



Transfer learning for radioactive particle tracking

Guilherme Lindner^a, Sai Shi^b, Slobodan Vučetić^b, Sanja Mišković^{a,*}

^a Norman B. Keevil Institute of Mining Engineering, University of British Columbia, Vancouver, Canada

^b Department of Computer and Information Sciences, Temple University, Philadelphia, USA

HIGHLIGHTS

- TL can reuse historical calibration data and give more accurate RPT predictions.
- In a few-shot case, TL predictions are superior to training models using only new data.
- TL is particularly useful when no calibration data are available for a new condition.
- If calibration data are plentiful, TL can be inferior to training models on new data.
- TL is very accurate when historical data are obtained under similar conditions as new.

ARTICLE INFO

Article history:

Received 12 May 2021

Received in revised form 5 October 2021

Accepted 7 October 2021

Available online 16 October 2021

Keywords:

Transfer learning

Radioactive particle tracking

Fluidized bed reactor

GEANT4

Machine learning

ABSTRACT

Radioactive particle tracking (RPT) is a non-invasive technique used to monitor opaque multiphase flow systems. Achieving highly accurate particle tracing is challenging and time-consuming because of the need to build a new RPT model from calibration data each time the experimental conditions change. This paper aims to examine if RPT calibration data under previous conditions can be leveraged with the help of transfer learning (TL) when creating an RPT model for a new condition. Several TL strategies for exploiting historical calibration data are evaluated in conjunction with Geant4 simulations to understand their applicability to RPT. The results show that when it is impractical to collect a lot of calibration data, TL is often superior to training an RPT model only on new data. Moreover, when new calibration data collection is not feasible, an RPT model trained on the historical data can be very accurate if the new condition is sufficiently similar to the historical conditions.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

Radioactive particle tracking (RPT), originally developed by Lin et al. (1985), is a non-invasive flow measurement technique that reconstructs the trajectory of a small gamma-emitting point radiation source acting as a tracer. RPT has been applied in the past to map flow fields inside a wide variety of multiphase units, such as fluidized beds (Lin et al., 1985), bubble column reactors (Devanathan et al., 1990), stirred tanks (Rammohan et al., 2000), and powder blenders (Perrault et al., 2010) among many others. In RPT, the tracer particle bearing gamma-emitting radionuclide, with similar physical properties to the phase being monitored, is introduced into a dynamic multiphase system surrounded by an array of gamma-ray detectors. During the experiment, the intensity and energy of the gamma rays reaching detectors are monitored and used to estimate the tracer's location over time.

One of the critical phenomena intrinsic to the technique is that gamma rays emitted by the tracer interact with the matter through which they pass and that those interactions impact detector readings. The number of gamma rays that reach a detector depends on the tracer activity (number of beams emitted over a unit of time), detector size and orientation, tracer to detector distance, and properties of the attenuating medium. The thicker and denser the attenuating media, the higher the probability of a photon interacting with it. Gamma rays interact with the matter by processes that may generate charged or uncharged secondary radiations and scatter. While many charged particles are adsorbed in the attenuating media, the scattered or secondary uncharged radiation may also reach the detectors. Due to these complex interactions between the emitted photons and the matter, reconstruction of the particle position from the detector readings is not a straightforward task. Many RPT particle reconstruction models (RPT models) have been proposed and implemented over the years to estimate the positions of the tracer particle, including Weighted Least

* Corresponding author.

E-mail address: sanja.miskovic@ubc.ca (S. Mišković).

Nomenclature

α_c	volume fraction of the continuous phase	x	Detector reading
α_p	volume fraction of the solids	y	Tracer position of particle
ϵ	margin of tolerance, hyperparameter of support vector regression	P_c	pressure of the continuous phase
γ	Kernel coefficient, hyperparameter of support vector regression	AI	All-Included transfer learning experiments
b	Bias of feedforward neural network	BL	Baseline single-task experiments
C	Regularization parameter, hyperparameter of support vector regression	FNN	Feedforward Neural Network
K	Kernel function, hyperparameter of support vector regression	GS	Grouped-Source transfer learning experiments
W	Weights of feedforward neural network	IGO	Include-Group-Only transfer learning experiments
μ_i	Mass attenuation coefficient of material i	LG	Leave-Group-Out transfer learning experiments
μ_{li}	Linear attenuation of material i	LO	Leave-one-Out transfer learning experiments
ρ_i	Density of material i	MEDE	Mean Euclidean Distance Error
D_i	Number of photons counted by detector i	ML	Machine Learning
E_i	Efficiency of detector i	PT	Paired-Task transfer learning experiments
u_c	velocity of the continuous phase	RBF	Radial Basis Function kernel
u_p	velocity of the solid phase	RMSE	Root Mean Squared Error
		RPT	Radioactive Particle Tracking
		SVR	Support Vector Regression
		TL	Transfer Learning

Squares (WLS) (Lin et al., 1985), Monte Carlo (MC) (Larachi et al., 1994), Support Vector Regression (SVR) (Yadav et al., 2019), and Artificial Neural Networks (ANN) (Yadav et al., 2020; de Freitas Dam et al., 2021). The modern Machine Learning (ML) models, such as SVR, are capable of reconstructing the tracer position with high accuracy, but they rely on learning from examples and require a lot of calibration data. In current practice, calibration of the RPT system for a new monitored domain is performed before an RPT experiment (Larachi et al., 1994; IAEA, 2008). Once all detectors in the RPT array have been placed in their desired positions, the calibration is performed by recording photon counts at each detector for a finite number of known tracer locations (for example, 150 in Larachi et al. (1994), 376 in Khane et al. (2016), 516 in Yadav et al. (2019), and 901 in Chen et al. (2001)).

The calibration step is the backbone of RPT and the accuracy of the RPT model relies heavily on the availability of high quality and high resolution calibration data (Yadav et al., 2017). However, collection of such data can be technically very challenging, time-consuming, and sometimes impossible. For some systems, such as high-pressure or high-temperature multiphase reactors (Khane and Al-Dahhan, 2017), calibration must be performed in air or empty reactor, when the exact characteristics of the monitored domain are not fully represented in the collected data. Another significant source of error, according to Devanathan (1991), is the dynamic change of density profiles for multiphase systems with high-density ratios, which cannot be captured fully in the standard calibration data. Hence, calibration data are typically limited in number, consider a relatively small number of grid points, and are unable to capture the true attenuation profile of dynamic systems.

This is further complicated by a need for collection of new calibration data each time the properties of the system change even slightly (IAEA, 2008), which then prompts the need for creation of a new RPT model. A significant change in the attenuation profiles of the system may be caused by a small change in the characteristics of the monitored domain and experimental setup, such as (i) density and thickness of the reactor wall, (ii) media density, (iii) number, type, size, and position of the detectors in the array, and (iv) tracer type and activity level.

In summary, there are two main strategies for calibration data collection used in practice. One is to collect new calibration data

for each new condition, which can be expensive or unfeasible. Another is to collect calibration data under a simplified scenario, such as an empty reactor, and hope that the RPT model trained on such data will be sufficiently accurate for a filled reactor. In this paper, we aim to explore if the current RPT practices could be improved by developing highly accurate RPT models in a cost-effective way.

Transfer learning (TL) (Pan and Yang, 2010), which is a popular ML technique, may address some of the aforementioned challenges associated to applying ML models to RPT. TL refers to the settings where data collected over a variety of conditions (called source tasks) are available when training an ML model that will be used over a desired condition (called the target task). For example, by applying TL to automatic speech recognition, models developed for English language can be upgraded with little additional data to work on similar languages such as German (Kunze et al., 2017). In an industrial engineering applications, TL has been used for fault detection, where data from a simulation model was exploited to improve accuracy of fault detection in a physical plant (Li et al., 2020).

The main motivation for the use of TL in the current application is in its potential to reduce the need to collect a large amount of calibration data to train a new RPT model each time the experimental conditions change. We also target cases where it is infeasible or challenging to collect sufficient calibration data. In order to reduce the errors and costs associated with the current RPT modelling practices, this paper explores the application of TL based on the availability of historical calibration data from multiple RPT experimental setups – which will be referred to as tasks in this work – relying on the assumption that there are underlying similarities between the historical and the current tasks. We aim to challenge the current convention of “a single calibration data for a single experiment”, based on the main hypothesis that historical calibration data can be exploited through application of TL to increase RPT accuracy for a new experimental condition.

Results of a detailed empirical study which evaluates the potential of TL in RPT are presented. The considered numerical setup is a lab-scale bubbling fluidized bed reactor, where different reactor wall and media materials commonly used in laboratories and industrial applications are simulated to mimic a wide range of attenuation profiles.

2. Methodology

2.1. Particle physics simulations

To generate the training data, Geant4, a Monte Carlo based platform for high energy physics, is used to develop a digital twin of the RPT system considered in this study and to perform a number of RPT measurements (tasks) for a range of reactor wall and fill media materials.

2.1.1. GIPPE-RPT interface

An in-house software, GIPPE-RPT, is developed to interface with Geant4 and help during all stages of the case setup, particle simulation, and tracer position reconstruction procedures. More specifically, GIPPE-RPT is built to enable easier manipulation of input and output data from Geant4 and to provide a user-friendly interface for a series of functions. All the relevant files used by Geant4 are manipulated directly via GIPPE-RPT interface, and no manual tuning of the simulation parameters is required by the user for a new Geant4 case.

The platform comprises a pre-processor interface, a Geant4 solver, and post-processing tools. A diagram demonstrating the general workflow of the GIPPE-RPT is given in Fig. 1. In the pre-processor block, parameters such as domain size, number of calibration points, tracer type and activity, wall thickness, reactor walls and media materials, and detector type and arrangement can be configured. At the solver stage, the modeled domain is compiled with the pre-processor parameters, and the simulation is performed for the desired calibration grid. The energy spectrum and counts of all photons collected by each detector are stored in a standard ROOT binary file developed by CERN (Rademakers et al., 2019) for each calibration point. After the simulation, each ROOT file is extracted into an ASCII file containing a matrix of simulated points (rows) and the counts for each detector (columns). The post-processing stage allows selection of the tracer position reconstruction methods and provides basic statistics, including mean Euclidian distance error (MEDE) and root mean squared error (RMSE), for each tracer location within the calibration grid. Visual representation of the numerical domain, calibration points, and calibration curves are displayed to ensure the proper definition of parameters.

2.1.2. Geant4 simulation setup

The gamma ray emitting tracer is modelled as a Geantino in the current application, which is included in the Geant4 library and is appended with the properties of the isotope ^{46}Sc . This step involves

definition of the atomic number and the total number of nucleons (protons + neutrons) for the desired radioisotope; the Geantino itself is a virtual particle that does not interact with the matter and handles the transportation processes only, acting as a placeholder for the real particle. A fixed number of beams, 5 million, is defined in the simulation for each tracer position. This parameter is selected for simplicity, as it corresponds to a source activity of 5 MBq - this results in the emission of about 10 million photons for ^{46}Sc , where about 5 million photons are emitted with 0.889 MeV and about 5 million with 1.121 MeV energy signatures. Once all case characteristics are defined within GIPPE-RPT, the simulation is generated and all the input files required for running the simulation in Geant4 are created. Multiple simulations considering different material combinations can be created, resulting in separate folders for each case.

