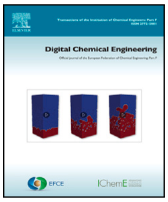
Digital Chemical Engineering 17 (2025) 100272

Contents lists available at ScienceDirect

Digital Chemical Engineering

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



Original article

Optimization-based framework for kernel parameter identification in multi-material population balance models

Haoran Ji, Lena Fuhrmann, Juan Fernando Meza Gonzalez, Frank Rhein *

Karlsruhe Institute of Technology (KIT), Institute of Mechanical Process Engineering and Mechanics (MVM), Strasse am Forum 8, Karlsruhe, 76131, Germany

ARTICLE INFO

Dataset link: github.com/pdhs-group/PSD_opt

Keywords:
Kernel parameter optimization
Population balance equation
Agglomeration
Breakage
Sensitivity analysis
Distributed computing framework

ABSTRACT

This study presents a robust, parallelized optimization framework for kernel parameter identification that is adaptable to any population balance equation (PBE) formulation and process type. The framework addresses the challenge of incomplete 2D particle size distribution (PSD) measurements in multi-material systems by combining a reduced 2D PSD with complementary 1D datasets. The framework was validated by using noisy synthetic PSD data and evaluating both the error in PSD and kernel values across eight kernel parameters. Hyperparameter and sensitivity analyses provided configuration recommendations and insights into the influence of individual parameters, thus guiding kernel model selection. Incorporating prior knowledge of one kernel parameter (e.g., through multi-scale simulations) mitigated non-unique solutions and enhanced noise tolerance, ultimately improving the framework's robustness and reliability. A case study based on experimental data from a dispersion process demonstrated the framework's flexibility and practical relevance.

1. Introduction

Population balance equations (PBEs) are widely used to model the dynamic evolution of particle size distributions (PSDs) in complex disperse systems encountered in many industrial processes. Accurate control of the PSD is critical for tailoring product properties to meet specific application requirements. For example, in the preparation of lithium-ion battery electrodes using carbon black, the PSD directly influences the electrode's conductivity and overall performance. A well-optimized PSD ensures a uniform distribution of active materials and conductive additives, leading to improved battery efficiency (Asylbekov et al., 2023). Among the various processes that can affect PSD dynamics, particle agglomeration and breakage are especially prominent. The accuracy of the PBE in representing such processes depends strongly on the underlying kernels, which quantify agglomeration rates, breakage rates, and fragment distributions.

Consequently, various approaches have been explored to determine the kernels. For agglomeration kernels, Jeldres et al. (2018) extensively studied methods based on mechanistic principles, such as the DLVO theory, and reviewed both classical and modern adaptations for diverse agglomeration mechanisms. Similar mechanistic approaches have also been employed in biological and nanoscale systems, notably in the modeling of viral agglomeration within biological media (Zhang et al., 2022), and in the identification of agglomeration regimes in charged nanoparticle suspensions under varying ionic strengths (Atmuri et al., 2013). Patruno et al. (2009) analyzed a range of empirical and mechanistic descriptions to capture the fragmentation dynamics of particle systems, emphasizing the importance of tuning these models to specific experimental conditions. Building on these foundational reviews, traditional methods for kernel determination often rely on theoretical frameworks combined with experimental validation. For example, Bemer (1979) developed a comprehensive framework for understanding the kinetics of agglomeration in suspensions, focusing on batch systems. His work introduced a method for calculating agglomeration rates by utilizing backscattered light to monitor particle growth in real-time. These experimental results were then combined with a population balance model (PBM) that accounted for both coalescence and breakage mechanisms. In this model, the coalescence and breakage rates were modeled as functions of particle size and experimental variables such as stirring speed and binder concentration. Chi and Sommerfeld (2002) employed a kinetic theory approach to simulate fictitious collision partners, evaluating the likelihood of particle agglomeration by comparing the normal relative velocity of colliding particles against a critical velocity threshold. Similarly, Zheng et al. (2019) calculated the agglomeration kernel by taking a root mean square of multiple agglomeration mechanisms, including orthokinetic, hydrodynamic, and Brownian mechanisms.

While these examples of more "traditional" methods have laid a solid foundation, recent advances focus on leveraging optimization

* Corresponding author.
E-mail address: frank.rhein@kit.edu (F. Rhein).

https://doi.org/10.1016/j.dche.2025.100272
Received 28 July 2025; Received in revised form 22 September 2025; Accepted 21 October 2025
Available online 27 October 2025

techniques and machine learning to determine kernels with greater precision and adaptability. Capece et al. (2011) proposed a back-calculation method based on non-linear optimization to determine breakage kernels, fitting experimental data to a non-linear PBM. Nielsen et al. (2020) introduced a hybrid machine learning framework trained with online and at-line sensor data, leveraging real-time sensor measurements to predict dynamic parameters such as agglomeration and breakage rates. These predictions were then integrated into the population balance equation (PBE) to continuously model particle behavior and kinetics in real time. Wang et al. (2022) determined the breakage probability by considering a unified breakage criterion that integrates various impact modes and evaluates the fraction loss of a particle under stressing events. Their breakage selection function was obtained by fitting experimental data using nonlinear regression within a PBM framework. Raponi and Marchisio (2024) employed a deep-learning approach to obtain all kernels in the PBE. Their method involved training a neural network, referred to as the "mirror model", with a synthetic dataset generated from PBE simulations. Once trained, the mirror model took as input the experimental particle size indicators along with the magnesium ion concentration, and then directly predicted the eight kinetic parameters without relying on iterative optimization.

While previous studies have developed various approaches for determining kernels, there remains a gap in the systematic evaluation and optimization of the methods themselves: Mostly, kernel values are presented together with the resulting deviations between experiment and model without additional details on the optimization procedure. To address this, this work develops a novel optimization framework for kernel determination and thoroughly evaluates its performance, adaptability, and computational efficiency, providing a solid foundation for further improvement and application. At the core of this framework is a dPBE solver, which iteratively solves multi-dimensional PBE during each optimization step to refine kernel estimation and ensure an accurate fit to the experimental data. This framework takes the time-evolving particle size distribution PSD_0 as input and outputs optimized kernel combinations that best fit the observed data. It is capable of handling PSD data involving two distinct materials, allowing for the extraction of kernels that describe their interactions. Since "true" kernel values are not measurable directly, the accuracy of the framework was validated by using synthetic data generated from predefined and therefore known kernel values. Additionally, a hyperparameter study identified the influence of factors such as the number of iterations, sampling algorithms, and optimization criteria. The performance of the framework was evaluated using both the error in PSD and in obtained kernels as key metrics. Their correlation was analyzed and surfaced interesting relationships. Through this parameter study, the framework itself was optimized, identifying well-performing parameter settings that serve as a reference for future use. By leveraging a Ray-based distributed computing framework, the computational efficiency was significantly enhanced, laying the groundwork for future large-scale kernel optimization and database development. In the final part of this work, a cross-validation was performed on experimental data from a battery paste production, i.e. dispersion, process employing a mixture of graphite, carbon black, and binder. The results highlight the framework's practical robustness and its strong tolerance to errors in raw experimental data.

2. Methodology

2.1. Population balance equation

The population balance equation serves as a fundamental tool for modeling the dynamic behavior of particulate systems, where the distribution and evolution of particle properties over time are of primary interest. In this study, we consider a spatially homogeneous system, assuming uniform physical properties and particle distribution throughout the domain. Consequently, spatial coordinates are omitted from the equation, significantly simplifying the discretization and computation. The time-dependent partial differential equation captures the agglomeration and breakage processes within the system. The equation takes the form of

$$\frac{\partial n(x,y,t)}{\partial t} = B_{\text{agg}}(x,y,t) - D_{\text{agg}}(x,y,t) + B_{\text{break}}(x,y,t) - D_{\text{break}}(x,y,t) \quad , \tag{1}$$

where the four terms represent the birth and death of agglomerates due to agglomeration and breakage, as determined by Eq. (2) through (5):

$$B_{\text{agg}}(x,y,t) = \frac{1}{2}\int_0^x\int_0^y r_{\text{A}}(x-x', y-y', x', y')\,n(x-x', y-y', t)\,n(x', y', t)\,dx'\,dy' \tag{2}$$

$$D_{\text{agg}}(x,y,t) = \int_0^\infty\int_0^\infty r_{\text{A}}(x,y,x',y')\,n(x,y,t)\,n(x',y',t)\,dx'\,dy' \tag{3}$$

$$B_{\text{break}}(x,y,t) = \int_x^\infty\int_y^\infty r_{\text{B}}(x',y')\,f_{\text{B}}(x,y,x',y')\,n(x',y',t)\,dx'\,dy' \tag{4}$$

$$D_{\text{break}}(x,y,t) = r_{\text{B}}(x,y)\,n(x,y,t) \tag{5}$$

Here, n denotes the number-density function that characterizes the distribution of agglomerates in terms of their internal composition. x and y are the internal variables denoting the partial volumes of two distinct materials within an agglomerate. The parameter r_{A} represents the agglomeration kernel. This kernel quantifies the rate of two particles merging to form a new agglomerate. The breakage rate kernel r_{B} represents the rate of a particle of size (x,y) undergoing breakage per unit time. Meanwhile, f_{B} is the breakage distribution function, which describes the probability distribution of a parent particle of size (x,y) breaking into smaller fragments of size (x',y').

2.1.1. Solution of the population balance equations

In general, Eq. (1) cannot be solved analytically without imposing specific constraints, so it is typically discretized and solved numerically. However, direct discretization only yields accurate results when the initial particle volume distribution in the system is monodisperse and a uniform grid is used. In systems where the particle volume varies widely, this approach demands an exceedingly large number of grid points, making the computation highly time-consuming and inefficient. This study therefore employs a geometric grid, where the node coordinates are defined as follows:

$$x_i = Sx_{i-1} \tag{6}$$

$$y_j = Sy_{j-1} \tag{7}$$

Here, i, j represents the node index, and S is a parameter used to adjust the grid spacing. By selecting an appropriate value for S, the geometric grid can cover a wide range of volumes with a significantly lower amount of nodes (N_S) compared to a uniform grid. Nevertheless, to minimize errors introduced by the non-uniform node spacing, the geometric grid requires the use of the cell average technique (CAT) as developed in Kumar et al. (2008). The CAT improves accuracy by averaging particle properties within each grid cell and redistributing them based on a mass-conserving weighting scheme.

