International Journal of Heat and Mass Transfer 177 (2021) 121557

Contents lists available at ScienceDirect

International Journal of Heat and Mass Transfer

journal homepage: www.elsevier.com/locate/hmt

Deep neural network-based heat radiation modelling between particles and between walls and particles

Josef Tausendschön, Stefan Radl

Institute of Process and Particle Engineering, Graz University of Technology, Inffeldgasse 13/III, Graz 8010, Austria

ARTICLE INFO

Article history:
Received 28 February 2021
Revised 17 May 2021
Accepted 1 June 2021

Keywords:
Deep Neural Networks
Discrete Element Method (DEM)
Radiative heat transfer
Radiation modelling
Machine learning
View factors
Wall radiation

ABSTRACT

We present a Deep Neural Network (DNN)-based view factor model to calculate radiative heat transfer rates between particles, as well as between particles and walls in Discrete Element Method (DEM)-based simulations. A systematic analysis of the most promising markers available in DEM simulations to be correlated with the view factor is performed. Subsequently, a neural network is trained, and its predictive performance is analyzed. View factors are studied for a variety of systems ranging from dilute particle systems to dense (i.e., settled under gravity) particle beds. It is demonstrated that the trained DNN-model can model view factors at higher accuracy and with significantly less computational effort than other literature models. A validation with experimental data is provided, showing that the implemented model can predict the total heat flux, as well as particle temperatures in a packed bed accurately.

1. Introduction

Radiative heat transfer is the dominating heat transfer mechanism in many high temperature processes. Such processes are typically operating at temperatures above 700 °C and appear in a widespread industrial sectors: from pebble bed nuclear reactors, chemical reactors [1], fluidized beds [2], solar energy exploitation [3,4], selective laser sintering in 3D printing [5,6], infrared drying [7,8] to thermal processes including rotary kilns [9–11]. Also, thermal transport under vacuum conditions in astronomical applications heavily relies on radiative heat transfer.

Radiative heat transfer naturally occurs among the other heat transfer mechanisms: conduction and convection. A separated experimental investigation of radiative heat transfer is very complex and due to that fact rarely performed. The lack of separated experimental data causes difficulties in validating numerical models for conduction, convection and radiation. Therefore, radiative heat transfer is often investigated as one of many contributions to the total transferred heat in an experiment. The same is often true for theoretical or numerical investigations that are then focusing on such experiments. This experimentally-driven approach typically yields correlations [9,12–14] to describe the total heat transfer rate.

For the above mentioned processes, Computational Fluid Dynamics (CFD) [15,16], the Discrete Element Method (DEM) [10,11,17,18], the combination of both CFD-DEM [19–22], as well as Two Fluid Models (TFM) [4,23] can be used to numerically describe these processes. De Beer et al. [16] presented a methodology to separate between conductive and radiative heat transfer in their experiments [24] based on CFD simulations. Thus, these more advanced simulation tools offer some advantage over an experimentally-driven approach.

CFD and TFM simulations are used with participating media and rely on solving the radiative transfer equation (RTE) [25]. The radiative transport is then typically described via finite volume Discrete Ordinate Methods (fvDOM) [26], or by (simple) differential approximations, e.g., the so-called “P1 model” [4,25]. If no media is contributing to the radiative heat transfer, or the contribution is neglectable, the surface-to-surface radiation can be described with view factors (in literature also called “configuration factors”). View factors model the radiative net exchange between two objects, based on the ratio of emitted and absorbed radiation between the two surfaces.

Another discrete possibility is the Discrete Transfer Radiation Model (DTRM), that predict the rays’ propagation from the emitting surface through the simulation domain. DTRM is typically based on Monte Carlo methods using a statistically significant (i.e., large) number of rays. This so-called “Monte Carlo ray-tracing” (MCRT) approach can either be used to calculate the radiative heat transfer between two surfaces directly [5,6], or also to determine view factors between two surfaces [27,28].

However, the discrete perspective of MCRT naturally fits to DEM-based simulations: Amberger et al. [29] introduced MCRT in

E-mail address: josef.tausendschoen@tugraz.at (J. Tausendschön).

https://doi.org/10.1016/j.ijheatmasstransfer.2021.121557

Nomenclature

Latin symbols

A	area [m²]
c_p	thermal capacity [J/(kg K)]
C	set of contacts [-]
d	diameter [m]
$\boldsymbol{F}$	force [N]
$\boldsymbol{g}$	gravity [m/s²]
h	activation function
I	moment of inertia [kg m²]
k	thermal conductivity [W/(m K)]
m	mass [kg]
MSE	mean squared error [-]
$\boldsymbol{n}$	normal vector [m]
n	number of a quantity [-]
N	set of found particles between an emitting and absorbing object [-]
$\boldsymbol{q}$	heat flux [W/m²]
$\dot{Q}$	heat transfer rate [W]
r	radius [m]
r_g	rank of a quantity [-]
r_S	Spearman rank [-]
R^2	coefficient of determination [-]
S	surface distance [m]
$\boldsymbol{S}$	surface-to-surface vector [m]
t	time [s]
T	temperature [K]
$\boldsymbol{T}$	torque [N m]
$\boldsymbol{v}$	velocity [m/s]
V	volume [m³]
$\boldsymbol{w}$	weights of a neural network [-]
w_0	bias of a neural network [-]
x_U, x_O	undersize and oversize boundary [m]
$\boldsymbol{x}$	particle vector [m]
X, Y	populations
Y^*	Young's modulus [Pa]
$\tilde{z}$	normalized prediction of the neural network [-]
z	denormalized prediction of the neural network [-]

Greek symbols

β	angle [-]
$\in$	emissivity [-]
ε	view factor [-]
$\rho_{X,Y}$	Pearson correlation coefficient [-]
σ	standard deviation
σ_{SB}	Stefan-Boltzmann constant [W/(m² K⁴)]
ϕ	particle volume fraction [-]
$\boldsymbol{\psi}$	quantity
$\tilde{\boldsymbol{\psi}}$	normalized quantity
$\boldsymbol{\omega}$	angular velocity [1/s]

Sub-/superscripts

cond	conduction
cont	contact
corr	corrected
cyl	cylinder
dot	dot product
DEM	Discrete Element Method
exp	experimental
hidd	hidden
i, j	particle indices
$i-j$	between i and j
loc	local
max	maximum
min	minimum
n	normal
norm	normalized
orig	original
p	particle
pib	particle in between
pred	prediction
PP	particle-particle
PW	particle-wall
rad	radiation
ray	rays per unit area
rel	relative
R	reference
t	tangential
tot	total
vor	voronoi

the DEM solver LIGGGHTS®. Unfortunately, the high computational cost of Monte Carlo Methods limits the applicability to a small number of particles. Nonetheless these approaches showed that radiative heat transfer modelling based on view factors can be a very powerful method. They also initiated approaches to decrease the computational cost when calculating view factors.

Feng and Han [1] also discussed a computationally less expensive ray-tracing method. A similar approach was used by Maliounski and Rabinovich [30], and applied to particle-wall view factors. Forgber and Radl [31] introduced the projection method, that decreased the effort to calculate view factors between particles significantly compared to MCRT. They predict precise view factors, unfortunately the computational demand is still too big to adopt the method in general for heat radiation modelling in DEM simulations.

Simple correlations for view factors [20,32–35] consume a negligible amount of computation effort, but are limited to the specified setup and are hence of limited applicability. An interesting idea was proposed by Wu et al. [17], who used a segregated approach: 'short-range' radiation is modelled with view factors based on Voronoi cells, and 'long-range' radiation with ray-tracing. The study was limited to dense beds and particle-wall radiation was not investigated. The radiation distribution (RD) factor is another concept that was presented by Liu et al. [36]: here, a ray-tracing method was used to calculate the radiation distribution factor, which is basically a view factor in dilute particle systems. Particle-wall RD-factors were not analyzed.

One can conclude that an accurate view factor calculation is still computationally expensive, while fast methods are (i) either limited to very specified systems, or (ii) oversimplify the radiative effects. This is particularly true for view factors between particles and walls.

As mentioned at the beginning, separating radiative effects is difficult and so is separating between radiation from walls and radiation among particles. Due to that, particle-wall radiation is often modelled combined with other heat transfer related quantities like the effective thermal conductivity or (effective) heat transfer coefficients. Reviews of the usage of correlations in heat transfer have been presented by Van Antwerpen et al. [12] and Dìaz-Heras et al. [14]. A further review with explicit focus on the near-wall region was published by De Beer et al. [37]. They conclude that no method correctly accounts for the near wall region. In summary, radiative heat transfer between walls and particles is also often abandoned, or modelled in a simplified fashion [34,38]. Recently, Oschmann and Kruggel-Emden [19] presented a method to describe the conductive heat transfer between particles and walls

for the CFD-DEM approach. They applied their ideas to a rotating drum and a packed bed. Particle-wall radiation was not addressed, though. All these studies triggered our interest in more efficient radiation modeling for DEM-based simulations, with a special focus on radiation between particles and walls.

The so-far discussed methods reflect the pragmatism typically encountered to handle the tradeoff between accuracy and (simulation) speed. In summary, an accurate consideration of radiative heat transfer in numerical studies is computationally very expensive. To overcome this tradeoff, we use Machine Learning (ML) techniques in our present work to create a pre-trained heat radiation model. This model will be based on a Deep Neural Network (DNN), that predicts accurate view factors at low computational cost.

The usage of ML techniques in simulation science has already gained a lot of interest in recent publications. Kutz [39] reviewed the usage of DNNs in CFD, showing the broad usage, while Lee et al. [40] provided a general overview of Deep Learning in the process systems engineering field. In CFD-DEM simulations of gas-particle flows, DNNs have been used with the focus on drag modelling [41,42]. Also Direct Numerical Simulations (DNS) have been enhanced by data-driven approaches [43–45]. Sparse regression for turbulence closures have been shown by Beetham et al. [46]. With a focus on radiative heat transfer Wu et al. [47] presented a matrix-model based on a DNN. Based on the particle position between the interacting particles and the neighbor particle configurations, view factors are predicted. They validated the effective thermal conductivity with experimental data [48], showing reasonable agreement. Particle-wall interaction was not accounted for. In a preceding contribution Wu and Hao [49] also used a DNN based on geometric configurations that predicts view factors to analyze the wall effect of particle thermal radiation in a pebble bed. Wall-particle view factors were not directly calculated, instead radiation not absorbed by the other particles was automatically attributed to the wall. Kang et al. [50] used a DNN based on the surface emissivity of the particles, the particles' diameter and the bed height to predict the transmissivity in packed beds. The target values where derived via MRCT, finding reasonable agreement between the DNN-based prediction and the target values. However, no dedicated view factors were calculated, and also particle-wall radiation was not investigated.

1.1. Goals

The aim of our present contribution is to present a methodology that can predict accurate view factors with low computational cost for particle-particle and particle-wall configurations. Therefore, possible input quantities of the neural network, so-called markers, that are available in basic DEM simulations are systematically investigated. Also, the applicability and the relevance of non-trivial markers, e.g. Voronoi cell volumes, is explored. Subsequently a feature selection of the explored markers is performed, followed by the development of a Deep Learning Model for the presented regression problem. To integrate neural network-based predictions into the DEM solver LIGGGHTS®, available code (https://github.com/pplonski/keras2cpp/) is adjusted and modified. After the integration in the simulation framework, the DNN-based view factor prediction is validated with experimental data.

1.2. Content

This paper is organized as follows: We present the computational method in Section 2. The marker analysis and selection, as well as the identified Deep Neural Network parameters are discussed in Section 3. In Section 4, the merits of the DNN-based view factor prediction are assessed and the validation with experimental data is shown. Our main findings are summarized in Section 5.

2. Computational method

In our present study we will focus on a system of uniformly sized particles that are not moving. However, an accurate description of the particle movement is essential to generate realistic particle beds. These particle beds are then used to create the input and the target data of the DNNs. The used DEM solver is LIGGGHTS® [51]. The motion of spherical particles in translational and rotational direction is described by:

$$m_{p,i}\,\frac{d\boldsymbol{v}_{p,i}}{dt} = \sum_j \left(\boldsymbol{F}^n_{cont,i-j} + \boldsymbol{F}^t_{cont,i-j}\right) + m_{p,i}\,\boldsymbol{g} \tag{1}$$

$$I_i\,\frac{d\boldsymbol{\omega}_i}{dt} = \boldsymbol{T}_i \tag{2}$$

The mass of particle i is notated with $m_{p,i}$. $\boldsymbol{v}_{p,i}$ is the translational velocity of particle i. $\boldsymbol{F}^n_{cont,i-j}$ and $\boldsymbol{F}^t_{cont,i-j}$ are the normal and tangential contact forces. $\boldsymbol{T}_i$ is the total torque (due to tangential contact forces) acting on particle i.

2.1. Heat transfer calculation in DEM simulations

In the performed DEM simulations heat can be transferred via heat conduction between particles, and between particles and walls. The radiative heat transfer is also considered among particles and between particles and walls. The temperature update of each particle is then performed according to:

$$m_{p,i}\;c_p\frac{dT_{P,i}}{dt} = \sum_{j\,\in\,C_i}\dot{Q}_{cond,i-j} + \sum\dot{Q}_{rad,i-j} \tag{3}$$

C_i notates the set of contacts of particle i. c_p is the thermal capacity of the particle and $T_{P,i}$ is the individual particle temperature. The conductive heat transfer rate $\dot{Q}_{cond,i-j}$ between two particles, as well as between a particle and a wall, is calculated via:

$$\dot{Q}_{cond,i-j} = \frac{4\;k_i\;k_j}{k_i+k_j}\left(\frac{A_{cont,i-j}}{\pi}\right)^{1/2}\Delta T_{i-j} \tag{4}$$

Where k_i and k_j are the thermal conductivity of the respective body and ΔT_{i-j} describes the temperature difference. $A_{cont,i-j}$ labels the contact area between the interacting bodies.

A view factor and subsequently the radiative heat transfer rate $\dot{Q}_{rad,i-j}$ is calculated as follows:

$$\dot{Q}_{rad,\,i-j} = \frac{\sigma_{SB}\left(T_i^4 - T_j^4\right)}{\frac{1-\epsilon_i}{A_i\epsilon_i} + \frac{1-\epsilon_j}{A_j\epsilon_j} + \frac{1}{\varepsilon_{i-j}A_i}} \tag{5}$$

Here σ_{SB} is the Stefan-Boltzmann constant, ϵ the emissivity, A the surface area of the emitting and absorbing particle and ε_{i-j} the view factor.