During a simulation, the number of absorbed photons with their respective energies are recorded for each detector. Detector photon counting efficiency, E , which is the ratio of the number of photons collected by the detector and a total number of photons emitted by the tracer, is then calculated and used as the main data input in TL models. Eq. 1 describes the efficiency relation:

$$E_i = \frac{D_i}{N} \quad (1)$$

where, E_i is the efficiency of detector i , D_i is the number of photons counted by the detector i , and N is the total number of photons emitted by the source.

Detector efficiency depends on the detector distance from the tracer, the detector orientation, and the linear attenuation coefficient of the media through which the emitted photons traverse. As the detector to tracer distance increases, the probability of photons reaching the detector decreases. Similarly, as the thickness and density of the attenuating media increase, photons have a higher probability of interacting with the matter and being scattered and attenuated by it.

2.1.3. Reactor walls and fill media properties

In this work, the multi-task RPT simulations consist of twelve distinct scenarios considering different combinations of wall and media materials. More specifically, air, water, glass, and stainless steel are considered as reactor fill media (material contained inside the reactor), and air, acrylic, sapphire glass, and stainless steel are considered as reactor wall materials. When defining the fill media, the assumption is made that the reactor holds a fully compacted bed of particles of the chosen material. However, we should note that the material can be more loosely dispersed within the reactor

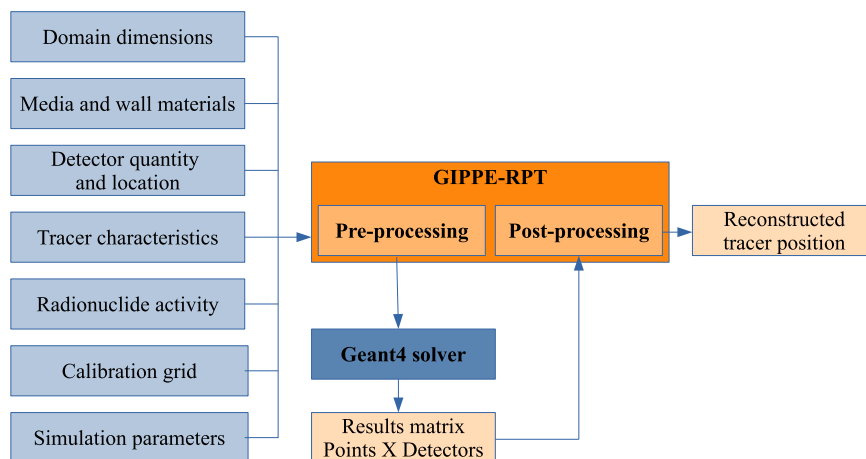


Fig. 1. GIPPE-RPT input and output scheme.

volume as is the case in many real world applications. Table 1 gives an overview of all combinations of fill media and wall materials considered in this study, with their corresponding abbreviations that will be used in the following sections. It is worth noting that the Air-Air condition represents a tracer in the open environment (in the absence of a reactor wall and media).

The chemical composition and attenuation coefficients of each material that is used in this work is presented in Table 2. The mass attenuation coefficients, μ_i , are taken from the National Institute of Standards and Technology (NIST) database (NIST Office of Data and Informatics), for the tracer energy level of 1 MeV. The linear attenuation, μ_{li} , is calculated using the following expression:

$$\mu_{li} = \mu_i \rho_i \quad (2)$$

where, ρ_i is the density of the material i . As could be noticed, the mass attenuation coefficients for the considered materials are similar in comparison, which is a known characteristic for tracer gamma energies between 800 and 1400 keV (McAlister, 2012). However, the linear attenuation varies significantly between materials and is directly proportional to their densities.

Physical phenomena associated with the photon transport through different attenuating media are investigated using MC simulations in Geant4, as shown in Fig. 2. Here, faith of 20 photons emitted from a tracer located in the middle of the simulated domain is given for 12 media/wall material pairs. ^{46}Sc radionuclide is chosen as a tracer in all experiments, which decays via β^- decay and, hence, gives off an electron (red paths), an electron antineutrino (gray paths), and gamma-rays (green paths). Perhaps intuitively, materials with higher densities are shown to have a higher gamma ray attenuating potential, whereas negligible attenuation is occurring when the tracer is suspended in the air.

To better understand the cumulative attenuation properties of each media-wall pair (task) considered in this study, the composite linear attenuation coefficients are determined using MC simulations in Geant4 and a collimator setup used by Akça and Erzeneoglu (2016). Here, a narrow photon beam directed toward the tested composite material is obtained by two collimators placed between the gamma emitting tracer ^{46}Sc with 100 MBq source activity. The thickness of the composite sample is chosen to represent the tracer position that is in the center of the fluidized bed, relative to the detector oriented parallel with the longer side of the reactor (parallel to the reactor width). The total thickness of the composite material through which the photons are allowed to traverse is 121 mm, where 115 mm corresponds to the filling media thickness and 6 mm is the thickness of the reactor wall. Task A-A is used as a reference case for all the calculations. The composite linear attenuation of all twelve material pairs is given in Table 3 and it is calculated from the combined energy levels (0.889 MeV and 1.121 MeV) using the Beer-Lambert's rule

$$\mu = -\left[\frac{\ln(I/I_0)}{t}\right] \quad (3)$$

where, I and I_0 are the attenuated and reference collected photons, μ is the linear attenuation coefficient, and t is thickness of the target sample.

Table 1
Fill media and wall material pairs considered in the current multi-task RPT simulations.

Media/ Wall	Air	Acrylic	Sapphire Glass	Stainless Steel
Air	A-A	A-AC	A-SG	A-S
Water	–	W-AC	W-SG	W-S
Glass	–	G-AC	G-SG	G-S
Stainless Steel	–	S-AC	S-SG	–

2.1.4. Simulation setup

The geometry selected for particle transfer simulations and tracer position reconstruction is a lab-scale rectangular gas-solid fluidized bed, with dimensions $0.23 \times 0.076 \times 1.2$ m based on the NETL SSCP I experiment geometry (Gopalan and Shaffer, 2012). Specifically, a smaller domain representing a lower segment of the column with the dimensions of $0.23 \times 0.076 \times 0.2$ m is in focus of the current study, and this is the domain from where all the calibration data are acquired. An array of sixteen $2'' \times 2''$ NaI(Tl) scintillation detectors is placed around the reactor as shown in Fig. 3a. The detectors are modelled as individual objects, which are covered by a 1 mm layer of aluminium, representing the Scio-nix Holland NaI(Tl) Gamma Scintillation Detector. The exact locations of the detectors relative to the fluidized bed and each other is given in Fig. 3b. Different materials of reactor wall and fill media, which represent materials commonly used in laboratories and industrial applications, are simulated to obtain a wide range of attenuation profiles. The reactor geometry, detector type, size, and placement, type and size of the calibration grid, and tracer type and activity are common for all simulations in this work.

2.1.5. Data size and format

The details of our simulated dataset are as follows. For each of the twelve simulated conditions, photon counts received by each of the sixteen detectors are collected for a total of 10527 tracer locations within the calibration grid. The tracer locations are populated in the domain with a spacing of 7 mm in each direction, resulting in a $33 \times 11 \times 29$ grid as shown in Fig. 4. The physical dimensions of the domain in x, y, z directions are 0.23 m, 0.076 m, and 0.2 m, respectively. The detector data for each simulated condition is organized as a 10527×16 matrix, with row representing 10527 tracer locations and columns representing 16 detector locations. The corresponding tracer locations are organized as 10527×3 matrix, with columns representing the x, y , and z coordinates of the tracer. The center of the domain is at (0, 0, 0) coordinate.

2.2. Machine learning approach for RPT

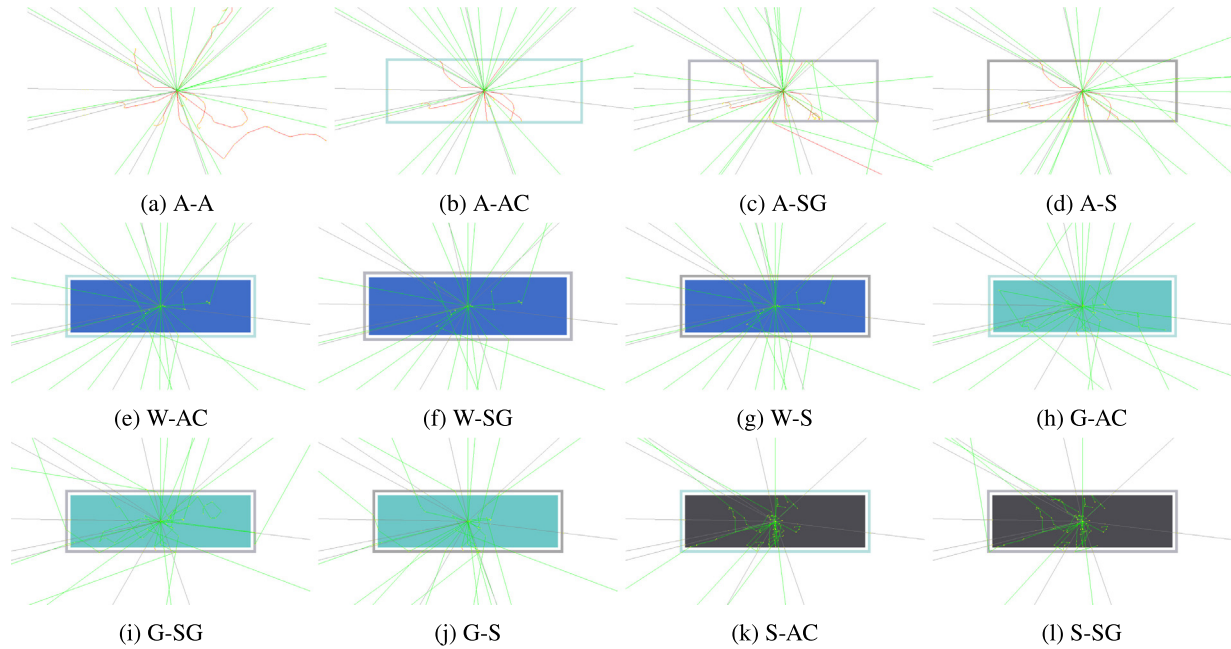
In this section we describe a machine learning (ML) approach that is applied to examine if the historical calibration data collected over a variety of conditions could be leveraged in RPT. ML uses an inductive learning approach to learn a function f that predicts tracer coordinates $\mathbf{y} = (y_1, y_2, y_3)$ along the 3 dimensions given detector readings $\mathbf{x} = (x_1, x_2, \dots, x_{16})$ for the 16 detectors shown in Fig. 3. The function is learned using a training data set $D = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1 \dots N\}$ consisting of N labeled data points $(\mathbf{x}_i, \mathbf{y}_i)$, where \mathbf{x}_i is a vector of the observed detector readings for the known tracer position \mathbf{y}_i .

To learn prediction function f from training data, ML models typically minimize an objective function that is a sum of prediction losses (e.g., squared errors) observed on the training data and a regularization term (e.g., a 2-norm of the weights of the prediction function) that controls the complexity of the prediction function. The purpose of the regularization term is to reduce overfitting, a phenomenon occurring when a predictor achieves high accuracy

Table 2

Chemical composition and attenuation coefficients of selected fill media and wall materials for the 1 MeV source at 14 °C and 101.325 kPa.

