RSC Mechanochemistry

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *RSC Mechanochem.*, 2025, **2**, 889

Linking mechanics and chemistry: machine learning for yield prediction in $NaBH_4$ mechanochemical regeneration

Santiago Garrido Nuñez, *[a] Dingena L. Schott [b] and Johan T. Padding [a]

Mechanochemical synthesis faces reproducibility and scale-up challenges due to complex parameter interactions. This study employs machine learning (ML) to predict $NaBH_4$ regeneration yield, integrating chemical experimental data and DEM (Discrete Element Method) derived invariant mechanical descriptors ($\bar{E}_n$, $\bar{E}_t$, f_{col}/n_{ball}). Various algorithms were evaluated, including a two-step modeling strategy to isolate the dominant effect of milling time in our process. Results demonstrate that a two-step Gaussian Process Regression (GPR) model achieves good predictive performance ($R^2 = 0.83$), significantly outperforming single-stage models and providing valuable uncertainty estimates. Tree-based ensembles (XGBoost, RF) also benefit from the two-step approach and can enhance interpretability. This work establishes a framework for using ML to optimize mechanochemical processes, reducing experimental cost and offering a method to link mechanical milling conditions to chemical outcomes, thereby enabling predictive mechanochemistry.

Received 11th June 2025
Accepted 1st September 2025

DOI: 10.1039/d5mr00076a

rsc.li/RSCMechanochem

1 Introduction

1.1 Mechanochemical reactions *via* high-energy ball milling

The advancement of mechanochemistry in the last two decades has seen the application and innovation of multiple tools and processes to achieve chemical and material synthesis that align with the principles of green chemistry.[1] Typically, mechanochemical processes at the lab scale rely on ball mills to supply the (mechanical) energy required to achieve a desired chemical reaction, although different methods have been explored to combine this with additional sources of energy, such as thermal energy, acoustic energy or electrical energy.[2] Pure mechanochemical ball milling is often characterized by intuitive process parameters that any ball mill can readily account for, namely rotational speed, filling ratio, ball-to-powder ratio (BPR), milling time, and additional physical material properties such as density of the milling balls.[3–7] Although these parameters certainly steer the overall behavior of the process, it has been observed that they are not sufficient to accurately characterize mechanochemical processes, leading to significant challenges in reproducibility and scaling up given the intrinsic differences in working principle that different machines have.[8,9]

It becomes clear that mechanochemistry involves a series of complex interactions that must be investigated systematically before layering on additional, non-intuitively tunable energy inputs, especially because both mechanical and chemical variables fundamentally dictate high yields. However, due to the relative novelty of the field, research has remained largely exploratory, employing one-variable-at-a-time (OVAT) studies that prove inadequate once scale-up or efficiency optimization becomes the goal.[10–12]

To tackle this challenge, the Discrete Element Method (DEM) has been employed to accurately characterize a high-energy ball mill's internal dynamics, effectively bypassing the dependency of the utilized mill or the aforementioned process variables. This is done by defining three key mechanical characterization properties: the mean normal energy dissipation per collision $\bar{E}_n$, the mean tangential energy dissipation per collision $\bar{E}_t$, and the specific collision frequency per ball $\frac{f_{col}}{n_{ball}}$.[8] This methodology can be applied to any milling machine of any scale, reducing the challenges in reproducibility and providing guidelines for the specifications needed in larger-scale equipment. Regardless, this numerical characterization remains mechanical and thus, cannot include the influence of the chemical variables of the system, such as the molar ratio, BPR, milling time, and their confounded influence with the rest of the mechanical variables.

The chemical characterization of the system can only be accomplished experimentally. In our target reaction, the mechanochemical regeneration of $NaBH_4$ from $NaBO_2 \cdot 4H_2O$ (see eqn (1)), the dependence on molar ratio is non-linear, while the influence of milling time is effectively linear within the investigated range.[3] Furthermore, the interaction between the BPR and molar ratio is statistically significant, indicating a complex interaction among operating parameters. These

[a]*Department of Process and Energy, Delft University of Technology, Leeghwaterstraat 39, Delft, The Netherlands. E-mail: s.garridonunez@tudelft.nl*

[b]*Department of Maritime and Transport Technology, Mekelweg 2, Delft, The Netherlands*

experiments were carried out under constant, albeit optimized mechanical conditions from the pure perspective of energy dissipation, ignoring the effect that changing the distribution of shear and normal stress can have on the system.[8] Thus, while DEM simulations can facilitate a mechanical characterization, its effectiveness can only be tested experimentally. To overcome this limitation, we investigate the use of different machine learning (ML) algorithms to predict the conversion yield, reducing the need for trial-and-error experiments.

$$NaBO_2 \cdot 4H_2O + 6MgH_2 \rightarrow NaBH_4 + 6MgO + 8H_2 \quad (1)$$

In other applications, Anglou *et al.*[13] employed linear regression to link DEM outputs (collision frequency and average kinetic energy of a milling ball) to the depolymerization of PET, obtaining a good fit ($R^2 = 0.966$). This result, however, holds only within a range of total energy given to the system before the linear condition is lost. The authors accurately point out that a non-linear model could be trained but the lack of data prohibits this. Furthermore, this study made use of a single milling ball in a 25 mL jar where only the milling frequency was varied. This configuration effectively simplifies many other operational parameters that lab-scale and industrial-scale ball milling processes can have.

