incur expensive and time-consuming data acquisition, design, optimization, and high inference times (of the order of tens of seconds, up to minutes). Deep Learning (DL) methods, on the other hand, owing to their accuracy and their inherently faster inference times (often only in the order of milliseconds), are suitable candidates for real-time prediction.

But unlike other industry settings, only a limited number of thermo-couples could be physically deployed within furnaces, making it infeasible to get even near-optimal, large-scale, good-quality real-world data to train a data-hungry DL model. The classical Hottel's zone method [7, 8, 9, 10, 1] provides an elegant way (and superior, as studied by Yuen and Takara [9]) to model the physical phenomenon in high-temperature processes inside reheating furnaces.

Zone method for DL: In a real-world furnace as in Figure 1a, the release of combustion materials (by burners, controlled via their firing rates) and movement of objects to be heated (slabs, or obstacles) from the left to the right (discharge end, with higher temperature), causes energy and mass flow. The zone method mathematically models this by dividing the furnace into a set of zones: i) G: Gas/ volume and ii) S: Surface (consisting of furnace walls *fur* and obstacle surfaces *obs*). The radiation interchange ($\leftharpoonup$ indicates the direction of flow) among all possible pairs (i, j) of zones: Gas to Gas ($\overleftharpoon{\boldsymbol{G_i G_j}}$), Surface to Surface ($\overleftharpoon{\boldsymbol{S_i S_j}}$), Surface to Gas ($\overleftharpoon{\boldsymbol{G_i S_j}}$), and Gas to Surface ($\overleftharpoon{\boldsymbol{S_i G_j}}$), can be modeled along with a set of **Energy-Balance (EB) equations**.

Hu et al. [10] has proposed a computational model of the zone method, which though highly accurate, is slower for real-time prediction. We use it, and simulate an offline, IID data set $\mathcal{X}_{IID}$=$\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{N}$ for DL training, by following their algorithmic flow. The advantage of this model is that the required input entities (e.g., ambient temperatures, set point temperatures, firing rates, walk-interval) are readily available without dependency on the physical placement of sensors in every relevant location where we want to collect data in the real-world.

We study the following two settings:

1. **Input Setting 1 - Without previous temperatures in the input vector:** Here, for time step instance i, we set: $\boldsymbol{x}^{(i)} = [fr(i)^\top, wi(i)^\top, sp(i)^\top]^\top$, and
$\boldsymbol{y}^{(i)} = [tG(i)^\top, tS\ fur(i)^\top, tS\ obs(i)^\top, fr(i+1)^\top]^\top$.
2. **Input Setting 2 - With previous temperatures in the input vector:** Here, for time step instance i, we set: $\boldsymbol{x}^{(i)} = [fr(i)^\top, wi(i)^\top, sp(i)^\top, tG(i-1)^\top, tS\ fur(i-1)^\top, tS\ obs(i-1)^\top]^\top$, and
$\boldsymbol{y}^{(i)} = [tG(i)^\top, tS\ fur(i)^\top, tS\ obs(i)^\top, fr(i+1)^\top]^\top$.

Here, $fr(i)$, $wi(i)$, $sp(i)$, $tG(i)$, $tS\ fur(i)$ and $tS\ obs(i)$ are respectively the vectors containing firing rates, walk-interval, set point temperatures, gas zone temperatures, surface zone temperatures for furnace walls, and surface zone temperatures for obstacles for a time step i. Also, $tG(i-1)$, $tS\ fur(i-1)$, and $tS\ obs(i-1)$ are respective vectors containing the corresponding temperatures from the previous time step. $fr(i+1)$ is a vector containing firing rates for the next time step.

Using $\mathcal{X}_{IID}$=$\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{N}$, we can estimate parameters θ of a Multi-Layer Perceptron (MLP) model $f_\theta(.)$ by training it to predict $\boldsymbol{y}^{(i)}$ given $\boldsymbol{x}^{(i)}$, for all i, as:

$$\theta^* \leftarrow \arg\min_{\theta} \mathbb{E}_{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) \in \mathcal{X}_{IID}} [||\boldsymbol{y}^{(i)} - f_\theta(\boldsymbol{x}^{(i)})||_2^2] \quad (1)$$

Then, we can obtain the required values of temperatures by extracting them from $f_{\theta^*}(\boldsymbol{x}^{(i)})$.

Zone method based PINN: DL models are not naturally good at generalizing to Out-Of-Distribution (OOD) instances [11]. In our context, such OOD data could belong to furnace configurations (operating conditions) not seen during training. To tackle this, we propose employing a novel Physics-Informed Neural Network (PINN) model [12] based on MLP. This is done by incorporating prior physical knowledge based on the zone method using a set of our novel proposed Energy-Balance regularizers.