Material Name	Material Density, g/cm ³	Mass Attenuation, cm ² /g	Linear Attenuation, cm ⁻¹	Chemical Composition, –
Air	0.00123	6.350E–02	0.00008	ArN ₇₆ O ₂₃
Water	0.9992	7.072E–02	0.0706	H ₂ O
Acrylic	1.1900	6.870E–02	0.0818	C ₅ H ₈ O ₂
Glass	2.5000	6.346E–02	0.1586	Ca ₁₁ Na ₉ O ₄₆ Si ₃₄
Sapphire Glass	3.9500	6.252E–02	0.2469	Al ₂ O ₃
Stainless Steel	8.0500	6.000E–02	0.4830	Cr ₁₉ Fe ₆₈ Mn ₂ Ni ₁₀ Si

**Fig. 2.** Top view of the results from Geant4 MC simulations with ⁴⁶Sc tracer and 20 emitted photons showing gamma ray attenuation by different media pairs representing twelve multi-task RPT experiments. The green paths are gamma rays, red paths are electrons, and gray paths are antineutrinos.**Table 3**Combined linear attenuation for composite materials using ⁴⁶Sc as gamma emitting tracer and 121 mm sample size (media thickness = 115 mm; reactor wall thickness = 6 mm).

Task	A-A	A-AC	A-SG	A-S	W-AC	W-SG	W-S	G-AC	G-SG	G-S	S-AC	S-SG
μ_l [cm ⁻¹]	ref	0.004	0.010	0.020	0.058	0.065	0.075	0.124	0.132	0.141	0.370	0.379

on training data but significantly lower accuracy on data unseen during training (test data). RPT is an example of a regression problem, because the tracer positions \mathbf{y} are real numbers. We use two ML models in this study: support vector regression (SVR), which is a representative of kernel ML methods, and feedforward neural networks (FNN), which is a representative of deep learning methods. They will be summarized next. After that we will introduce TL and explain how it relates to RPT. We will end this section by defining the accuracy measures used in this work.

2.2.1. Support Vector Regression (SVR)

Support vector machine (SVM) (Cortes and Vapnik, 1995) is an ML model that was originally proposed for binary classification, but the idea has been extended to other supervised learning problems, including regression, when the resulting model is called the support vector regression (SVR) (Drucker et al., 1997). The SVR prediction function has the following form

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i \cdot K(\mathbf{x}_i, \mathbf{x}) + b \quad (4)$$

where $K(\mathbf{x}_i, \mathbf{x})$ is the kernel distance between \mathbf{x}_i and \mathbf{x} , K is the kernel function, α_i is the weight associated with the i -th training data point, and b is a scalar bias term. Typically, many or most of the α_i weights are zeros and the training data points $(\mathbf{x}_i, \mathbf{y}_i)$ having non-zero α_i are called the support vectors. Thus, to make a prediction for a new data point \mathbf{x} , the SVR needs to find its kernel distances from all the support vectors. Intuitively, SVR finds a weighted average of labels of its neighboring support vectors as defined by distance K .

To determine the prediction function f , which involves determining weights α_i and bias b , SVR training model minimizes the sum of the ϵ -insensitive loss over training data points and a scaled regularization term with the scaling hyperparameter C . The ϵ -insensitive loss for labeled data point $(\mathbf{x}_i, \mathbf{y}_i)$ is defined as $|\mathbf{y}_i - f(\mathbf{x}_i)|_\epsilon$, where $|x|_\epsilon = 0$ if $|x| \leq \epsilon$ and $|x|_\epsilon = |x| - \epsilon$ if $|x| > \epsilon$, and ϵ is a hyperparameter.

To train an SVR predictor, a user needs to specify kernel function K and hyperparameters C and ϵ . In this study, we experimented with 4 types of kernel functions: linear, polynomial, RBF, and sigmoid. For example, the RBF kernel function is

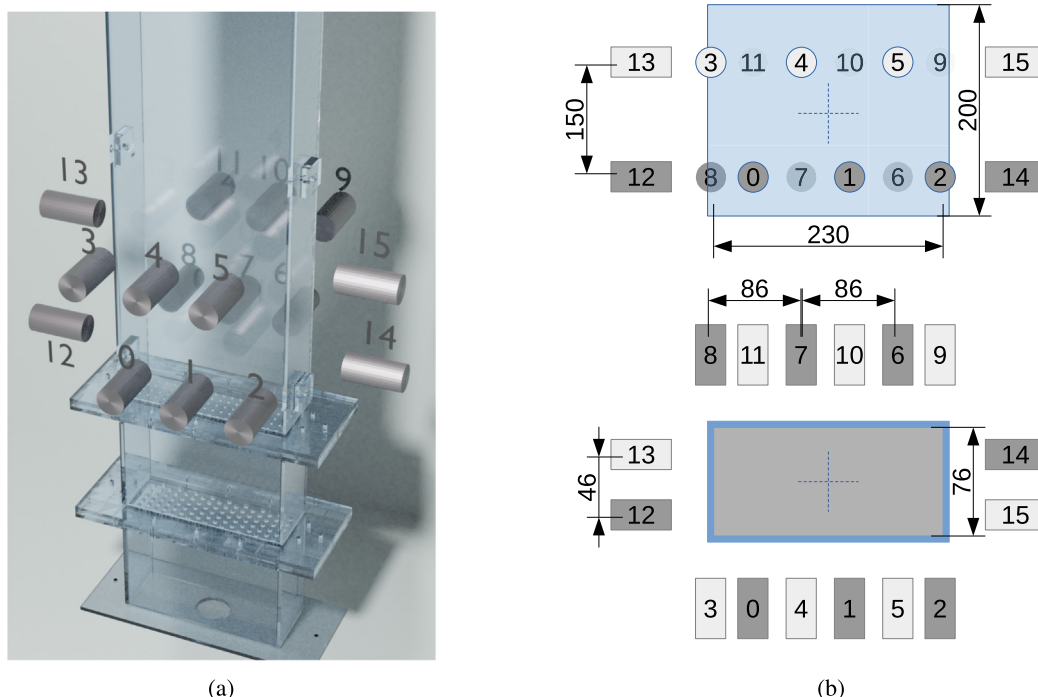


Fig. 3. Geometry of the fluidized bed reactor showing relative placement of 16 detectors with their corresponding numbers/names (a), and side and top view of the reactor domain and detector placement with exact dimensions (b). All dimensions are given in mm.

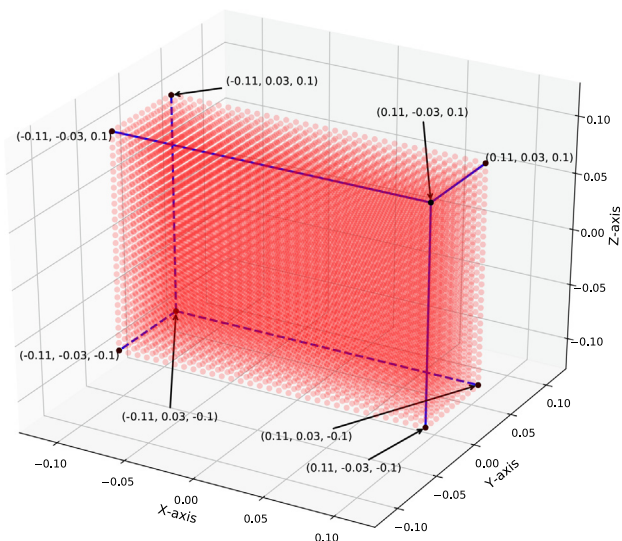


Fig. 4. Tracer locations in the calibration grid.

$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\gamma \|\mathbf{x} - \mathbf{x}_i\|^2)$, where γ is a hyperparameter. All kernels excluding the linear kernel have one or more hyperparameters. To determine the right combination of SVR hyperparameters, a grid search is performed over a set of choices for each hyperparameter. For a given selection of hyperparameters, a random subset of training data points (called validation data) are excluded from training data and used to test prediction accuracy. The set of hyperparameters that results in the highest accuracy on the validation data is selected as the best choice.

We observe that the tracer position \mathbf{y} has three coordinates. Since the SVR in its original form assumes there is only one output, we train three separate SVR prediction functions, one for each of the coordinates.

2.2.2. Feedforward Neural Networks (FNN)

Artificial neural networks (Dreyfus, 1990) are a class of ML models originally developed a few decades ago that have grown in popularity with recent methodological advances that allowed them to reach state of the art accuracy on a range of very challenging applications. In this study we consider a particular type of neural network called the feedforward neural network (FNN). The basic unit of FNN is a neuron, which is a function that transforms an input vector \mathbf{x} into a scalar output $o = g(\mathbf{w}^T \mathbf{x} + b)$, where g is a single-valued transfer function of the dot product between the input vector \mathbf{x} and a vector of neuron weights \mathbf{w} increased by the scalar bias term b . In this study, we rely on two types of transfer functions: linear, where $g(x) = x$, and RELU, where $g(x) = x$ if $x \geq 0$ and $g(x) = 0$ otherwise.

In FNN, an input vector \mathbf{x} of length M (in our case, $M = 16$ detector readings) is passed to H_1 neurons (called the first hidden layer), each with a different set of weights. The H_1 outputs provided by those neurons are concatenated to create an output vector \mathbf{o}_1 , which is a nonlinear transformation of the input vector \mathbf{x} . The output vector can be written in a matrix form as $\mathbf{o}_1 = g(W_1 \cdot \mathbf{x} + \mathbf{b})$, where W_1 is an $H_1 \times M$ matrix whose j -th row are weights of the j -th neuron, and \mathbf{b} is a vector of bias terms of the H_1 neurons. Output \mathbf{o}_1 can then be used as an input to the second hidden layer with H_2 neurons, each with its own set of weights. This process can be repeated over a total of h hidden layers. Output of the h -th hidden layer is used as an input to the output neurons. Since in our application we are predicting 3 coordinates of the tracer location, we use FNN with three output neurons having linear transfer function.

To learn all the FNN weights, the mean squared error (squared error is defined as $(y - f(x))^2$ over the training data points is minimized using backpropagation (Rosenblatt, 1958), which is a computationally efficient implementation of the gradient descent algorithm. Rather than introducing a regularization term in the objective function to prevent overfitting, modern FNN algorithms rely on a range of approaches, including dropout (Srivastava

et al., 2014), batch normalization (Ioffe and Szegedy, 2015), and early stopping of backpropagation.

FNNs have a range of hyperparameters that need to be selected. This includes the FNN architecture with choices such as the number of hidden layers, number of neurons per hidden layer, and types of transfer functions. There are also hyperparameters related to FNN training such as properties of dropout, number of epochs (iterations of gradient descent), batch size (number of training data points used for one step of gradient descent), and parameters of gradient descent such as algorithms type and learning rate. To determine all those hyperparameters, practitioners rely on a mix of best practices (common default choices) and grid search using validation data (Liashchynskiy and Liashchynskiy, 2019). This process will be further discussed in the results section.