The total radiative heat flux (in [W/m²]) is calculated by summing up the radiative fluxes per particle in a reference volume V_R (which equals the simulation domain):

$$\boldsymbol{q}_{rad} = \frac{1}{V_R}\sum_i\sum_j^{j\,\neq\,i}\dot{Q}_{rad,i-j}\boldsymbol{x}_{i-j} \tag{6}$$

where $\boldsymbol{x}_{i-j}$ is the vector between the center of body i and j. The same can be done to determine the total conductive heat flux:

$$\boldsymbol{q}_{cond} = \frac{1}{V_R}\sum_i\sum_j^{j\in C}\dot{Q}_{cond,i-j}\boldsymbol{x}_{i-j} \tag{7}$$

The total heat flux is then simply:

$$\boldsymbol{q}_{tot} = \boldsymbol{q}_{rad} + \boldsymbol{q}_{cond} \tag{8}$$

2.2. Monte Carlo ray-tracing

The view factors that represent the targets of the DNN are calculated with the so-called RayFactor-tool that is based on a Monte Carlo ray-tracing algorithm from Walker et al. [28]. RayFactor calculates view factors between geometrical objects via combination of so-called "primitives", which represent simple geometries like spheres, planes, cylinders and so on. In principle a defined (i.e. large) number of rays starts from random points with random angles on the object into the domain, where the first intersection with another object of each ray is accounted. The view factor is then the fraction of rays leaving the emitting object and intersecting with the absorbing object. We use RayFactor to determine view factors between spheres, which represent idealized particles, and between spheres and plane walls. Apparently important for the accuracy of the derived view factors is the number of started rays, which is defined via the ray density per unit area. In the Section "Targets" the influence of the ray density is discussed and the distribution for particle-particle as well as particle-wall view factors can be seen.

3. Deep Neural Network

The used artificial neural networks (ANN) are feedforward networks with multiple layers. Such a multilayer perceptron (MLP) is often notated as Deep Neural Network (DNN). These DNNs consist of an input layer, an arbitrary number of hidden layers, and an output layer. Each layer consists of an arbitrary number of nodes. Therefore, two important decisions need to be made when implementing a DNN: the number of hidden layers and the number of nodes for each of these layers.

Basically, a neural net with zero hidden layers can represent linear separable functions or decisions. However, without a hidden layer the neural net is not considered to be "deep" and cannot model nonlinear behavior. With one hidden layer a neural net can approximate any function that contains a continuous mapping from one finite space to another. Using two hidden layers enables the DNN to model an arbitrary decision boundary to an arbitrary accuracy [52]. With an increasing number of hidden layers, the complexity of training (and hence the computational effort) rises [52], therefore we focus on one hidden layer in our present study. Since the structure of the targeted view factors differs significantly between particle-particle and particle-wall view factors, two separate DNNs are necessary (see Section "Targets").

3.1. Marker definitions

A basic quantity that is used in correlations to describe view factors between particles is the surface to surface distance $|\boldsymbol{S}_{i-j}|$ [31,34]. Since particles in DEM simulations can be overlapping, the norm of the center to center vector $\boldsymbol{x}_{i-j}$ is used (see Fig. 1). The solid angle between the interacting objects is an important quantity in the analytical integration of view factors [1]. For that reason, the solid angle will be investigated as marker in a simplified way by using the angle between the center vectors β_{i-j}. However, the definition of the solid angle depends on the origin of the coordinate system and hence this marker is only helpful if the coordinate system is fixed.

Representative for the local particle volume fraction the volume of Voronoi cells achieved from Voronoi tessellation by Rycroft [53] is also considered as a marker. This is motivated by the fact that view factor correlations based on Voronoi cells have been investigated previously [17]. We use the Voronoi volume relative to the emitting particle's Voronoi volume to create non-constant training data from a single particle bed. Otherwise, the Voronoi cell data would reflect a constant input marker connected to the target.

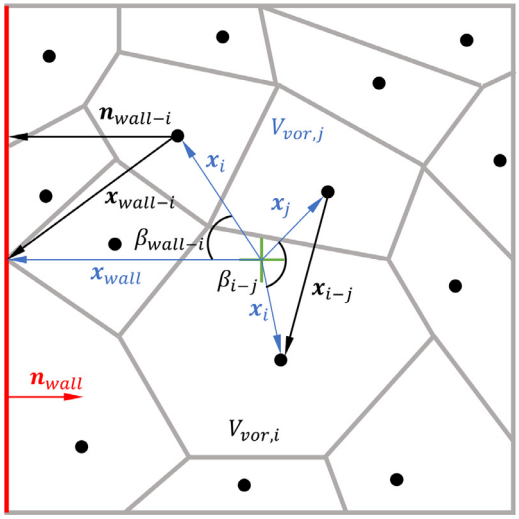

Fig. 1. Schematic 2D-illustration of the calculated markers (black) and the necessary parameters (blue). The black dots mark the centers of particles, and the grey lines are the edges of the Voronoi cell. Exemplarily, a wall is shown in red on the left side, with the corresponding unit normal vector. The origin of the coordinate system is shown as a green cross. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Shadows between particles significantly influence radiative heat transfer. Therefore, an algorithm that detects particles that are between the emitting and the absorbing particle was implemented as well. This was done by considering a hypothetical cylinder with radius r_{cyl} located with its axis on the line connecting the two particles. Specifically, if every of the following three equation is true, a particle with center vector $\boldsymbol{x}$ is within this hypothetical cylinder whose top and bottom surface are located at the emitting and absorbing particle:

$$(\boldsymbol{x}-\boldsymbol{x}_i)\cdot\left(\boldsymbol{x}_j-\boldsymbol{x}_i\right)>0 \tag{9}$$

$$\left(\boldsymbol{x}-\boldsymbol{x}_j\right)\cdot\left(\boldsymbol{x}_j-\boldsymbol{x}_i\right)<0 \tag{10}$$

$$\frac{\left|(\boldsymbol{x}-\boldsymbol{x}_i)\times(\boldsymbol{x}_j-\boldsymbol{x}_i)\right|}{\left|\boldsymbol{x}_j-\boldsymbol{x}_i\right|}\leq r_{cyl} \tag{11}$$

As r_{cyl} the particle radius was chosen, also to keep the additional computational effort low. The identified set of particles for each absorbing particle is notated as N. The identified particles are summed up to calculate the number $n_{pib,i-j}$ of particles between the emitting and the absorbing particle.

As an additional marker to consider the local particle volume fraction, the ratio of the particle volume and the Voronoi cell volume (of the particles in set N) are summed up, and a "local" particle volume fraction $\phi_{loc,i-j}$ is calculated (see Table 1 for details). Note that this "local" particle volume fraction is not bounded by unity but expresses the product of the mean particle volume fraction (in the hypothetical cylinder) and the number of particles between the emitting and the absorbing particle.

The principles of how to calculate $n_{pib,i-j}$, $|\boldsymbol{x}_{i-j}|$, and β_{i-j} for particle-particle interactions can be directly applied to particle-wall interactions, yielding the markers $n_{pib,wall-i}$, $|\boldsymbol{x}_{wall-i}|$, and β_{wall-i}. Referring to Eqs. (9) to (11) for particle-wall interaction, the cylinder is spanned from the center of the wall to the respective particles. However, radiation from the wall to the particle bed typically evolves in layers, so the norm of the normal distance vector

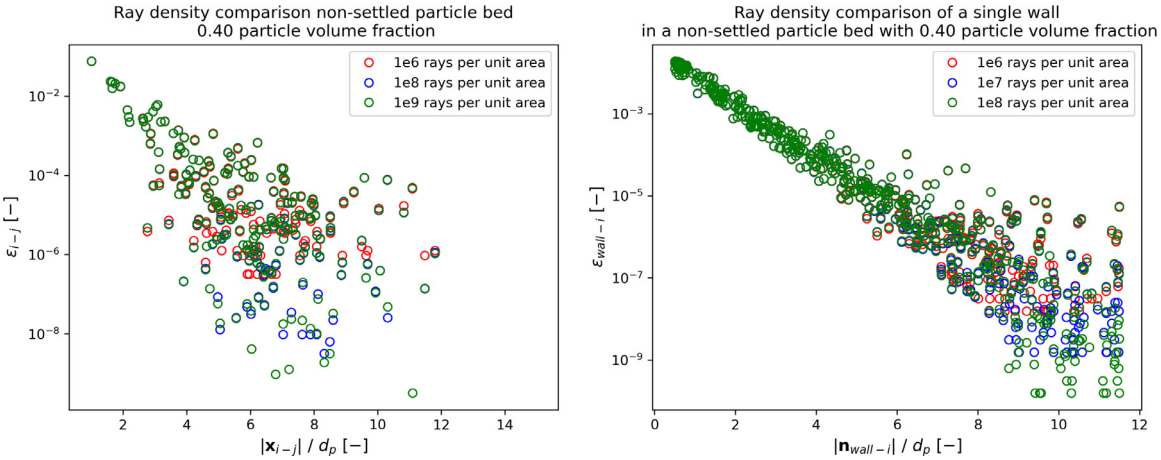

Fig. 2. View factors for different number of rays per unit area in non-settled particle beds. Left panel: for a particle bed, right panel: for particle-wall radiation.

Table 1
Overview of the investigated markers for particle-particle and particle-wall interaction and their definition.

Marker Particle-Particle	Definition
$\lvert\boldsymbol{x}_{i-j}\rvert$	$\sqrt{(x_i-x_j)^2+(y_i-y_j)^2+(z_i-z_j)^2}$
β_{i-j}	$\arccos(\frac{\boldsymbol{x}_i\cdot\boldsymbol{x}_j}{\lvert\boldsymbol{x}_i\rvert\ \lvert\boldsymbol{x}_j\rvert})$
$n_{pib,i-j}$	based on Eqs. (9) to (11)
$V_{vor,i-j}$	$\frac{V_{vor,j}}{V_{vor,i}}$
$\phi_{loc,i-j}$	$\frac{V_{p,i}}{V_{vor,i}}+\sum_{n\in N}\frac{V_{p,n}}{V_{vor,n}}$
$x_{dot,i-j}$	$\boldsymbol{x}_i\cdot\boldsymbol{x}_j$
Marker Particle-Wall	
$\lvert\boldsymbol{x}_{wall-i}\rvert$	$\sqrt{(x_{wall}-x_i)^2+(y_{wall}-y_i)^2+(z_{wall}-z_i)^2}$
$\lvert\boldsymbol{n}_{wall-i}\rvert$	$\lvert\boldsymbol{x}_{wall-i}\cdot\boldsymbol{n}_{wall}\rvert$
β_{wall-i}	$\arccos(\frac{\boldsymbol{x}_{wall}\cdot\boldsymbol{x}_i}{\lvert\boldsymbol{x}_{wall}\rvert\ \lvert\boldsymbol{x}_i\rvert})$
$n_{pib,wall-i}$	based on Eqs. (9) to (11)
$V_{vor,i}$	Rycroft [53]

pointing from the wall to the particle $|\boldsymbol{n}_{wall-i}|$ is explored additionally. The Voronoi cell volumes are directly used to account for particle-wall interactions, since different bed configurations lead to different cell structures.

Table 1 summarizes all possible markers and their definitions. Also, the inverse values of $|\boldsymbol{x}_{i-j}|$, $|\boldsymbol{x}_{wall-i}|$ and $|\boldsymbol{n}_{wall-i}|$ are investigated as potential markers. From the definitions in Table 1, one can see that $x_{dot,i-j}$, β_{wall-i}, and β_{i-j} involve the absolute coordinates and hence are not independent of the coordinate origin. Also, $|\boldsymbol{x}_{wall-i}|$ involves the center point of the wall, which can be defined arbitrarily. In the following Sections the coordinate origin was located at the center of the simulation domain and the point at the wall was located at the wall's center.

In the presented approach we focus on monodisperse systems. In order to extend our present results to polydisperse systems additional markers, e.g., the ratio of the radii could be used. However, this is not explored further in our present study.

3.2. Targets

As previously described the ray density per unit area n_{ray} is decisive for the view factor determination. The maximum number of rays per unit area used by Walker et al. [28] was 1e8. One has to note the maximum number of rays that can be considered with RayFactor is 1e9.

Fig. 2 schematically shows the view factor distribution for particle-particle and particle-wall interaction, as well as view factors for different number of rays per unit area. The mean squared error (*MSE*) between the calculate view factors for particle-particle interaction when using n_{ray} equal to 1e6 and 1e9 is 2.428e-10. For particle-wall interaction the corresponding *MSE* when using n_{ray} equal to 1e6 and 1e8 is 2.191e-11. Clearly, both values are very small compared to the found view factor values.

As can be seen qualitatively in Fig. 2, the number of non-zero view factors found is increasing significantly with an increasing number of rays. Quantitively the number of detected view factors for the particle-particle case are 177, 203, and 207 of 586 possible interactions (for each particle) for ray densities of 1e6, 1e8, and 1e9, respectively. The respective number of detected particle-wall view factors for ray densities of 1e6, 1e7, and 1e8 are 492, 530, and 547 of 587 possible interactions.

If we consider a threshold value of 5e-6 as done by Forgber and Radl [31], the tradeoff between accuracy and increasing computational cost becomes worse with increasing number of rays. Therefore, the ray density used to create the non-settled target datasets was 1e6 in both scenarios. For the detailed analysis of a settled particle bed case and comparison to the shown non-settled case the interested reader is referred to Appendix A.1.

3.3. Marker selection

Two different parameters are used to describe the quality of correlation between the marker and the target. The first one is the Pearson correlation coefficient $\rho_{X,Y}$ that describes the linear relation between two populations X and Y by:

$$\rho_{X,Y}=\frac{Cov(X,Y)}{\sigma_X\ \sigma_Y} \tag{12}$$

σ_X and σ_Y are the standard deviations of the respective population. The second parameter is the Spearman rank correlation coefficient r_S that is defined via:

$$r_S=\frac{Cov\left(r_{g,X},r_{g,Y}\right)}{\sigma_{r_{g,X}}\ \sigma_{r_{g,Y}}} \tag{13}$$

Here $r_{g,X}$ and $r_{g,Y}$ are the ranks of a quantity, where the smallest value has rank 1. $\sigma_{r_{g,X}}$ and $\sigma_{r_{g,Y}}$ are the standard deviation of the respective ranking. The above marker selection equations were evaluated using the Pandas® framework in Python®. Fig. 3 shows the so-called heatmap, which ranks the Pearson correlation between the markers and the target, as well as the correlation among each other. The left panel shows the heatmap for particle-particle, and the right panel for particle-wall interactions. Histograms of each investigated quantity can be found in Appendix A.2.