2.1.2. Agglomeration and breakage kernels

In many previous studies, the agglomeration kernels have been mathematically characterized as the product of two distinct factors

$$r_{\text{A},abij} = \alpha_{abij}\beta_{abij} \quad , \tag{8}$$

where α_{abij} and β_{abij} represent the collision efficiency and collision frequency, respectively. The indices ab and ij denote the two agglomerates involved in the collision. For collision frequency, several formulations exist depending on the specific scenario. In this work, we employed the general shear-induced collision model:

$$\beta_{abij} = \beta_{\text{corr}} G\left(R_{ab} + R_{ij}\right)^3 \tag{9}$$

Here, β_{corr} serves as a correction factor that accounts for deviations between idealized model assumptions and complex real-world physical conditions, such as non-spherical agglomerate shapes and non-ideal shear flow profiles, which would otherwise require a detailed resolved numerical simulation (see e.g. Trunk et al. (2018)). For example, Chin et al. (1998) used a value of 2.3, while Ruan et al. (2022) adopted a value of $\frac{1}{6}$. In addition, since the current PBE model neglects spatial inhomogeneity, all physical parameters (such as G) are represented by their mean values over the computational domain ($\overline{G}$), and β_{corr} can further compensate for inaccuracies introduced by this averaging.

For the collision efficiency, we adopted the collision case model proposed in our previous work (Rhein et al., 2019). This model estimates the collision efficiency based on the material properties and partial volumes within the colliding agglomerates. The collision case model simplifies collisions between hetero-agglomerates into material-to-material interactions, making it highly suitable for multi-dimensional PBEs. In the 2D case, there are three possible material contact scenarios, so α_{abij} can be expressed as a vector $\boldsymbol{\alpha}_{\mathrm{corr}}$ containing three elements. Specifically, $\alpha_{\mathrm{corr},11}$ and $\alpha_{\mathrm{corr},22}$ represent the collision efficiencies of each material with itself, while $\alpha_{\mathrm{corr},12}$ denotes the collision efficiency between the two different materials. Although mechanistic equations, e.g. developed from the DLVO theory, exist, unavailable material data and unsuitable assumptions result in $\boldsymbol{\alpha}_{\mathrm{corr}}$ being empirical factors that need calibration to the investigated system. Since the collision frequency and collision efficiency are always used as a product in the PBE, the current 2D agglomeration model has three kernel parameters that require identification:

$$\mathbf{k}_{\mathrm{corr}} = \boldsymbol{\alpha}_{\mathrm{corr}}\beta_{\mathrm{corr}} = \begin{pmatrix} \alpha_{\mathrm{corr},11} \\ \alpha_{\mathrm{corr},12} \\ \alpha_{\mathrm{corr},22} \end{pmatrix} \beta_{\mathrm{corr}} \tag{10}$$

The breakage rate is treated in a manner similar to the collision frequency. In this study, we utilized the adapted two-component Power Law model proposed by Pandya and Spielman (1983), which is particularly suitable for capturing the effects of shear rate and particle volume on the breakage process.

$$r_{\mathrm{B},ij} = P_1 G \left(\frac{x_i}{x_{\mathrm{mean}}}\right)^{P_2} + P_3 G \left(\frac{y_j}{y_{\mathrm{mean}}}\right)^{P_4} \tag{11}$$

The parameters P_1 and P_3 in Eq. (11) act as correction factors for the shear rate and adjust the weighting of different materials' influence on the breakage rate. P_2 and P_4 describe how particle volume affects the breakage rate. These parameters reflect both the intrinsic mechanical strength of the materials and the energy input to agglomerate breakage from the shear flow. x_{mean} and y_{mean} are weighted particle volumes derived from the initial particle distribution, used to adjust the scale of the equations. Their impact on the calculation can be effectively captured by parameters P_1 and P_3. For simplicity, both x_{mean} and y_{mean} are set to $1000\,\mu\mathrm{m}^3$.

The breakage function must be carefully designed to ensure computational stability and mass conservation, maintaining a physically consistent fragment distribution and preserving the total mass before and after breakage. Given these considerations, we adopted a two-dimensional extension of the Parabolic distribution function by Diemer and Olson (2002).

$$f_{B,ijkl} = \frac{\theta_{ijkl}}{x_i y_j} \tag{12}$$

$$\theta_{ijkl} = (\upsilon + 2)(\upsilon + 1)\, z_{ijkl}^{(\upsilon-1)} \left(1 - z_{ijkl}\right) \frac{\upsilon}{2} \tag{13}$$

$$z_{ijkl} = \frac{x_k y_l}{x_i y_j} \tag{14}$$

The parameter υ within Eq. (13) is typically not considered a material-specific parameter, but is associated with the particular breaking mechanism. υ governs the trend of the fragment distribution, determining whether the breakage process favors the production of larger or smaller fragments. Moreover, υ also influences the number of fragments generated from a single breakage event. Note that the selected kernel models and PBE formulation serve primarily as a demonstrative basis for the optimization framework. These choices are not the focus of this work and can be readily replaced with alternative kernels or solution strategies, depending on the specific application. Although different kernel models will likely alter the dynamics of the optimization procedure, the results of this study still provide general insights into the optimization procedure itself. Considering experimental data, kernel models must of course allow for a good representation of the true behavior for the optimization to be successful.

2.2. *Optimization framework*

Fig. 1 illustrates the external structure of the general workflow of the optimization framework, with the dashed box indicating the internal workings of the optimizer, which are detailed in Fig. 4. The framework begins with a time-evolving particle size distribution PSD_0 as input, which is derived from experiments or simulations. Through iterative optimization, the algorithm explores various kernel combinations $\mathrm{k}'_{\mathrm{opt}}$ to generate a predicted $\mathrm{PSD}'_{\mathrm{opt}}$. The difference between $\mathrm{PSD}'_{\mathrm{opt}}$ and PSD_0, referred to as PSD_δ, guides the optimizer in selecting subsequent kernel combinations. After reaching the specified number of iterations, the optimal kernels $\mathrm{k}_{\mathrm{opt}}$, associated with the minimum PSD_δ, are considered approximations of the true kernels underlying the original data. To validate the performance of the optimization framework, this study generated PSDs from a dPBE model with known kernels k_0. Before use, these PSDs were smoothed and noise was added to better match real-world experimental data. The difference between the optimized kernels $\mathrm{k}_{\mathrm{opt}}$ and the original kernels k_0, denoted as k_δ, served as one of the performance metrics for the optimization framework.

The optimization framework supports both 1D and 2D PSD data as input. Due to the higher number of kernels involved in the 2D PBE equation, a single set of 2D PSD data may not provide sufficient information for the optimization process to converge effectively. This is particularly true when the kernels exhibit symmetry, as in Eq. (11). In such cases, the optimization framework may mistakenly swap the kernels for two different materials, when identical grid spacing is used for both materials. Another limitation is that in real-world experiments, obtaining a true 2D PSD is nearly impossible, as it would require measuring the partial volumes of each material within every agglomerate. Typically, only the total particle volume can be measured, resulting in a reduced form of the 2D PSD that lacks the detailed, material-specific information, as illustrated in Fig. 2. To address these challenges, we proposed to expand the data set as depicted in Fig. 3: by using the initial kernels for each material separately in a 1D dPBE model, two additional 1D PSDs were generated. These 1D PSDs, along with the reduced 2D PSD, were fed into the optimization framework to simultaneously determine the kernels that describe the interactions within each material and between the two materials. In practical experiments, these two 1D PSDs were obtained by testing each material separately under the same experimental conditions.

Given the high computational cost associated with solving the dPBE equation for the substantial number of iterations required, this work implemented a parallel optimization algorithm based on Liaw et al. (2018). Ray is an open-source distributed computing framework, and Ray Tune is a Python library within Ray specifically designed for hyperparameter tuning. In the optimization process, as shown in Fig. 4, each iteration is referred to as a "trial". A trial involves using a candidate set of kernels $\mathrm{k}'_{\mathrm{opt}}$ in the dPBE to compute the predicted $\mathrm{PSD}'_{\mathrm{opt}}$, and then evaluating the corresponding PSD_δ. An optimization task for a set of PSD data is managed by a Tuner, which executes multiple trials in parallel. For algorithms that require statistical analysis of existing results to guide subsequent testing, such as Gaussian process-based Bayesian optimization, the Tuner dynamically or periodically collects results to update its statistical models. Ray also supports the simultaneous execution of multiple Tuners, allowing the optimization framework to handle multiple datasets concurrently. Within the entire

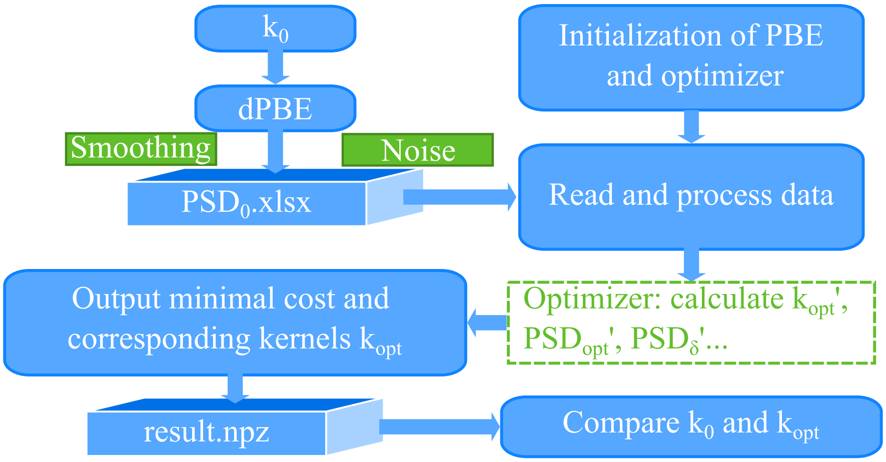

k0
dPBE
Smoothing
Noise
PSD0.xlsx
Initialization of PBE and optimizer
Read and process data
Optimizer: calculate kopt', PSDopt', PSDδ'...
Output minimal cost and corresponding kernels kopt
result.npz
Compare k0 and kopt

Fig. 1. Workflow of the optimization framework with synthetic data.

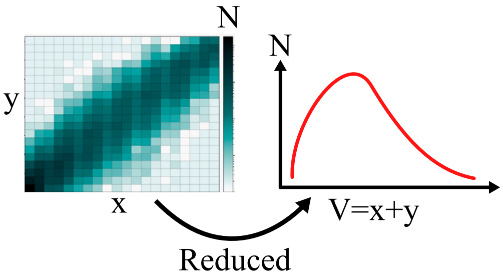


Fig. 2. Due to experimental limitations, the material distribution within individual agglomerates cannot be measured, and only the total particle volume distribution is obtained, resulting in an information reduction of 2D PSDs.