2.2.3. Transfer learning for RPT

The standard assumption in machine learning is that the training data are a random sample from an underlying distribution and that a prediction model trained on such data will be used on samples from the same distribution. However, in many applications this standard assumption is not valid. This has motivated research in TL (Pan and Yang, 2010), where the overall question is whether training data obtained by sampling from one distribution can be leveraged when building a prediction model that will be used on samples from a different distribution. In this paper, we study RPT as an example of a TL problem. As explained in Section 2.1 and as will be demonstrated in Section 3.1, every time the simulated conditions change the distribution of detector readings \mathbf{x} changes and the relationship between those readings and the tracer location \mathbf{y} changes. Thus, the TL question in RPT is whether historical calibration data obtained previously under different simulated conditions can be leveraged to achieve accurate particle tracking over the current condition. For this study, as explained in 2.1, we simulated calibration data corresponding to 12 different properties of the wall and fill materials. Our simulated data allow us to thoroughly evaluate the potential of TL in RPT applications.

In the TL terminology, the training data corresponding to a fixed simulated condition are referred to as the task. Let us denote the training data set from the j -th task as $D^j = \{(\mathbf{x}_i^{(j)}, \mathbf{y}_i^{(j)}), i = 1 \dots N_j\}$, where N_j is the number of training data points obtained from the j -th simulated condition. If we are interested in building an accurate RPT model for the j -th task, we refer to it as the target task. The remaining tasks are called the source tasks. A common approach to TL is to minimize an objective function that is a weighted average of prediction losses on source and target tasks together with an optional regularization term,

$$\min_f \left[\sum_{ij} \alpha_{ij} \cdot \text{loss}(\mathbf{y}_i^{(j)}, f(\mathbf{x}_i^{(j)})) + \text{regularization}(f) \right] \quad (5)$$

where α_{ij} is the weight given to the i -th training data point from the j -th task. The success of this optimization task is measured by the accuracy of the resulting prediction function f on data points from the target task. Weights α_{ij} are treated as hyperparameters. For example, if only data from k -th source task are used for predictor training, $\alpha_{ij} = 1$ if $j = k$ and $\alpha_{ij} = 0$ otherwise.

Interestingly, unless the target task has a large number of training data points and the source tasks are very dissimilar from the target task, this TL approach often results in a more accurate prediction function than when training only on data points from the target task. It should be added that this TL approach is applicable under the so-called zero-shot scenario (Xian et al., 2017; Wang et al., 2019), where training data points are not available for the target task. The zero-shot scenario in RPT would arise when it is not feasible to collect calibration data for the target task either

due to technical reasons or because experimental conditions (e.g., properties of the fill material) are changing too rapidly. Another TL scenario relevant to RPT is the few-shot scenario (Wang et al., 2020), where target (and, possibly, source) tasks have only a limited number of training data points. For example, in RPT applications it is possible to easily obtain thousands of calibration data points by simulation, but it is much more difficult to obtain them under experimental conditions. Also, it is much easier to obtain calibration data under air-air (A-A) condition than in the presence of walls and dense fill material.

We explore several TL scenarios for RPT in this study as illustrated in Fig. 5. In each illustrated scenario, we assume that the target task corresponds to the S-AC simulated condition. The scenarios differ by what training data are used to build a prediction model for the target tasks: (i) Baseline (BL) uses only data from the target task; (ii) Paired-Task (PT) uses training data from a single source task (e.g., A-AC); (iii) Leave-One-out (LO) uses training data from all source tasks but not from the target task; (iv) All-Included (AI) uses training data points from both source and target tasks; and (v) Group-Source (GS) uses only a subset of source tasks. PT, LO, and GS are all zero-shot TL scenarios.

2.2.4. Accuracy measures

To evaluate the RPT prediction functions we use three types of measures. The first is the Mean Euclidean Distance Error (MEDE) defined as

$$\text{MEDE} = \frac{\sum_{i=1}^N \sqrt{(\mathbf{y}_i - f(\mathbf{x}_i))^T (\mathbf{y}_i - f(\mathbf{x}_i))}}{N}, \quad (6)$$

where N is the number of labeled test data points that were not used for predictor training. The second is the Root Mean Squared Error (RMSE) which, in the case of a scalar prediction, is defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2}{N}}. \quad (7)$$

Because in the RPT application the prediction are three coordinates, we can report the RMSE for each coordinate. We can also report the average RMSE over the three coordinates. The third measure is R^2 score, which is also calculated for scalar outputs and can be derived from RMSE as

$$R^2 = 1 - \frac{\text{MSE}}{\text{Var}(y)}, \quad (8)$$

where $\text{MSE} = \text{RMSE}^2$ and $\text{Var}(y)$ is the sample variance of variable y over the test data points. Thus, R^2 can be reported separately for each coordinate or it could be averaged over the three coordinates.

For all the experiments performed and results presented in the following section, the unit of measure is millimeter.

3. Results & discussion

This section starts with analysis of detector efficiencies (readings) collected for 12 distinct material pairs, 16 detectors, and 10527 tracer locations. The main distinctions between gamma attenuation properties of different materials and their effect on photon count and energy spectrum received by detectors will be highlighted. The TL results obtained for a range of settings will then follow. All the results consider the entire energy spectrum, without discrimination between coherent and incoherent Compton scattered rays and other phenomena.

Baseline (BL)												
Source	A-A	A-AC	W-AC	G-AC	S-AC	A-SG	W-SG	G-SG	S-SG	A-S	W-S	G-S
Target					S-AC							
Paired-Tasks experiment (PT)												
Source	A-A	A-AC	W-AC	G-AC	S-AC	A-SG	W-SG	G-SG	S-SG	A-S	W-S	G-S
Target					S-AC							
Leave-One-out experiment (LO)												
Source	A-A	A-AC	W-AC	G-AC	S-AC	A-SG	W-SG	G-SG	S-SG	A-S	W-S	G-S
Target					S-AC							
All-Included experiment (AI)												
Source	A-A	A-AC	W-AC	G-AC	S-AC	A-SG	W-SG	G-SG	S-SG	A-S	W-S	G-S
Target					S-AC							
Grouped Source experiment (GS)												
Source	A-A	A-AC	W-AC	G-AC	S-AC	A-SG	W-SG	G-SG	S-SG	A-S	W-S	G-S
Target					S-AC							

Fig. 5. TL scenarios for RPT. Assuming S-AC is the target task (dark orange), the selected source tasks are highlighted in light orange and the ignored source tasks are shown in blue.

3.1. RPT data

Photon counting efficiencies of detectors 13 and 14 are shown in Fig. 6 to help demonstrate how material properties (tasks) and tracer to detector geometry affect the results. Each point in the figures represents detector efficiencies of the two detectors for one tracer location. From a total of 10527 tracer locations over the grid, 200 random locations are selected to enable clearer visualization of data from multiple tasks. The lines connecting the points representing the same tracer location for different tasks are also shown.

A $x = y$ reference line is shown in each subfigure of Fig. 6, which represents a relationship between the efficiencies of two perfectly overlapping detectors. There are two limit cases that should be considered here. First, for the tracer location that is at an infinite distance from the detectors, the resulting pair of efficiencies will reach zero (0,0 on the reference line). In the case where the tracer is positioned right at the detector face, and no photon attenuation is occurring, the efficiency pair is at its highest limit (second limit of the reference line). The points in the figures that are close to the $x = y$ reference line correspond to the tracer locations with a similar distance from the two detectors. On the other hand, the points furthest away from the reference line, with high efficiency for one detector and low efficiency for the other detector, represent tracer locations that are near the detector with high efficiency and far from the other detector.

From Figs. 6a and b, similar efficiency curves are obtained for test cases with the same media material properties - air in the left figure and water in the right figure, due to their similar cumulative attenuation potential. This is particularly true for water as a fill media, where efficiency curves are practically overlapping. It can be concluded that density of the wall material has a smaller impact on the number of collected photons due to its small thickness relative to the media. In other words, a significant correlation in efficiency responses could be observed for this subgroup of media-wall pairs (tasks).

Figs. 6c and d demonstrate how the change in the material properties of the fill media affects the efficiencies of the same pair of detectors. Here, an increase in media density results in the shift of the detector efficiency curves toward the axes (gaining steeper exponential decay response). For these subgroups of experiments,

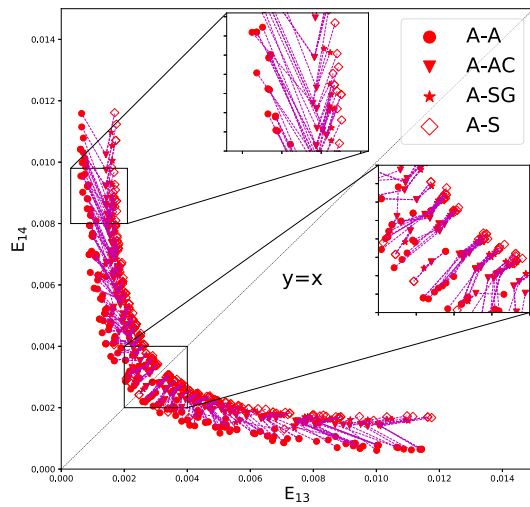
the higher composite attenuation coefficient of the media-wall pair results in the reduced correlation between the tasks.

Focusing on the tasks with the most distinct media-wall attenuation potentials, namely A-AC and S-AC, a full range of possible detector efficiencies collected by the 13–14 detector pair can be seen in Fig. 6d. A significant shift of the detector efficiency curve toward the axes can be observed again, with the extreme case being the S-AC task where the curve transitions from an exponential decay to a Dirac-like response.

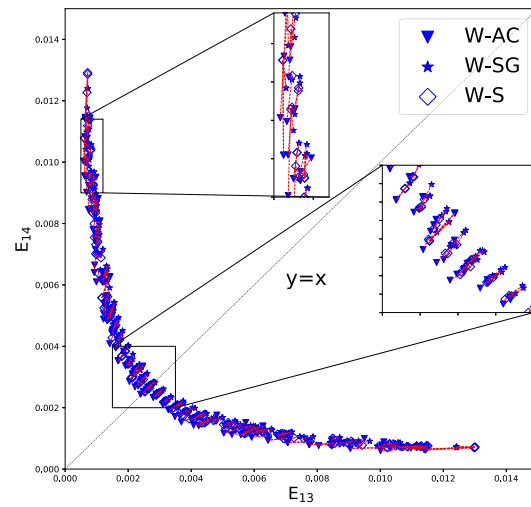
Somewhat non-intuitive trend could be observed when analyzing efficiency clusters for the given detector pair (13–14) and two material pairs with distinctly different gamma ray attenuation potential, such as A-AC and S-AC in Fig. 6d. It may be expected that the efficiency for stainless steel fill media (S-SG task) will be significantly lower than that for air, but the opposite relationship is found at the extremes. This can be observed by the outward stretching of the connecting lines from the task with lower density media (A-AC) to the task with the heavier media (S-AC).

To help us better explain this phenomena, the full energy spectrum of the received photons at detectors 13 and 14 and for four different tasks, namely A-SG, W-SG, G-SG, S-SG, is given in Fig. 7. The tracer in these experiments is located at the closest proximity from detector 13 (near the wall) and at the opposite side from detector 14. As can be seen in the detailed view of Fig. 7a, the increase in the number of low energy photons with an increase in gamma attenuating potential of the fill media, in A-W-G-S order, come as a result of the Compton scattering. These low energy photons contribute to the high efficiency with the stainless steel media when the tracer is near the detector.