Table 2 summarizes all calculated correlation coefficients. As can be seen, using the inverse distance norms leads to very high

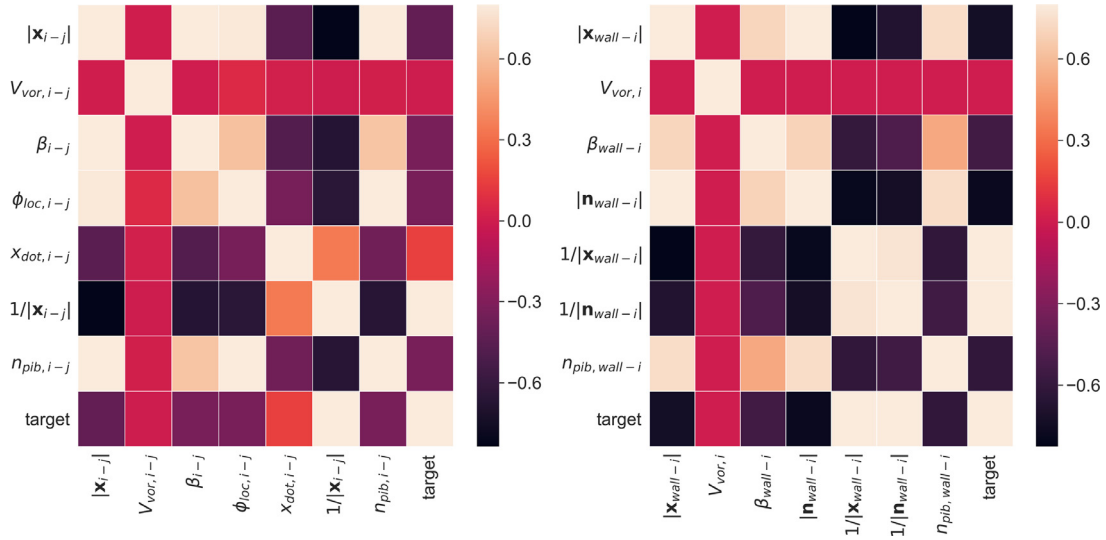

Fig. 3. Correlation heatmaps based on the Pearson coefficient. Left side: particle-particle interaction, right side: particle-wall.

Table 2
Found Pearson and Spearman coefficients.

Particle-Particle	Pearson coeff.	Spearman rank coeff.
$1/\lvert\boldsymbol{x}_{i-j}\rvert$	0.8038	0.7937
$x_{dot,i-j}$	0.1485	0.3489
$V_{vor,i-j}$	-0.0088	-0.0049
$\phi_{loc,i-j}$	-0.3336	-0.7462
β_{i-j}	-0.3349	-0.6572
$n_{pib,i-j}$	-0.3358	-0.7635
$\lvert\boldsymbol{x}_{i-j}\rvert$	-0.4174	-0.7937
Particle-Wall		
$1/\lvert\boldsymbol{n}_{wall-i}\rvert$	0.9464	0.9640
$1/\lvert\boldsymbol{x}_{wall-i}\rvert$	0.8389	0.9565
$V_{vor,i}$	-1.2e-5	-1.4e-4
β_{wall-i}	-0.5402	-0.6799
$n_{pib,wall-i}$	-0.6163	-0.8030
$\lvert\boldsymbol{x}_{wall-i}\rvert$	-0.7426	-0.9565
$\lvert\boldsymbol{n}_{wall-i}\rvert$	-0.7767	-0.9640

correlations for particle and particle-wall interaction. The benefit of inverse distances is to link high input values with high target values (see Figs. 5 and 7). As expected, the normal distance yields a better correlation for the description of particle-wall view factors.

Based on the found coefficients and consideration of the necessary computational time of each feature, the following marker were chosen: $1/|\boldsymbol{x}_{i-j}|$ and $n_{pib,i-j}$ for the particle-particle DNN and $1/|\boldsymbol{n}_{wall-i}|$ and $n_{pib,wall-i}$ for the particle-wall DNN. It is interesting to note that $\phi_{loc,i-j}$ gives a slightly weaker correlation compared to $n_{pib,i-j}$, despite the fact that former contains additional information (i.e., the mean particle volume fraction).

β_{i-j} and β_{wall-i} are dismissed since they are not independent of the coordinate origin (and hence would be only useful in the same coordinate system), and anyhow would yield only mediocre correlations. Since the Voronoi cell information shows no significant correlation and the Voronoi cell calculation is associated with a significant computational expense, $n_{pib,wall-i}$ is preferred over $V_{vor,i}$.

Additionally, the possibility of calculating $n_{pib,wall-i}$ based on the normal distance vector $\boldsymbol{n}_{wall-i}$ instead of the center distance vector $\boldsymbol{x}_{wall-i}$ was investigated. The Pearson and Spearman coefficients in that scenario are then -0.4284 and -0.3910, both values reflect a weaker correlation to the target. One has to note that the investigated particle clouds are very homogeneously distributed towards the wall. In other situations with heterogeneously distributed particle clouds the usage of $\boldsymbol{n}_{wall-i}$ over $\boldsymbol{x}_{wall-i}$ might be advantageous.

3.4. Neural network training and testing

The investigated non-settled particle bed contained 587 particles with a total particle volume fraction of 0.40. The total dataset therefore consists of 586∗587 view factors values. The total dataset was divided into a training dataset, a test dataset, and a validation dataset with the respective relative amounts of 0.8, 0.1, and 0.1. For particle-wall interaction, the top and the bottom walls of the particle bed were used for the creation of the particle-wall dataset. 396 different particle beds based on random seeds were generated to increase the size of the particle-wall data set. The total particle-wall dataset then consists of 792∗587 view factor values. The split ratios (for training, test, and validation) were the same as for particle-particle interactions.

The creation and the training of all DNNs was performed within the Keras® framework in Python®. The preparation of the raw data and the randomized search optimization were performed using the Scikit-learn® environment. One has to note that very small view factors lead to negligibly small heat fluxes and are set to zero if they are below a threshold value of 5e-6. The thresholding is applied to the training data and also to the prediction of the DNN. To increase the overall efficiency of the DNN training process, mini-batch gradient descent is used as variation of the gradient descent algorithm [54]. The mini-batch size is then an additional hyperparameter of the DNN. However, the number of the particles that can absorb radiation from the emitting object in the bed will be chosen as mini-batch size. Thus, the mini-batch size is not part of the randomized search approach.

Feature scaling (or normalization) of the input markers is typically performed before being fed into a neural net [52,55]. The normalization process of a quantity $\boldsymbol{\psi}$ follows:

$$\tilde{\boldsymbol{\psi}} = \left(\boldsymbol{\psi} - \psi_{min}\right)/\left(\psi_{max} - \psi_{min}\right) \tag{14}$$

where $\tilde{\boldsymbol{\psi}}$ reflects a normalized input marker. If one considers a simple single non-linear neuron the outcome $\tilde{z}$ is defined by:

$$\tilde{z} = h\left(\boldsymbol{w}^T\ \tilde{\boldsymbol{\psi}} + w_0\right) \tag{15}$$

$\boldsymbol{w}$ is the vector that contains the weights, w_0 is the bias, and h is the applied activation function. Consequently, if the input is nor-

Table 3
Investigated hyperparameters of the DNN.

Hyperparameter	Searched options	Found optimum
n_{hidd}	25, 50, 75, 100, 125, 150, 175, 200	175
Init weights	"he uniform" [60], "he normal" [60], "zeros"	"he uniform"
Optimizer	Adam [61], SGD, Adagrad [62], RMSprop [63]	Adam
Learning rate	1e-3, 8e-4, 5e-4, 3e-4, 1e-4	8e-4

malized, the outcome is also normalized. To get the true prediction z that reflects the actual view factors, the prediction needs to be denormalized with:

$$z = \tilde{z}(\varepsilon_{\max} - \varepsilon_{\min}) + \varepsilon_{\min} \tag{16}$$

Before normalization the available data is separated into training-, validation- and testing-data set. The necessary respective min- and max-values are then searched in the training data and used to normalize (and denormalize) the validation and the testing data to avoid data leakage [56]. If a pretrained DNN is used for prediction, the input data is normalized with min-and max-values based on the new input data. The prediction is then denormalized by using the analytical maximum view factor as ε_{max} and 0 as ε_{min}.

3.5. Neural network design and hyperparameters

The input layer is defined by the features of the training data, i.e., the selected markers. Consequently, the number of nodes for the input layer is the number of selected markers of your training data. The output layer is then defined by the desired output (i.e. the single target view factor) and the number of nodes. The choice of the activation layers and the used optimizer are also considered as neural network model design parameters. Beside the standard Stochastic Gradient Descent (SGD) optimizer, different advanced options are examined (see Table 3). In regression tasks typically no activation function is applied to the output layer [52]. The hidden layer (characterized by the number of nodes n_{hidd}) is typically followed by the Rectifying Linear Unit (ReLU) activation function in regression tasks [57].

To avoid overfitting, the early stopping method is used with a patience parameter of zero [58]. The learning rate is a so-called hyperparameter of the neural network. Furthermore, the choice of the weight distribution at initialization is also a hyperparameter, and is therefore investigated. The model design and the hyperparameters form the neural net model parameters. To achieve the best possible generalization with the DNN-based model, it is essential to optimize these model parameters. The optimization is performed by the randomized search approach [59], and Table 3 summarizes all searched possibilities. To evaluate the amount of training data and the found parameters, we also performed the randomized search approach with changed amounts of training, validation and testing data. Additionally, we trained neural networks based on the found optimum in Table 3 with different amount of training data. In summary, the initially stated data split appears to be sufficient and the found optimum is reasonable. A detailed description is shown in Appendix A.3.

The choice of the cost function is one of the key tasks when creating a DNN. In the performed approach the Mean Squared Error (*MSE*) is chosen, since zeros are part of the input and as the target of the DNN no logarithmic or relative error measure can be taken. Therefore, the *MSE* is also a key metric when evaluating the quality of the DNN-based prediction. The *MSE* is obtained by the following relation:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(z_{pred} - z_{target})^2 \tag{17}$$

Because the *MSE* is significantly more influenced by large view factors and does not reflect the overall quality of the prediction, the coefficient of determination (R^2) is used to describe this correlation between prediction and target:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(z_{target} - z_{pred})^2}{\sum_{i=1}^{n}(z_{target} - \bar{z}_{target})^2} \tag{18}$$

Another quantitative measure is the coefficient of determination (${R_{spread}}^2$) for particle-particle (or particle-wall normal) distances within a certain range (defined by the lower boundary x_U and the upper boundary x_O):

$$R^2_{spread}(x_U, x_O) = 1 - \frac{\sum_{i=1}^{n}\left(z_{target}\left(\left|\boldsymbol{x}_{i-j}\right|\right) - z_{pred}\left(\left|\boldsymbol{x}_{i-j}\right|\right)\right)^2}{\sum_{i=1}^{n}\left(z_{target}\left(\left|\boldsymbol{x}_{i-j}\right|\right) - \bar{z}_{target}\left(\left|\boldsymbol{x}_{i-j}\right|\right)\right)^2} \quad for\, x_U \le \left|\boldsymbol{x}_{i-j}\right| \le x_O \tag{19}$$

In the following Sections x_U equals half of the domain size minus 1.5 d_p, while x_O is half of the domain size plus 1.5 d_p. Fig. 4 schematically illustrates the final structure and the input and output markers of the created Deep Neural Networks.

4. Results and discussion

A single test dataset contains the view factors of one emitting particle or wall, which is also the mini-batch size as described in Section 3.4. The total test dataset contains several single test datasets.

4.1. Effect of the particle bed configuration

As shown the Monte Carlo-based view factors determined with RayFactor [28] are dependent to some degree on the ray density and are also varying over the number of calculations. The view factor distribution is influenced by the particle volume fraction (see Appendix A.1). However, the influence of the ray density on the view factors is smaller in dilute particle beds than in dense particle beds (for details see Appendix A.1). To quantify the influence of the particle volume fraction on the DNN-prediction, the derived particle-particle DNN was also tested with different particle volume fraction in the same simulation domain and with a different simulation setup (the detailed predictions and results can be seen in Appendix B).

Applying the DNN to a dilute system with 0.10 particle volume fraction, yields an *MSE* of 1.238e-5, R^2 is 0.7352 and R^2_{spread} of 0.2518. All measures are significantly worse than the trained case with 0.40 particle volume fraction. When slightly increasing the particle volume fraction to 0.20 the quality of the prediction increases compared to the dilute case. The *MSE* is then 3.194e-6, R^2 is 0.9327 and R^2_{spread} is 0.5312. However, the prediction quality of the basic training case is not reached. If we increase the particle volume fraction and additionally change the particle and the domain size, the *MSE* is 2.728e-6, R^2 is 0.9765, and R^2_{spread} is 0.6078. In this scenario the change in volume fraction to the trained system is smaller than in the other scenarios. For this reason, the quality of the prediction is the closest to the actual trained system.

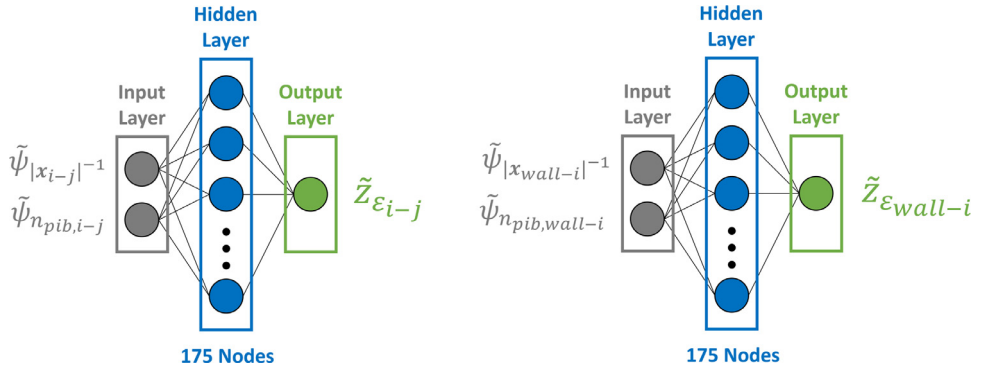

Fig. 4. Schematic illustration of the Deep Neural Network for particle-particle interaction on the left side and for particle-wall interaction on the right side.

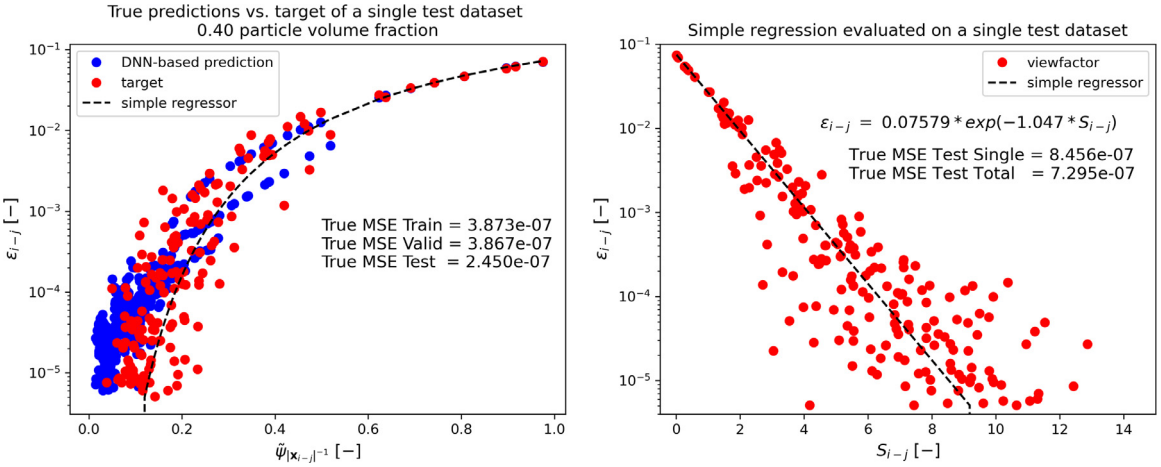

Fig. 5. The comparison of the prediction of the DNN-based model (left panel; versus the normalized input marker) and of the simple regression model (right panel) for particle-particle view factors.