To explain our PINN (see Figure 1b), let us use eq(1) and denote: $\mathcal{L}_{sup} = \mathbb{E}_{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}) \in \mathcal{X}_{IID}}[||\boldsymbol{y}^{(i)} - f_\theta(\boldsymbol{x}^{(i)})||_2^2]$ as the standard *supervised term.* Then, the overall PINN loss is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{sup} + \lambda_{ebv}\mathcal{L}_{ebv} + \lambda_{ebs}\mathcal{L}_{ebs} \tag{2}$$

Here, $\lambda_{ebv}, \lambda_{ebs} > 0$ are hyper-parameters corresponding to $\mathcal{L}_{ebv}$ and $\mathcal{L}_{ebs}$, such that $\mathcal{L}_{ebv}$=$||\text{normalize}(\boldsymbol{v}_g)||_2^2$ is our proposed regularizer term corresponding to the **E**nergy-**B**alance equations for the **V**olume zones (**EBV**) using the zone method. Similarly, $\mathcal{L}_{ebs}$=$||\text{normalize}(\boldsymbol{v}_s)||_2^2$ is our proposed regularizer term corresponding to the **E**nergy-**B**alance equations for the **S**urface zones (**EBS**). Normalizing an output vector of a neural network in a regression task is standard practice for ensuring convergence. In our work, we use: $\text{normalize}(\boldsymbol{v}) = \boldsymbol{v}/\max(\boldsymbol{v})$, where $\max(\boldsymbol{v})$ is the maximum value from among all components in $\boldsymbol{v}$. We propose to represent $\boldsymbol{v}_g$ and $\boldsymbol{v}_s$ as:

$$\begin{aligned} \boldsymbol{v}_g &= (\boldsymbol{g}_{(g)arr} + \boldsymbol{s}_{(g)arr} - 4\boldsymbol{g}_{leave} + \boldsymbol{h}_g) \in \mathbb{R}^{|G|} \\ \boldsymbol{v}_s &= (\boldsymbol{s}_{(s)arr} + \boldsymbol{g}_{(s)arr} - \boldsymbol{s}_{leave} + \boldsymbol{h}_s) \in \mathbb{R}^{|S|} \end{aligned} \tag{3}$$

Here, $|G|/|S|$ denotes the number of Gas/ Surface zones. Intuitively, $\boldsymbol{v}_g$ and $\boldsymbol{v}_s$ are vector representatives corresponding to Energy-Balance equations for gas and surface zones respectively.

Having discussed $\boldsymbol{v}_g$ and $\boldsymbol{v}_s$, we now define the terms used to compute them. Let, $\boldsymbol{g}_{(g)arr} \in \mathbb{R}^{|G|}$ be a vector whose i^{th} entry represents the amount of radiation arriving at the i^{th} gas zone from all the other gas zones, $\boldsymbol{s}_{(g)arr} \in \mathbb{R}^{|G|}$, a vector whose i^{th} entry represents the amount of radiation arriving at the i^{th} gas zone from all the other surface zones, $\boldsymbol{g}_{leave} \in \mathbb{R}^{|G|}$, a vector whose i^{th} entry represents the amount of radiation leaving the i^{th} gas zone, and $\boldsymbol{h}_g \in \mathbb{R}^{|G|}$ a heat term. Also, let $T_{g,j}$ (or T_g) and $T_{s,j}$ (or T_s) denote the j^{th} gas and surface zone temperatures respectively. Then, following EBV equations, the i^{th} entries of $\boldsymbol{g}_{(g)arr}$, $\boldsymbol{s}_{(g)arr}$, $\boldsymbol{g}_{leave}$ and $\boldsymbol{h}_g$ can be computed as:

$$\begin{aligned} \boldsymbol{g}_{(g)arr}(i) &= \sum_j^{|G|} \overleftharpoon{\boldsymbol{G_i}\boldsymbol{G_j}}\sigma T_{g,j}^4 \\ \boldsymbol{s}_{(g)arr}(i) &= \sum_j^{|S|} \overleftharpoon{\boldsymbol{G_i}\boldsymbol{S_j}}\sigma T_{s,j}^4 \\ \boldsymbol{g}_{leave}(i) &= \sum_n^{|N_g|} a_{g,n}(T_{g,i})k_{g,n}\sigma V_i T_{g,i}^4 \\ \boldsymbol{h}_g(i) &= -(\dot{Q}_{conv})_i + (\dot{Q}_{fuel,net})_i + (\dot{Q}_a)_i + \boldsymbol{q}_i \end{aligned} \tag{4}$$

Here, the constants (known apriori) $(\dot{Q}_{conv})_i$, $(\dot{Q}_{fuel,net})_i$, and $(\dot{Q}_a)_i$ respectively denote the convection heat transfer, heat release due to input fuel, and thermal input from air/ oxygen. An enthalpy vector $\boldsymbol{q} \in \mathbb{R}^{|G|}$ is computed using the flow-pattern obtained via polynomial curve fitting during simulation. σ is the Stefan-Boltzmann constant, V_i is volume of i^{th} gas zone.

Let, $\boldsymbol{s}_{(s)arr} \in \mathbb{R}^{|S|}$, be a vector whose i^{th} entry represents the amount of radiation arriving at the i^{th} surface zone from all the other surface zones, $\boldsymbol{g}_{(s)arr} \in \mathbb{R}^{|S|}$, a vector whose i^{th} entry represents the amount of radiation arriving at the i^{th} surface zone from all the other gas zones, $\boldsymbol{s}_{leave} \in \mathbb{R}^{|S|}$, a vector whose i^{th} entry represents the amount of radiation leaving the i^{th} surface zone, and $\boldsymbol{h}_s \in \mathbb{R}^{|S|}$ a heat term. Then, following EBS equations, the i^{th} entries of $\boldsymbol{s}_{(s)arr}$, $\boldsymbol{g}_{(s)arr}$, $\boldsymbol{s}_{leave}$ and $\boldsymbol{h}_s$ can be computed as:

$$\begin{aligned} \boldsymbol{s}_{(s)arr}(i) &= \sum_j^{|S|} \overleftharpoon{\boldsymbol{S_i}\boldsymbol{S_j}}\sigma T_{s,j}^4 \\ \boldsymbol{g}_{(s)arr}(i) &= \sum_j^{|G|} \overleftharpoon{\boldsymbol{S_i}\boldsymbol{G_j}}\sigma T_{g,j}^4 \\ \boldsymbol{s}_{leave}(i) &= A_i\epsilon_i\sigma T_{s,i}^4 \\ \boldsymbol{h}_s(i) &= A_i(\dot{q}_{conv})_i - \dot{Q}_{s,i} \end{aligned} \tag{5}$$

For a surface zone i, the constants (known apriori) $A_i(\dot{q}_{conv})_i$ and $\dot{Q}_{s,i}$ respectively denote the heat flux to the surface by convection and heat transfer from it to the other surfaces. Here, A_i is the area,

Table 1: Comparison of our proposed method against naive baseline and MLP without physics-based regularizer, in an IID evaluation setting.

Without previous temperatures as inputs			
	Baseline Methods		Proposed Physics-Informed Method
Performance Metric	Naive Avg	MLP Baseline	EBV+EBS
RMSE tG (↓)	58.63	10.27	**10.04**
RMSE tS fur (↓)	53.03	8.94	**7.95**
RMSE tS obs (↓)	68.19	**30.94**	31.64
MAE tG (↓)	39.04	7.31	**7.19**
MAE tS fur (↓)	34.45	5.97	**5.58**
MAE tS obs (↓)	42.27	**14.95**	15.13
R^2 tG (↑)	-0.031	0.954	**0.961**
R^2 tS fur (↑)	-0.042	0.948	**0.959**
R^2 tS obs (↑)	-0.065	**0.886**	0.885
mMAPE fr (↓)	155.30	7.41	**6.84**

With previous temperatures as inputs			
	Baseline Methods		Proposed Physics-Informed Method
Performance Metric	Naive Avg	MLP Baseline	EBV+EBS
RMSE tG (↓)	58.63	5.75	**4.91**
RMSE tS fur (↓)	53.03	4.77	**4.24**
RMSE tS obs (↓)	68.19	**17.18**	17.39
MAE tG (↓)	39.04	3.21	**3.01**
MAE tS fur (↓)	34.45	3.09	**2.74**
MAE tS obs (↓)	42.27	**4.80**	5.81
R^2 tG (↑)	-0.031	0.984	**0.989**
R^2 tS fur (↑)	-0.042	0.983	**0.989**
R^2 tS obs (↑)	-0.065	**0.966**	**0.966**
mMAPE fr (↓)	155.30	7.86	**6.87**

Table 2: Comparison of our proposed method against MLP without physics-based regularizer, in an auto-regressive evaluation setting.

Without previous temperatures as inputs			
Metric/ Method	MLP	EBV+EBS	EBV+EBS improvement over MLP (in %)
RMSE tG (↓)	28.6	**27.3**	4.2
RMSE tS fur (↓)	10.1	**9.6**	4.8
RMSE tS obs (↓)	**42.7**	44.0	-3.1
MAE tG (↓)	17.1	**16.1**	5.8
MAE tS fur (↓)	7.8	**7.3**	6.5
MAE tS obs (↓)	**20.0**	20.2	-1.1
mMAPE fr (↓)	69.2	**63.5**	8.2

With previous temperatures as inputs			
Metric/ Method	MLP	EBV+EBS	EBV+EBS improvement over MLP (in %)
RMSE tG (↓)	74.1	**36.8**	50.3
RMSE tS fur (↓)	74.5	**25.8**	65.4
RMSE tS obs (↓)	83.3	**65.3**	21.5
MAE tG (↓)	48.8	**29.3**	39.9
MAE tS fur (↓)	49.7	**20.8**	58.2
MAE tS obs (↓)	53.6	**42.0**	21.6
mMAPE fr (↓)	96.2	**40.6**	57.8

and ϵ_i is the emissivity of the i^{th} surface zone. In eq(4), since the computations are being done for learning the gas zone related terms, the T_g terms after being obtained from $f_\theta(\boldsymbol{x})$ ($\boldsymbol{x}$: input tensor to the PINN) are kept associated with the computational graph for back-propagating, but not the T_s terms. The reverse is true in eq(5) where we are learning for the surface zone related terms, i.e., T_s terms are kept in the computational graph for back-propagating, but not T_g terms. In addition, eq(4) and eq(5) also contain the DFAs ($\overleftarrow{\boldsymbol{GS}} \in \mathbb{R}^{|G|\times|S|}$, $\overleftarrow{\boldsymbol{SS}} \in \mathbb{R}^{|S|\times|S|}$, $\overleftarrow{\boldsymbol{GG}} \in \mathbb{R}^{|G|\times|G|}$, and $\overleftarrow{\boldsymbol{SG}} \in \mathbb{R}^{|S|\times|G|}$) and terms such as $a_{g,n}(T_{g,i}), k_{g,n}$, which can be referred from Hu et al. [10].