Similarly, Yu *et al.*[14] utilized polynomial regression to analyze different milling parameters and predict target particle sizes while ball milling alumina ceramics. Although no chemical processes were involved in this study, the authors point out the same challenge mentioned before: most studies focus on optimizing milling parameters, varying only one process variable and keeping the rest constant, which severely limits the applicability of data-based methods to gain a more profound understanding of their impact on the process. In the same context of pure milling, Li *et al.*[15] trained a convolutional neural network (CNN) to predict the grinding rate and size distribution of a rotating drum mill, achieving high accuracy ($R^2 > 0.95$) and good transfer learning results. This indicates that deep neural networks can capture the complex physics of milling when sufficient training data exists.

However, deep models such as these remain unviable when applied directly to experimental mechanochemical data, simply because such large, labeled datasets do not exist. Furthermore, the creation of these datasets requires extensive experimental work that necessitates a significant amount of time. For instance, a typical experiment involving the regeneration of $NaBH_4$ takes at least 72 hours from sample preparation to yield quantification.

To address these issues, a shared mechanochemical reaction database has been created,[16] allowing researchers to pool results and push machine learning approaches that connect milling conditions with chemical outcomes, something experts believe could revolutionize the field.[17] However, because different groups study different reactions in different mills, detailed data for any one process remain scarce, and most characterization methods only work on the specific equipment for which they were developed as stated before. Moreover, the "black-box" nature of ML models adds another limitation. While some methods like random forests offer feature-importance insights, other methods like deep neural networks and support–vector machines hardly explain why a given parameter set succeeds or fails. This makes it hard to build a mechanistic understanding or plan experiments beyond the model's training scope. Furthermore, while previous applications of machine learning in milling have often focused on either purely physical outcomes (*e.g.*, particle size prediction) or utilized limited operational parameters for chemical yield, a comprehensive approach integrating detailed, DEM-derived mechanical descriptors with a broader set of chemical process variables to predict yield for complex reactions remains unexplored. Finally, practical challenges persist: producing large, high-quality datasets demands extensive experimentation, run-to-run variability can introduce noise, and fitting sensors inside a sealed milling jar to gather real-time data is technically difficult.[18–20] Altogether, these five factors keep ML-driven ball milling mechanochemistry at a very early stage.

Within the broader landscape of data-driven reaction discovery and optimization, machine learning has not only accelerated condition search but also changed how chemists learn from experiments. In solution-phase synthesis, high-throughput experimentation (HTE) and automation provide the dense, standardized datasets that enable multivariate modeling and closed-loop optimization.[21,22] Multivariate linear models extract quantitative structure–reactivity/selectivity relationships that rank which variables matter and why, enabling prospective design.[23] Orchestration and active-learning platforms (*e.g.*, ChemOS; LabMate.ML) close the loop between Bayesian decision-making and automated execution, reaching high-yielding conditions in tens of experiments while handling mixed categorical/continuous spaces.[24,25] Beyond single substrates, closed-loop protocols now optimize for generality across substrate matrices, identifying condition sets that transfer across chemotypes.[26] Recent systems show that optimization can produce knowledge on-the-fly, integrating interpretable/physics-informed models with automation to uncover mechanistic factors during optimization.[27,28] These developments motivate our study, but also highlight two distinctions specific to mechanochemistry: data throughput is typically much lower than in plate- or flow-based solution platforms,[21] and controllable variables necessarily include mechanical/process descriptors of mechanical stressing and energy transfer, which are absent from most solution phase models.[8,29]

Motivated by recent ML-driven progress in solution-phase optimization, we lay the groundwork for a mechanochemistry-specific framework. We take advantage of a DEM-based mechanical characterization that establishes a commonality between mills, enabling unified datasets. We compare modeling families and map their strengths and limitations to use cases in small-data, high-cost regimes. Ultimately, we show that combining mechanical and chemical operating variables can accurately predict the mechanochemical yield. The dataset spans 27 experiments with wide ranges in both chemical and mechanical factors and, although compact, constitutes the most extensive open-access operating space for $NaBH_4$

regeneration to date, positioning this study as a practical starting point for predictive mechanochemistry.