One can conclude that if the particle volume fraction significantly differs from the trained configuration, i.e. much higher or smaller particle volume fraction, a separate DNN and consequently a changed training dataset is necessary. Whereas the domain or particle size does not harm the prediction of the DNN in principle, since the input markers are normalized. Thus, a more advanced DNN could incorporate the domain-average particle volume fraction as an additional marker, but would cause additional complexity when applied (i.e., heterogeneities in the particle volume fraction, computational expense).

4.1.1. Particle-particle interaction

Fig. 5 shows the predicted view factors of a single test dataset of the trained DNN in comparison to a simple regressor (see right panel in this Figure for the simple expression used). The simple regressor is based on the relative surface-to-surface distance S_{i-j} that is defined as $|\boldsymbol{S}_{i-j}|/r_i$. As can be seen the *MSE* evaluated on the total test dataset is very small for the DNN-based model and the simple regression model. This is caused by the fact that the *MSE* is significantly more influenced by big values than by small values. On the right side in Fig. 5 it can be seen that the big view factors are predicted very accurately by the simple regression. In contrast, for the rest of the widespread view factors, there is only one prediction for each distance possible with the simple regressor. However, the *MSE* of the total test dataset of the DNN-based prediction is significantly smaller.

Fig. 6 displays the additional parameters that were introduced to classify the quality of the prediction. The total coefficient of determination of the simple regressor is 0.9829, which is significantly smaller than the 0.9934 of the DNN-based prediction. For the calculation of R^2_{spread} the boundaries are $x_U = 2.5$ [m] and $x_O = 5.5$ [m]. The value of R^2_{spread} of the simple regressor is 0.03614, while DNN-based it is 0.7009 and therefore significantly higher.

4.1.2. Particle-wall interaction

As previously mentioned, a separate DNN to predict particle-wall or wall-particle view factors is necessary. Similar to the simple regressor for particle-particle view factors, the particle-wall regressor is derived via the relative normal surface distance S_{wall-i}, which is determined by $(|\boldsymbol{n}_{wall-i}| - r_j)/r_j$. Fig. 7 shows the predicted view factors of the DNN-based model on the left side and the prediction of the simple regression model on the right side. Compared to Fig. 5 more data points can be seen in Fig. 7, since more non-zero view factors were determined via Monte Carlo ray-tracing in particle-wall cases than in particle-particle cases. The *MSE* on the total test dataset could be reduced by an order of magnitude from 2.111e-7 to 5.911e-8 when using the DNN compared to a simple regressor.

In Fig. 8 the additional parameters R^2 and R^2_{spread} can be seen for the predicted wall-particle view factors. The total coefficient of determination of the simple regression on the shown single test dataset is 0.8248, which is significantly smaller than the 0.9506 of the DNN-based prediction. The value of R^2_{spread} of the simple regressor is 0.03065 (i.e., an extremely low correlation), while for the DNN-based model it is 0.6003 and therefore significantly higher.

R^2 when describing the particle-wall interaction is significantly smaller than in particle-particle interaction. This is led by the fact that the particle-particle view factor distribution is more beneficial

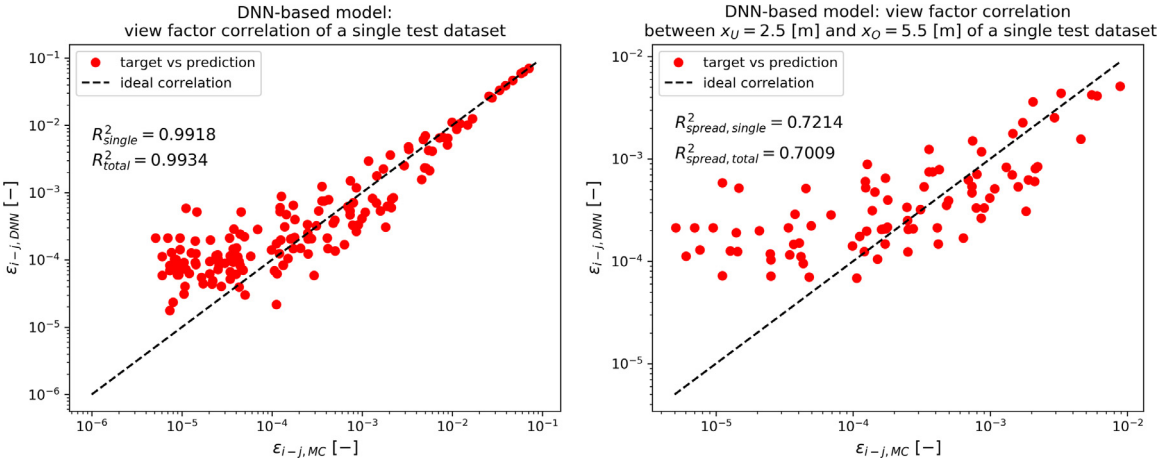

Fig. 6. The correlation of the predicted particle-particle view factors and the target values. On the left side all predicted view factors are correlated, while on the right side only view factors within x_U and x_O are considered.

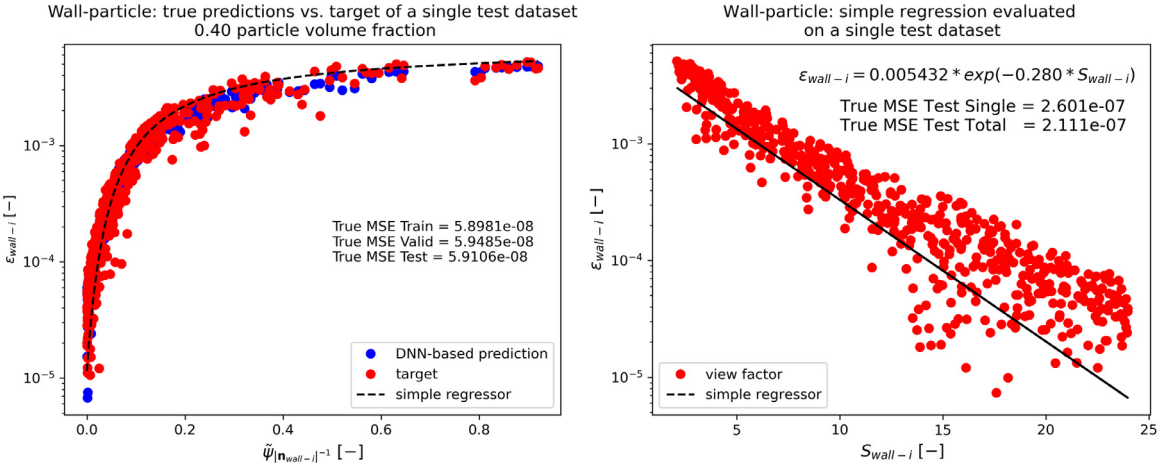

Fig. 7. The comparison of the prediction of the DNN-based model (left panel; versus the normalized input marker) and of the simple regression model (right panel) for wall-particle view factors.

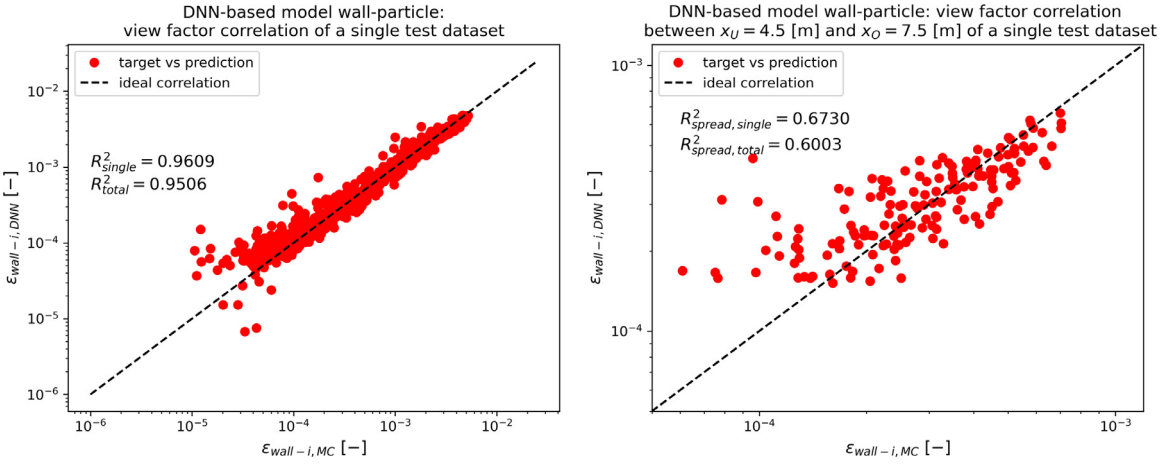

Fig. 8. The correlation of the predicted wall-particle view factors and the target values. On the left side all predicted view factors are correlated, while on the right side only view factors within x_U and x_O are considered.

for training of the DNN or regression in general. View factors that occur at small distances, e.g. values of larger than 0.6 for the inverse normalized distance, which significantly influence R^2, show more variability if we compare the left panels of Figs. 5 and 7. Quantitively the number of these high view factors, compared to the mean value, that closely located is 8 in the shown particle-particle single test dataset, while for the shown single particle-wall test data set it is 15. The variance of these particle-particle view factors is 0.04261 and the variance of the particle-wall view factors is 7.719e-3. The smaller number of targets and the higher variance

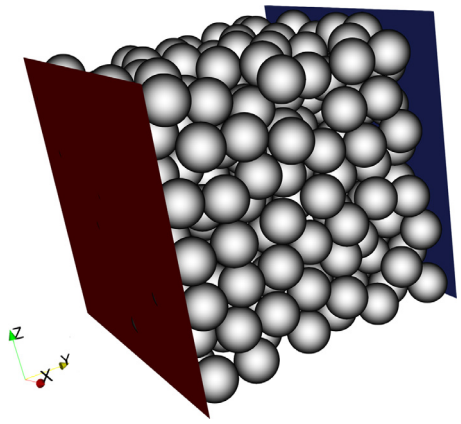

Fig. 9. The created particle bed, that is used for validation and a schematic representation of the investigated setup. The (left) red wall represents the hot inner reflector, while the (right) blue wall embodies the cold outer reflector. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

are both advantageous in favor of regression of particle-particle interaction.

4.2. Validation with experimental data

Experimental investigation of radiative heat transfer related to particles is a complex and therefore rarely performed task. However, De Beer [24] investigated radiative heat transfer in particle beds in a Near-Wall Thermal Conductivity Test Facility (NWTCTF). They considered experiments with various heating wall temperatures. In our presented study the results of the testcases with 400 and 800 °C are used, since they are the lowest and highest explored temperatures by De Beer [24].

4.2.1. Experimental and DEM simulation setup

Fig. 9 schematically shows the DEM simulation setup. The graphite radiator of the NWTCTF is equal to the hot inner reflector wall shown in red in Fig. 9, and the graphite reflector equals the cold outer reflector wall in blue. The measured temperatures that were used to determine the effective thermal conductivity in [16,24] were located very close to the so-called “center layer”, where the highest particle temperatures can be expected. If one would investigate all individual particle temperatures in such an experimental case (where the other surrounding wall have a significantly lower temperature than the heating wall), the particle temperature normal to the main heat flux direction would have a significant spread.

De Beer et al. [16] based on the earlier work of De Beer [24] presented a methodology to separate the effects of conduction and radiation based on CFD simulations. They included the insulated side walls in their CFD simulations with a decreased emissivity of 0.68. Heat transfer in the CFD simulations was calculated based on Surface-to-Surface Radiation and a “Segregated Solid Energy model” [16]. In Surface-to-Surface Radiation models the exchanged radiative heat is also based on view factors, but between individual (flat) faces on the computational surface mesh. Segregated Solid Energy models do not couple the solution of the energy equation to the solution of the governing equation of the fluid. However, they also used the hottest particle layer for their analysis of the CFD simulations. Therefore, only the inner reflector wall and the outer reflector wall are considered in the DEM simulation. In the other directions (i.e. z and x) the boundaries were considered periodic in our present study. This will result in a particle layer that contains more particles than in the experiment, where the temperature spread normal to the intended heat flux direction will be rather small (i.e. see the right panel of Fig. 11, where the mean difference along the fitted curve between the max and the minimum is 30.23 [K]).

Table 4
Simulation parameters.

Parameter	Value
Particle diameter	0.06 [m]
Particle density (Graphite)	2100 [kg/m³]
Number of particles	332
Simulation domain	0.42 [m] x 0.42 [m] x 0.42 [m]
Young's modulus bed generation	4.8e6 [Pa]
Poisson ratio	0.2
$T_{P,i,start}$	293.2 [K]
Thermal capacity graphite at 20 °C	710 [J/(kg K)]
Emissivity particles and reflector walls	0.8
Δt_{DEM} for bed generation	1e-5 [s]
Δt_{DEM} for heat transfer simulation	1 [s]

De Beer [24] defined the total heat transfer rate that is entering the system as the sum of the total heat losses and the heat flux that was necessary to keep the temperature of the outer radiator wall constant. Since no heat loss through side walls appears in the DEM simulations, the experimentally measured heat transfer rate $\dot{\boldsymbol{Q}}_{out,exp}$ will be used to calibrate the area correction in our DEM simulation (see Section 4.2.4). The total experimental heat flux of the system can be calculated with:

$$\boldsymbol{q}_{tot,exp} = \frac{\dot{\boldsymbol{Q}}_{out,exp}}{A_{out}} \tag{20}$$

The parameters of the simulation can be found in Table 4. The temperature dependence of the thermal conductivity k of the graphite particles and the reflector walls is considered via the correlation provided by De Beer [24]:

$$\begin{aligned} k(T) = \; & 1.595\text{e}-14\,T^5 - 7.255-11\,T^4 + 6.566\text{e}-8\,T^3 \\ & + 1.403\text{e}-4\,T^2 - 0.3179\;T + 235.01 \end{aligned} \tag{21}$$

The thermal capacity is also a temperature-dependent material property. However, the effect of a temperature dependent thermal capacity is irrelevant for the steady-state simulation result we focus on. Therefore, the thermal capacity is treated as constant. Since the particles were arrested in the performed simulation, a timestep Δt_{DEM} of 1 [s] was identified as suitable for the thermal DEM simulation. The simulations were performed until the total thermal energy of the particles and the total heat flux reached a steady-state. Since the particles are not moving, the view factors are only calculated in the initial time step and are not updated.