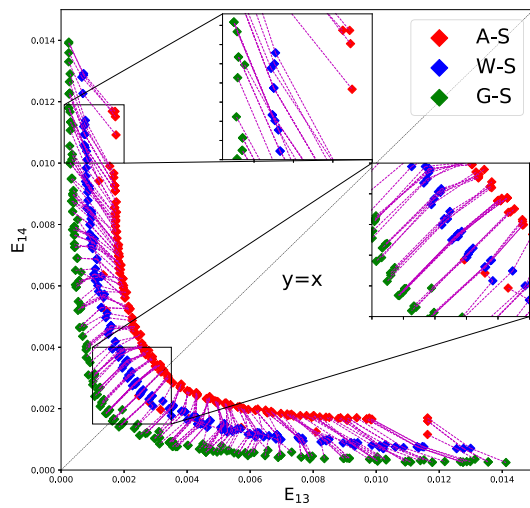
Fig. 8 shows the fate of 200 photons, which are emitted from a tracer located at the same position as in Fig. 7 for two extreme cases of media-wall material pairs, namely A-A and S-SG. Minimal gamma attenuation can be observed when tracer is suspended in air, which is when the efficiency is merely a function of detector distance from the tracer Fig. 8a. Here, the energy spectrum of photons collected by the detector is similar to the energy spectrum of the particles emitted by the tracer. As shown in Fig. 8b, the Compton scattering in the second case results in the generation of secondary deflected photons with lower energy signatures, which are in turn contributing to the higher overall efficiency of the detector that is the closest to the tracer (blue detector in Fig. 8).



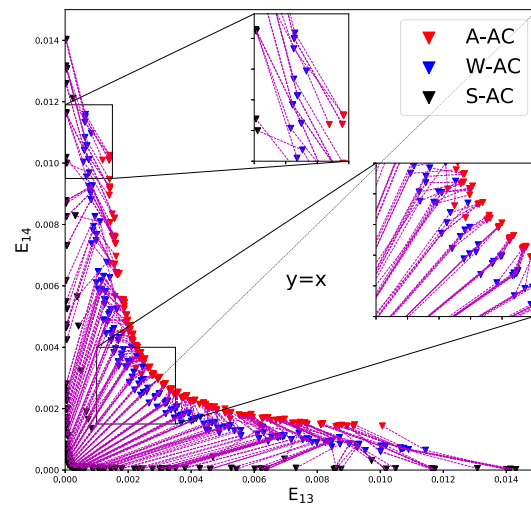
(a) A-A vs. A-AC vs. A-SG vs. A-S



(b) W-AC vs. W-SG vs. W-S

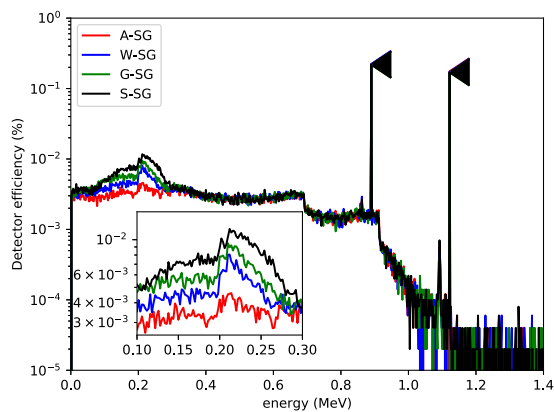


(c) A-S vs. W-S vs. G-S

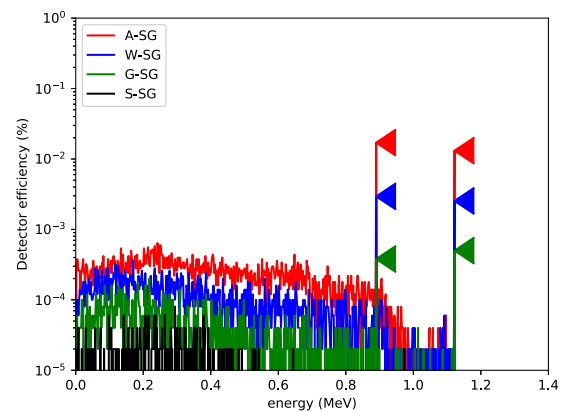


(d) A-AC vs. W-AC vs. S-AC

Fig. 6. Photon counting efficiencies of detectors 13 and 14 - each point represents detector efficiencies of the two detectors (we randomly sampled 200 from 10527 calibration points of each task). The lines are connecting the points from the same tracer locations of two specific tasks).



(a) Detector 13



(b) Detector 14

Fig. 7. Energy spectrum of the tracer near the left wall ($x = -0.112$, $y = 0.0209$, $z = 0.077$). The significant photon counts for ^{46}Sc are represented by triangles located at the peak energy levels.

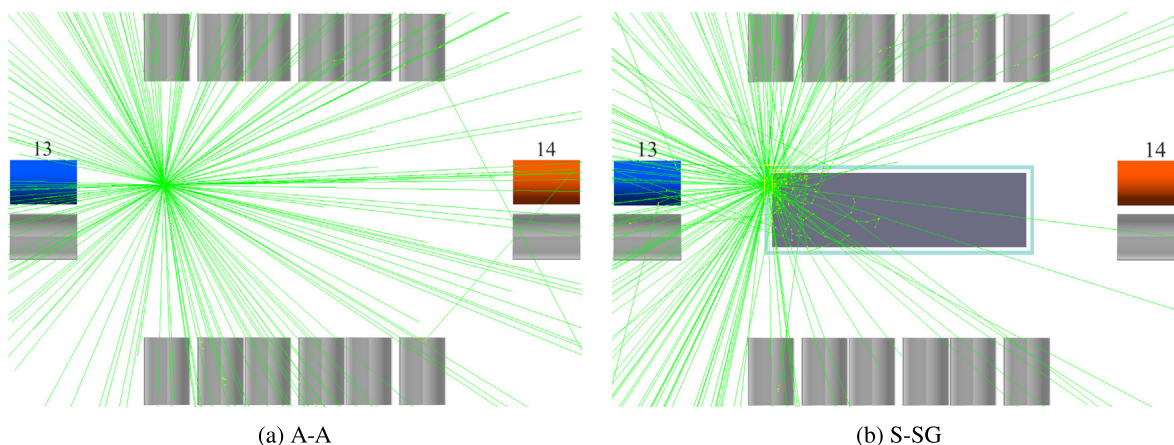


Fig. 8. Visualization of 200 photons for A-A (a) and S-SG (b) material pairs emitted from the tracer at the left wall of the reactor ($x=-0.112$, $y=0.0209$, $z=0.077$). Detector 13 is shown in blue and Detector 14 in orange.

For the same tracer location, and relative to detector 14 (red detector in Fig. 8), the photons have a higher probability to interact with the matter and be attenuated by it before they can reach the detector. This can be observed through significantly lower number of beams that are reflected toward red detector in Fig. 8b. A significant reduction in detector efficiency in this scenario is followed by the energy spectrum shift toward low energy photons (Fig. 7b). The number of high energy photons being attenuated by the media is directly proportional to the density of the media. A gradual drop of peaks over the high range of photon energies with increase in media density, until their full disappearance in S-SG case, can be observed in Fig. 7b.

In order to further demonstrate the effect of detector location (*feature*) and reactor material (*task*) on detector efficiencies constituting the training data, alpha shapes (Edelsbrunner et al., 1983) are calculated using a complete set of detector efficiency data (10527 points per feature per task), which show the spread of efficiencies for 12 considered tasks and 4 distinct detector pairs as shown in Fig. 9. The first row of plots in Fig. 9 show the alpha contours of two neighbour detectors, specifically 0 and 1, and a good correlation between them is found. This is in line with the previous observation that the efficiencies of two perfectly overlapping detectors would generate a response close to $x = y$ line.

In contrast, the alpha contours of the two most remote detectors (long diagonal), namely detector 13 and 14, which were also in focus during previous discussions, are given in the last row of Fig. 9. A distinct L-shape of detector efficiencies can be noticed again. Two additional detector pairs are chosen, where the first pair (detectors 0 and 8) represents detectors that are across of each other, and the second pair (detectors 0 and 9) detectors that are at the shorter diagonal relative to the monitored domain. In general, as the distance between the detectors is increasing their efficiencies are having more negative correlation.

3.2. Machine learning models for RPT

In this subsection we provide a comparison between SVR and neural network models for RPT. Both SVR and neural networks have a large number of hyperparameters that need to be determined through a trial and error process. We will first describe the hyperparameter search procedure. Then, we will compare accuracies of the resulting SVR and neural network models.

As mentioned in 2.2.1, to train an SVR predictor, a user needs to specify kernel function, C and ϵ . To find the right combination of the hyperparameters, we performed a grid search over a range of

choices for each hyperparameter: $kernel = ['linear', 'polynomial', 'sigmoid', 'RBF']$, $C = [1, 2, 4, \dots, 2048, 4096]$, $\epsilon = [2^{-9}, 2^{-10}, 2^{-11}, \dots, 2^{-20}]$. Kernel functions other than linear kernel also have one or more hyperparameters. For the RBF kernel, we explored kernel width hyperparameter $\gamma = [1, 2, 4, 8, \dots, 2048, 4096]$. For the polynomial kernel we explored $degrees = [1, 3, 5, 7, 9]$, and for the sigmoid kernel we explored $coef0 = [0, 0.1, 1, 10, 100, 1000]$. All other hyperparameters were the default values used in scikit-learn Python package release 0.24. To evaluate each combination of hyperparameters, we trained SVR on RPT data from A-A task because our assumption was that a set of hyperparameters that works well on A-A should also work well on other combinations of wall and medium materials.

Since the choice of hyperparameters depends on training data size, we tuned hyperparameters separately on training sizes of $N = [20, 50, 100, 500, 1000, 5000]$. In particular, we selected 2000 data points randomly from 10527 generated tracer locations in A-A task and reserved them for testing. Of the remaining 8,527 data points, we selected N randomly for training and the remaining $8,527 - N$ for validation. For each set of hyperparameters, we trained SVR on the training data and measured the accuracy on the validation data. The set of hyperparameters with the lowest MEDE on the validation data for each choice of N was selected as the best choice and used in all experiments presented in the following subsections.

The results showed that RBF kernel is superior to other kernel functions we tried. Fig. 10a shows the best combination of hyperparameters of SVR model with RBF kernel for different training sizes N . It could be seen that the hyperparameter choice is sensitive to the size of the training data set. The corresponding accuracies of the best SVR models for each training data set size N , as measured on 2000 test data points, are listed in Table 4. It could be seen that MEDE improves rapidly as the training data size increases, although the difference in accuracy between $N = 1000$ and $N = 5000$ is not very large, demonstrating that accuracy started converging over that range. In Fig. 10b we illustrate sensitivity of SVR to hyperparameters for $N = 100$ as all but one hyperparameter is fixed at its optimal values, $C = 512$, $\epsilon = 2^{-13}$, $\gamma = 256$. It could be seen that MEDE changes gradually with a change in the hyperparameters.