2 Experimental Results

PINN vs MLP vs Naive Baseline: In our experiments, we compare our proposed PINN against a baseline MLP with the same architecture as our PINN, but without the physics-based EB regularizers (architecture and training details in appendix). We also compare a naive baseline, which, for a test instance, simply predicts the average value of a target variable using the training data. To evaluate the methods, we make use of an IID dataset (details in Appendix). For each test instance, we have input and output ground-truth values. We cast the prediction as a regression problem, and hence we can make use of the following standard regression performance evaluation metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Coefficient of determination (R^2). We separately compute a model's performance for gas zone temperatures, furnace surface zone temperatures and obstacle surface zone temperatures, thus resulting in the metrics RMSE tG, RMSE tS fur, RMSE tS obs, MAE tG, MAE tS fur, MAE tS obs, R^2 tG, R^2 tS fur, and R^2 tS obs.

We train all models in the IID training split, tune hyper-parameters using the validation split, and report all performance metrics on the test split. We also predict the next firing rates, and because they are in practice within the normalized range $[0, 1]$ [10], we make use of a modified Mean Absolute Percentage Error (mMAPE), by adding a small value $\epsilon = 0.05$ to the denominator of the MAPE computation (to scale up the metric values). A lower value of RMSE, MAE, and mMAPE indicates a better performance (indicated by $\downarrow$), while a higher value of R^2 indicates a better performance (indicated by $\uparrow$). The best obtained metric by a method shall be shown in **bold** in the result tables.

From Table 1, we noticed the superior performance of our proposed PINN over the MLP, as it better respects the underlying physics. When previous temperatures are provided in the inputs, due to additional signals, performance of both improves, but ours becomes better. In all cases, both the MLP and our PINN performs significantly better than the naive baseline, thereby, highlighting that learning based methods indeed help in this scenario.

We additionally perform evaluation on the test data in an Auto-Regressive (AR) manner: Only for the test data set, rearrange all the test instances of a furnace configuration according to their time step values. Now, use the model checkpoint obtained in an IID manner on the training data, to infer on the test data set for a configuration, and compute performance metrics. The metrics across all configurations in the test split are then averaged.

Specifically, in the inference time step t of the AR evaluation, instead of providing the trained model the input values of firing rates, gas and surface zone temperatures obtained during simulation, we rather use the model predicted values obtained in the inference time step $t-1$. This is similar to a real-world operation, where a deployed model would be expected to continuously predict different values and use them as inputs for the next model predictions.

In the IID evaluation setting (Table 1), at each inference step, the data is sampled IID, and the model inputs are those that are obtained via the simulation, which would be correct, as per the zone model. However, in the AR evaluation, over time, the model inputs being provided by its own predictions done earlier, are prone to cumulative error propagation. Thus, we can see that in Table 2, the values of performance metrics have degraded compared to the metrics obtained in the IID evaluation setting (Table 1). Even then, our PINN outperforms the baseline MLP, on an average.

Also, when previous temperatures are provided as inputs, this makes the AR evaluation more challenging. This is because there are now more input entities which could be predicted by the model sub-optimally. In this case, performance of the MLP deteriorates significantly. This might be because it merely learns to memorize the training data, without really understanding the underlying physical phenomenon. On the other hand, our method, being aware of the underlying physics, is more generalizable and hence performs significantly better than the MLP baseline (up to 50-65% improvements). In the appendix, we provide additional experimental results, including the in-depth analysis of our PINN.

Limitations of our PINN: The original data obtained by the computational model of [10] is time dependent in nature, i.e., data of a time step is dependent on the previous time step outputs. However, conforming to IID nature of data upon which standard ML/DL/MLP models are trained for regression, we simplified the structure of the data, to make it IID. However, one could keep the time dependent nature of the data intact and make use of a Recurrent Neural Network (RNN) to model the data during training. However, doing so is best justified as a separate future work.

At the same time, in Tables 1-2, we noticed that our PINN is unable to consistently improve the performance on the predicted obstacle surface zone temperatures ($tS\ obs$ performance). This is because our regularizer does not treat furnace and obstacle surfaces separately. Instead, the $\mathcal{L}_{ebs}$ term optimizes a collective loss across all surfaces (furnace and obstacles). The number of obstacle surfaces are higher in practice, than the number of furnace surfaces. Temperatures poorly predicted for a few of obstacle surfaces can increase the average errors. Our PINN regularizer does not proactively address this. Also, geometry and other features of the slabs (e.g., material quality) are not taken into account. While being geometry-agnostic is favorable to our framework in terms of simplicity and generalizability, there is a trade-off between formulating a generic vs specific model. These avenues can be addressed as independent works in the future.

3 Conclusion

While some loosely related prior works have touched upon aspects of radiative heat transfer, exchange area calculation [13, 14], genetic algorithm for nonlinear dynamic systems [15], neural network for absorption coefficients [16], view factor modeling with DEM-based simulations [17], near-field heat transfer or close regime [18], and some on non neural network based temperature profiling in reheating furnaces [19, 20, 21, 22, 23, 24, 25], casting the temperature prediction task in reheating furnaces as a regression task, and modeling via explicit physics-constrained regularizers as done in our work, is a first of its kind. In the future, our work could be extended for newer avenues, such as incorporating additional furnace geometries via transfer learning and continual learning.