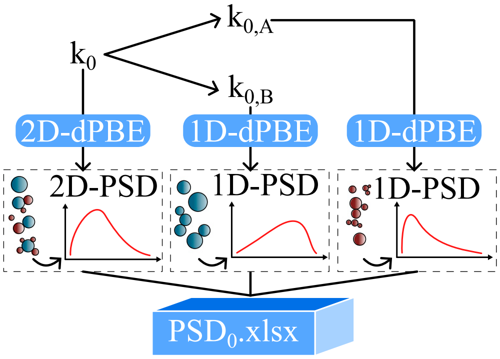


Fig. 3. Multi-data input strategy: A reduced 2D PSD with two additional, material-specific 1D PSDs serve as input data PSD_0.

framework, nearly all computational time is consumed by solving the dPBE equations. The overhead from the optimizer itself—including parallel communication and algorithm-related computations—is negligible by comparison. A detailed discussion based on specific values is provided in the SI.

Since "true" kernel values cannot be determined experimentally, we used simulation-generated quasi-experimental data to validate the framework. To comprehensively evaluate the optimization framework's effectiveness in identifying kernels under different conditions, we adopted a full factorial design of all possible combinations of the eight kernel parameters to generate the original PSD data set. However, due to computational constraints, only two values were sampled for each parameter. The sampling range was chosen empirically based on two criteria: First, the dPBE must remain numerically stable, and second, the PSD must evolve meaningfully within a defined time frame. The resulting sampling values for each parameter are listed in Table 1. It is important to note that the value ranges of kernel parameters are determined by several factors, including the set process time. If the PSD reaches a steady state early in the process, e.g. due to large kernel values, a longer process time generates a large amount of stationary data, which in turn dilutes the meaningful (dynamic) data. Since this negatively impacts the optimization process, such parameter sets were considered unsuitable for testing under the current simulation duration. The optimizer's search bounds were defined slightly broader than the sampling ranges, as shown in Table 2.

Table 1
The value ranges of kernels during generation of the synthetic dataset.

Parameter	Sampling values
$\mathbf{k}_{\mathrm{corr}}$ [0], $\mathbf{k}_{\mathrm{corr}}$ [1], $\mathbf{k}_{\mathrm{corr}}$ [2]	$[10^{-3}, 10^{-1}]$
P_1, P_3	$[10^{-4}, 10^{-2}]$
P_2, P_4	$[0.5, 2.0]$
υ	$[1.0, 1.5]$

Table 2
The value ranges of kernels during optimization (search bounds).

Parameter	Lower bound	Upper bound
$\mathbf{k}_{\mathrm{corr}}$ [0], $\mathbf{k}_{\mathrm{corr}}$ [1], $\mathbf{k}_{\mathrm{corr}}$ [1]	10^{-4}	10^{0}
P_1, P_3	10^{-5}	10^{-1}
P_2, P_4	0.3	3.0
υ	0.5	2.0

The original data were first smoothed using a kernel density estimation algorithm and then noise was introduced in a multiplicative form

$$\mathrm{PSD}'_{V,t} = \mathrm{PSD}_{V,t}(1 + \delta_{\mathrm{PSD}}) \tag{15}$$

to simulate measurement errors typically encountered in experimental data. The value of δ_{PSD} follows a normal distribution with a mean of 0 and a standard deviation of 0.1. $\mathrm{PSD}_{V,t}$ is either a 1D PSD or the reduced 2D PSD obtained from the dPBE, while V refers to a specific particle size at a particular point in time t. All process- and grid-specific parameters were held constant throughout this study at the values presented in Table 3.

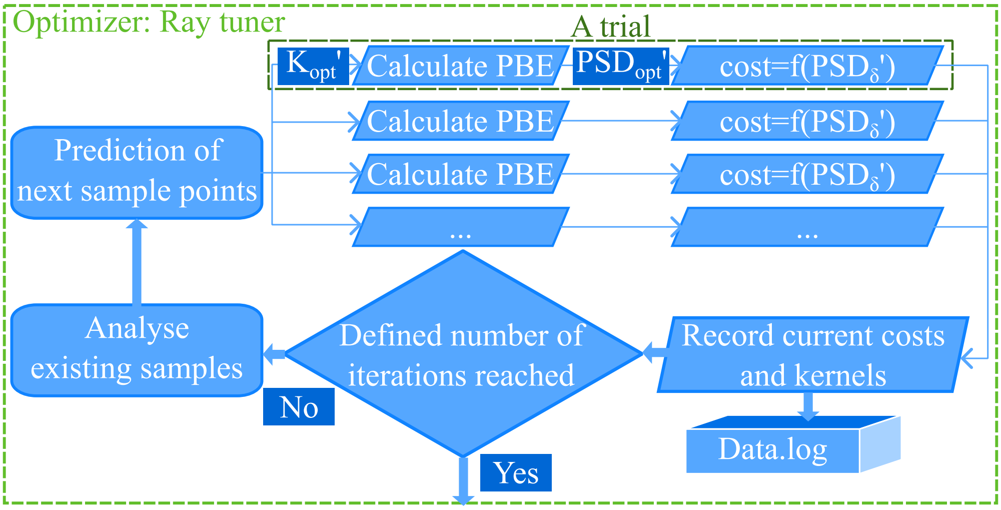


Fig. 4. Optimization process with Ray tune.

Table 3
Process- and grid-specific parameters used throughout this study.

Symbol	Definition [Unit]	Setting
N_S	Node count in each direction for dPBE grid [-]	10
S	Scaling factor for the spacing between nodes in a non-uniform grid [-]	4
$R_{0,x}$	Minimum particle radius in the x-direction [μm]	1.1
$R_{0,y}$	Minimum particle radius in the y-direction [μm]	1.1
t_{total}	physical process time of PBE [s]	600

PSD_δ serves as the optimization target. However, a PSD can be described in various ways, e.g. using the differential particle size distribution by volume q_3, the cumulative particle size distribution by volume Q_3, or the median particle size x_{50}. To facilitate a comparison, PSD_δ was characterized by the mean squared error (MSE) of q_3 between the optimized PSD and the original PSD, and was scaled accordingly. The scaled error $\mathrm{MSE}_{\mathrm{q}3}$ is defined as follows:

$$\mathrm{MSE}^*_{\mathrm{q}3} = \frac{1}{PT}\sum_{p=1}^{P}\sum_{\mathrm{t}=1}^{T}\left(q^*_{3,p,t,0} - q_{3,p,t,0}\right)^2 \tag{16}$$

$$\mathrm{MSE}_{\mathrm{q}3} = \frac{\mathrm{MSE}'_{\mathrm{q}3}}{\mathrm{MSE}^*_{\mathrm{q}3}} \tag{17}$$

Here, P denotes the total number of discretization points in the system, and T represents the number of time points considered in the analysis. $\mathrm{MSE}'_{\mathrm{q}3}$ follows the same form of definition as $\mathrm{MSE}^*_{\mathrm{q}3}$ but denotes the PSD error corresponding to the test kernel parameters during the optimization process. $q^*_{3,0}$ refers to the PSD obtained by directly inputting k_0 into the dPBE solver. Therefore, the scale factor $\mathrm{MSE}^*_{\mathrm{q}3}$ represents the error solely due to noise and smoothing effects, and also signifies the theoretical minimum of $\mathrm{MSE}'_{\mathrm{q}3}$.

Similarly, k_δ undergoes a scaling process to ensure comparability between the errors of different kernels, which may vary in magnitude and dimension. The scaling is based on the formula:

$$k^*_m = \max\left(\mathrm{K}_{m,0}\right) - \min\left(\mathrm{K}_{m,0}\right) \tag{18}$$

$$k_\delta = \frac{1}{M}\sum_{m=1}^{M}\frac{k_{m,\mathrm{opt}} - k_{m,0}}{k^*_m} \tag{19}$$

Here, m denotes the different kernels being optimized, and M represents the total number of kernel parameters. In this formula, $\mathrm{K}_{\mathrm{m},0}$ denotes the set of values taken by a specific kernel in all experiments, and k^* represents the difference between the maximum and minimum values within the set $\mathrm{K}_{\mathrm{m},0}$. In this work, five sets of PSD data with added noise were generated for each parameter setting to simulate repeated experimental measurements. In the subsequent analysis of the parameter study results, the overall performance was assessed using the average kernel error, $\overline{k}_\delta$, and the average PSD error, $\overline{\mathrm{MSE}}_{\mathrm{q}3}$, across all PSD datasets.

3. Theoretical study based on synthetic data

3.1. Convergence trends of $\overline{MSE}_{q3}$ and $\overline{k}_\delta$

To assess the fundamental performance of the optimization framework and evaluate whether it can identify the correct kernels, i.e. if $\overline{k}_\delta$ converges to a minimum, we first examined its behavior across different iteration counts. The number of iterations represents the distinct kernel combinations tested by the optimization framework. In Fig. 5(a), the plot illustrates changes in $\overline{\mathrm{MSE}}_{\mathrm{q}3}$ and $\overline{k}_\delta$ as the number of iterations increases from 50 to 3200, with the standard error of the results also marked. The red dashed line represents $\overline{\mathrm{MSE}}_{\mathrm{q}3} = 1$, the theoretical minimum of $\overline{\mathrm{MSE}}_{\mathrm{q}3}$. As the iteration count increases, we observed that $\overline{\mathrm{MSE}}_{\mathrm{q}3}$ gradually decreased, approaching this theoretical minimum, while $\overline{k}_\delta$ was also consistently reduced. This trend confirms the effectiveness of the optimization framework in refining kernel estimates over multiple iterations. Interestingly, while the $\overline{\mathrm{MSE}}_{\mathrm{q}3}$ had already approached its theoretical minimum around 800 iterations and remained nearly constant thereafter, a substantial reduction in $\overline{k}_\delta$ was still observed between 800 and 1600 iterations. This highlights the risk of relying solely on $\overline{\mathrm{MSE}}_{\mathrm{q}3}$ as a stopping criterion, as doing so could lead to premature termination of the optimization. On the other hand, once $\overline{\mathrm{MSE}}_{\mathrm{q}3}$ stabilizes at its theoretical minimum, $\overline{k}_\delta$ also levels off at a non-zero value and ceases to improve further. This behavior can be attributed to the complexity of optimizing in high-dimensional parameter spaces. In such cases, it is often possible for different combinations of kernel parameters to produce very similar results in terms of predicted PSD. Since the breakage rate formula exhibits clear symmetry, as shown in Eq. (11), it is highly likely that multiple sets of values for these four parameters can produce the same breakage rate. This phenomenon can prevent the optimizer from ever reaching the exact original parameters,

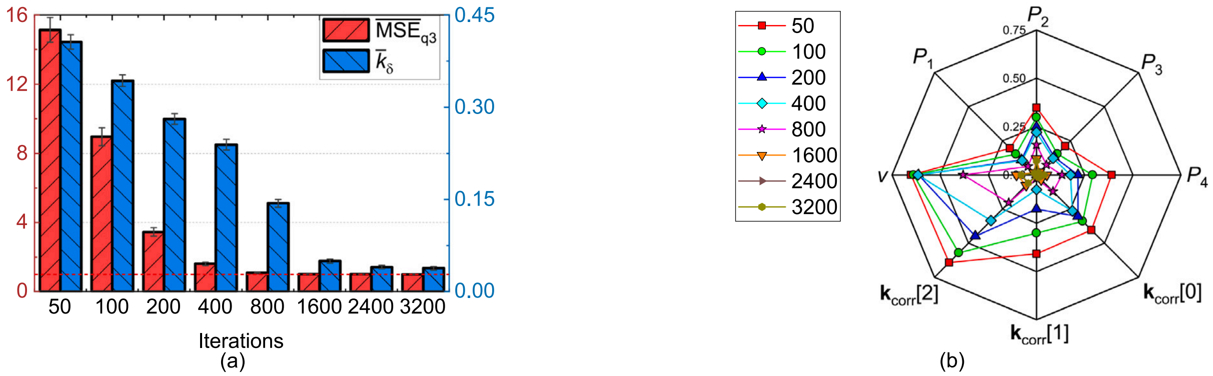


Fig. 5. Optimization results with CMA-ES sampler across different iteration counts. (a) Trend of $\overline{\text{MSE}}_{\text{q3}}$ and $\bar{k}_{\delta}$ as the number of iterations increases. (b) Error evolution of individual kernel parameters with increasing iterations.