2 Methodology

This section details the methodology employed to predict the experimental yield using machine learning techniques. The workflow encompasses data acquisition, feature engineering, model training, hyperparameter optimization, and evaluation. All analyses were performed using Python 3.9.

2.1 Data acquisition

The dataset (Table 1) utilized in this work combines two previously published components: experimental yields for regeneration of $NaBH_4$ using the Emax high-energy ball mill,[3] and a DEM-based methodology to mechanically characterize ball milling conditions.[8] In the present study, we derive device-independent descriptors for all experimental cases and assemble an ML-ready dataset that supports comparison and transfer across ball-milling devices. This is achieved by defining three key parameters: the mean normal energy dissipation per collision $\bar{E}_n$, the mean tangential energy dissipation per collision $\bar{E}_t$, and the specific collision frequency per ball $\frac{f_{col}}{n_{ball}}$. We note that the variables modeled in solution phase yield prediction studies typically comprise solvent, base, ligand/catalyst, temperature, concentrations, and time, often explored at scale *via* HTE or flow with inline analytics. In mechanochemistry, outcome-relevant variables also include mill type, jar/ball materials and sizes, ball-to-powder ratio, fill ratio, and milling frequency, and thus require abstraction *via* the aforementioned mechanical descriptors of energy transfer to compare between devices. The results presented in our previous work can be readily used to arrive to these key parameters in the Emax, but the methodology can be applied to any ball mill.[8]

Table 1 $NaBH_4$ regeneration dataset. BPR is the ball-to-powder ratio (mass basis), "Mol ratio" is the molar ratio $MgH_2 : NaBO_2 \cdot 4H_2O$, and "Time" is the milling time. $\bar{E}_n$ and $\bar{E}_t$ are the mean normal and tangential energy dissipated per collision obtained from DEM simulations; f_{col}/n_{ball} is the specific collision frequency per ball. Yields are onversion percentages through reaction (1). Experimental details;[3] DEM details[8]

Case	BPR	Mol ratio	Time [h]	$\bar{E}_n$ [μJ]	$\bar{E}_t$ [μJ]	f_{col}/n_{ball} [s^{-1}]	Yield [%]
0	10	8	5.0	221	500	400	12
1	10	8	12.5	382	888	533	22
2	10	8	20.0	613	1391	667	30
3	10	10	5.0	221	500	400	28
4	10	10	12.5	613	1391	667	39
5	10	10	20.0	221	500	400	45
6	10	12	5.0	613	1391	667	40
7	10	12	12.5	221	500	400	61
8	10	12	20.0	382	888	533	73
9	30	8	5.0	382	888	533	26
10	30	8	12.5	613	1391	667	37
11	30	8	20.0	221	500	400	42
12	30	10	5.0	613	1391	667	50
13	30	10	12.5	221	500	400	71
14	30	10	20.0	382	888	533	88
15	30	12	5.0	221	500	400	21
16	30	12	12.5	382	888	533	32
17	30	12	20.0	613	1391	667	49
18	50	8	5.0	613	1391	667	25
19	50	8	12.5	221	500	400	62
20	50	8	20.0	382	888	533	74
21	50	10	5.0	221	500	400	31
22	50	10	12.5	382	888	533	73
23	50	10	20.0	613	1391	667	90
24	50	12	5.0	382	888	533	41
25	50	12	12.5	613	1391	667	62
26	50	12	20.0	221	500	400	57

Experimentally, hydrated sodium metaborate ($NaBO_2 \cdot 4H_2O$) (≥99%) was purchased from Sigma-Aldrich, while magnesium hydride (MgH_2) (≥99.9%, ≤50 μm) was sourced from Nanoshel. All reactants were used without further purification. The sample preparation for all ball milling experiments was carried out in a glove box under an argon atmosphere, with oxygen and water concentrations maintained below 0.1 ppm. For a detailed description of the quantification of the chemical yield and equipment cleaning to preserve similar conditions for all experimental cases, we refer to our previous work.[3]

2.2 Feature engineering

To facilitate the capture of non-linear relationships and interactions, several feature engineering steps were applied to the initial feature set:

(1) Quadratic term: our previous results indicate that the molar ratio has a significant quadratic relationship with the chemical yield.[3] Thus, a new feature containing this quadratic term was added.

(2). Interaction term: we have found a strong interaction between the BPR and the molar ratio.[3] Therefore, we also include an additional feature composed of the product of these 2 variables.

(3) Sigmoid transformation: to account for potential saturation effects, sigmoid transformations ($1/(1 + \exp(-x))$) were applied to the 'Time', and 'BPR' features, creating new features while retaining the original features. This allows the model to plateau rather than grow indefinitely (or turn negative) as these variables change.

The resulting set of features constituted the final engineered feature matrix used for model training.