Acknowledgments

The authors wish to acknowledge the Transforming Foundation Industries Network+ (EPSRC grant EP/V026402/1) for funding this work. Corresponding author's email: yukun.hu@ucl.ac.uk.

References

[1] Yukun Hu, CK Tan, John Niska, Jahedul Islam Chowdhury, Nazmiye Balta-Ozkan, Liz Varga, Paul Alun Roach, and Chunsheng Wang. Modelling and simulation of steel reheating processes under oxy-fuel combustion conditions–technical and environmental perspectives. *Energy*, 185:730–743, 2019.

[2] Gregor D Wehinger. Radiation matters in fixed-bed cfd simulations. *Chemie Ingenieur Technik*, 91(5):583–591, 2019.

[3] M De Beer, CG Du Toit, and PG Rousseau. A methodology to investigate the contribution of conduction and radiation heat transfer to the effective thermal conductivity of packed graphite pebble beds, including the wall effect. *Nuclear Engineering and Design*, 314:67–81, 2017.

[4] Heather N Emady, Kellie V Anderson, William G Borghard, Fernando J Muzzio, Benjamin J Glasser, and Alberto Cuitino. Prediction of conductive heating time scales of particles in a rotary drum. *Chemical Engineering Science*, 152:45–54, 2016.

[5] Tobias Oschmann and Harald Kruggel-Emden. A novel method for the calculation of particle heat conduction and resolved 3d wall heat transfer for the cfd/dem approach. *Powder Technology*, 338:289–303, 2018.

[6] Jan Marti, Andreas Haselbacher, and Aldo Steinfeld. A numerical investigation of gas-particle suspensions as heat transfer media for high-temperature concentrated solar power. *International Journal of Heat and Mass Transfer*, 90:1056–1070, 2015.

[7] HC Hottel and ES Cohen. Radiant heat exchange in a gas-filled enclosure: Allowance for nonuniformity of gas temperature. *AIChE Journal*, 4(1):3–14, 1958.

[8] Hoyt C Hottel and Adel F Saforim. *Radiative transfer*. McGraw-Hill, 1967.

[9] Walter W Yuen and Ezra E Takara. The zonal method: A practical solution method for radiative transfer in nonisothermal inhomogeneous media. *Annual review of heat transfer*, 8, 1997.

[10] Yukun Hu, CK Tan, Jonathan Broughton, and Paul Alun Roach. Development of a first-principles hybrid model for large-scale reheating furnaces. *Applied Energy*, 173:555–566, 2016.

[11] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2020.

[12] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[13] Hadi Ebrahimi, Akbar Zamaniyan, Jafar S Soltan Mohammadzadeh, and Ali Asghar Khalili. Zonal modeling of radiative heat transfer in industrial furnaces using simplified model for exchange area calculation. *Applied Mathematical Modelling*, 37(16-17):8004–8015, 2013.

[14] Matthieu Melot, Jean-Yves Trépanier, Ricardo Camarero, and Eddy Petro. Comparison of two models for radiative heat transfer in high temperature thermal plasmas. *Modelling and Simulation in Engineering*, 2011, 2011.

[15] Kang Li. Eng-genes: a new genetic modelling approach for nonlinear dynamic systems. *IFAC Proceedings Volumes*, 38(1):162–167, 2005.

[16] Walter W Yuen. Rad-nnet, a neural network based correlation developed for a realistic simulation of the non-gray radiative heat transfer effect in three-dimensional gas-particle mixtures. *International Journal of Heat and Mass Transfer*, 52(13-14):3159–3168, 2009.

[17] Josef Tausendschön and Stefan Radl. Deep neural network-based heat radiation modelling between particles and between walls and particles. *International Journal of Heat and Mass Transfer*, 177:121557, 2021.

[18] Juan José García-Esteban, Jorge Bravo-Abad, and Juan Carlos Cuevas. Deep learning for the modeling and inverse design of radiative heat transfer. *Physical Review Applied*, 16(6):064006, 2021.

[19] Jong Gyu Kim and Kang Y Huh. Prediction of transient slab temperature distribution in the re-heating furnace of a walking-beam type for rolling of steel slabs. *ISIJ international*, 40(11):1115–1123, 2000.

[20] Jung Hyun Jang, Dong Eun Lee, Man Young Kim, and Hyong Gon Kim. Investigation of the slab heating characteristics in a reheating furnace with the formation and growth of scale on the slab surface. *International Journal of Heat and Mass Transfer*, 53(19-20):4326–4332, 2010.

[21] Man Young Kim. A heat transfer model for the analysis of transient heating of the slab in a direct-fired walking beam type reheating furnace. *International Journal of Heat and Mass Transfer*, 50(19-20):3740–3748, 2007.

[22] Guangwu Tang, Bin Wu, Dengqi Bai, Yufeng Wang, Rick Bodnar, and Chenn Q Zhou. Modeling of the slab heating process in a walking beam reheating furnace for process optimization. *International Journal of Heat and Mass Transfer*, 113:1142–1151, 2017.

[23] Yukun Hu, CK Tan, Jonathan Broughton, Paul Alun Roach, and Liz Varga. Nonlinear dynamic simulation and control of large-scale reheating furnace operations using a zone method based model. *Applied Thermal Engineering*, 135:41–53, 2018.

[24] Guojun Li, Wenchao Ji, Linyang Wei, and Zhi Yi. A novel fuel supplies scheme based on the retrieval solutions of the decoupled zone method for reheating furnace. *International Communications in Heat and Mass Transfer*, 141:106572, 2023.