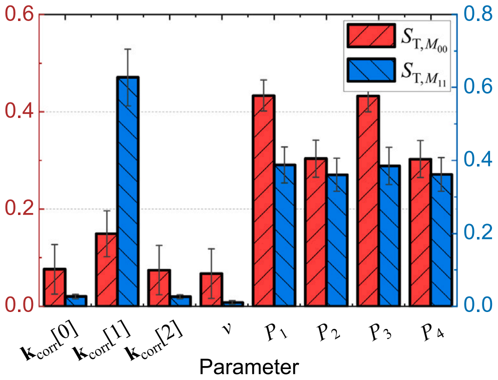


Fig. 6. Variance-based sensitivity analysis.

as multiple solutions appear equally valid. A potential solution to address the issue of non-uniqueness will be discussed in subsequent paragraphs.

Fig. 5(b) displays the optimization results for individual kernel parameters as the number of iterations increases, showing their relative errors as defined similarly to Eq. (19). It is evident that despite minor fluctuations, the errors for all kernel parameters generally decreased as the iteration count increased, indicating an overall improvement in accuracy with additional iterations. Two aspects merit further analysis: First, because a full factorial design was used and identical parameters, such as particle diameter and initial distribution, were applied for materials A and B in the dPBE, the optimization results for $\mathbf{k}_{\text{corr}}$ [2] and $\mathbf{k}_{\text{corr}}$ [0] should ideally be very similar. The observed asymmetry between these parameters, however, results from the fixed random seed. In the CMA-ES algorithm, the random seed influences the initialization process by determining the initial selection of kernel parameters to test and generating the first probabilistic model. The current seed led to an initial misjudgment for $\mathbf{k}_{\text{corr}}$ [2], which required a high number of iterations to correct. This highlighted that the initial guess can impact results at lower iteration counts, as was verified by further testing (see SI). At higher iteration counts, this influence diminished. Second, even when disregarding the effect of the random seed, the convergence rates among different kernel parameters remained uneven. The agglomeration-related parameters exhibited greater errors at lower iteration counts compared to the breakage-related parameters, with $\mathbf{k}_{\text{corr}}$ [2] and $\mathbf{k}_{\text{corr}}$ [0] converging more slowly than $\mathbf{k}_{\text{corr}}$ [1]. The parameter v showed consistently slow convergence, with a final relative error noticeably higher than that of other parameters.

3.2. Global sensitivity analysis of kernel parameters

The varying convergence speeds of different parameters in the optimization process can be analyzed in terms of each parameter's "importance" to the overall outcome, which is captured through sensitivity analysis. The Sobol' method (Sobol', 2001) – a variance-based approach – was used to decompose output variance and quantify each input parameter's contribution. The total-effect index S_{T} was used as the sensitivity metric, as it captures not only the direct impact of each parameter but also its interactions with other parameters. Specifically, S_{T} represents the fraction of the output variance that can be attributed to the parameter in question, including all higher-order interaction effects involving that parameter. A larger value of S_{T} indicates greater overall influence on the model output. Samples for the Sobol' sensitivity analysis were generated using Saltelli's extension of the Sobol' sequence, a quasi-random low-discrepancy method that ensures more uniform coverage of the input space than purely random sampling. The base sample size was empirically set to $N = 2^{17} = 131{,}072$, which balances convergence accuracy and computational cost. Given eight input parameters, this resulted in a total of $N \times (2D + 2) = 2{,}359{,}296$ samples, following the standard Saltelli formulation. The sampling space was defined by the parameter bounds listed in Table 2. Two metrics were used as outputs for the Sobol' method: the zeroth moment M_{00} and the first cross moment M_{11} of the 2D PSD. The zeroth moment M_{00} reflects the total particle count and captures the rate of the process. The first cross moment M_{11} represents the distribution of both materials within the agglomerates and serves as an indicator of PSD changes. Fig. 6 shows that for both moments, the Total-effect index S_{T} of $\mathbf{k}_{\text{corr}}[1]$, P_1, P_2, P_3, and P_4 were significantly higher than those of $\mathbf{k}_{\text{corr}}[0]$, $\mathbf{k}_{\text{corr}}[2]$, and v. Notably, the sensitivity of M_{11} to $\mathbf{k}_{\text{corr}}[1]$ is especially pronounced, indicating that the PSD dynamics are most responsive to changes in the hetero-agglomeration rate. This is consistent with the rapid convergence of this parameter during optimization. Among all parameters, v consistently showed the lowest S_{T}, particularly for M_{11}, indicating minimal influence on PSD dynamics. This is in line with its observed slow convergence in the optimization process, and further underscores the challenge of accurately identifying parameters with low sensitivity. In contrast, while the breakage rate kernel parameters exhibited high total-effect indices, their convergence rates were not clearly superior to those of the agglomeration rate kernel parameters. This may be attributed to the structure of the data: the optimization utilized a combined 2D and 1D dataset, whereas the sensitivity analysis was conducted on the 2D PBE alone. Because the collision case model provides a physically motivated link between 1D and 2D agglomeration, information from the 1D dataset can be reliably transferred to the 2D agglomeration parameters. However, the breakage kernel model

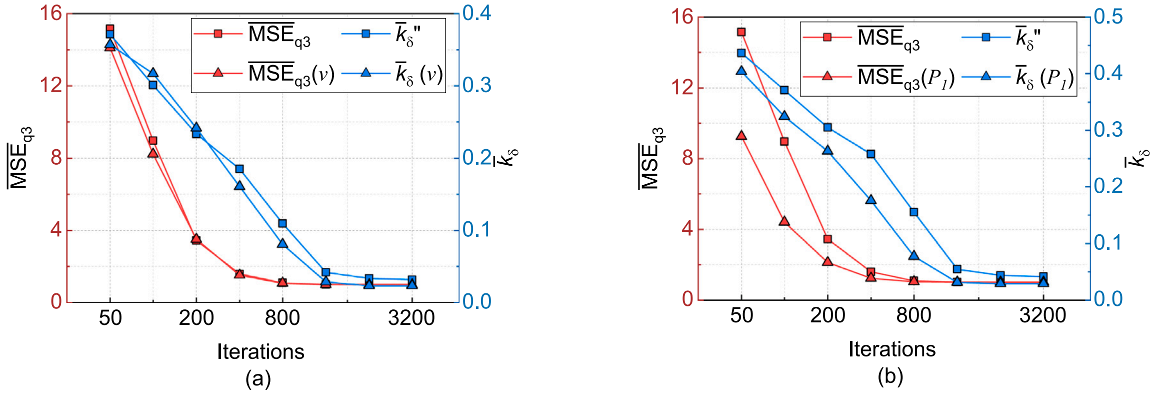


Fig. 7. Comparison of optimization results with CMA-ES sampler between general cases and known kernel parameter v (left) or P_1 (right).

used here is based on a straightforward extension of the power-law form, and it lacks direct mechanistic justification. Consequently, the support provided by the 1D data for breakage parameter optimization may be limited, and despite their high sensitivity, the breakage parameters may not benefit from the same convergence advantage as the agglomeration parameters. This issue could be addressed by adopting a more physically reliable multi-material breakage rate model.

3.3. Case study: Impact of known parameter on optimization

One possible approach to address the issues posed by low-impact parameters and non-unique solutions is to optimize or streamline the kernel models before the main optimization process, reducing the number of parameters and ensuring that each remaining parameter has a more consistent impact on the results. However, this requires a high level of model refinement and presents its own set of complexities. Another approach is to directly obtain values for certain parameters, e.g. through particle breakage experiments or simulation methods such as DEM. These parameter values are then considered known within the optimization framework, thereby reducing the model's degrees of freedom. To test this effect, several kernel parameters were tested for pre-determination. v and P_1 were selected as representative cases due to their distinct impacts and the results for the other parameters are given in the SI. For each case, the respective kernel parameter was set to its true value while optimizing for the remaining ones. To ensure a meaningful comparison, the reference $\overline{k}_\delta$ was recalculated in each case excluding the known parameter and denoted as $\overline{k}_\delta$". As shown in Fig. 7, after 800 iterations, the values of $\overline{\text{MSE}}_{q3}$ were nearly identical with or without predetermined parameters. In contrast, $\overline{k}_\delta$ showed consistent improvement when a parameter was known in advance. For v, a parameter with minimal impact on PSD, fixing its value helped the optimizer avoid expending effort on matching an inherently insensitive and hard-to-fit parameter, improving optimization efficiency. In the case of P_1, pre-determination resolved the symmetry issue in the breakage kernel function discussed earlier, leading to clearer convergence. These results suggest that selecting parameters for pre-estimation should be guided by whether they introduce difficulties that the optimizer cannot resolve on its own. In practice, however, low-impact parameters like v are more feasible choices, as they can be roughly estimated without compromising overall accuracy.