2.3 Train-test split and feature scaling

The dataset, comprising the engineered features and the target yield, was divided into training (80%) and testing (20%) sets. A fixed random state was used to ensure reproducibility of the split. With 27 experimental cases, this corresponds to 21 training and 6 test samples. To ensure every algorithm is evaluated on the same examples, we used a single, predetermined 21/6 partition created by shuffling once and then locking that partition for all analyses. All model fitting and hyperparameter selection used only the training data; the test set was held back until the final evaluation.

Feature scaling was applied to ensure that features with larger ranges did not disproportionately influence the model's sensitivity to feature magnitude, such as distance-based algorithms (Support Vector Regression (SVR), Gaussian Process Regression (GPR)) and linear models. Specifically, standardization was employed, where each feature was transformed to have zero mean and unit variance according to eqn (2):

$$x_{\text{scaled}} = \frac{x - \mu_{\text{train}}}{\sigma_{\text{train}}} \quad (2)$$

where x is the original feature value and μ_{train} and σ_{train} are the mean and standard deviation of that feature calculated exclusively from the training data partition.

The train-test split ensures that no information from the test set influences the transformation applied during the training phase (preventing data leakage) and preserves the integrity of the test set for unbiased model evaluation. The same training set parameters (μ_{train}, σ_{train}) were then used to standardize the corresponding features of the training set and the test set. Models requiring scaled data (Linear Regression, GPR, SVR) utilized these standardized features for both training and prediction. In contrast, tree-based models (Random Forest, XGBoost), which are less sensitive to feature scaling, were trained and evaluated using the original, unscaled engineered features.

2.4 Weighted loss function

To prioritize accurate prediction of higher yields, which are often of greater experimental interest, a custom weighted mean squared error (MSE) loss function was defined:

$$\text{Weighted MSE} = \frac{1}{N}\sum_{i=1}^{N} w_i \left(y_{\text{true},i} - y_{\text{pred},i}\right)^2 \quad (3)$$

where N is the number of samples, $y_{\text{true},i}$ and $y_{\text{pred},i}$ are the true and predicted yields for sample i, respectively. The weight w_i was set to 2.0 if $y_{\text{true},i} > 70\%$, and 1.0 otherwise. This weighted MSE was used as the primary scoring metric during hyperparameter optimization and for comparing model performance.

2.5 Modeling approach motivation

Informed by previous research,[3] the milling time feature alone was found to account for approximately 50% of the observed variance in yield. A primary concern was that this dominant predictor could mask the influence of the remaining process parameters. To address this potential overshadowing effect and better capture the contributions of the remaining features, we implemented a specialized two-step modeling strategy. The approach involves:

(1) Training a first model using only the 'time' feature to predict the yield.

(2) Calculating the residuals (actual yield minus the first model's prediction) on the training data.

(3) Training a second model using all other engineered features (excluding 'time') to predict these residuals.

(4) The final prediction is the sum of the predictions from the time model and the residual model.

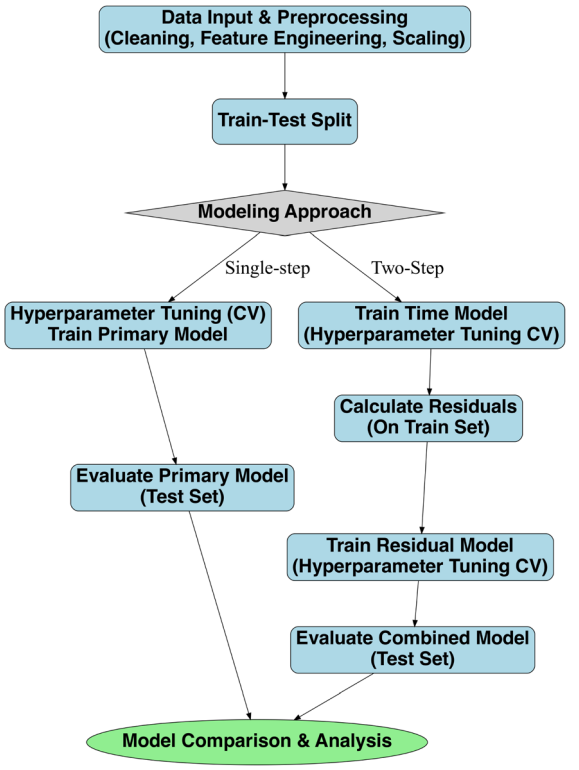


Fig. 1 Methodology workflow overview.

This allows the residual model to focus on explaining the yield variation not captured by the primary time trend. To rigorously assess the benefit of this specialized strategy, we also trained corresponding models directly on the full set of engineered features for direct performance comparison against their two-step counterparts. The general methodology is visualized in Fig. 1.