Using the same experimental design as for SVR, we determined the best set of hyperparameters for a neural network model. Neural networks have hyperparameters that are related to model architecture (number of hidden layers, number of neurons per layer, and neuron type) and optimization (optimization algorithm,

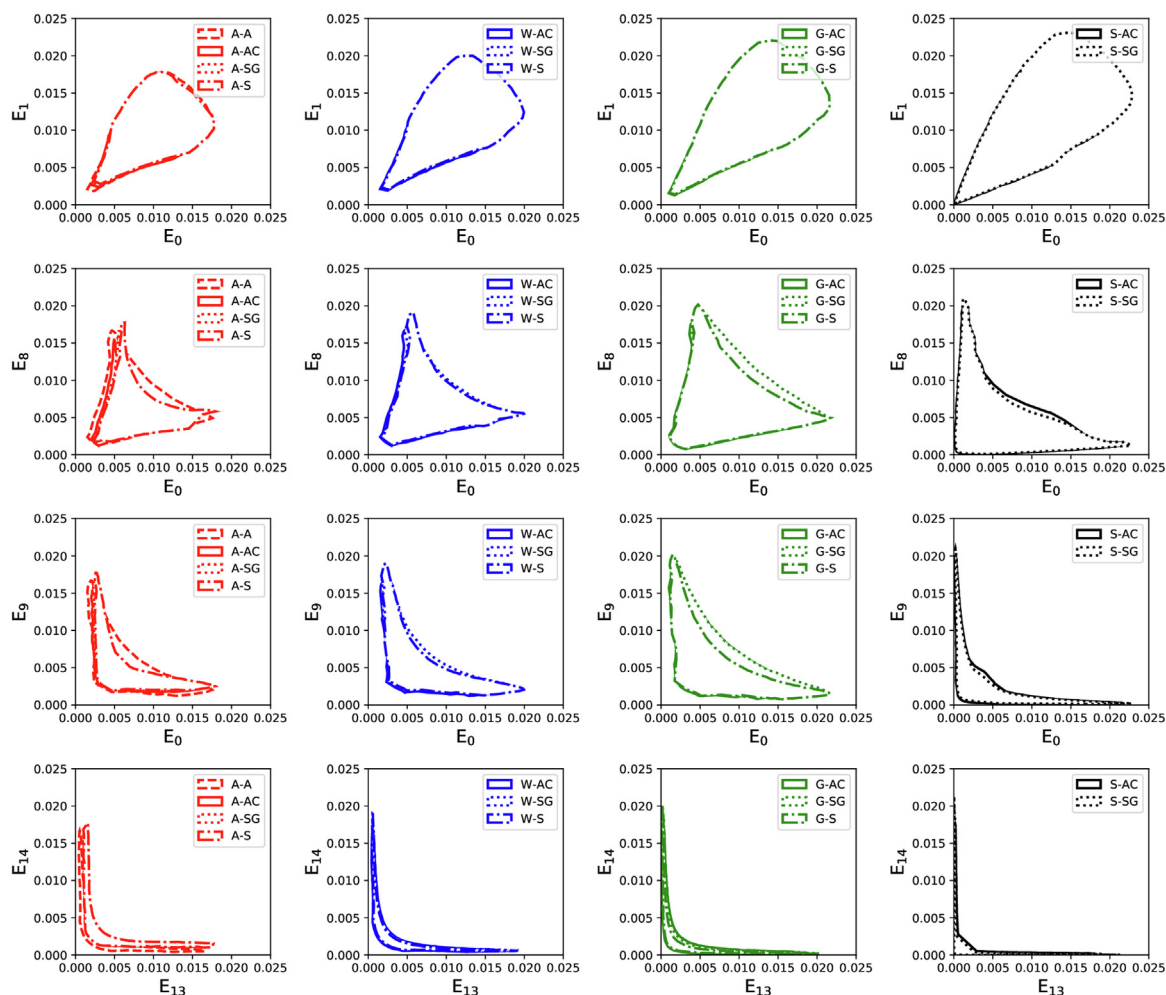
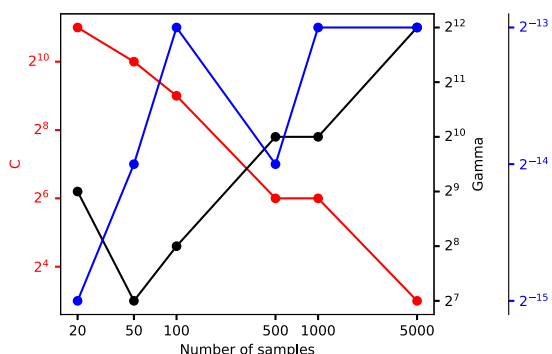
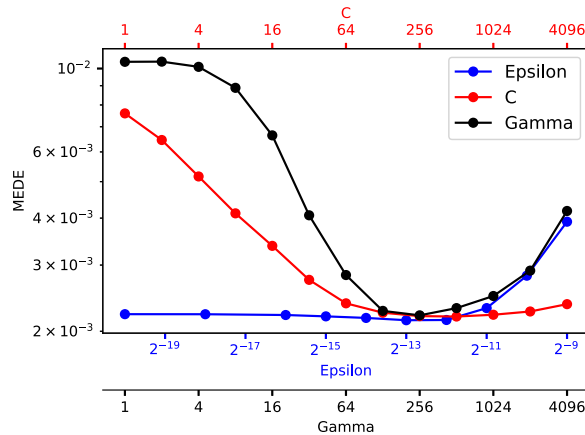


Fig. 9. Contour plots of 12 material pairs with all calibration points (detector efficiency) paired with different detectors.



(a) Best hyperparameter choice vs. sample size



(b) Sensitivity of MEDE to hyperparameter values

Fig. 10. SVR hyperparameter results. Note: Plot (b) was generated by changing one hyperparameter while keeping all other parameters at their best values for the training set size $N = 100$.

learning rate, batch size, and use of batch normalization). We fixed the number of hidden layers to 4 and the output layer had 3 neurons, one for each of the 3 dimensions for the tracer location. We explored 4 different combinations of number of neurons per hidden layer: [512, 256, 256, 256], [256, 128, 128, 128],

[128, 64, 64, 64], [64, 32, 32, 32]. Neuron type choices were 'sigmoid', 'tanh', and 'relu', optimizer choices were 'SGD', 'RMSprop', 'Adam', 'Adagrad', learning rate choices were [0.01, 0.001, 0.0001, 0.00001], and the batch size choices were [8, 16, 32, 64, 128]. All other hyperparameter choices were taken as default values in

Table 4
SVR vs. FNN model performance comparison (A-A baseline experiment)

Model	Metric	N = 20	N = 50	N = 100	N = 500	N = 1000	N = 5000
SVR	MEDE	16.8188	2.2825	1.5301	0.6250	0.5510	0.4250
	R ²	0.9515	0.9983	0.9990	0.9999	0.9999	0.9999
	RMSE	12.0154	1.9557	1.2869	0.4230	0.3883	0.2752
FNN	MEDE	84.6136	30.1503	14.1318	12.3326	2.4132	0.7701
	R ²	0.0291	0.7214	0.9134	0.9491	0.9985	0.9998
	RMSE	51.7733	19.8457	9.4635	9.6908	1.6458	0.4970

keras python package release 2.4.3, which we used for neural network training.

Table 4 lists the neural network accuracies with the best choice of hyperparameters for each training data set size N , as measured on 2000 test data points. As could be seen from the Table, SVR was consistently more accurate than neural network for every choice of N . Thus, in the remainder of this paper, we will discuss only results we obtained using the SVR model for RPT.

3.3. TL experiment

This section presents results of a comprehensive set of experiments exploring viability of TL in RPT. Results shown in Fig. 11 as matrix heatmaps summarize several types of TL experiments as explained next. Each column in the figure denotes the target task and each row in the figure denotes the source task(s). The values in the matrices correspond to MEDE accuracy in mm of the SVR models trained on one or more tasks and tested on a target task.

Before running any experiment, we reserved 2000 randomly selected data points from each task as the test data set to estimate the accuracy of RPT models. Data points from test data were not seen while training any of the SVR models. The remaining data points from each task were available for SVR training. Depending on the setup, we were selecting a subset of the remaining data points randomly to train an SVR model.

3.3.1. Baseline single-task experiments (BL)

To establish a baseline, referred to as BL in Fig. 5, diagonal entries in Fig. 11 correspond to single-task experiments where SVR is trained on data from a particular task and tested on data from the same task. For example, the top-left entry in Fig. 11a refers to accuracy on A-A task of an SVR model trained on 20 data points from A-A task. The diagonals in the four tables in Fig. 11 differ by the number of data points used to train SVR models, specifically $N = 20, 50, 100, 1000$. There are two dominant observations. First, the SVR accuracy improves as the training data size grows across all 12 tasks. Fig. 12a demonstrates this behavior more clearly, where MEDE on each task is shown for training sizes $N = 20, 50, 100, 200, 500, 1000$. Second, the tasks have different difficulty, with A-A, A-AC, W-AC, W-SG being among the easiest and with S-AC and S-SG being the most difficult.

3.3.2. Paired-Task experiments (PT)

In the paired-task experiments, denoted as PT in Fig. 5, an SVR model is trained on data from one task (e.g., A-A) and tested on data from another task (e.g., S-AC). Those results are shown in off-diagonal entries in Fig. 11. In the machine learning literature, this experiment corresponds to the *zero-shot* TL (Wang et al., 2019) because no training data points from the target task are available for RPT model training. *Zero-shot* TL is relevant to practical scenarios where an RPT model needs to be used under a novel experimental condition, where it is unfeasible to collect any calibration data.

As could be seen from Fig. 11, with a few exceptions, PT SVR models result in significantly lower accuracy than BL models. The

contrast is particularly stark between the two tasks with the stainless steel media (S-AC and S-SG) and other 10 tasks. At the other extreme, accuracy of model trained on W-AC and tested on A-A, and vice versa, results in almost identical accuracy to the BL model, indicating that RPT calibration data between those two tasks are very similar and that TL between those two tasks is very appropriate. The PT results are consistent with contours shown in Fig. 9 where pairs of tasks that have similar contours also result in better transfer learning.

Another interesting observation is the impact of source task training data size. As can be seen in Figs. 11a and 11d, for each of the 12 target tasks there is a PT model trained on 1000 data points from a source task that has higher accuracy than a BL model trained on 20 data points from the target task. For example, SVR trained on 1000 data points from A-A task has MEDE = 4 mm on W-SG test data, while SVR trained on 20 data points from W-SG task has MEDE = 7.4 mm on the same test data. This result strongly indicates that TL is a promising strategy for RPT modelling in scenarios where it is unfeasible to collect calibration data under a new experimental condition.

3.3.3. Leave-one-Out experiments (LO)

In this group of experiments, denoted as LO in Fig. 5, we were training SVR models on data from all tasks, excluding data from the target task. The first entry in the LO row in Fig. 11a refers to SVR models trained on a data set that contains $N = 20$ data points from every task excluding A-A task (220 training data points in total) and tested on A-A task. Similar to the PT scenario, this setup corresponds to the *zero-shot* TL scenario because it does not need any training data points from the target task.

The results in Fig. 11 show that the accuracy of any of the PT models for all choices of $N = 20, 50, 100, 1000$ and for all target tasks is better than its corresponding LO accuracy, excluding S-AC and S-SG. For S-AC target task, SVR trained on S-SG task is only slightly more accurate than the LO model. The same holds for SVR trained on S-AC task and tested on S-SG task. Those results are a strong indicator that combining all available calibration data from multiple simulated conditions might be overall superior to hand-picking calibration data from any single simulated condition.