3.4. Comparison of sampling strategies in optimization

To further enhance the performance and robustness of the optimization framework, a comprehensive hyperparameter study was conducted. For ease of comparison, the CMA-ES (Covariance Matrix Adaptation Evolutionary Strategy (Hansen, 2023)) sampler paired with an MSE optimization objective was used as the baseline. Figs. 8(a) and 8(b) provide an analysis of various sampling algorithms used in the optimization framework, evaluating their effectiveness in minimizing $\overline{\text{MSE}}_{q3}$ and $\overline{k}_\delta$ across increasing iteration counts. To ensure the reproducibility of the test results, the random seed for all samplers was fixed at 1. As shown in Fig. 8(a), all samplers demonstrated the capability to reduce the objective function, although with varying convergence speeds. Among them, CMA-ES achieved the fastest reduction in $\overline{\text{MSE}}_{q3}$, reaching near-optimal values within 1600 iterations. The GP(traditional Gaussian Process Bayesian Optimization), TPE(Tree-structured Parzen Estimator (Watanabe, 2023)), and NSGA(Nondominated Sorting Genetic Algorithm (Deb and Jain, 2013; Jain and Deb, 2013)) samplers showed similar performance, all steadily decreasing the objective, albeit at a slower pace than CMA-ES. In contrast, QMC exhibited the slowest convergence across all tested iterations. Fig. 8(b) reveals more pronounced differences in terms of kernel parameter recovery. CMA-ES outperformed other methods significantly in reducing $\overline{k}_\delta$, particularly after 400 iterations. As $\overline{\text{MSE}}_{q3}$ approached its theoretical minimum, CMA-ES continued to drive $\overline{k}_\delta$ sharply downward, indicating its effectiveness in accurately identifying kernel parameters even in late-stage optimization. Conversely, QMC showed little improvement in $\overline{k}_\delta$ throughout the optimization process. This suggests that QMC's highly stochastic nature makes it less suitable for high-dimensional optimization tasks that require precise and consistent parameter tuning, especially when operating under limited iteration budgets.

3.5. Effect of objective function choice on optimization accuracy

The calculation of PSD_δ depends on the representation of the PSD. Additionally, different cost functions such as MSE, RMSE, and MAE are commonly used, along with the Kullback–Leibler Divergence (KL), which measures the divergence between two probability distributions. To quantify these influences, various cost functions were employed during optimization. Since iteration count had a minimal impact on the relative differences in optimization results across various targets, only the results at 800 iterations were shown in Fig. 9 for clarity. As shown in Fig. 9, using q_3 and Q_3 as optimization targets yielded significantly better results compared to x_{50}. This is to be expected since x_{50} is a single numerical value and only offers very limited information. In certain cases, different PSDs can have the same median volume diameter, making the solution non-unique and potentially misleading. For the cost functions applied to both q_3 and Q_3 targets, the performances of MSE, RMSE, and MAE were very similar. This can be attributed to the fact that both q_3 and Q_3 represent full distribution information, making them inherently robust against small deviations in individual particle sizes. The results further indicated that using KL divergence with q_3 as the target could yield slightly better results, though the improvement was not pronounced. KL divergence is particularly effective in distinguishing between probability distributions by focusing on

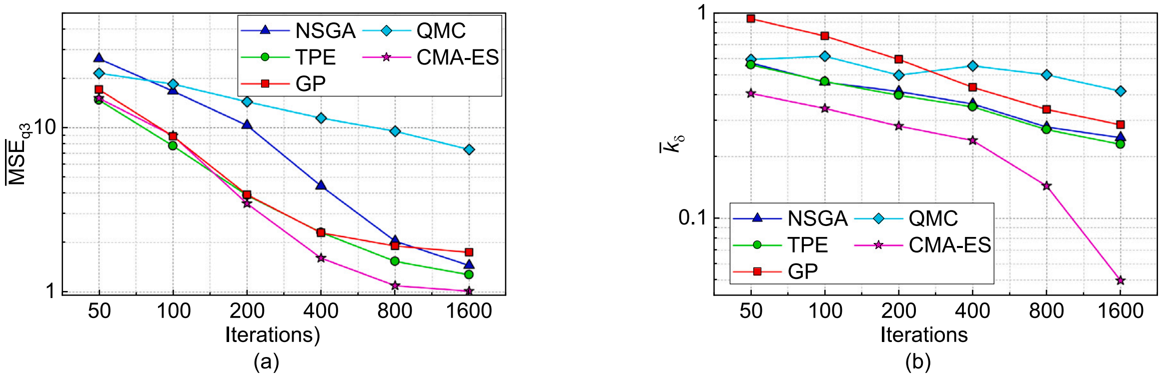


Fig. 8. Comparison of optimization results using different samplers. (a) Variation of $\overline{\text{MSE}}_{q3}$ with the number of iterations for different samplers. (b) Variation of $\overline{k}_\delta$ with the number of iterations for different samplers.

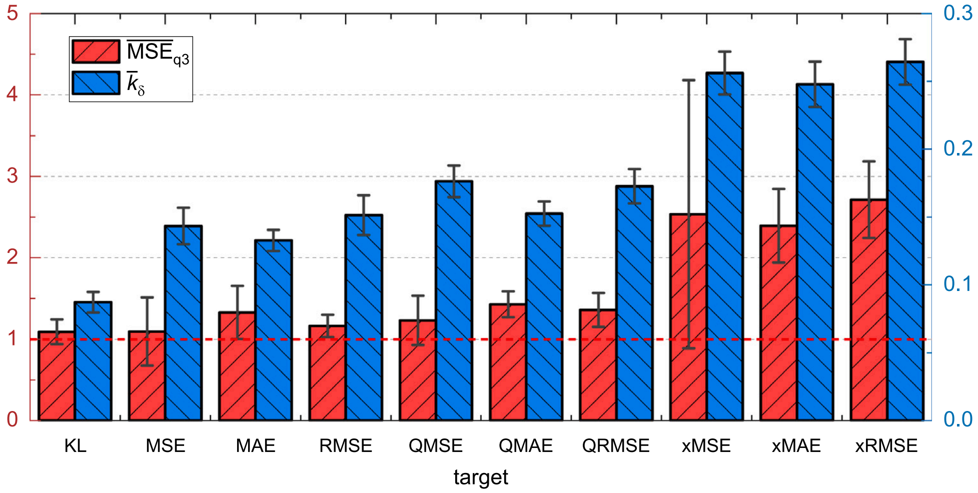


Fig. 9. Comparison of optimization results under different objectives using CMA-ES sampler with 800 iterations.

areas where they diverge, such as the distribution tails. This sensitivity allows KL divergence to capture subtle differences between the estimated and target distributions more effectively. However, this same sensitivity also increases the influence of small discrepancies and noise, potentially amplifying variance in the results. Consequently, while KL divergence can improve kernel estimation accuracy, the improvement remains modest and may lead to minor instability due to its heightened responsiveness to distributional nuances.

3.6. Evaluation of multi-data input strategy

As mentioned in Section 2.2, the optimization framework employs a multi-data input approach. To quantify its effectiveness, the results with and without this approach were shown in Fig. 10. The plot illustrated the differences in performance between the two methods, with respect to both $\overline{\text{MSE}}_{q3}$ and $\overline{k}_\delta$. Here, $\overline{\text{MSE}}_{q3}$" represents post-processed metrics for the multi-data input approach, in which the error contributions from 1D data were excluded to ensure comparability with the single-data case. Initially, the multi-data input method showed higher $\overline{\text{MSE}}_{q3}$ than the single-data approach. This is likely because, in the multi-data strategy, the optimizer minimizes the combined error across all datasets. At low iteration counts, the optimization may initially focus more on reducing the error of the 1D datasets. However, as the number of iterations increases, the 2D data error—reflected in $\overline{\text{MSE}}_{q3}$"—also decreases rapidly. For $\overline{k}_\delta$, the multi-data input method demonstrated a clear advantage over the single-data approach. As iterations increased, the multi-data method consistently reduced kernel error at a faster rate. Conversely, the single-data method converged more slowly over time, potentially plateauing before achieving the desired accuracy in kernel estimation. The underlying reason for this issue lies in the symmetry in the kernel model and the dPBE grid, as discussed in Section 2.2.

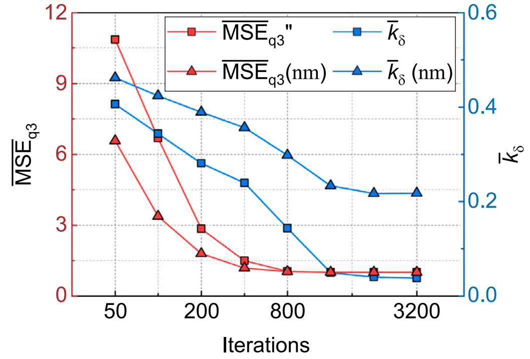


Fig. 10. Comparison of optimization results with and without multi-data input (nm) using CMA-ES sampler.

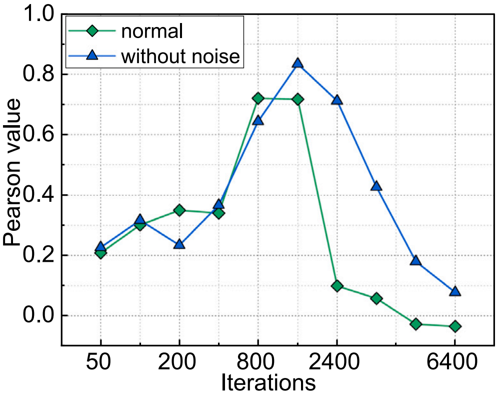


Fig. 11. Pearson value under different iteration counts and external conditions.

Consequently, the multi-data input approach appears more robust and effective.

3.7. Correlation between $\overline{MSE}_{q3}$ and $\bar{k}_{\delta}$

The primary objective of the optimization framework is to minimize the difference between the input PSD_0 and the optimized $\mathrm{PSD}_{\mathrm{opt}}$, which in turn helps identify the kernel set $\mathrm{k}_{\mathrm{opt}}$ that is closest to the true kernel k_0. This relationship suggests that minimizing $\mathrm{MSE}_{\mathrm{q3}}$ should also minimize k_δ, which forms the foundation of the framework's design. To assess this assumption, we conducted a correlation study between $\mathrm{MSE}_{\mathrm{q3}}$ and k_δ at various iteration steps using the Pearson correlation coefficient. The iteration range up to 6400 was divided into ten non-uniform intervals, and for each interval, ten evenly spaced results were extracted per dataset. Pearson correlation coefficients were computed independently for each dataset within each interval, and the final reported values represent the average across all datasets. The Pearson value quantifies the linear relationship between two variables, with values ranging from −1 to 1. Positive values indicate a direct correlation, while negative values suggest an inverse relationship. A value near 0 indicates no correlation. The optimization results in Fig. 11 were obtained using the CMA-ES sampler. The figure presents two scenarios: the green line represents results under normal conditions without additional constraints; the blue line represents results with data free from noise, simulating an ideal scenario without measurement errors.