2.6 Machine learning algorithms and hyperparameter optimization

The machine learning algorithms described below were selected on the basis of their suitability and applicability to the current state of available experimental mechanochemistry data. It is worth highlighting that the feature engineering and two-step approach described in previous sections are omitted for the linear regression model, which is included solely as a baseline for comparison. For the GPR (Gaussian Process Regression), RF (Random Forest), SVR (Support Vector Regression), and XGBoost models, hyperparameter optimization was performed using the Tree-structured Parzen Estimator (TPE) algorithm implemented in the Hyperopt library. The objective was to minimize the weighted MSE, evaluated using repeated k-fold cross-validation of the training data with $k = 5$ splits and $n = 3$ repeats. We chose $k = 5$ as a pragmatic bias–variance compromise.[30] Given 21 samples, 5-fold yields validation folds of 4–5 samples (training folds of 16–17), whereas 10-fold would leave 2–3 per validation and leave-one-out CV only 1, both of which increase variance in hyperparameter comparisons.

A total of 100 function evaluations were assigned for most models, while SVR, known to be potentially slower to tune, was assigned 500 evaluations to ensure a thorough search.[31] The best hyperparameters found during this process were used to train the final model on the entire training set.

2.6.1 Linear regression. Linear regression is included, given its simplicity and the ability to assess how well the relationship between input and output variables can be captured with a linear relationship. Linear models have been widely used to relate physical–organic descriptors to outcomes and selectivity in reaction development.[23,32] The relationship between target Y and input variables X_i can be described as:[33]

$$Y = \beta_0 + \sum_{i=1}^{p} X_i \beta_i \tag{4}$$

2.6.2 Gaussian Process Regression (GPR). Gaussian Process Regression (GPR) is a supervised learning method that models a distribution over possible functions rather than fitting a single function directly.[34] At its core, GPR assumes that any set of observed points is drawn from a joint Gaussian distribution characterized by a mean function and a covariance (kernel) function. The kernel function defines how closely related any two points are, which in turn governs the smoothness and shape of the functions in the model. Thus, GPR is an attractive alternative, as it produces not only a prediction value but also a distribution, effectively giving confidence intervals for the outcome. Moreover, these functions can adapt as more data is collected for training, making it particularly applicable for small data sets. Given the scarce data available currently in mechanochemistry, it is a clear candidate until more data can be collected for deep models. For background on GPR in chemistry, including kernel design and uncertainty quantification in small-data settings, see the general overview in *Chem. Rev.*[35] and recent catalysis-focused reviews.[36]

In our formulation, we assume a zero mean function. This common choice is made when no strong prior knowledge about the mean exists; any systematic trends are then captured by the covariance (kernel) function, allowing the model to focus on the underlying correlation structure. Additionally, we use a composite kernel (eqn (5)) consisting of a constant scaling factor (C), a Matérn kernel (with smoothness parameter ν fixed at 1.5) since it is effective in modeling physical processes,[37] and a white noise kernel ($\sigma_n{}^2$) to account for observation noise.

$$k = Ck_{\text{Matern}}(\nu = 1.5) + k_{\text{WhiteKernel}}(\sigma_n{}^2), \tag{5}$$

$$k_{\text{Matern}}(r) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{l} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu} r}{l} \right), \tag{6}$$

where $r = |x - x'|$ is the distance between two inputs, l is the length scale, σ^2 is the signal variance, ν controls the smoothness, $\Gamma(\cdot)$ is the Gamma function, and $K_\nu(\cdot)$ is the modified Bessel function of the second kind.

Given a training set $\mathbf{X} = \{x_1, \ldots, x_N\}$ with outputs $\mathbf{y} = \{y_1, \ldots, y_N\}$ and a test set $\mathbf{X}^* = \{x_1^*, \ldots, x_m^*\}$, the joint distribution of the training outputs and the latent function values $\mathbf{f}^*$ at the test points is modeled as:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{f}^* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_n{}^2\mathbf{I} & K(\mathbf{X}, \mathbf{X}^*) \\ K(\mathbf{X}^*, \mathbf{X}) & K(\mathbf{X}^*, \mathbf{X}^*) \end{pmatrix}\right), \tag{7}$$

where σ_n2 denotes the noise variance. K denotes the covariance function computed from the composite kernel and is used to construct the covariance matrices for both the training data and the test data.