[25] Silvia Maria Zanoli, Crescenzo Pepe, and Lorenzo Orlietti. Multi-mode model predictive control approach for steel billets reheating furnaces. *Sensors*, 23(8):3966, 2023.

Appendix

Table 3: Dataset details. A total of 50 configurations have been used which are categorized as normal or abnormal.

Normal Behaviour Configurations (SP1<SP2<SP3)			
Type 1 (Varying SP1 only)	**Type 2 (Varying SP2 only)**	**Type 3 (Varying SP3 only)**	**Type 4 (Varying WI only)**
905_1220_1250_750.csv (Training)	955_1170_1250_750.csv (Training)	955_1220_1230_750.csv (Training)	955_1220_1250_675.csv (Training)
915_1220_1250_750.csv (Val)	955_1180_1250_750.csv (Val)	955_1220_1240_750.csv (Val)	955_1220_1250_690.csv (Val)
925_1220_1250_750.csv	955_1190_1250_750.csv	955_1220_1250_750.csv	955_1220_1250_705.csv
935_1220_1250_750.csv (Training)	955_1200_1250_750.csv (Training)	955_1220_1260_750.csv (Training)	955_1220_1250_720.csv (Training)
945_1220_1250_750.csv (Val)	955_1210_1250_750.csv (Val)	955_1220_1270_750.csv (Val)	955_1220_1250_735.csv (Val)
965_1220_1250_750.csv	955_1230_1250_750.csv	955_1220_1280_750.csv	955_1220_1250_765.csv
975_1220_1250_750.csv (Training)	955_1240_1250_750.csv (Training)	955_1220_1290_750.csv (Training)	955_1220_1250_780.csv (Training)
985_1220_1250_750.csv (Val)		955_1220_1300_750.csv	955_1220_1250_795.csv (Val)
995_1220_1250_750.csv			955_1220_1250_810.csv
			955_1220_1250_825.csv (Training)

Abnormal Behaviour Configurations/ Arbitrary SPs				
Type 1 (start@955-incr-dec/const)	**Type 2 (start@1220-incr-dec)**	**Type 3 (start@1220-dec-inc)**	**Type 4 (start@1250-dec-inc)**	**Type 5 (start@1250-dec-inc)**
955_1220_1200_750.csv (Training)	1220_1250_955_750.csv (Training)	1220_955_1250_750.csv (Training)	1250_955_1220_750.csv (Training)	1250_1220_955_750.csv (Training)
955_1220_1210_750.csv (Val)	1220_1250_955_795.csv	1220_955_1250_780.csv	1250_955_1220_825.csv	1250_1220_955_810.csv
955_1220_1220_750.csv				
955_1250_1220_750.csv (Training)				
955_1250_1220_765.csv (Val)				
955_1250_1250_750.csv				
955_1260_1250_750.csv (Training)				
955_1270_1250_750.csv				

Data set: For the IID data set generation, we make use of a FORTRAN code provided by the authors of [1], to represent various furnace configurations of the real-world furnace shown in Figure 1a. We consider 50 different configurations, and create disjoint train-val-test splits in such a way that there is no overlap in the data across different splits. Also, each configuration could belong to either of train/val/test split. As val/test data belong to furnace configurations different from that of training, it naturally makes the test data OOD in nature. Each configuration can be defined by set point temperatures and the walk-interval. Set point temperatures are essentially the desired temperatures that the furnace is expected to achieve at different stages/ zones.

We represent a configuration as: `SP1_SP2_SP3_WI`, where `SP1`, `SP2`, `SP3` and `WI` respectively denote the set point 1, set point 2, set point 3, and walk interval. Note that we consider configurations with both normal conditions (`SP1<SP2<SP3`, as naturally occurring in practice), as well as abnormal ones (arbitrary set points). The details are present in Table 3. Here, each configuration is represented by a .csv file containing 1500 time steps (and with the appropriate training/val label in parenthesis, and no label for a test split). Within a configuration, each time step is sampled with a 15s delay, to account for conduction analysis.

In Algorithm 1, we outline the key steps required in the data generation step, for a particular configuration. Please refer [10] for details on the flow. Here, the major entities as discussed in our formulation are mentioned: $fr(t), wi(t), sp(t), tG(t), tS\ fur(t), tS\ obs(t)$. In addition, we also make use of auxiliary entities such as enthalpy $\boldsymbol{q}^{(t)}$, heat-flux $\boldsymbol{w}^{(t)}$, and node temperatures $\boldsymbol{n}^{(t)}$.