The heat transfer simulations typically took about 7e4 steps to reach the steady-state and the necessary computation time was then approximately 671 [s] on a single core. A single core was used since multithreading distributes particle information to different cores, which denies to calculate a view factor between particles on different threads. Consequently, long-range radiation effects would have not been accounted for in such a scenario. Of course, neighbor lists can be built over different threads, but then the influence of the neighbor list range must be treated as additional fitting parameter of the simulation. Too large search ranges result then in the same situation as using a single thread.

4.2.1.1. DEM particle bed generation and area correction. Due to hydrostatic pressure, rough surfaces, and a combination of random placement and particle insertion by hand in the experimental case [16], an exact reproduction of the particle contact areas is unlikely in our DEM. Unfortunately, the contact area between two bodies i and j is decisive for the conducted heat $\dot{Q}_{cond,i-j}$ (see Eq. (4)). For

the used particle bed in the performed DEM simulation 332 random positioned particles are inserted and then settled (see Fig. 9) to approximate the experimental conditions. The bed is generated with a comparably low value of the Young's Modulus Y^* that is typically used in DEM simulations (see Table 4). Since this leads to larger contact areas, the contact area is corrected by adjusting an original Young's Modulus Y^*_{orig} that is typical for graphite materials. The contact area that is then used to calculate the conducted heat is corrected with:

$$A_{cont,i-j,corr} = A_{cont,i-j} \frac{Y^*}{Y^*_{orig}} \tag{22}$$

However, using the Young modulus of graphite gave unsatisfactory results due to other influence factors (e.g., surface roughness), making a calibration of this parameter necessary. To calibrate the area correction, the experiment with 400 °C is used, since the heat radiation effects are the smallest of all performed experiments. One has to note that the experimental measured heat fluxes include radiation effects also at this comparably low temperature. Thus, radiation effects were considered in the calibration simulations using our DEM-based model as well.

We note in passing that De Beer et al. [16] used a fixed contact area among the particles and between particles and walls in their CFD simulations. This was realized using a fillet between each object with a radius of 0.2 [%] of the particle diameter. This results in a contact area with a radius of 1.2e-4 [m] and consequently a circular area of 4.525e-8 [m²] per contact.

4.2.2. View factor calculation validation case

The experimental particle bed is very dense and settled. This leads to much higher particle-wall view factors than in situations where particles are not in contact with the wall. In addition, in particle beds that are bounded by walls layering occurs. Therefore, a dataset for the particle-wall view factor was created based on the specified simulation set-up. Based on 396 random created particle beds the total particle-wall dataset size was 792∗332. Since the increased particle volume fraction impacts the DNN-based view factor prediction, as shown in Appendix B, a dedicated dataset for particle-particle view factors was also created for the experimental validation. The size of the dataset was increased by considering different particle beds, resulting into 12∗332∗331 marker-target pairs for the particle-particle view factors. The split ratios for training, testing, and validation were the same as described Section 3.4.

As previously mentioned, the principles of calculating view factors via Monte Carlo simulations lead to fluctuations in the view factor values. The effect of these fluctuations increases when the particles volume faction increases and particles start to overlap (see Appendix A.1). Since real particles are not overlapping, the analytical maximum view factor is used to denormalize (see Eq.(16)) the DNN-predicted view factor. Therefore, the target value is also normalized with the analytical maximum (see Eq. (14)).

According to [64] the analytical view factor for two contacting particles of equal size is 0.07180. The respective analytical view factor between a square wall with a length of 7 times the particle diameter and a contacting particle is 0.02797 (from the particle to the wall it is 0.4362) [65]. Fig. 10 shows the particle-particle and particle-wall view factors of the actual simulation set-up that is used in the validation. The left panel shows the prediction for particle-particle view factors, while the right panel shows the view factors between particles and the walls.

R^2 of the total test dataset for particle-particle interaction is 0.9898 and for particle-wall it is 0.9570. R^2_{spread} between $x_U = 0.12$ [m] and $x_O = 0.30$ [m] for particle-particle is then 0.6205. R^2_{spread} for particle-wall is 0.4272.

Table 5
Overview of experimental measurements of the 400 °C case.

Parameter	Value
$T_{in,reflector}$	673.2 [K]
$T_{out,reflector}$	332.5 [K]
$k_{in,reflector}$	91.92 [W/(m K)]
$k_{out,reflector}$	146.4 [W/(m K)]
$\dot{Q}_{out,exp}$	287.4 [W]
$\boldsymbol{q}_{tot,exp}$	1629 [W/m²]

4.2.3. Simulations with 400 °C heating wall temperature without area correction

Table 5 shows the experimental results and the necessary parameters to perform the DEM simulations related to the 400 °C experiment. The wall temperatures are fixed in the performed simulations.

The particle temperatures are standardized by the temperatures of the hot-and cold reflector with:

$$T^* = \frac{T - T_{out,reflector}}{T_{in,reflector} - T_{out,reflector}} \tag{23}$$

As first validation step simulations without area correction are performed. This testcase was performed with the basic particle bed shown in Fig. 9. The mean number of contacts in the generated bed was 4.018 and the mean contact area per particle was 9.975e-5 [m²]. In separated simulations (i.e., only considering conduction or radiation) the total conductive heat flux $\boldsymbol{q}_{cond,DEM}$ equals 11,572 [W/m²] and the total radiative heat flux $\boldsymbol{q}_{rad,DEM}$ equals 806.9 [W/m²]. In the combined heat transfer simulation $\boldsymbol{q}_{tot,DEM}$ is 12,396 [W/m²] and therefore an order of magnitude higher compared to the experimental data. This leads to the conclusion that the mean contact area is significantly too high and that the area correction is absolutely necessary.

4.2.4. Simulations with 400 °C heating wall temperature and area correction

The closest reproduction of the total experimental heat flux without losses was found with an area correction value of Y^*_{orig}= 9.04e8 [Pa]. The mean contact area per particle was reduced to 5.357e-7 [m²] resulting in a mean contact area per contact of 1.333e-7 [m²]. In a separated simulation the conductive heat flux $\boldsymbol{q}_{cond,DEM}$ is 847.5 [W/m²] (the separated radiative heat flux is 806.9 [W/m²], i.e., same as above).

Fig. 11 shows the individual particle temperatures $T_{P,i}$, the mean temperature per space increment $T_{mean,bound}$, and the experimental data $T_{measured,Exp}$. On the left panel the radiation-only simulation can be seen, while the right panel shows the combined heat transfer simulation. The total calibrated heat flux $\boldsymbol{q}_{tot,DEM}$ fits the experimentally found heat flux.

As can be seen, the fitted temperature curves are very close. The mean squared error between the fitted temperature curve of the experimental data and the simulation temperature is 5.218e-4. A small underestimation of the temperatures near the heating wall can be seen.

Performing simulations with the area-corrected Youngs modulus would impose severe limitations on the time step, and hence would yield a sub-optimal simulation duration. Hence, we have performed simulations with somewhat (i.e., 2x) harder particles, and used the calibrated Youngs modulus determined for the area correction. Specifically, the changed Young's modulus was 1e7 [Pa], an the area corrected value remained at 9.04e8 [Pa].

The total heat flux $\boldsymbol{q}_{tot,DEM}$ for the new particle bed (featuring the 2x harder particles) is 1580 [W/m²] compared to the 1629 [W/m²] of the original bed used for calibration (i.e., 3.0 [%] lower).

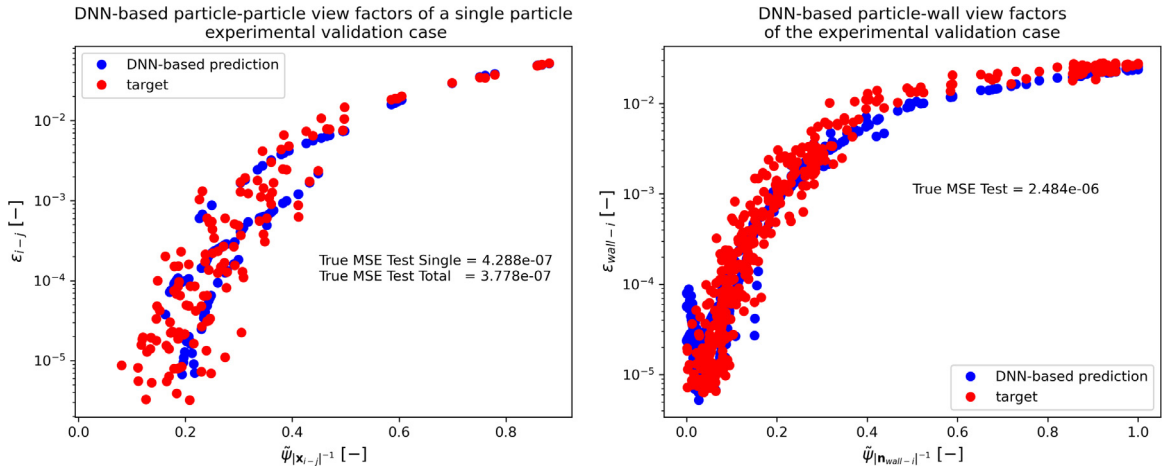

Fig. 10. DNN-based view factors of the experimental validation test case over the normalized distance input markers. Left panel: particle-particle, right panel: wall-particle.

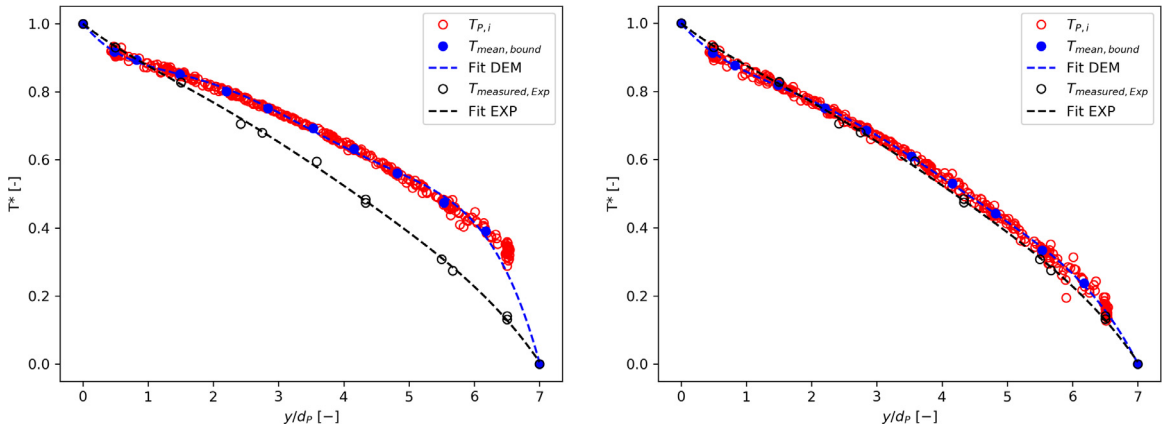

Fig. 11. Temperature distributions with area correction Y^*_{orig}=9.04e8 [Pa]. Left side: radiation-only, right side: combined heat transfer.

The *MSE* between the fitted is curve is 2.269e-3 and therefore significantly higher than the 5.218e-4 of the calibrated bed. The total radiative heat flux $\boldsymbol{q}_{rad,DEM}$ equals 915.6 [W/m²] and is significantly higher (i.e., +13.5 [%]) than 806.9 [W/m²] for the original bed. This shows that the specific bed configuration has a certain influence on the radiative heat transfer rate, however, the total heat flux is only marginally affected.

One has to note that, although the same area correction value was applied, the mean contact area per particle for the case of 2x harder particles (i.e., 2.389e-07 [m²]) was significantly smaller than that of the original calibrated case (i.e., 5.357e-7 [m²]). Therefore, the calibrated area correction value is tied to the particle bed configuration, because changed particle positions and changed stiffness lead to different contact areas (i.e., changed conductive heat fluxes), and different view factors (i.e. changed radiative heat fluxes). It also reflects the complexity of reproducing experimental data of radiative heat transport in a particle bed for which only sparse information is typically available.

Fig. 12 shows the outcome of the radiation-only and the combined heat transfer simulation, using the derived DNNs in Section 3 and the found area correction. In the radiation-only simulation the total heat flux $\boldsymbol{q}_{rad,DEM}$ equals 2087.3 [W/m²] and in the combined heat transfer simulation $\boldsymbol{q}_{tot,DEM}$ is 2910.7 [W/m²]. In both simulations the heat flux is significantly higher and the temperature distributions significantly differ from the experimental data and the outcome based on the DNNs for settled particle beds.

When comparing the view factor prediction as done in Section 4.2.2, it can be seen that the DNN for particle-particle interaction works comparably well with a *MSE* of 4.477e-7, an R^2-value of 0.9879 and an R^2_{spread}-value of 0.5800 on the total test dataset. The DNN for particle-wall interaction in contrast does not predict accurate view factors: the *MSE* is 1.382e-5, R^2 is 0.7605 and R^2_{spread} is below zero on the total test dataset. This is caused by the fact that the non-settled dataset was created within a cuboid domain, which view factor distribution significantly differs from the view factor distribution of a cubic domain (see the right side of Figs. 2 and A.1). When comparing the left side of Figs. 2 and A.1, one can see that this does not apply to particle-particle view factors. This also resembles the influence of the size ratio of the finite wall and the particle ensemble.

4.2.4. Applying the found correction values to the 800 °C experiment

Next, the found area correction value is applied to the high temperature experiment. Table 6 summarizes the necessary parameters to reproduce the experimental test case performed with 800 °C by a DEM simulation.

As addition, in Fig. 13 the temperature profile from the CFD simulation of De Beer [24] is plotted in green. Clearly, the predictions made with the present model are much closer to the experimental data than the CFD simulation of De Beer [24].

For the found area correction values the heat flux of the conduction-only simulation is $\boldsymbol{q}_{cond,DEM}$ is 1466 [W/m²]. In the radiation-only simulation the heat flux $\boldsymbol{q}_{rad,DEM}$ is 5421 [W/m²].