Conditioning on the training data, the predictive distribution for the latent function values at the test points is Gaussian with mean and covariance given by

$$\bar{\mathbf{f}}^* = K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n{}^2\mathbf{I}]^{-1}\mathbf{y}, \tag{8}$$

$$\text{cov}(\mathbf{f}^*) = K(\mathbf{X}^*, \mathbf{X}^*) - K(\mathbf{X}^*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_n{}^2\mathbf{I}]^{-1}K(\mathbf{X}, \mathbf{X}^*). \tag{9}$$

The kernel hyperparameters (l, $\sigma_n{}^2$, ν) and the noise variance $\sigma_n{}^2$ are estimated by maximizing the log marginal likelihood:

$$\begin{aligned} \text{Log } p(\mathbf{y}|\mathbf{X}) = &-\frac{1}{2}\mathbf{y}^\top \left[K(\mathbf{X}, \mathbf{X}) + \sigma_n{}^2\mathbf{I}\right]^{-1}\mathbf{y} \\ &-\frac{1}{2}\log\left|K(\mathbf{X}, \mathbf{X}) + \sigma_n{}^2\mathbf{I}\right| - \frac{N}{2}\log(2\pi). \end{aligned} \tag{10}$$

This structure allows flexibility in modeling the signal variance, smoothness, feature relevance, and noise level. The key hyperparameters optimized *via* Hyperopt are detailed in Table 2.

2.6.3 Random forest. A random forest (RF) is an ensemble algorithm that makes use of multiple decision trees to enhance performance and reduce over-fitting. Each tree is fed with different samples of the training data (*i.e.* a bootstrap) and at each node a random subset of features are used for decision making.[38] This introduces variability across trees and thus, errors made across different trees are averaged out in the final prediction. Given the confounded nature of variables in mechanochemical processes, tree-based algorithms are appealing due to the 'if-then' working principle, which can capture non-linear relationships. For instance, tree-based ensembles such as random forest are standard in chemoinformatics[39] and QSAR.[40] The overall prediction is given by eqn (11).[41]

$$\hat{f}(x) = \frac{1}{M}\sum_{m=1}^{M} h(x;\ \Theta_m), \tag{11}$$

where $\hat{f}(x)$ is the ensemble prediction for x, M is the number of trees in the forest, and $h(x; \Theta_m)$ denotes the prediction of the m-

Table 2 Hyperparameter search space for Gaussian-process regression (GPR)

Hyperparameter	Search space/value
Constant scaling (C)	$\mathcal{U}(0.1,\ 1000)$
Base length-scale (l)	$\mathcal{U}(0.05,\ 10)$
Noise variance (σ_n2)	$\mathcal{U}(10^{-5},\ 1.5)$
Matérn smoothness (ν)	Fixed at 1.5

Table 3 Hyperparameter search space for random-forest (RF) regression

Hyperparameter	Search space/value
Number of trees ($n_{\text{estimators}}$)	$\mathcal{U}(50,\ 300)$
Maximum depth	$\mathcal{U}(5,\ 30)$
Minimum samples split	$\mathcal{U}(2,\ 20)$
Minimum samples leaf	$\mathcal{U}(1,\ 10)$

Table 4 Hyperparameter search space for support–vector regression (SVR)

Hyperparameter	Search space/value
Regularization (C)	Log $\mathcal{U}(0.1,\ 100)$
Epsilon (ε)	$\mathcal{U}(0.001,\ 1)$
Gamma (γ)	Log $\mathcal{U}(0.001,\ 1)$

th tree. Here, Θ_m represents the random factors, such as the bootstrap sample and random feature selection, used in constructing the m-th tree.

Key hyperparameters were tuned to optimize performance as detailed in Table 3.

2.6.4 Support vector regression. A support Vector Regression (SVR) aims to find a function $f(x)$ that deviates from the target values y_i by a value no greater than ε for all training points, while remaining as flat as possible.[42] The resulting regression function takes the form:

$$\hat{f}(x) = \sum_{i=1}^{N} \left(\alpha_i - \alpha_i^*\right) K(x_i,\ x) + b, \tag{12}$$

where x_i are the training points (support vectors), α_i, α_i^* are non-negative Lagrange multipliers determined during optimization, and b is a bias term. The choice of the kernel function $K(x_i, x)$ allows capturing non-linear relationships. The optimization process finds these multipliers subject to constraints, including the crucial box constraint $0 \le \alpha_i, \alpha_i^* \le C$, where C is the regularization hyperparameter. This parameter $C > 0$ controls the trade-off between the flatness of $f(x)$ and the tolerance for errors larger than ε; a larger C allows less error but potentially a more complex function.

In SVR, the support vectors are defined by training data points that lie on or outside the boundary of the ε-insensitive tube. This characteristic of SVR is particularly advantageous in the current state of mechanochemical processes, where experimental data is scarce and studies typically explore the effect of only one or two parameters on yield. By concentrating on the most informative data points, it can uncover subtle nonlinear dependencies among multiple process parameters. SVR has long been part of the chemometrics toolkit for nonlinear calibration and classification.[43,44] The Radial Basis Function (RBF) kernel was employed:

$$K(x_i,\ x) = \exp\left(-\gamma \|x_i - x\|^2\right) \tag{13}$$

where the kernel parameter γ controls the influence of a single training example. The key hyperparameters C, ε, and γ, which influence the model's complexity, error tolerance, and kernel shape, respectively, were optimized using Hyperopt as detailed in Table 4.