Algorithm 1 Data generation algorithm

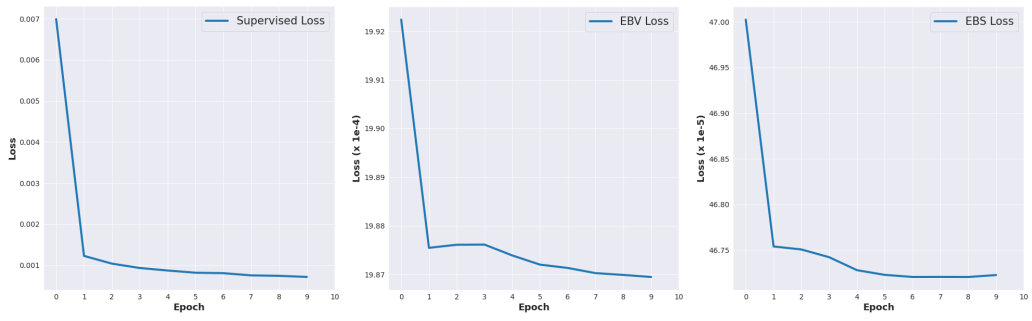
```
1: Initialize a furnace configuration via set points and walk interval.
2: Initialize $\mathcal{X} = \{\}, T > 0$ (max no. of steps).
3: Initialize $tG(0), tS\ fur(0), tS\ obs(0)$ with ambient temperatures, and $fr(0)$.
4: for t=1 to $T$ do                                                              ▷ t: time step
5:     $fr(t) \leftarrow$ update firing rates($fr(t-1)$, set point temperatures, $tG(t-1), tS\ fur(t-1), tS\ obs(t-1)$)
6:     $\boldsymbol{q}^{(t)} \leftarrow$ Enthalpy(Flow-pattern($fr(t)$))
7:     $\overleftarrow{\boldsymbol{GG}}^{(t)}, \overleftarrow{\boldsymbol{GS}}^{(t)}, \overleftarrow{\boldsymbol{SG}}^{(t)}, \overleftarrow{\boldsymbol{SS}}^{(t)} \leftarrow$ DFA($tG(t-1), tS\ fur(t-1), tS\ obs(t-1), \boldsymbol{GG}, \boldsymbol{GS}, \boldsymbol{SG}, \boldsymbol{SS}$)
8:     $tG(t) \leftarrow$ EBV($\boldsymbol{q}^{(t)}, \overleftarrow{\boldsymbol{GG}}^{(t)}, \overleftarrow{\boldsymbol{GS}}^{(t)}$)
9:     $\boldsymbol{w}^{(t)} \leftarrow$ heat-transfer($tG(t), tS\ fur(t-1), tS\ obs(t-1), \overleftarrow{\boldsymbol{SS}}^{(t)}, \overleftarrow{\boldsymbol{SG}}^{(t)}$)
10:    $tS\ fur(t), tS\ obs(t) \leftarrow$ EBS(conduction($\boldsymbol{w}^{(t)}$)), $\boldsymbol{n}^{(t)} \leftarrow$ conduction($\boldsymbol{w}^{(t)}$)
11:    $\mathcal{X}_t \leftarrow \{fr(t)$, Flow-pattern($fr(t)$), $\boldsymbol{q}^{(t)}, tG(t), tS\ fur(t), tS\ obs(t), \boldsymbol{w}^{(t)}, \boldsymbol{n}^{(t)}\}$
12:    $\mathcal{X} \leftarrow \mathcal{X} \cup \mathcal{X}_t$
13: end for
14: return $\mathcal{X}$
```

As the temperatures predicted in a time step influence the firing rates for the next time step, there is a time dependency among the data in $\mathcal{X}$. However, most standard off-the-shelf Machine Learning (ML)/ DL models suitable for regression require the data in an Independent and Identically Distributed (IID) format, that could be loaded in a tabular form (with each row being an instance and the columns representing the attributes). Thus, to convert $\mathcal{X}$ to $\mathcal{X}_{IID}$, we essentially add new columns and shift the entries.

Particularly, we add a new column `firing_rates_next` by shifting the original firing rates column a step back and then dropping the last row. Likewise, we add new columns for *prev* temperatures by shifting the original temperature columns a step forward and then dropping the first row. After rearranging the data as IID, we consolidate all the 20 training, 12 validation, and 18 test configurations (with 1500 minus 2 time steps per configuration), resulting in 29960 train, 17976 val, and 26964 test time steps/ IID samples. The 2 time steps are subtracted to account for the shift operations discussed during the IID data creation.

Training details and model architecture: We train our PINN model EBV+EBS for 10 epochs using PyTorch, with early stopping to avoid over-fitting. For the EB equations, we perform the same normalization for enthalpy, flux, and temperatures, as in the final neural network output as discussed earlier. We found a learning rate of 0.001 with Adam optimizer and batch size of 64 to be optimal, along with ReLU non-linearity.

We pick the [50,100,200] configuration for hidden layers, i.e., 3 hidden layers, with 50, 100, and 200 neurons respectively. We use $\lambda_{ebv} = \lambda_{ebs} = 0.1$. In general, a value lesser than 1 is observed to be better, otherwise, the model focuses less on the regression task. Following are values of other variables: $|G| = 24$, $|S| = 178$ (76 furnace surface zones and 102 obstacle surface zones), $N_g = 6$, and Stefan-Boltzmann constant=5.6687e-08. Unless otherwise stated, this is the setting we use to report any results for our method, for example, while comparing with other methods.



(a) (b) (c)

Figure 2: Convergence behaviour of our method, considering: a) Supervised, b) EBV, and c) EBS terms.

Table 4: Performance of EBV+EBS (ReLU) variant of our method against varying hidden layer configurations.