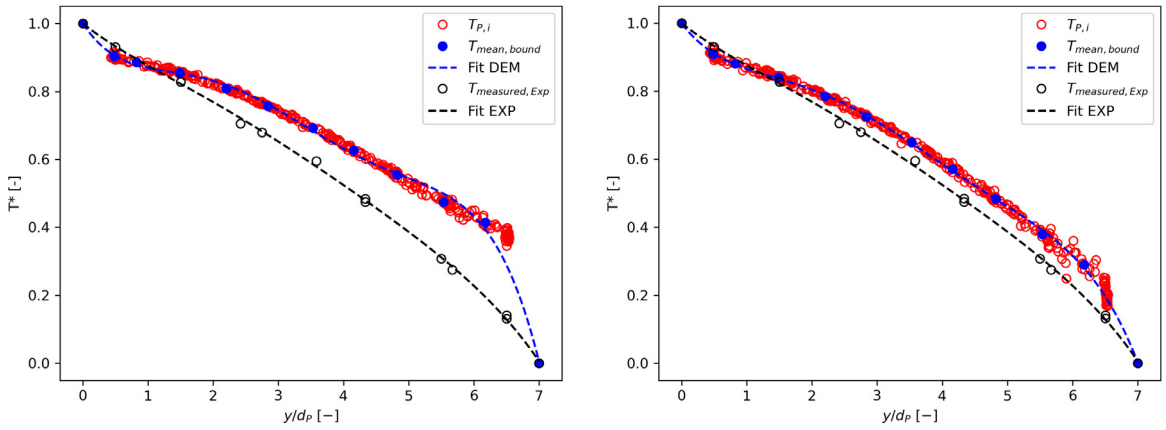

Fig. 12. Temperature distributions with area correction Y^*_{orig}=9.04e8 [Pa] using the derived DNNs of the non-settled system. Left side: radiation-only, right side: combined heat transfer.

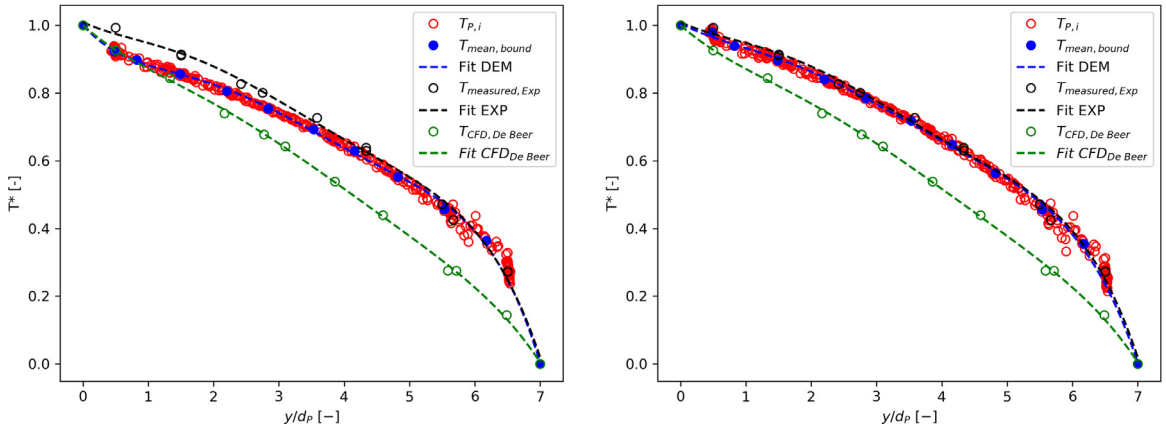

Fig. 13. Temperature distributions: with constant area correction Y^*_{orig}=9.04e8 [Pa] (left panel), and area correction between the heating wall and the particles reduced to 2e5 [Pa] as well as for all other contacts to 3e8 [Pa] (right panel).

Table 6
Overview of experimental measurements of the 800 °C case.

Parameter	Value
$T_{in,reflector}$	1073 [K]
$T_{out,reflector}$	436.2 [K]
$k_{in,reflector}$	63.05 [W/(m K)]
$k_{out,reflector}$	126.1 [W/(m K)]
$\dot{Q}_{out,exp}$	1658 [W]
$\boldsymbol{q}_{tot,exp}$	9397 [W/m²]

In the combined heat transfer simulation $\boldsymbol{q}_{tot,DEM}$ is 6733 [W/m²], which results in a deviation of 28.35 [%] to the experimental data. The *MSE* between the experimental fitted curve and the DEM fitted curve is 1.370e-3.

With increasing temperature, the material stiffness decreases in general. This will lead to increased contact areas in the experiment. Especially the contact areas near the high temperature wall will increase since the temperatures in this region are around 800 °C. This needs to be accounted for in the performed DEM simulations. In the right panel of Fig. 13 the outcome of a simulation is shown when calibrating the area correction between the heating wall and the particle to minimize the difference between the fitted curves. The closest *MSE* between the two curves was found at a value of Y^*_{orig}= 2e5 [Pa], which is actual smaller than Y^* used in the DEM simulation that created the particle bed. The area correction value for particle-particle contacts, as well as the particles contact with the cooling wall was also reduced to Y^*_{orig}= 3e8 [Pa]. As can be seen the temperatures near the heating wall are significantly higher, and the overall temperatures yield a very close reproduction of the experimental data. The total heat flux is then 8245 [W/m²], and hence much closer to the experimental observation. Also, the *MSE* between the fitted curves is significantly reduced to 1.215e-4.

Without the additional area correction, the *MSE* is 1.370e-3 and thus still small, although the deviation of the heat flux mismatch is increased significantly compared to the 400 °C case. An increased underprediction of the wall energy input without additional area correction can be clearly seen. When considering temperature effects on the contact areas, a nearly perfect fit of experimental temperatures could be achieved. However, the total heat flux is still significantly underpredicted by approximately 12 [%]. One can speculate that this disagreement might be caused by the stagnant remaining gas around the particles in the experimental facility, which can contribute significantly to the total heat flux [66].

We finalize with simulation snapshots of both validated cases, displayed in Fig. 14. It is important to note that no gradient in z- and x-direction (i.e., parallel to the walls) can be seen, which is a consequence of the walls being insulated in these directions.

When comparing the temperature curves of the low and higher temperature experiment, not only the ratio of conductive and radiative heat transfer is changing, also the ratio of particle-particle and wall-particle heat radiation is changing. The former can be seen in the change of a close-to-linear temperature curve to a non-linear curve, while the latter is observable by the change of the temperature gradient at the wall.

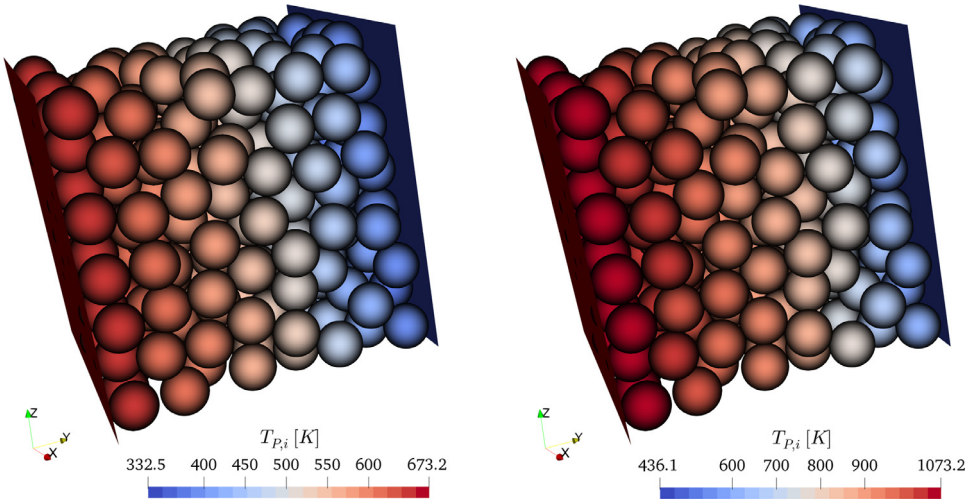

Fig. 14. Simulation snapshots. Left panel: 400 °C test case, right panel: 800 °C test case.

5. Conclusions

A methodology to calculate radiative heat transfer among particles and between particle and walls based on Deep Neural Networks (DNNs) was introduced. There exists only one recent study [35] that has examined particle-wall view factors that are integrable into a DEM-based simulation framework. However, we show in our present study that our DNN model is able to significantly improve view factor predictions compared to a simple correlation with distance: in contrast to this previous work, our DNN correctly predicts a non-zero variance of the view factors at a given particle-particle (or particle-wall) distance which is observed in ray-tracing simulations.

Input data for DNNs that is accessible in DEM simulations was investigated systematically if they are suitable as a marker for particle-particle, as well as particle-wall interaction. From this analysis, and the investigation of the available target data, it can be stated that the crucial point for the prediction accuracy is the quality of the input and target data. In a negative sense this is often quoted with “garbage in garbage out”.

Also, our predictions for the view factors and subsequently the radiation modelling in DEM simulations was validated with experimental data. The differences in the presented results between the simulations and the experiment can be argumented with differences between real view factors and their numerical counterpart based on ray-tracing, as well as the effect of temperature on the material stiffness and hence contact area. The DNN-based prediction cannot achieve better results than the training database derived from ray-tracing and the DEM’s heat conduction prediction is very sensitive to the material stiffness. However, the scope of this work is not to address the differences between numerically determined view factors and their real-world counterparts. The objective was to present a fast and accurate DNN-based method to calculate radiative heat transfer in DEM simulations. In addition, heat transfer effects of the surrounding fluid that still remain after applying vacuum conditions to the experimental setup cannot be completely excluded. Hence, they were not considered in our DEM simulation.

The DNN-based view factor prediction itself achieved high accuracy with a low computational cost compared to methods with comparable accuracy. A substantial reduction of the overall computational cost could be achieved by, e.g., using reduced particle neighbor lists for the $n_{pib,i-j}$ and $n_{pib,wall-i}$ calculation instead of the total particle bed. Thinking the other way around, long-range modeling is limited in performance by the cubic increase in potential neighbors with increasing distance. However, this demands further sensitivity analysis and could be addressed in future studies. By the validation it was also shown that the achieved spread-prediction quality is sufficient to accurately predict temperatures and heat fluxes for DEM-based simulations of dense settled particle beds.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ijheatmasstransfer.2021.121557. The supplementary material contains all trained and used Deep Neural Networks. The DNNs are available in common version (.json-file for the structure and .h5-file for weights and bias) and in a “dumped” version that can directly be used via https://github.com/pplonski/keras2cpp/ .

Appendix A.1: Targets of settled beds

Similar to the Section “Targets” Fig. A.1 exemplarily shows the view factors dependent on the ray density for particle-particle and particle-wall view factors of a random settled particle bed. The particle beds again were generated by a DEM simulation. In difference to the non-settled particle bed overlapping particles are typical in settled beds. The *MSE* between n_{ray} equal to 1e6 and 1e9 is 8.606e-8 and therefore significantly higher than the 2.428e-10 in the non-settled case. For particle-wall interaction the *MSE* between n_{ray} equal to 1e6 and 1e8 is 2.087e-9, also significantly higher than the 2.386e-12 of the non-settled case.

If we consider the maximum view factor per particle over the entire particle bed, the standard deviation and the variance for non-settled case are 0.004962 and 2.462e-5, while for the settled particle bed they are 0.005515 and 3.042e-5. One can therefore conclude that with increasing particle volume fraction and occurring overlaps the fluctuations increase.

Again, the number of found view factors found is increasing significantly with an increasing number of rays. The number of detected view factors are 43, 81, and 105 of 331 possible interactions

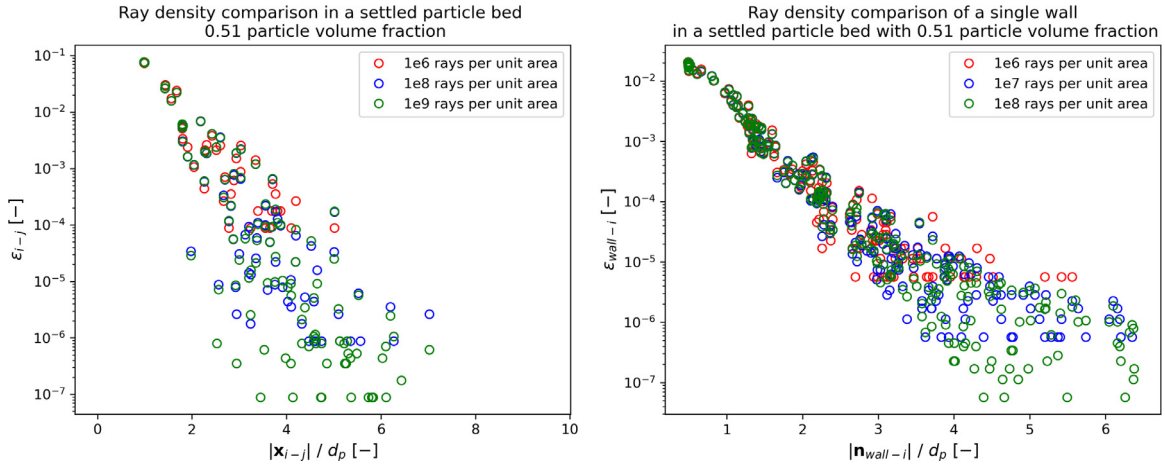

Fig. A.1. View factors for different number of rays per unit area in settled particle beds. Left panel: for a particle bed, right panel: for particle-wall radiation.

Table A.1
Results of randomized search with different amount of test, validation and test data.

Split ratios	n_{hidd}	True *MSE* Test	R^2_{total}	$R^2_{spread,total}$
0.50/0.25/0.25.	200	2.938e-7	0.9921	0.6755
0.60/0.20/0.20	175	2.468e-7	0.9934	0.6963
0.70/0.15/0.15	75	2.490e-7	0.9933	0.7067
0.80/0.10/0.10	175	2.450e-7	0.9933	0.7009

Table A.2
Results of training the original structure with different amount of test, validation and test data.

Split ratios	n_{hidd}	True *MSE* Test	R^2_{total}	$R^2_{spread,total}$
0.50/0.25/0.25.	175	3.673e-7	0.9915	0.6920
0.60/0.20/0.20	175	3.333e-7	0.9922	0.7100
0.70/0.15/0.15	175	2.997e-7	0.9928	0.6294
0.80/0.10/0.10	175	2.450e-7	0.9933	0.7009

for ray densities of 1e6, 1e8, and 1e9, respectively. Compared to the settled particle bed, the relative number of particle-particle interactions is significantly smaller. This is caused by bigger gaps between particles in the non-settled case that do not prevent rays from reaching particles at large distances. The number of found particle-wall view factors are 200, 247, and 271 of 332 possible interactions for ray densities of 1e6, 1e7, and 1e8, respectively. Finally, for the creation of target datasets of the dense particle bed (i.e., particle volume fraction of 0.51) a ray density of 1e8 is chosen.

Appendix A.2: Histograms marker selection

Fig. A.2 shows the histogram of the markers and the target for particle-particle view factors on the top and for particle-wall view factors on the bottom. The particle diameter in the analyzed system was 1 [m].

Appendix A.3: Influence of training data

In addition to the performed randomized search with a training, validation and testing split of 0.80/0.10/0.10, we performed the following splits for the non-settled particle-particle scenario: 0.70/0.15/0.15, 0.60/0.20/0.20 and 0.50/0.25/0.25. The results of this analysis on the respective test dataset are summarized in Table A.1.