2.6.5 XGBoost. XGBoost is another tree-based algorithm that, in contrast to RF which constructs independent trees and averages them, builds an ensemble of regression trees in a sequential form.[45] Its efficiency, ability to capture complex non-linear relationships and feature interactions, and sophisticated regularization techniques make it a powerful choice for predictive modeling tasks, particularly with structured or tabular data often encountered in chemical process optimization. The final prediction $\hat{f}(x)$ is the sum of the predictions from all M trees:

$$\hat{f}(x) = \sum_{m=1}^{M} f_m(x), \tag{14}$$

where $f_m(x)$ represents the prediction of the m-th tree, and M corresponds to the number of estimators.

The training process iteratively adds trees by minimizing an objective function $\mathcal{L}(\phi)$ that combines a loss term (measuring the difference between predictions and actual values) and a regularization term Ω (penalizing model complexity), summed over all trees:

$$\mathcal{L}(\phi) = \sum_{i=1}^{N} l\left(y_i,\ \hat{y}_i^{(M)}\right) + \sum_{m=1}^{M} \Omega(f_m), \tag{15}$$

where $l(y_i, \hat{y}_i^{(M)})$ is the loss for sample i after M trees (*e.g.*, squared error for regression), and $\hat{y}_i^{(M)}$ is the cumulative prediction. The regularization term for a single tree f is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2. \tag{16}$$

here, T is the number of leaves in the tree, and w is the vector of scores (weights) at the leaves. The hyperparameter γ represents the minimum loss reduction required to make a further partition on a leaf node, acting as a tree pruning mechanism. The term $\frac{1}{2}\lambda\|w\|^2$ is an L2 regularization on the leaf weights, where λ (typically fixed, *e.g.*, $\lambda = 1$ by default in XGBoost, and not tuned in this study) helps to prevent overfitting by shrinking the leaf scores.

The structural complexity of each individual tree f_m is primarily controlled by its maximum depth. The boosting process, which dictates how the ensemble is built, is further refined by several key hyperparameters: the learning rate (often denoted as η) scales the contribution of each new tree, reducing the impact of individual trees and preventing overfitting. The subsample specifies the fraction of training instances randomly sampled to grow each tree, introducing stochasticity and improving generalization. Sample-by-tree defines the fraction of features randomly sampled when constructing each tree (or each split), which further diversifies the trees and helps manage feature collinearity. These parameters, along with the number of estimators and γ, were optimized *via* Hyperopt. This careful tuning of the gradient boosting process, combined with its inherent regularization strategies, allows XGBoost to achieve

Table 5 Hyperparameter search space for extreme-gradient boosting (XGBoost)

Hyperparameter	Search space/value
Maximum depth (d_{max})	$\mathcal{U}_{int}(3, 10)$
Learning rate (η)	Log $\mathcal{U}(0.01, 0.3)$
Number of estimators (M)	$\mathcal{U}_{int}(100, 500)$
Row subsample (subsample)	$\mathcal{U}(0.5, 1)$
Column subsample (colsample_bytree)	$\mathcal{U}(0.5, 1)$
Gamma (γ)	$\mathcal{U}(0, 5)$

high accuracy while effectively mitigating overfitting.[46] For chemical best practices with XGBoost on tabular reaction/molecular data, we refer to the *J. Cheminf.* guidelines[47] and recent domain reviews in catalysis science.[36]

The main hyperparameters tuned are listed in Table 5.

3 Results and discussion

The primary objective of this study is to develop accurate predictive models that can link mechanical and chemical operational parameters to experimental mechanochemical yield. To assess this, we evaluate several machine learning algorithms using two distinct modeling strategies:

(1) A primary modeling approach, where each algorithm was trained directly on the full set of engineered features (either scaled or unscaled, as appropriate for the specific model).

(2) A two-step modeling approach, designed to address the potentially dominant influence of the 'time' feature. This involved first modeling the yield based on 'time' alone, and then modeling the residuals (the difference between actual yield and the time-model's prediction) using the remaining engineered features. The final prediction was the sum of the outputs from these two component models.

This dual approach allows for a comprehensive assessment of how well different algorithms capture the underlying relationships in the data, particularly concerning the prominent role of reaction time. It should be reiterated that the weighted MSE calculations discussed here reflect a configuration in which the yields above 70% were given a weight of 2.0, and all other yields a weight of 1.0.