As shown in Fig. 11 both the normal and noise-free cases exhibit similar trends in Pearson correlation. Prior to 400 iterations, Pearson values remained low and slightly fluctuated, as many datasets were still in the early exploration phase and lacked a strong linear relationship between $\mathrm{MSE}_{\mathrm{q3}}$ and k_δ. After approximately 800 iterations, most datasets entered a significant convergence phase, resulting in a sharp increase in correlation. This trend is also consistent with the observation in Section 3.1, where even a small reduction in $\mathrm{MSE}_{\mathrm{q3}}$ between 800 and 1600 iterations led to a substantial decrease in k_δ. Beyond 2400 iterations, the majority of datasets had already converged, and the correlation between $\mathrm{MSE}_{\mathrm{q3}}$ and k_δ naturally diminished. In the later stages, the presence of noise led to a more rapid drop in the Pearson value, even falling slightly below zero. This suggests that noise has a small impact on final optimization precision, making it harder for the optimizer to further reduce k_δ through minimizing $\mathrm{MSE}_{\mathrm{q3}}$.

4. Experimental case study

4.1. Experimental setup and data

To provide the experimental foundation of the kernel model, a water-based anode slurry with a solid content of 43 wt% was prepared in a 400 mL beaker using a EUROSTAR 40 digital overhead stirrer (IKA) with a 50 mm dissolver disk. The formulation compromising 93 wt% graphite, 1.4 wt% carbon black (CB), 1.87 wt% carboxymethyl cellulose (CMC) and 3.73 wt% styrene-butadiene rubber (SBR) was based on compositions reported in the literature Meza Gonzalez et al. (2024). CMC (Carl Roth) was first dispersed in deionized water at 1200 rpm for 70 min, followed by the addition of SBR (Nanografi Nano Technology) and continued stirring for 5 min. CB (Super C65, Nanografi) and graphite (Mechano-Cap 1P1, HC Carbon GmbH) were subsequently added at 600 rpm. Complete wetting was assumed after 15 min of stirring, defining the initial time point t_0. After reaching t_0, the stirring speed was set to 600, 900, 1200, 1500, and 1800 rpm in individual experimental runs. At each set stirring speed, three samples were collected from the center of the beaker at defined time points (0, 5, 10 and 45 minutes after t_0). The samples were diluted (1:100) with deionized water and the number-based particle size distribution (Q_0) was analyzed using a LUMiSizer (LUM GmbH). Due to its significantly larger particle volume and much lower number density compared to carbon black, the contribution of graphite to the overall Q_0 distribution is negligible. We also performed a separate stirring experiment using graphite alone and found that, under the current experimental conditions, graphite particles exhibit almost no breakage. Based on this, we truncated the Q_0 distribution at the minimum observed graphite volume, and the remaining portion was considered to represent the carbon black distribution exclusively. This truncated distribution was then used as input for the 1D-PBE model.

4.2. SPH simulations and post-processing

To quantify the mechanical stress induced by the mixer, numerical simulations of the flow behavior under different conditions were performed. For this purpose, we implemented the open-source software DualSPHysics (Domínguez et al., 2022), which uses the smoothed particle hydrodynamics (SPH) method to solve the Lagrangian formulation of the continuity and momentum conservation equations governing the flow.

The shear rate within the mixer was determined based on local velocity gradients derived from SPH. As depicted in Fig. 12, the computed velocity field exhibits a clearly heterogeneous flow pattern. This spatial variation in velocity gives rise to corresponding gradients in shear stress across the domain. The local shear rate was computed by first evaluating the strain rate tensor $\dot{\gamma}$ from the velocity gradient field, using the following equation (Meza Gonzalez and Nirschl, 2023):

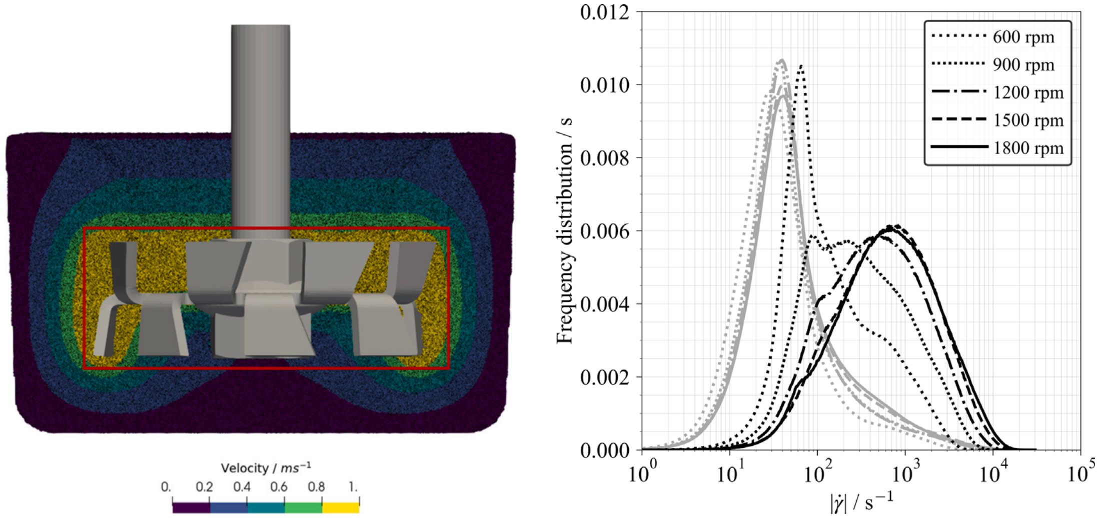
$$\dot{\gamma} = \nabla\vec{v} + \nabla\vec{v}^{\mathrm{T}} \tag{20}$$

$$|\dot{\gamma}| = \sqrt{\frac{1}{2}\left(\mathrm{tr}(\dot{\gamma})^2 - \mathrm{tr}(\dot{\gamma}^2)\right)} \tag{21}$$

The distribution of the shear rate magnitude $|\dot{\gamma}|$ is illustrated in Fig. 12(b). To ensure statistical robustness, local shear rate values were sampled over 20 consecutive time frames, corresponding to one second of physical simulation time. From this dataset, both mean and median values were computed to characterize effective shear rates under different assumptions. To account for spatial heterogeneity in the flow field, shear rate evaluations were conducted in two distinct regions. The global evaluation considered all sampling points within the entire mixer domain, while the local evaluation was restricted to the high-shear region adjacent to the dissolver disk. This localized region, highlighted by a red rectangle in Fig. 12(b), corresponds to a cylindrical volume with a height of 16 mm and a diameter of 50 mm.

Accordingly, four effective shear rate metrics were defined:

- the global mean shear rate, $\bar{G}_{\mathrm{global}}$;
- the global median shear rate, $G_{\mathrm{global,50}}$;
- the local mean shear rate, $\bar{G}_{\mathrm{local}}$;



(a) Fluid velocity snapshot

(b) Shear rate distribution

Fig. 12. Flow field and shear rate characteristics in the stirring. The red rectangular in (a) indicates the locally defined high-shear region near the stirrer. In (b), the gray curves shows the global shear rate distributions, while the black curve highlights the distributions restricted to the local region.

- the local median shear rate, $G_{\text{local},50}$.

These metrics were subsequently used as input variables in the optimization framework to investigate their influence on the accuracy of kernel parameter identification. The resulting numerical values are listed in Table D.1 in the SI.

4.3. Optimization procedure

The kernel models introduced in Section 2.1.1 were employed here in a 1D form. As discussed previously, these kernel parameters are determined by the material properties and the type of shear applied. On this basis, it is reasonable to assume that, within the batch experimental setup, all experimental groups share the same set of kernel parameters. The effect of varying stirrer speed on agglomeration and breakage is captured by the corresponding shear rate G. Relevant grid and model parameters were adjusted to match the experimental values and are listed in Table D.2 in the SI.

As noted earlier, the true kernel parameters underlying the experimental data are unknown; consequently, $\overline{k}_\delta$ cannot be computed to directly quantify the effectiveness of the optimization framework. Instead, a five-fold cross-validation strategy was adopted: in each fold, one group served as the test set while the remaining four were used for training. For each fold, a set of kernel parameters was optimized by minimizing the average cost function across all training datasets. The resulting parameters were then evaluated on the held-out test set. For each fold, the training loss (the mean value of the minimized objective over the training sets), test loss (the cost on the test set) and optimal parameters were recorded. The differences observed among the folds thus provide a practical measure of the optimization error and model generalizability.

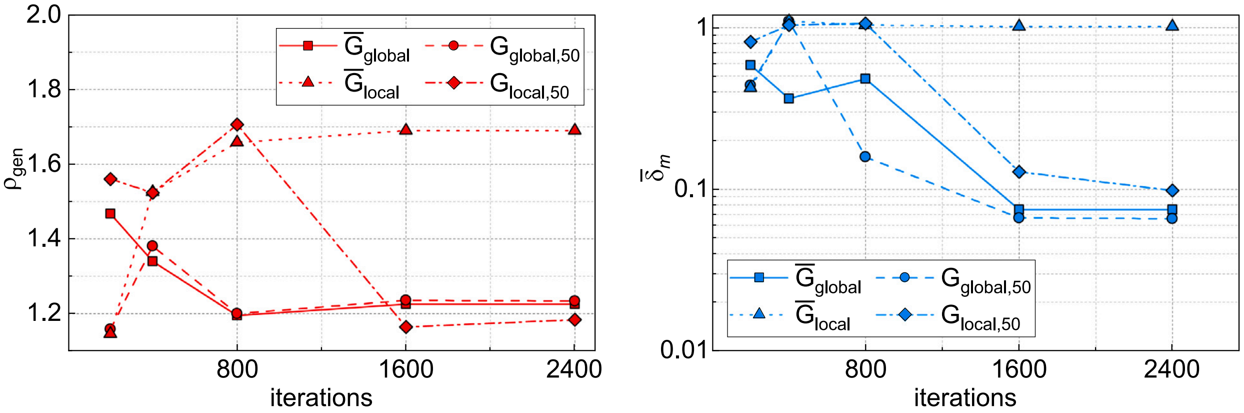
In this cross-validation scheme, relative measures are preferred over absolute values to assess optimization performance. One key indicator is the ratio of test loss to training loss

$$\rho_{\text{gen}} = \frac{\mathcal{L}_{\text{test}}}{\mathcal{L}_{\text{train}}} \quad , \tag{22}$$

which quantifies the model's generalization ability. A ratio close to unity suggests that the optimized parameters perform equally well on unseen data, indicating strong generalizability. Another important metric is the standard deviation of the optimized kernel parameters across the folds. When this deviation approaches zero, it implies that all cross-validation runs have converged to nearly identical solutions, reflecting both reliable convergence and the suitability of the kernel model for the experiment. For comparability across parameters of different magnitudes, each standard deviation was further normalized by dividing by the corresponding mean parameter value.

$$\delta_m = \frac{\sigma_{k_{m,\text{opt}}}}{\overline{k}_{m,\text{opt}}} \tag{23}$$

The selection of search boundaries for the optimization parameters is critical in practical applications. For the kernel models used in this study, two scenarios can be distinguished. First, for parameters such as v and P_2, empirical knowledge allows for relatively narrow bounds. In the breakage function, v is limited to the interval $(0, 2]$, with smaller values corresponding to a greater number of fragments. A lower bound of 0.1 is sufficient, allowing for up to 21 fragments, which covers most realistic cases. P_2, representing the exponential influence of relative particle volume on the breakage rate, is typically restricted to positive values and rarely reaches extreme magnitudes in practice, so a range of [0.1, 5] is considered adequate. In contrast, parameters such as k_{corr} and P_1 may span several orders of magnitude, making it difficult to set a priori search limits. In this work, suitable bounds were identified by conducting multiple test optimizations with 400 iterations each. Because the optimization framework relies on heuristic algorithms, inappropriate boundaries can cause the sampling strategy to drive the search rapidly toward one edge of the interval. This signaled the need to extend or shift the boundary in that direction. Conversely, if the optimization converged well within the chosen range after increasing the iteration count, the boundary selection was likely appropriate. The final parameter ranges used in this case study are summarized in Table 4. Additionally, in our tests, we also observed that defining an extremely large search range for the parameters did lead to a noticeable slowdown in convergence. For example, expanding the search range for k_{corr} to 10^{-16}–10^4, for P_1 to 10^{-16}–10^4, and for P_2 to 0.01–50 resulted in a total required iteration count of approximately 4000–6400 steps for convergence. However, the optimization results eventually matched those obtained with narrower search ranges. Therefore, when



(a) (b)

Fig. 13. Evolution of ρ_{gen} and $\bar{\delta}_m$ under different shear rate assumptions.