As can be seen in Table A.1 a significant reduction of training data from 0.8 to 0.5 decreases the overall quality of the prediction. With at least 0.6 relative amount of training data, no significant differences in terms of prediction quality can be seen anymore. In terms of the found number of hidden nodes, high numbers where found for split ratios of 0.50/0.25/0.25 and 0.60/0.20/0.20, confirming the high number of hidden nodes of the initial approach. For a split of 0.70/0.15/0.15 a significantly smaller number of nodes was found. This shows that due to some degree of randomness in performing a randomized search, a single true optimal number of hidden nodes cannot be found.

In Table A.2 the results of the respective test dataset of trained DNNs based on the found hyperparameters with 0.80/0.10/0.10 split, with the above-mentioned adjusted split ratios are shown. One has to note that in this scenario the DNNs are trained using the basic principle, while the randomized search is based on cross-validation.

One can see that a significant reduction in training data increases the *MSE* on the test dataset. Overall the DNNs derived from basic training have, as expected, higher *MSE* values than cross-validated DNNs if we compare the results from Table A.1 and Table A.2. The results of both additional investigations show that the initially stated training, validation and testing splits and the found DNN model parameter are reasonable.

Appendix B: Predictions for different particle bed configurations

Fig. B.1 shows the prediction of the DNN when the particle bed configuration changes. The training data was created with 0.40 particle volume fraction in an 8 × 8 × 12 [m] domain and 0.5 [m] particle radius. The top and middle rows display the same domain with smaller particle volume fractions, while the bottom row displays results for a different domain size (0.42 × 0.42 × 0.42 [m]), particle radius (0.03 [m]) and a somewhat higher particle volume fraction. R^2_{spread} for the volume fractions 0.10, 0.20 and 0.51 are 0.2518, 0.5312 and 0.6078, respectively.

Appendix C: Computational cost

Table C.1 shows the performance of predicting view factors of the DNN-based model for the settled particle bed case in dependency of the number of nodes in the hidden layer. The boundaries for the determination of $R^2_{spread,PW,test}$ are 0.12 [m] and 0.30 [m] and for calculating $R^2_{spread,PP,test}$. The last entries in Tables C.1 and C.2 reflect the outcome of the DNNs that were used for validation.

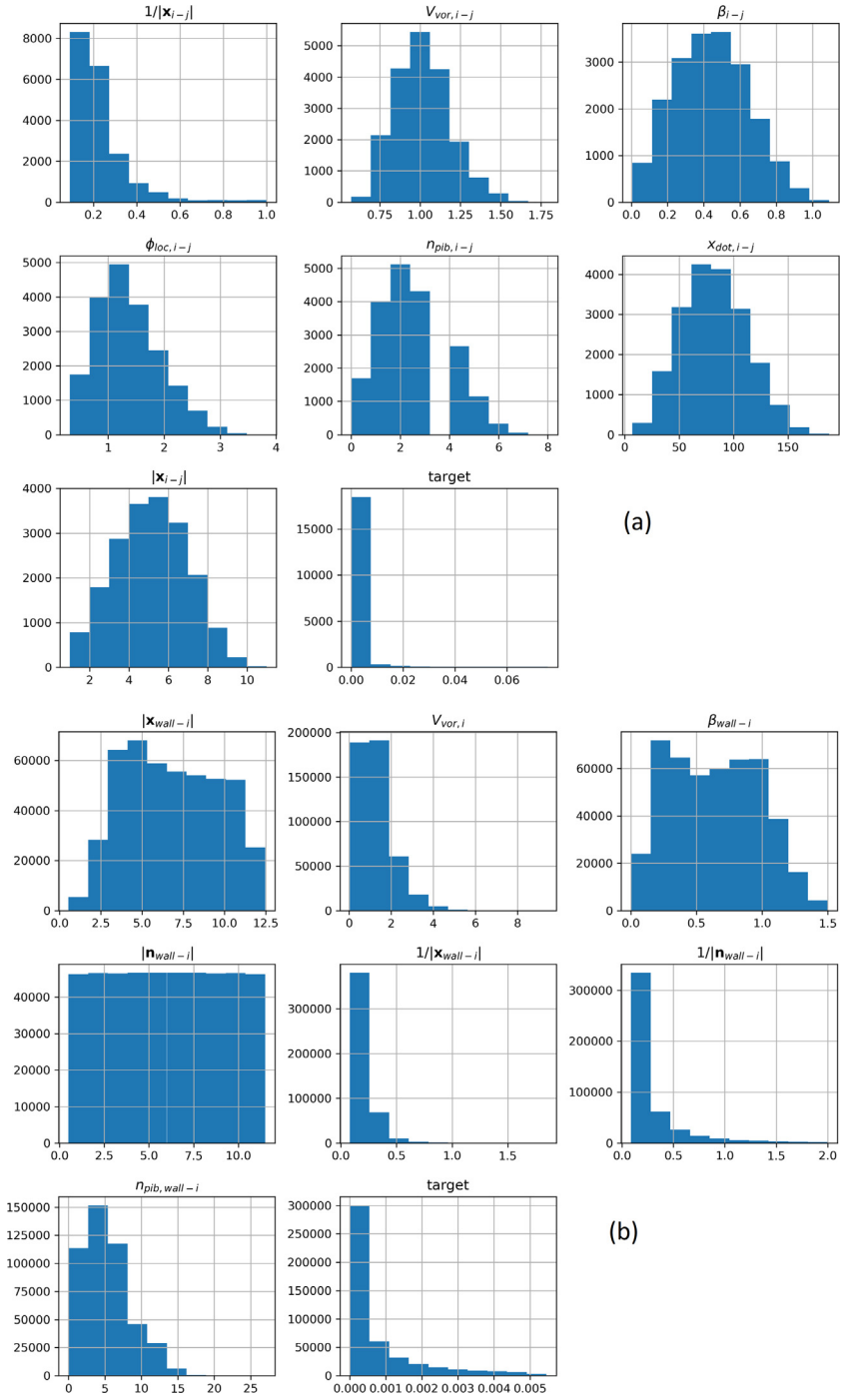

Fig. A.2. Histogram overview of investigated markers and targets. Top panel: particle-particle interaction ($\phi_{loc,i-j}$ is the accumulated particle volume fraction between particle i and j), bottom panel: particle-wall interaction.

As can be the number of hidden nodes does not significantly influence the performance of DNN that predicts particle-particle view factors. In the particle-wall case an increasing number of hidden nodes results in an increasing $R^2_{spread,PW,test}$-value.

Since the particle bed is not moving, the view factors are calculated only during the initialization of the simulation. The effect of the number of nodes on the necessary computation time can be seen in Table C.2. The shown loop time includes the marker calculation, the loading of the DNN, the calculation of the view factors, and the first timestep of the heat transfer DEM simulation. The number of nodes of the DNN-based model for particle-particle view factors and particle-wall view factors were equal in every test. As can be seen the major part of the loop time is occupied by the view factor calculation. The time needed for calculating $n_{pib,i-j}$ and $n_{pib,wall-i}$ takes up around 95 [%] of the shown view factor calculation time. This indicates that the calculation of $1/|\boldsymbol{x}_{i-j}|$ and $1/|\boldsymbol{n}_{wall-i}|$ and the loading and the evaluation of the DNN require very little computational effort. The Monte Carlo-based RayFactor-tool in comparison needed 1541 [s] to calculate the view factors in this scenario.

Overall the reduction of the computational cost is not worth the decrease of $R^2_{spread,PW,test}$. However, finding the optimal hyper-

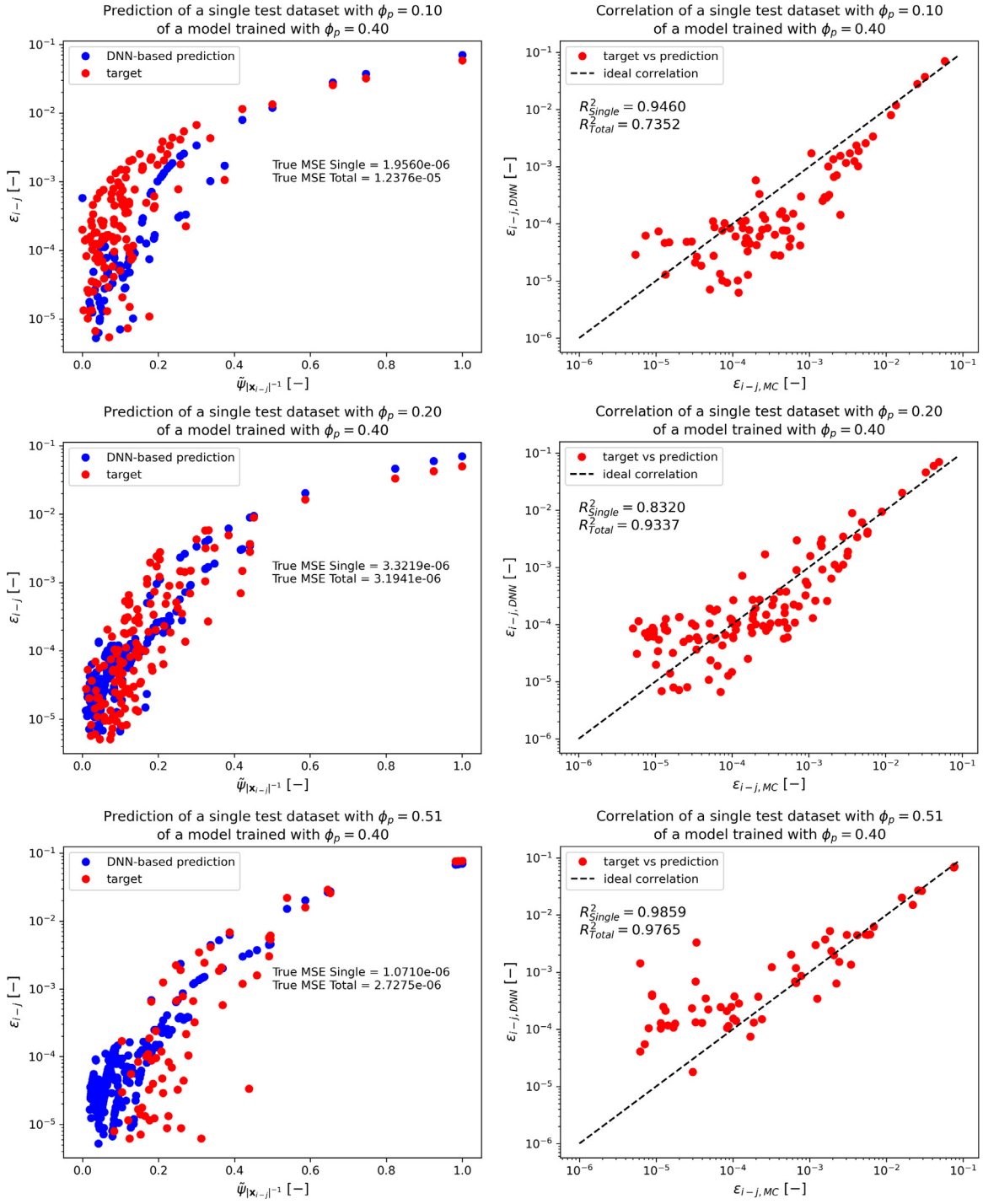

Fig. B.1. Predictions for different particle bed configurations. The left column shows the prediction of a single test dataset over the normalized distance input marker, while the right column shows the according correlation plot.

Table C.1
Performance of the DNN-based models in dependency of the number of nodes in the hidden layer.

	$MSE_{PP,test}$ [-]	$R^2_{PP,test}$ [-]	$R^2_{spread,PP,test}$ [-]	$MSE_{PW,test}$ [-]	$R^2_{PW,test}$ [-]	$R^2_{spread,PW,test}$ [-]
10	4.503e-7	0.9879	0.5881	2.553e-6	0.9558	0.3007
25	4.291e-7	0.9884	0.5621	1.953e-6	0.9661	0.0848
50	4.104e-7	0.9889	0.5915	2.024e-6	0.9649	-0.0277
100	4.420e-7	0.9881	0.5655	2.033e-6	0.9648	0.0753
150	3.986e-7	0.9893	0.6310	1.718e-6	0.9702	0.2932
175	3.778e-7	0.9898	0.6205	2.484e-6	0.9570	0.4272

Table C.2
Computational cost in dependency of the number of nodes in the hidden layer.

	View factor calculation time [s]	Loop time total [s]	$q_{tot,DEM}$ [W/m²]
10	2.813	2.836	565.9
25	2.953	2.968	654.3
50	2.938	2.967	668.7
100	2.984	3.007	654.6
150	3.000	3.011	557.7
175	3.031	3.052	560.9

parameters, including the number of nodes, is not an trivial task and the performed randomized search is favorable for many DNN-applications [59].

An additional option to reduce the computational cost when evaluating DNNs is pruning as shown by Blalock et al. [67]. We have tested pruning of our selected DNNs for the settled particle bed case. The total number of parameters could be reduced from 1231 to 701 for the particle-particle and the particle-wall DNN. In terms of prediction quality: $MSE_{PP,test}$ remained unchanged while $MSE_{PW,test}$ was reduced to 1.937e-6, $R^2_{PP,test}$ and $R^2_{PW,test}$ could be increased to 0.9936 and 0.9664, respectively, and $R^2_{spread,PP,test}$ and $R^2_{spread,PW,test}$ were also increased to 0.6889 and 0.4870, respectively. However, the size reduction and the associated lower computational costs are only available in the Python®-environment. Only with additional implementation they would be available in LIGGGHTS®, but this would limit the flexibility of our approach. This more challenging transfer of the DNN, the limited flexibility, and the for such small neural networks small speed improvement prevented us from implementing pruning in our present study.

CRediT authorship contribution statement

Josef Tausendschön: Conceptualization, Methodology, Software, Validation, Formal analysis, Visualization, Data curation, Writing – original draft, Writing – review & editing. **Stefan Radl:** Conceptualization, Methodology, Writing – review & editing.

References

[1] Y.T. Feng, K. Han, An accurate evaluation of geometric view factors for modelling radiative heat transfer in randomly packed beds of equally sized spheres, Int. J. Heat Mass Transf. 55 (2012) 6374–6383, doi:10.1016/j.ijheatmasstransfer.2012.06.025.

[2] P. von Zedtwitz, W. Lipiński, A. Steinfeld, Numerical and experimental study of gas-particle radiative heat exchange in a fluidized-bed reactor for steam-gasification of coal, Chem. Eng. Sci. 62 (2007) 599–607, doi:10.1016/j.ces.2006.09.027.