It is interesting to observe that for $N = 20$ LO achieves higher accuracy than BL on all but the S-AC target task. This is an impressive result showing that *zero-shot* TL is superior to BL when the number of available calibration data points for the target task is small. For $N = 50$, LO is most often superior to BL, while for $N = 100$ and $N = 1000$ it is very competitive to BL. In fact, LO with $N = 1000$ is most often superior to training SVR on $N = 100$ data points from the target task.

3.3.4. All-Included experiments (AI)

In this group of experiments, denoted as AI in Fig. 5, we were training SVR models on data from all tasks, including data from the target task. The first entry in the AI row in Fig. 11a refers to SVR models trained on a data set that contains $N = 20$ data points from every task (240 training data points in total) and tested on A-A task. Unlike PT and LO scenarios, this setup corresponds to the

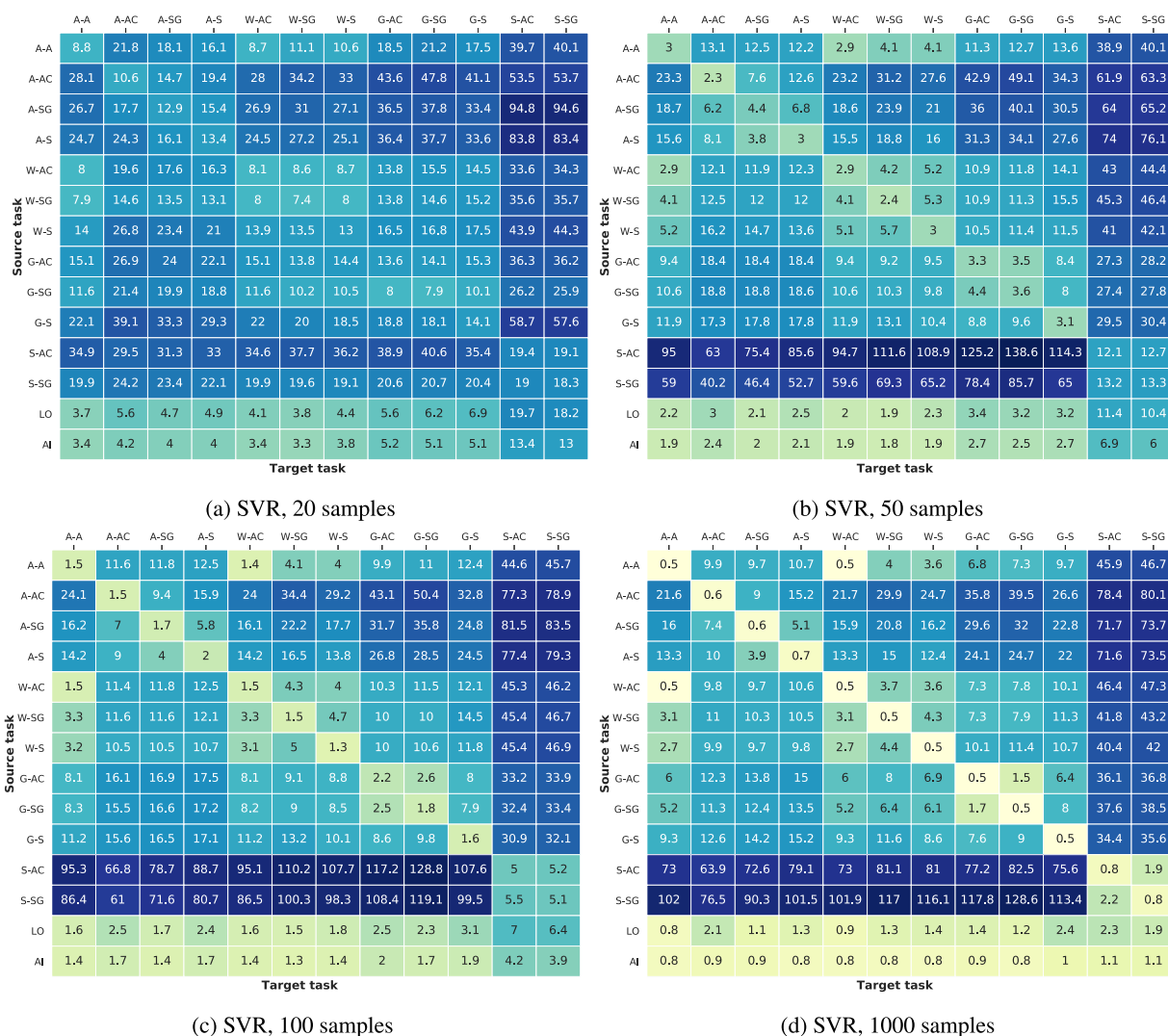


Fig. 11. MEDE of SVR TL models considering 12 different material pairs (tasks). MEDE results are shown for BL, PT, LO, and AI experiments using 20, 50, 100 and 1000 sample size.

few shot TL scenario because it combines a small number of target task data points with a larger number of data points from source tasks. The hope is that the source task data can help boost the accuracy compared to the BL model trained only on the target task data.

The results in Fig. 11 show that AI is superior to LO, demonstrating that adding a small number of training data points from the target task to the source task data can substantially improve the accuracy. For $N = 20$ and $N = 50$, AI is also superior to the BL (single-task) model across all 12 tasks, indicating that calibration data from other simulated conditions can be very helpful when only a small number of calibration data points are available. For $N = 100$, AI is more accurate than the BL models on all target tasks excluding A-AC, W-S, and G-S. However, for $N = 1000$, the BL models are more accurate than the AI model across all 12 tasks. Thus, if the calibration data are plentiful for a particular simulated condition, the results indicate that adding data from other simulated conditions is not productive.

3.3.5. Grouped-Source experiments (GS)

Based on the results shown in Fig. 11, the accuracy of PT models trained on calibration data from one simulated condition (source task) and tested on data from a different condition (target task) highly depends on the degree of similarity between these condi-

tions. For example, in the case where the target task is with steel media (S-AC or S-SG tasks; the last two columns in the matrices in Fig. 11), selection of any source task with media density different than steel results in the significant reduction of the PT model accuracy. In order to further explore the impact of task similarity, we performed Leave-Group-Out (LG) experiments. As shown in Fig. 13, we created 10 groups of tasks based on the media and wall material types. For example, the first group listed contains S-AC and S-SG tasks. Each column in the figure corresponds to the accuracy on the listed task. Rows BL and LO are the same as BL and LO results in Fig. 11.c.

Row LG in the Fig. 13 corresponds to MEDE of the SVR model trained on all tasks but the selected group and testing on the tasks from the selected group. Thus, LG corresponds to the zero-shot TL. Value 24 in the first slot of the LG row in Fig. 13.a is MEDE of an SVR model trained on 100 training data points from all but S-AC and S-SG tasks (for a total of 1000 training data points) and tested on data points from S-AC. It is interesting to observe that the corresponding MEDE of 7 for the LO model is significantly lower. The only difference between LO and LG models is that training data for the LO model also included 100 training data points from the S-SG task. This strongly indicates that the success of TL in RPT is very beneficial for at least one of the simulated conditions from the

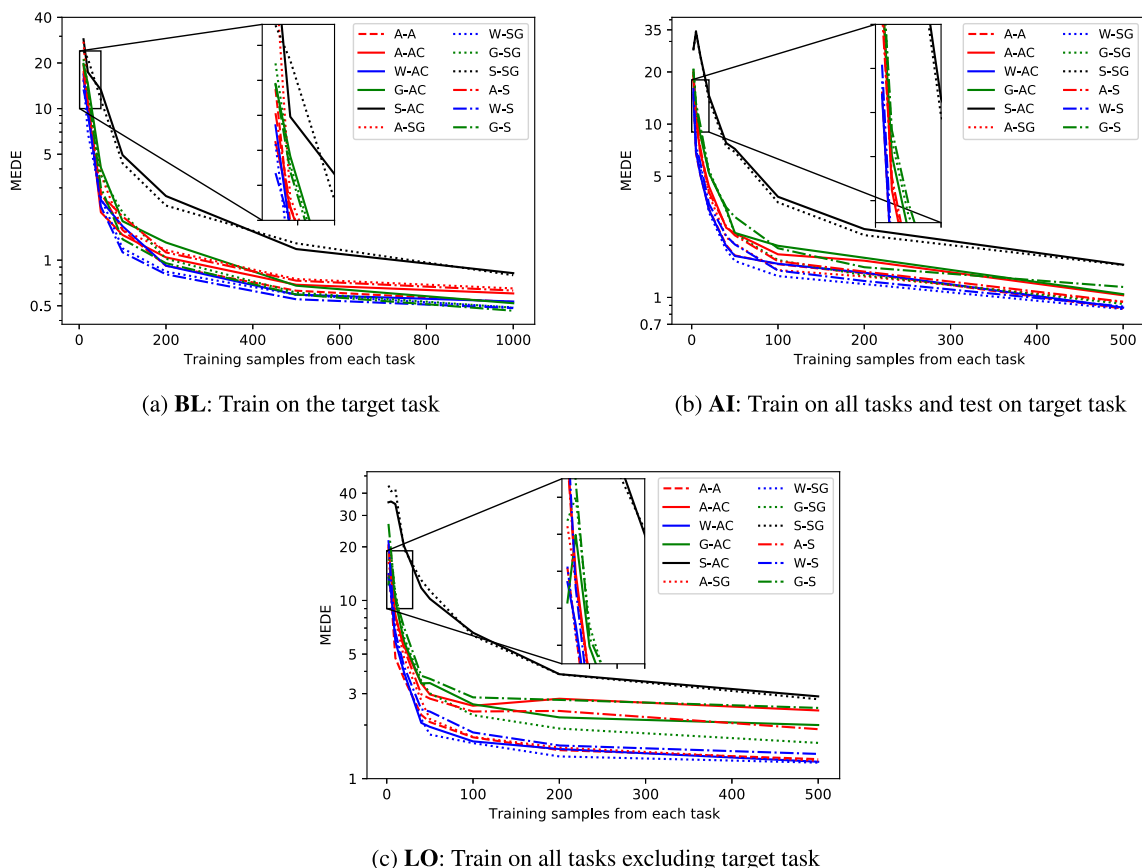


Fig. 12. SVR model MEDE prediction error as a function of training data size.

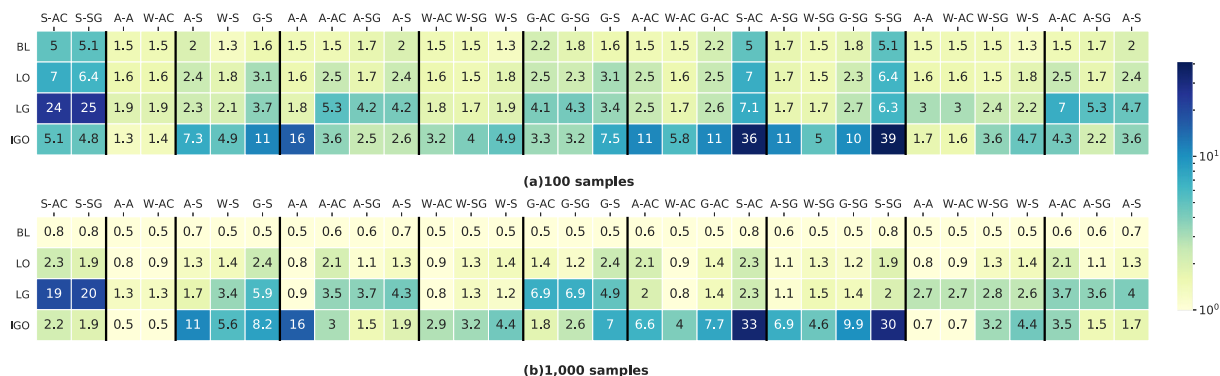


Fig. 13. MEDE of SVR TL models for LG and IGO experiments trained on (a) 100 and (b) 1000 randomly selected samples (BL and LO are shown for comparisons).

source tasks to be similar to the conditions from the target task. Fig. 13 shows that the difference in accuracy between LO and LG models is particularly large for {S-AC, S-SG}, {A-A, A-AC, A-SG, A-S}, {G-AC, G-SG, G-S}, and {A-AC, A-SG, A-S} groups. This indicates that the simulated conditions among tasks in those groups are substantially different than the conditions in the remaining tasks.