Table 4
The value ranges of kernels in case study.

Parameter	Lower bound	Upper bound
$\mathbf{k}_{\text{corr}}$	10^{-8}	10^{2}
P_1	10^{-10}	10^{-2}
P_2	0.1	5.0
υ	0.1	2.0

computational resources are sufficient, it remains feasible to employ a broad search range in combination with a higher iteration count.

4.4. Results and discussion

In this case study, the fastest optimizer CMA-ES was again employed. As in the theoretical study, the optimization objective can be formulated using various representations of the PSD in combination with different cost functions. All combinations were evaluated here as well, and the results generally corroborate the findings from the synthetic data. One notable observation was that, for experimental data, using the cumulative distribution Q_0 yielded slightly better performance than the density distribution q_0. This can be attributed to two factors. First, the measurement instruments directly provide the Q_0 distribution. Converting Q_0 to q_0 introduces discretization errors, which can reduce the precision of q_0. Second, experimental noise tends to be more pronounced and uneven, leading to large local fluctuations in q_0 that are difficult for the PBE solver to reproduce. In contrast, Q_0 is an integrated measure and is less susceptible to such noise-induced variations. Therefore, in the following discussion of the experimental case, we focused on the results obtained using Q_0 and MSE as a cost function.

Subsequently, various definitions of the effective shear rate were incorporated into the optimization framework and evaluated independently. The corresponding results are depicted in Fig. 13. As illustrated in Fig. 13(a), the use of $\bar{G}_{\text{local}}$ consistently resulted in elevated values of ρ_{gen} throughout the optimization process, indicative of reduced generalization performance. A similar pattern was evident in Fig. 13(b) for $\bar{\delta}_m$, where $\bar{G}_{\text{local}}$ again yielded the largest deviations, reflecting suboptimal convergence of the identified kernel parameters. In contrast, the remaining three definitions demonstrated substantially improved and mutually comparable performance. Among these, $G_{\text{local,50}}$ produced the lowest values of ρ_{gen}, whereas $G_{\text{global,50}}$ achieved the smallest $\bar{\delta}_m$, suggesting enhanced generalization and parameter stability, respectively. Given that ρ_{gen} serves as a more direct and comprehensive indicator of the model's generalization capability, $G_{\text{local,50}}$ was selected as the effective shear rate input for all subsequent analyses.

The results are summarized in Fig. 14. The horizontal axis represents the number of optimization iterations. The left vertical axis shows the generalization ratio ρ_{gen}, while the right vertical axis presents the normalized standard deviation δ_m for each kernel parameter. The generalization ratio ρ_{gen} converged to approximately 1.18 after 1600 iterations and remained constant. This indicates that as the optimizer reduced the PSD error on the training sets, the error on the test data was also effectively minimized. The final optimized parameters yielded similar performance on both the training and test groups, demonstrating good generalizability. Furthermore, as the number of iterations increased, the δ_m values for all kernel parameters initially decreased and then stabilized at a certain level, with no subsequent increase at higher iteration counts. This trend indicated that the kernel parameters converged effectively and that the optimizer found nearly identical solutions for each cross-validation fold. For all parameters except υ, further reduction in δ_m beyond approximately 2400 iterations was limited by measurement noise and uncertainties in the estimated shear rate. For the kernel parameters k_{corr} and P_1, the residual deviations were the largest. This can be attributed to the fact that both parameters appear as direct multiplicative factors with the shear rate G in the kernel models, so any error in G is directly propagated and limits their attainable accuracy. In contrast, P_2, which characterizes the dependence of the breakage rate on relative particle volume, is influenced only by errors in the measured PSD and thus achieves a higher level of precision. The parameter υ exhibited a distinct behavior: with increasing iterations, the optimizer consistently drove υ toward its upper boundary of 2 for all datasets, resulting in its standard deviation approaching zero. $\upsilon = 2$ corresponds to a parabolic breakage function producing two fragments. In industrial practice, binary breakage functions are often adopted to simplify the model, as in the work of Ruan et al. (2022). This observation suggests that, in shear-dominated industrial processes, it may be reasonable to fix $\upsilon = 2$ when optimizing against experimental data, thereby reducing the complexity of the optimization procedure.

5. Conclusion

This paper presents an optimization framework for accurately determining unknown kernel parameters in the dPBE model. The framework's performance, robustness, and adaptability were rigorously tested. For the theoretical study, synthetic data with added noise was used, while reliability was quantitatively assessed using two key metrics: the mean-squared error, $\overline{\text{MSE}}_{\text{q3}}$, which indicates discrepancies in particle volume distribution, and the mean kernel error, $\overline{k}_{\delta}$, which reflects the deviation between optimized and true kernel parameters. The testing results reveal that the CMA-ES sampler outperformed others, exhibiting faster and more stable convergence to minimal error. For the

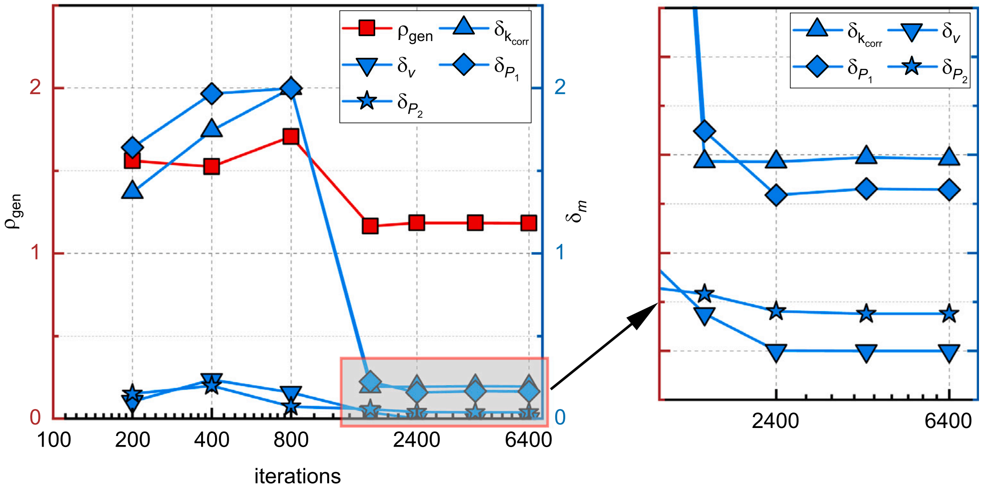


Fig. 14. Evolution of ρ_{gen} and δ_m with iteration count in the case study, using shear rates calculated by the $G_{\text{local},50}$ method.

current model with eight target parameters, CMA-ES achieved optimal results within 800–1600 iterations. Although slightly less optimal, HEBO, NSGA, and TPE samplers were viable alternatives, as they could also achieve comparable performance with an increased number of iterations, offering flexibility in optimization strategies.

One major challenge addressed in this study is the difficulty of obtaining true 2D particle volume distributions in multi-material systems, where only reduced agglomerate volume data is typically available. Optimization results suggested that relying solely on such reduced data could cause slow or even non-converging solutions. To address this, the study introduced a “multi-data” input approach, which incorporates one reduced 2D dataset alongside two separate 1D datasets, significantly enhancing the accuracy and efficiency of kernel parameter optimization in multi-material contexts.

The study also explored the relationship between MSE_{q3} and k_δ, demonstrating that non-unique solutions in high-dimensional parameter spaces can hinder precise kernel parameter identification. However, integrating at least one kernel parameter obtained through alternative methods, such as CFD-DEM, can effectively address this issue. Additionally, while this approach improves the framework’s tolerance to noise, the negative impact of noise on optimization results cannot be ignored. Enhancing measurement precision or employing multiple measurements to mitigate noise effects remains essential for achieving higher accuracy and more consistent optimization outcomes.

Finally, a case study was conducted using PSD data from a dispersion experiment. The cross-validation results demonstrated the fundamental reliability of the optimization framework, even with a limited amount of experimental data and confirmed the generalizability of the current kernel models. The analysis highlighted the influence of external factors such as measurement errors and the estimation of physical parameters on the optimization outcomes. Moreover, the results supported the practical assumption of a binary fragment distribution for the breakage function. As shown in the theoretical study, this simplification can significantly enhance the framework’s performance. It should be noted that although the case study employed only a 1D-PBE, the core purpose of the optimization framework is to identify kernel parameters that best fit a given PBM with respect to the available input data. The dimensionality mainly affects the number of parameters to be estimated, rather than the framework’s structure itself. Therefore, this experimental validation still supports the practical usability of the proposed framework.

In conclusion, the proposed optimization framework demonstrates high accuracy in identifying kernel parameters and offers a flexible, adaptable solution for kernel parameter estimation in complex particle systems. This study not only validates the framework’s performance but also outlines potential directions for further enhancement, particularly in improving its efficiency and accuracy across varied applications.