[3] C.L. Muhich, B.D. Ehrhart, I. Al-Shankiti, B.J. Ward, C.B. Musgrave, A.W. Weimer, A review and perspective of efficient hydrogen generation via solar thermal water splitting, Wiley Interdiscip. Rev. Energy Environ. 5 (2016) 261–287, doi:10.1002/wene.174.

[4] J. Marti, A. Haselbacher, A. Steinfeld, A numerical investigation of gas-particle suspensions as heat transfer media for high-temperature concentrated solar power, Int. J. Heat Mass Transf. 90 (2015) 1056–1070, doi:10.1016/j.ijheatmasstransfer.2015.07.033.

[5] H.C. Tran, Y.L. Lo, Heat transfer simulations of selective laser melting process based on volumetric heat source with powder size consideration, J. Mater. Process. Technol. 255 (2018) 411–425, doi:10.1016/j.jmatprotec.2017.12.024.

[6] J. Zhou, Y. Zhang, J.K. Chen, Numerical simulation of laser irradiation to a randomly packed bimodal powder bed, Int. J. Heat Mass Transf. 52 (2009) 3137–3146, doi:10.1016/j.ijheatmasstransfer.2009.01.028.

[7] S.B. Pawar, V.M. Pratape, Fundamentals of infrared heating and its application in drying of food materials: a review, J. Food Process Eng. (2017) 40, doi:10.1111/jfpe.12308.

[8] K. Krishnamurthy, H.K. Khurana, J. Soojin, J. Irudayaraj, A. Demirci, Infrared heating in food processing: an overview, Compr. Rev. Food Sci. Food Saf. 7 (2008) 2–13, doi:10.1111/j.1541-4337.2007.00024.x.

[9] A.S. Bongo Njeng, S. Vitu, M. Clausse, J.L. Dirion, M. Debacq, Wall-to-solid heat transfer coefficient in flighted rotary kilns: experimental determination and modeling, Exp. Therm. Fluid Sci. 91 (2018) 197–213, doi:10.1016/j.expthermflusci.2017.10.024.

[10] H.N. Emady, K.V. Anderson, W.G. Borghard, F.J. Muzzio, B.J. Glasser, A. Cuitino, Prediction of conductive heating time scales of particles in a rotary drum, Chem. Eng. Sci. 152 (2016) 45–54, doi:10.1016/j.ces.2016.05.022.

[11] B. Yohannes, H. Emady, K. Anderson, I. Paredes, M. Javed, W. Borghard, F.J. Muzzio, B.J. Glasser, A.M. Cuitiño, Scaling of heat transfer and temperature distribution in granular flows in rotating drums, Phys. Rev. E. 94 (2016) 1–5, doi:10.1103/PhysRevE.94.042902.

[12] W. Van Antwerpen, C.G. Du Toit, P.G. Rousseau, A review of correlations to model the packing structure and effective thermal conductivity in packed beds of mono-sized spherical particles, Nucl. Eng. Des. 240 (2010) 1803–1818, doi:10.1016/j.nucengdes.2010.03.009.

[13] E.S. Huetter, N.I. Koemle, G. Kargl, E. Kaufmann, Determination of the effective thermal conductivity of granular materials under varying pressure conditions, J. Geophys. Res. Planets 113 (2008) 1–11, doi:10.1029/2008JE003085.

[14] M. Díaz-Heras, J.F. Belmonte, J.A. Almendros-Ibáñez, Effective thermal conductivities in packed beds: review of correlations and its influence on system performance, Appl. Therm. Eng. 171 (2020) 115048, doi:10.1016/j.applthermaleng.2020.115048.

[15] G.D. Wehinger, Radiation matters in fixed-bed CFD simulations, Chem. Ing. Tech. 91 (2019) 583–591, doi:10.1002/cite.201800179.

[16] M. De Beer, C.G. Du Toit, P.G. 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, Nucl. Eng. Des. 314 (2017) 67–81, doi:10.1016/j.nucengdes.2017.01.010.

[17] H. Wu, N. Gui, X. Yang, J. Tu, S. Jiang, Effect of scale on the modeling of radiation heat transfer in packed pebble beds, Int. J. Heat Mass Transf. 101 (2016) 562–569, doi:10.1016/j.ijheatmasstransfer.2016.05.090.

[18] H. Wu, N. Gui, X. Yang, J. Tu, S. Jiang, Particle-scale investigation of thermal radiation in nuclear packed pebble beds, J. Heat Transfer. 140 (2018) 1–7, doi:10.1115/1.4039913.

[19] T. Oschmann, H. Kruggel-Emden, A novel method for the calculation of particle heat conduction and resolved 3D wall heat transfer for the CFD/DEM approach, Powder Technol. 338 (2018) 289–303, doi:10.1016/j.powtec.2018.07.017.

[20] B. Krause, B. Liedmann, J. Wiese, S. Wirtz, V. Scherer, Coupled three dimensional DEM-CFD simulation of a lime shaft kiln-Calcination, particle movement and gas phase flow field, Chem. Eng. Sci. 134 (2015) 834–849, doi:10.1016/j.ces.2015.06.002.

[21] H. Wu, N. Gui, X. Yang, J. Tu, S. Jiang, Numerical simulation of heat transfer in packed pebble beds: CFD-DEM coupled with particle thermal radiation, Int. J. Heat Mass Transf. 110 (2017) 393–405, doi:10.1016/j.ijheatmasstransfer.2017.03.035.

[22] H. Wu, N. Gui, X. Yang, J. Tu, S. Jiang, A smoothed void fraction method for CFD-DEM simulation of packed pebble beds with particle thermal radiation, Int. J. Heat Mass Transf. 118 (2018) 275–288, doi:10.1016/j.ijheatmasstransfer.2017.10.123.

[23] K. Agrawal, W. Holloway, C.C. Milioli, F.E. Milioli, S. Sundaresan, Filtered models for scalar transport in gas-particle flows, Chem. Eng. Sci. 95 (2013) 291–300, doi:10.1016/j.ces.2013.03.017.

[24] M. De Beer, Characterisation of Thermal Radiation in the Near - Wall Region of a Packed Pebble Bed, North-West University, 2014.

[25] M.F. Modest, Radiative Heat Transfer - Second Edition, Academic Press, 2003, doi:10.1017/CBO9781107415324.004.

[26] E. Cho, A comparison of the Monte Carlo method to the Discrete Ordinates Method in Fuent for Calculating Radiation Heat Transfer in a Particle Solar Receiver, San Diego State University, 2017.

[27] J.W.C. Tseng, Y. Xia, W. Strieder, Monte Carlo calculations of wall-to-random-bed view factors: impenetrable spheres and fibers, AIChE J. 38 (1992) 955–958, doi:10.1002/aic.690380616.

[28] T. Walker, S.C. Xue, G.W. Barton, Numerical determination of radiative view factors using ray tracing, J. Heat Transfer. 132 (2010) 1–6, doi:10.1115/1.4000974.

[29] S. Amberger, C. Kloss, S. Pirker, Using Ray Tracing to Model Thermal Radiation in LIGGGHTS, 6th International Conference on Discrete Element Methods (DEM6), 2013 http://lammps.sandia.gov/workshops/Aug13/Amberger/presentation.pdf.

[30] A. Malinouski, O. Rabinovich, Evaluation of particle-scale radiative heat transfer in polydisperse beds, in: Proceedings of the International Conference Part. Methods V, 2017.

[31] T. Forgber, S. Radl, A novel approach to calculate radiative thermal exchange in coupled particle simulations, Powder Technol. 323 (2018) 24–44, doi:10.1016/j.powtec.2017.09.014.

[32] F. Qi, M.M. Wright, Particle scale modeling of heat transfer in granular flows in a double screw reactor, Powder Technol. 335 (2018) 18–34, doi:10.1016/j.powtec.2018.04.068.

[33] G. Ruiz, N. Ripoll, N. Fedorova, A. Zbogar-Rasic, V. Jovicic, A. Delgado, M. Toledo, Experimental and numerical analysis of the heat transfer in a packed bed exposed to the high thermal radiation flux, Int. J. Heat Mass Transf. 136 (2019) 383–392, doi:10.1016/j.ijheatmasstransfer.2019.03.009.

[34] M.L. Pitso, Characterisation of long range radiation heat transfer in packed pebble beds, North-West University, 2011.

[35] E.F. Johnson, İ. Tarı, D. Baker, Radiative heat transfer in the Discrete Element Method by distance based approximations of radiation distribution factors, Powder Technol. (2020), doi:10.1016/j.powtec.2020.11.050.

[36] B.K. Liu, J.M. Zhao, L.H. Liu, Radiative heat transfer in densely packed spherical particle system by Monte Carlo method, in: Proceedings of the International Heat Transfer Conference, 2018, pp. 8439–8446, doi:10.1615/ihtc16.rti.024195. 2018-August.

[37] M. De Beer, P.G. Rousseau, C.G. Du Toit, A review of methods to predict the effective thermal conductivity of packed pebble beds, with emphasis on the

near-wall region, Nucl. Eng. Des. 331 (2018) 248–262, doi:10.1016/j.nucengdes.2018.02.029.
[38] Y. Gan, M. Kamlah, Thermo-mechanical modelling of pebble bed-wall interfaces, Fusion Eng. Des. 85 (2010) 24–32, doi:10.1016/j.fusengdes.2009.05.003.
[39] J.N. Kutz, Deep learning in fluid dynamics, J. Fluid Mech. 814 (2017) 1–4, doi:10.1017/jfm.2016.803.
[40] J.H. Lee, J. Shin, M.J. Realff, Machine learning: overview of the recent progresses and implications for the process systems engineering field, Comput. Chem. Eng. 114 (2018) 111–121, doi:10.1016/j.compchemeng.2017.10.008.
[41] Y. Jiang, J. Kolehmainen, Y. Gu, Y.G. Kevrekidis, A. Ozel, S. Sundaresan, Neural-network-based filtered drag model for gas-particle flows, Powder Technol. 346 (2019) 403–413, doi:10.1016/j.powtec.2018.11.092.
[42] L. He, D.K. Tafti, A supervised machine learning approach for predicting variable drag forces on spherical particles in suspension, Powder Technol. 345 (2019) 379–389, doi:10.1016/j.powtec.2019.01.013.
[43] G. Akiki, W.C. Moore, S. Balachandar, Pairwise-interaction extended point-particle model for particle-laden flows, J. Comput. Phys. 351 (2017) 329–357, doi:10.1016/j.jcp.2017.07.056.
[44] W.C. Moore, S. Balachandar, G. Akiki, A hybrid point-particle force model that combines physical and data-driven approaches, J. Comput. Phys. 385 (2019) 187–208, doi:10.1016/j.jcp.2019.01.053.
[45] A. Weiner, D. Hillenbrand, H. Marschall, D. Bothe, Data-driven subgrid-scale modeling for convection-dominated concentration boundary layers, Chem. Eng. Technol. 42 (2019) 1349–1356, doi:10.1002/ceat.201900044.
[46] S. Beetham, R.O. Fox, J. Capecelatro, Sparse identification of multiphase turbulence closures for coupled fluid–particle flows, ArXiv (2020) http://arxiv.org/abs/2004.08997.
[47] H. Wu, N. Gui, X. Yang, J. Tu, S. Jiang, A matrix model of particle-scale radiative heat transfer in structured and randomly packed pebble bed, Int. J. Therm. Sci. 153 (2020) 106334, doi:10.1016/j.ijthermalsci.2020.106334.
[48] K. Nasr, R. Viskanta, S. Ramadhyani, An experimental evaluation of the effective thermal conductivities of packed beds at high temperatures, J. Heat Transfer. 116 (1994) 829–837, doi:10.1115/1.2911455.
[49] H. Wu, S. Hao, A Deep Neural Network model of particle thermal radiation in packed bed, in: Proceedings of the Thirty-Fourth AAAI Conference Artif. Intelligence, Association for the Advancement of Artificial Intelligence, 2020, pp. 1029–1036, doi:10.1609/aaai.v34i01.5452.
[50] H.H. Kang, M. Kaya, S. Hajimirza, A data driven artificial neural network model for predicting radiative properties of metallic packed beds, J. Quant. Spectrosc. Radiat. Transf. 226 (2019) 66–72, doi:10.1016/j.jqsrt.2019.01.013.
[51] C. Kloss, C. Goniva, A. Hager, S. Amberger, S. Pirker, Models, algorithms and validation for opensource DEM and CFD-DEM, Prog. Comput. Fluid Dyn. Int. J. 12 (2012) 140, doi:10.1504/PCFD.2012.047457.
[52] J. Heaton, Artficial Intelligence For Humans, Volume 3: Deep Learning and Neural Networks, Heaton Research, Inc., 2015.
[53] C.H. Rycroft, VORO++: A three-dimensional Voronoi cell library in C++, Chaos 19 (2009) 2008–2009, doi:10.1063/1.3215722.
[54] M. Li, T. Zhang, A.J. Chen, Yuqiang Smola, Efficient mini-batch training for stochastic optimization, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 661–670.
[55] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, 32nd Int, in: Proceedings of the Conference Machine Learning ICML, 2015, 2015, pp. 448–456.
[56] S. Kaufman, S. Rosset, C. Perlich, Leakage in data mining: Formulation, detection, and avoidance, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011, pp. 556–563, doi:10.1145/2020408.2020496.
[57] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, The MIT Press, 2016. http://www.deeplearningbook.org/.
[58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958, doi:10.1016/0370-2693(93)90272-J.
[59] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.
[60] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference Computing Vis. 2015 Inter, 2015, pp. 1026–1034, doi:10.1109/ICCV.2015.123.
[61] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, 3rd Int, in: Proceedings of the Conference Learning Represent. ICLR 2015 - Conference Track Proc, 2015, pp. 1–15.
[62] J.C. Duchi, P.L. Bartlett, M.J. Wainwright, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (2011) 2121–2159, doi:10.1109/CDC.2012.6426698.
[63] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude, COURSERA Neural Netw. Mach. Learn. 4 (2) (2012) 26–31.
[64] J.D. Felske, Approximate radiation shape factors between two spheres tube wall temperatures of an eccentrically located horizontal tube within a narrow annulus, J. Heat Transfer. 100 (1978) 547–548, doi:10.1115/1.3450848.
[65] A. Feingold, K.G. Gupta, New analytical approach to the evaluation of configuration factors in radiation from spheres and infinitely long cylinders, J. Heat Transfer. 92 (1970) 66–69, doi:10.1115/1.3449647.
[66] S. Bu, J. Wang, W. Sun, Z. Ma, L. Zhang, L. Pan, Numerical and experimental study of stagnant effective thermal conductivity of a graphite pebble bed with high solid to fluid thermal conductivity ratios, Appl. Therm. Eng. 164 (2020) 114511, doi:10.1016/j.applthermaleng.2019.114511.
[67] D. Blalock, J.J.G. Ortiz, J. Frankle, J. Guttag, What is the state of neural network pruning? ArXiv (2020) https://arxiv.org/abs/2003.03033v1.