Row IGO in Fig. 13 is the accuracy of the SVR model trained on data points from tasks in the listed group and tested on a tasks from the same group. Thus, IGO corresponds to the few-shot TL. For example, value 5.1 from the first slot in of the IGO row in Fig. 13.a is MEDE of the SVR model trained on 100 data points from tasks {S-AC, S-SG} (for a total of 200 training data points) and

tested on data points from S-AC. As could be seen from Fig. 13.a, IGO model accuracy is sometimes lower and sometimes higher than the BL model. This is another indication that similarity between source and target tasks is a critical factor in determining the success of TL. Results in Fig. 13.b show that BL is consistently more accurate than IGO, confirming that if calibration data are plentiful for a particular target task it is counterproductive to use data points from other tasks to train the SVR model.

The comparison between LG and IGO experiments provide more insights into the effect of data distribution on TL performance. As for a particular target task, if its LG error is higher than its IGO error (for example, S-AC in the group {S-AC, S-SG}), then the target task

shares more similarities with tasks from the same group compared to tasks from other groups. On the other hand, if the LG error of a particular target task is lower than the IGO error (for example, A-S in the group {A-S, W-S, G-S}), then the target task distribution is distinct from other tasks in its group. Overall, we can infer that the similar tasks are: {S-AC, S-SG}, {A-A, W-AC}, {A-AC, A-SG, A-S}, and {G-AC, G-SG} and this result is consistent with similar contours of these groups shown in Fig. 9. This important finding gives us intuition about which source tasks to choose under zero-shot scenario.

4. Conclusions

In this paper, we show for the first time that it is possible to reduce the errors and costs associated with the current RPT practices by applying the TL approach based on the availability of historical RPT calibration data. The approach has been proven particularly effective in scenarios where it is unfeasible or challenging to collect sufficient calibration data for an unseen/new task. Using an in-house Geant4 powered software, GIPPE-RPT, we generated high-resolution calibration data sets (10527 tracer locations in the calibration grid) with a total of 12 unique wall and filling media combinations to perform RPT model benchmarking. Two machine learning techniques, SVR and FNN, and several distinct TL strategies were tested, and their effectiveness in exploiting historical calibration data was evaluated. Our empirical evaluation demonstrated that the historical calibration data collected under various simulated conditions can be exploited through TL when training an RPT model under a new condition. The historical data are particularly useful for the *zero-shot* scenario where it is not possible to obtain calibration data for the new condition and for the *few-shot* scenario where it is possible to obtain only a limited number of new calibration data points. Also, our results show that TL is more appropriate when historical calibration data correspond to conditions similar to the current simulated condition. A similar condition, in this application, relates to the physical properties of the attenuation material of the monitored system, i.e. materials of the reactor wall and media that were the focus of this study. If sufficient calibration data for a new condition is available, the inclusion of historical data via TL is not recommended, as less accuracy was observed in reconstructing the tracer's position. We demonstrated that the TL approach is a viable solution to improve RPT accuracy on systems that are hard to calibrate, but are similar to previously calibrated systems.

CRedit authorship contribution statement

Guilherme Lindner: Methodology, Investigation, Validation, Writing – original draft, Software. **Sai Shi:** Methodology, Investigation, Validation, Writing – original draft, Software. **Slobodan Vučetić:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Sanja Mišković:** Conceptualization, Methodology, Supervision, Writing – review & editing.

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.

Acknowledgements

This research was made possible with the help of the University of British Columbia and Temple University for the fellowship and

Compute Canada for the computational resources used in the simulations.

References

- Akça, Burcu, Erzeneoglu, Salih Zeki, 2016. Measurement of linear attenuation coefficients of compounds of some essential major elements. *J. Multidiscipl. Eng. Sci. Technol.* 3 (6).
- Chen, Jinwen, Rados, Novica, Al-Dahhan, Muthanna H., Dudukovic, Milorad P., Nguyen, Duyen, Parimi, Krishniah, 2001. Particle motion in packed/ebullated beds by CT and CARPT. *AIChE J.* 47 (5), 994–1004.
- Cortes, Corinna, Vapnik, Vladimir, 1995. Support-vector networks. *Mach. Learn.* 20 (3), 273–297.
- de Freitas Dam, Roos Sophia, Carvalho dos Santos, Marcelo, Santana Moreira do Desterro, Filipe, Luna Salgado, William, Schirru, Roberto, Marques Salgado, César, 2021. A novel radioactive particle tracking algorithm based on deep rectifier neural network. *Nucl. Eng. Technol.* 53 (7), 2334–2340.
- Devanathan, Narasimhan, 1991. Investigation of liquid hydrodynamics in bubble columns via a computer automated radioactive particle tracking (CARPT) facility (Ph.D.). Washington University in St. Louis, United States – Missouri.
- Devanathan, N., Moslemian, D., Dudukovic, M.P., 1990. Flow mapping in bubble columns using CARPT. *Chem. Eng. Sci.* 45 (8), 2285–2291.
- Dreyfus, Stuart E., 1990. Artificial neural networks, back propagation, and the kelly-bryson gradient procedure. *J. Guidance Control Dyn.* 13 (5), 926–928.
- Drucker, Harris, Burges, Chris J.C., Kaufman, Linda, Smola, Alex, Vapnik, Vladimir, 1997. Support vector regression machines. *Adv. Neural Inform. Process. Syst.* 9, 155–161.
- Edelsbrunner, H., Kirkpatrick, D., Seidel, R., 1983. On the shape of a set of points in the plane. *IEEE Trans. Inf. Theory* 29 (4), 551–559.
- Gopalan, Balaji, Shaffer, Franklin, 2012. A new method for decomposition of high speed particle image velocimetry data. *Powder Technol.* 220, 164–171.
- IAEA, 2008. Industrial Process Gamma Tomography.
- Ioffe, Sergey, Szegedy, Christian, 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift.
- Khane, Vaibhav, Al-Dahhan, Muthanna H., 2017. Hybrid dynamic radioactive particle tracking (RPT) calibration technique for multiphase flow systems. *Meas. Sci. Technol.* 28 (5), 055904.
- Khane, Vaibhav, Said, I.A., Al-Dahhan, Muthanna H., 2016. Experimental investigation of pebble flow dynamics using radioactive particle tracking technique in a scaled-down Pebble Bed Modular Reactor (PBM). *Nucl. Eng. Des.* 302, 1–11.
- Kunze, Julius, Kirsch, Louis, Kurenkov, Ilia, Krug, Andreas, Johannsmeier, Jens, Stober, Sebastian, 2017. Transfer learning for speech recognition on a budget.
- Larachi, Faïçal, Kennedy, Gregory, Chaouki, Jamal, 1994. A gamma-ray detection system for 3-D particle tracking in multiphase reactors. *Nucl. Instrum. Methods Phys. Res., Sect. A* 338 (2), 568–576.
- Liashchynskiy, Petro, Liashchynskiy, Pavlo, 2019. Grid search, random search, genetic algorithm: A big comparison for nas. *ArXiv, abs/1912.06059*.
- Li, Weijun, Sai, Gu., Zhang, Xiangping, Chen, Tao, 2020. Transfer learning for process fault diagnosis: Knowledge transfer from simulation to physical processes. *Comput. Chem. Eng.* 139, 106904.
- Lin, J.S., Chen, M.M., Chao, B.T., 1985. A novel radioactive particle tracking facility for measurement of solids motion in gas fluidized beds. *AIChE J.* 31 (3), 465–473.
- McAlister, Daniel R., 2012. Gamma ray attenuation properties of common shielding materials. Technical report. PG Research Foundation.
- NIST Office of Data and Informatics. NIST Chemistry WebBook. Publisher: National Institute of Standards and Technology.
- Pan, S.J., Yang, Q., 2010. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* 22 (10), 1345–1359.
- Perrault, M., Bertrand, F., Chaouki, J., 2010. An investigation of magnesium stearate mixing in a v-blender through gamma-ray detection. *Powder Technol.* 200 (3), 234–245.
- Rademakers, Fons, Canal, Philippe, Naumann, Axel, Couet, Olivier, Moneta, Lorenzo, Vassilev, Vassil, Linev, Sergey, Piparo, Danilo, Ganis, Gerardo, Bellenot, Bertrand, Guiraud, Enrico, Amadio, Guilherme, Wverkerke, Mato, Pere, TimurP, Tadel, Matevž, Wlav, Tejedor, Enric, Blomer, Jakob, Gheata, Andrei, Hageboeck, Stephan, Roiser, Stefan, Marsupial, Wunsch, Stefan, Shadura, Oksana, Bose, Anirudha, CristinaCristescu, Valls, Xavier, Iseman, Raphael, 2019. root-project/root: v6.18/02.
- Rammohan, A.R., Kemoun, A., Al-Dahhan, M.H., Dudukovic, M.P., 2000. A Lagrangian description of flows in stirred tanks via computer-automated radioactive particle tracking (CARPT). *Chem. Eng. Sci.* 56 (8), 2629–2639.
- Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65 (6).
- Srivastava, Nitish, Hinton, Geoffrey, Krizhevsky, Alex, Sutskever, Ilya, Salakhutdinov, Ruslan, 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15 (56), 1929–1958.
- Wang, Wei, Zheng, Vincent W., Yu, Han, Miao, Chunyan, 2019. A survey of zero-shot learning: Settings, methods, and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* 10 (2), 1–37.
- Wang, Yaqing, Yao, Quanming, Kwok, James T., Ni, Lionel M., 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv. (CSUR)* 53 (3), 1–34.

Xian, Y., Schiele, B., Akata, Z., 2017. Zero-shot learning – the good, the bad and the ugly. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3077–3086.

Yadav, Ashutosh, Ramteke, Manojkumar, Pant, Harish Jagat, Roy, Shantanu, 2017. Monte Carlo real coded genetic algorithm (MC-RGA) for radioactive particle tracking (RPT) experimentation. *AIChE J.* 63 (7), 2850–2863.

Ashutosh Yadav, Harish Jagat Pant, and Shantanu Roy. Velocity measurements in convective boiling flow using Radioactive Particle Tracking (RPT) technique. *AIChE Journal*, 0(ja):e16782, 2019.

Yadav, Ashutosh, Gaurav, Tuntun Kumar, Pant, Harish J., Roy, Shantanu, 2020. Machine learning based position-rendering algorithms for radioactive particle tracking experimentation. *AIChE J.* 66 (6), e16954.