Abbreviations

CFD-DEM	Computational Fluid Dynamics-Discrete Element Method
CMA-ES	Covariance Matrix Adaptation Evolutionary Strategy
GP	traditional Gaussian Process Bayesian Optimization sampler
HEBO	Heteroscedastic Evolutionary Bayesian Optimization
KL	Kullback–Leibler Divergence
MSE	Mean Squared Error
NSGA	Nondominated Sorting Genetic Algorithm
PSD	Particle Size Distribution
QMC	Quasi Monte Carlo
RMSE	Root Mean Squared Error
TPE	Tree-structured Parzen Estimator

Notation

B	birth rate of particles [$\mu\text{m}^{-4}\text{s}^{-1}$]
D	death rate of particles [$\mu\text{m}^{-4}\text{s}^{-1}$]
f_B	breakage distribution function [-]
G	shear rate [s^{-1}]
$\text{K}_{m,0}$	set of all test values for a kernel parameter [-]
$\mathbf{k}_{\text{corr}}$	correction factor for agglomeration kernel [-]
k_m^*	length of the kernel parameter set [-]
k_δ	average error of kernel parameters[-]
M_{00}	zeroth moment of PSD [-]
M_{11}	first cross moment of PSD [-]
n	particle number density [μm^{-4}]
P_1, P_2, P_3, P_4	kernel parameters for breakage rate
$\text{PSD}_{V,t}$	PSD value of particles with volume V at time t [-]
PSD_0	initial particle size distribution [-]
PSD_{opt}	optimized particle size distribution [-]
Q_3	Cumulative particle size distribution by volume [-]
q_3	Differential particle size distribution by volume [-]

r_A	agglomeration kernel [-]
r_B	breakage rate kernel [-]
R	radii of agglomerate [μm]
S_T	total-effect index in Sobol method [-]
t	time [s]
v	shape parameter for fragment distribution [-]
x, y	volume of two distinct materials in an agglomerate [μm^3]
$x_{\text{mean}}, y_{\text{mean}}$	weighted particle volumes derived from the initial PSD [-]
x_{50}	median particle size [μm^3]
z_{ijkl}	adapted self-similar daughter size [-]
α_{corr}	collision efficiency [-]
β_{abij}	collision frequency [s^{-1}]
β_{corr}	correction factor for shear rate [-]
δ_{PSD}	noise introduced into PSD [-]
θ_{ijkl}	self-similar daughter distribution [-]

CRediT authorship contribution statement

Haoran Ji: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Lena Fuhrmann:** Writing – original draft, Validation, Investigation, Data curation. **Juan Fernando Meza Gonzalez:** Writing – original draft, Validation, Investigation, Data curation. **Frank Rhein:** Writing – review & editing, Supervision, Software, Project administration, Methodology, Funding acquisition, Conceptualization.

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.

Acknowledgments

The authors acknowledge the funding by the DFG (German Research Foundation) under grant number 520278921.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.dche.2025.100272.

Data availability

The optimization framework is publicly available at github.com/pdhs-group/PSD_opt. Experimental data is available upon request.

References

Asylbekov, Ermek, Mayer, Julian, Nirschl, Hermann, Kwade, Arno, 2023. Modeling of carbon black fragmentation during high-intensity dry mixing using the population balance equation and the discrete element method. Energy Technol. (ISSN: 21944296) 11, http://dx.doi.org/10.1002/ente.202200867.

Atmuri, Anand K., Henson, Michael A., Bhatia, Surita R., 2013. A population balance equation model to predict regimes of controlled nanoparticle aggregation. Colloids Surfaces A: Physicochem. Eng. Asp. 436, 325–332. http://dx.doi.org/10.1016/j.colsurfa.2013.07.002.

Bemer, G.G., 1979. Agglomeration in suspension: A study of mechanisms and kinetics. https://repository.tudelft.nl/record/uuid:ccad7dfa-fb9a-4088-927f-6aad8cf1a5e3.

Capece, Maxx, Bilgili, Ecevit, Dave, Rajesh, 2011. Identification of the breakage rate and distribution parameters in a non-linear population balance model for batch milling. Powder Technol. (ISSN: 00325910) 208, 195–204. http://dx.doi.org/10.1016/j.powtec.2010.12.019.

Chi, Ho Anh, Sommerfeld, Martin, 2002. Modelling of micro-particle agglomeration in turbulent flows. Chem. Eng. Sci. 57, 3073–3084. http://dx.doi.org/10.1016/S0009-2509(02)00172-0.

Chin, Ching-Ju, Yiacoumi, Sotira, Tsouris, Costas, 1998. Shear-induced flocculation of colloidal particles in stirred tanks. J. Colloid Interface Sci. (ISSN: 0021-9797) 206, 532–545. http://dx.doi.org/10.1006/jcis.1998.5737.

Deb, Kalyanmoy, Jain, Himanshu, 2013. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints. http://dx.doi.org/10.1109/TEVC.2013.2281535.

Diemer, R.B., Olson, J.H., 2002. A moment methodology for coagulation and breakage problems: Part 3—generalized daughter distribution functions. Chem. Eng. Sci. (ISSN: 0009-2509) 57, 4187–4198. http://dx.doi.org/10.1016/S0009-2509(02)00366-4.

Domínguez, Jose M, Fourtakas, Georgios, Altomare, Corrado, Canelas, Ricardo B, Tafuni, Angelo, García-Feal, Orlando, Martínez-Estévez, Ivan, Mokos, Athanasios, Vacondio, Renato, Crespo, Alejandro JC, et al., 2022. DualSPHysics: from fluid dynamics to multiphysics problems. Comput. Part. Mech. 9 (5), 867–895. http://dx.doi.org/10.1007/s40571-021-00404-2.

Hansen, Nikolaus, 2023. The CMA evolution strategy: A tutorial. https://arxiv.org/abs/1604.00772.

Jain, Himanshu, Deb, Kalyanmoy, 2013. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: Handling constraints and extending to an adaptive approach. http://dx.doi.org/10.1109/TEVC.2013.2281534.

Jeldres, Ricardo I., Fawell, Phillip D., Florio, Brendan J., 2018. Population balance modelling to describe the particle aggregation process: A review. Powder Technol. (ISSN: 1873328X) 326, 190–207. http://dx.doi.org/10.1016/j.powtec.2017.12.033.

Kumar, Jitendra, Peglow, Mirko, Warnecke, Gerald, Heinrich, Stefan, 2008. An efficient numerical technique for solving population balance equation involving aggregation, breakage, growth and nucleation. Powder Technol. (ISSN: 00325910) 182, 81–104. http://dx.doi.org/10.1016/j.powtec.2007.05.028.

Liaw, Richard, Liang, Eric, Nishihara, Robert, Moritz, Philipp, Gonzalez, Joseph E, Stoica, Ion, 2018. Tune: A research platform for distributed model selection and training. arXiv preprint arXiv:1807.05118.

Meza Gonzalez, Juan Fernando, Nirschl, Hermann, 2023. Numerical investigation of the local shear rate in a twin-screw extruder for the continuous processing of Li-ion battery electrode slurries. Energy Technol. 11 (6), 2201517. http://dx.doi.org/10.1002/ente.202201517.

Meza Gonzalez, Juan Fernando, Nirschl, Hermann, Rhein, Frank, 2024. Continuous anode slurry production in twin-screw extruders: Effects of the process setup on the dispersion. Batteries 10 (5), 145. http://dx.doi.org/10.3390/batteries10050145.

Nielsen, Rasmus Fjordbak, Nazemzadeh, Nima, Sillesen, Laura Wind, Andersson, Martin Peter, Gernaey, Krist V., Mansouri, Seyed Soheil, 2020. Hybrid machine learning assisted modelling framework for particle processes. Comput. Chem. Eng. (ISSN: 00981354) 140, http://dx.doi.org/10.1016/j.compchemeng.2020.106916.

Pandya, J.D., Spielman, L.A., 1983. Floc breakage in agitated suspensions: Effect of agitation rate. Chem. Eng. Sci. (ISSN: 0009-2509) 38, 1983–1992. http://dx.doi.org/10.1016/0009-2509(83)80102-X.

Patruno, L.E., Dorao, C.A., Svendsen, H.F., Jakobsen, H.A., 2009. Analysis of breakage kernels for population balance modelling. Chem. Eng. Sci. (ISSN: 00092509) 64, 501–508. http://dx.doi.org/10.1016/j.ces.2008.09.029.

Raponi, Antonello, Marchisio, Daniele, 2024. Deep learning for kinetics parameters identification: A novel approach for multi-variate optimization. Chem. Eng. J. (ISSN: 1385-8947) 489, 151149. http://dx.doi.org/10.1016/j.cej.2024.151149.

Rhein, Frank, Russ, Felix, Nirschl, Hermann, 2019. Collision case model for population balance equations in agglomerating heterogeneous colloidal systems: Theory and experiment. Colloids Surfaces A: Physicochem. Eng. Asp. (ISSN: 18734359) 572, 67–78. http://dx.doi.org/10.1016/j.colsurfa.2019.03.089.

Ruan, Zhuen, Wu, Aixiang, Bürger, Raimund, Betancourt, Fernando, Ordoñez, Rafael, Wang, Jiandong, Wang, Shaoyong, Wang, Yong, 2022. A population balance model for shear-induced polymer-bridging flocculation of total tailings. Backfilling Mater. Undergr. Min. 12 (1), 40. http://dx.doi.org/10.3390/min, https://www.mdpi.com/2075-163X/12/1/40.

Sobol', I.M., 2001. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. Math. Comput. Simulation (ISSN: 0378-4754) 55, 271–280. http://dx.doi.org/10.1016/S0378-4754(00)00270-6, The Second IMACS Seminar on Monte Carlo Methods.

Trunk, Robin, Marquardt, Jan, Thäter, Gudrun, Nirschl, Hermann, Krause, Mathias J., 2018. Towards the simulation of arbitrarily shaped 3D particles using a homogenised lattice Boltzmann method. Comput. & Fluids (ISSN: 0045-7930) 172, 621–631. http://dx.doi.org/10.1016/j.compfluid.2018.02.027, URL https://www.sciencedirect.com/science/article/pii/S0045793018300823.

Wang, Li Ge, Ge, Ruihuan, Chen, Xizhong, 2022. On the determination of particle impact breakage in selection function. Particuology (ISSN: 22104291) 65, 117–132. http://dx.doi.org/10.1016/j.partic.2021.08.003.

Watanabe, Shuhei, 2023. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. https://arxiv.org/abs/2304.11127.

Zhang, Dong, Li, Qingjian, Prigiobbe, Valentina, 2022. Population balance modeling of homogeneous viral aggregation. Chem. Eng. Sci. 247, 117035. http://dx.doi.org/10.1016/j.ces.2021.117035.

Zheng, Jianxiang, Li, Yukai, Wan, Zongqun, Hong, Wenpeng, Wang, Long, 2019. Modification of the agglomeration kernel and simulation of the flow pattern in acoustic field with fine particles. Powder Technol. (ISSN: 1873328X) 356, 930–940. http://dx.doi.org/10.1016/j.powtec.2019.09.022.