Metric/ Hidden layer configuration	[100]	[50,100]	**[50,100, 200]**	[50,100, 200,200]	[50,100, 200,200, 205,205]
RMSE tG (↓)	11.64	17.25	**10.04**	10.84	14.27
RMSE tS fur (↓)	10.05	15.23	7.95	**7.83**	12.46
RMSE tS obs (↓)	34.82	37.62	**31.64**	33.57	36.42
mMAPE fr (↓)	8.76	9.15	**6.84**	8.06	7.51

Table 5: Performance of the proposed EBV+EBS variant using different batch sizes.

Metric	EBV+EBS ReLU bsz=32	EBV+EBS ReLU bsz=64	EBV+EBS ReLU bsz=128
RMSE tG (↓)	12.70	**10.04**	10.73
RMSE tS fur (↓)	9.14	**7.95**	9.69
RMSE tS obs (↓)	39.75	**31.64**	31.79
mMAPE fr (↓)	**5.24**	6.84	8.29

Table 6: Effect of individual regularizer terms in our method.

Metric	EBV only	EBS only	EBV+EBS
RMSE tG (↓)	11.85	11.66	**10.04**
RMSE tS fur (↓)	10.36	11.07	**7.95**
RMSE tS obs (↓)	32.46	32.04	**31.64**
mMAPE fr (↓)	**6.42**	7.53	6.84

Table 7: Performance of our method using different activation functions in the underlying network.

Metric	EBV+EBS ReLU	EBV+EBS GeLU	EBV+EBS SiLU	EBV+EBS Hardswish	EBV+EBS Mish
RMSE tG (↓)	**10.04**	13.57	10.07	15.26	10.16
RMSE tS fur (↓)	7.95	8.86	8.02	14.02	**7.71**
RMSE tS obs (↓)	31.64	39.65	31.64	36.23	**31.63**
mMAPE fr (↓)	6.84	**5.88**	6.23	7.03	6.33

Table 8: Comparison of our proposed method against classical ML baselines, in an IID evaluation setting.

Without previous temperatures as inputs

	Classical ML Baselines			Proposed Physics-Informed Method
Performance Metric	DT	RF	H-GBoost	EBV+EBS
RMSE tG (↓)	12.84	12.24	14.06	**10.04**
RMSE tS fur (↓)	9.42	8.97	10.09	**7.95**
RMSE tS obs (↓)	42.86	42.06	42.73	**31.64**
R^2 tG (↑)	0.943	0.948	0.925	**0.961**
R^2 tS fur (↑)	0.951	0.957	0.934	**0.959**
R^2 tS obs (↑)	0.788	0.798	0.763	**0.885**
mMAPE fr (↓)	5.50	5.30	**2.32**	6.84

With previous temperatures as inputs

	Classical ML Baselines			Proposed Physics-Informed Method
Performance Metric	DT	RF	H-GBoost	EBV+EBS
RMSE tG (↓)	11.17	6.96	5.00	**4.91**
RMSE tS fur (↓)	10.24	6.15	6.12	**4.24**
RMSE tS obs (↓)	43.05	32.81	23.01	**17.39**
R^2 tG (↑)	0.925	0.979	**0.989**	**0.989**
R^2 tS fur (↑)	0.915	0.977	0.983	**0.989**
R^2 tS obs (↑)	0.729	0.890	0.937	**0.966**
mMAPE fr (↓)	6.98	8.09	**0.76**	6.87

In-depth analysis of our PINN: To study our PINN in detail, we vary different aspects of our method (e.g., the impact of individual loss/regularization terms, hidden layer configuration, batch size, and activation functions). At a time, we vary one focused aspect, and fix all other hyper-parameters as per the default setting prescribed above.

Firstly, we study the empirical convergence of the default setting of our method. Fig 2 plots the convergence behaviour of each of the loss terms individually (supervised, EBV, and EBS). Our method, as shown, enjoys a good convergence. In Table 4, we report the performance of our method by varying the hidden layer configurations (e.g., $[100]$ denotes one hidden layer with 100 neurons, $[50, 100]$ denotes two hidden layers with 50, and 100 neurons respectively, and so forth). The maximum values for each row (corresponding to a metric) are shown in bold. We found that it suffices to use [50, 100, 200] configuration for a competitive performance.

In Table 5, we vary the batch size in our method. We found a batch size of 64 to provide an optimal performance for our experiments. In Table 6, we study the effect of individual physics-based regularization terms used in our method. We found that using both instances of volume and surface zone based regularizers together leads to better performance as compared to either EBV or EBS in isolation.

In Table 7, we vary the underlying activation functions throughout our model. While we observed the benefits of using ReLU, SiLU, and Mish over others, there is no clear winner. All three lead to competitive performance. But when it comes to consistent performance across batch sizes, we noticed from our experiments that ReLU is more robust. Thus, we could recommend using the basic ReLU as de facto in our experiments.

Additional comparisons of our PINN against classical ML techniques: In addition to DL, we also compare our method against the classical ML baselines (in an IID evaluation setting): i) Decision Tree (DT), ii) Random Forest (RF), and iii) Histogram Gradient Boosting (H-GBoost). When it comes to only the classical ML baselines, the performances are as per the expectation. For instance, with previous temperatures as inputs, the performance of DT, RF, and H-GBoost increases. However, being an ensemble learning method, RF performs superior to DT. At the same time, by virtue of boosting, among all the three classical methods, H-GBoost performs the best. We observe superior performance of our model against the classical baselines as well, as reported in Table 8. H-GBoost, though competitive, is significantly slower for our studied case of multi-output regression.