3.1 Model performance evaluation

The performance of all trained models was evaluated on a held-out test set. To provide a multifaceted view of predictive accuracy, several standard regression metrics were employed, in addition to the weighted MSE already defined in eqn (3). These metrics are:

• Root Mean Squared Error (RMSE): this metric calculates the square root of the average of the squared differences between predicted and actual values. It is sensitive to large errors due to the squaring term. The RMSE is given by:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{\mathrm{true},i} - y_{\mathrm{pred},i}\right)^2} \tag{17}$$

where N is the total number of samples in the test set, $y_{\mathrm{true},i}$ is the actual yield for sample i, and $y_{\mathrm{pred},i}$ is the predicted yield for sample i. Lower RMSE values indicate better fit, and the metric shares the same units as the target variable (yield).

• Mean Absolute Error (MAE): MAE measures the average magnitude of errors in a set of predictions. It is the average over the test sample of the absolute differences between prediction and actual observation.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_{\mathrm{true},i} - y_{\mathrm{pred},i}\right| \tag{18}$$

MAE is less sensitive to outliers compared to RMSE and provides a straightforward interpretation of the average error magnitude, also in the units of the target variable.

• Mean Absolute Percentage Error (MAPE): MAPE expresses the average absolute difference between predicted and actual values as a percentage of actual values. This makes it a scale-independent metric, useful for comparing performance across datasets or models with different output scales.

$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_{\mathrm{true},i} - y_{\mathrm{pred},i}}{y_{\mathrm{true},i}}\right| \tag{19}$$

where $y_{\mathrm{true},i} \neq 0$. Lower MAPE values are desirable.

• Coefficient of determination (R^2): the R^2 score indicates the proportion of the variance in the dependent variable (yield) that is predictable from the independent variables (features).

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_{\mathrm{true},i} - y_{\mathrm{pred},i}\right)^2}{\sum_{i=1}^{N}\left(y_{\mathrm{true},i} - \bar{y}_{\mathrm{true}}\right)^2} \tag{20}$$

where $\bar{y}_{\mathrm{true}}$ is the mean of the true yield values in the test set. R^2 values range from $-\infty$ to 1, where 1 indicates a perfect fit, 0 indicates the model performs no better than predicting the mean of the target, and negative values indicate poorer performance than predicting the mean.

3.2 Comparison of modeling strategies and algorithm performance

The performance metrics for all evaluated models on the test set are summarized in Table 6, and the predictions can be visualized in Fig. 2.

The Linear Regression model, utilizing a selected subset of scaled features, registered a weighted MSE of 395.29 and an R^2 of 0.53. While simple and interpretable, its linear nature inherently limits its ability to capture the complex, non-linear dynamics typical of chemical reactions, including those in mechanochemistry. Nonetheless, it should be noted that when examining the primary (single-stage) versions of the more complex algorithms, most struggled to significantly outperform this baseline. For instance, the GPR (primary) model achieved a weighted MSE of 260.82 and an R^2 of 0.51, while RF (primary) yielded a weighted MSE of 354.31 and R^2 of 0.52, and SVR (primary) resulted in a weighted MSE of 346.48 and R^2 of 0.52. On average, these primary models offered only a modest

Table 6 Comparison of regression models on several performance metrics

Model	RMSE	MAE	MAPE [%]	R^2	Weighted MSE
Linear regression	15.50	14.11	37.49	0.53	395.29
GPR (primary)	15.88	11.09	31.39	0.51	260.82
GPR (two-step)	9.43	7.48	26.59	0.83	93.37
RF (primary)	15.65	15.01	49.86	0.52	354.31
RF (two-step)	12.52	10.29	27.42	0.69	177.17
SVR (primary)	15.66	14.66	48.57	0.52	346.48
SVR (two-step)	15.36	14.73	50.20	0.54	320.32
XGBoost (primary)	12.65	11.20	32.68	0.69	194.37
XGBoost (two-step)	11.06	8.79	24.79	0.76	139.56

reduction in weighted MSE (approximately 20–35% improvement over baseline) and showed R^2 values very close to, or even slightly below, that of the simpler linear model. The XGBoost (primary) model was an exception, showing a marked improvement with a weighted MSE of 194.37 and an R^2 of 0.69; thus, when no prior domain knowledge is available, it should be the first-line choice, providing both competitive accuracy and an initial, data-driven ranking of influential variables. This general difficulty of the primary models to substantially advance beyond the linear regression baseline underscores the dominant influence of the 'time' variable, which, when not explicitly addressed, appears to overshadow the contributions of other features in these conventional modeling approaches.

Thus, a clear and consistent finding from these results is a significant benefit of the two-step modeling approach for several algorithms. The GPR two-step model stands out, achieving the lowest weighted MSE (93.37), MAE (7.48), and RMSE (9.43), alongside the highest R^2 value (0.83) among all models tested. Beyond its strong predictive accuracy, GPR can provide uncertainty estimates (confidence intervals) for its

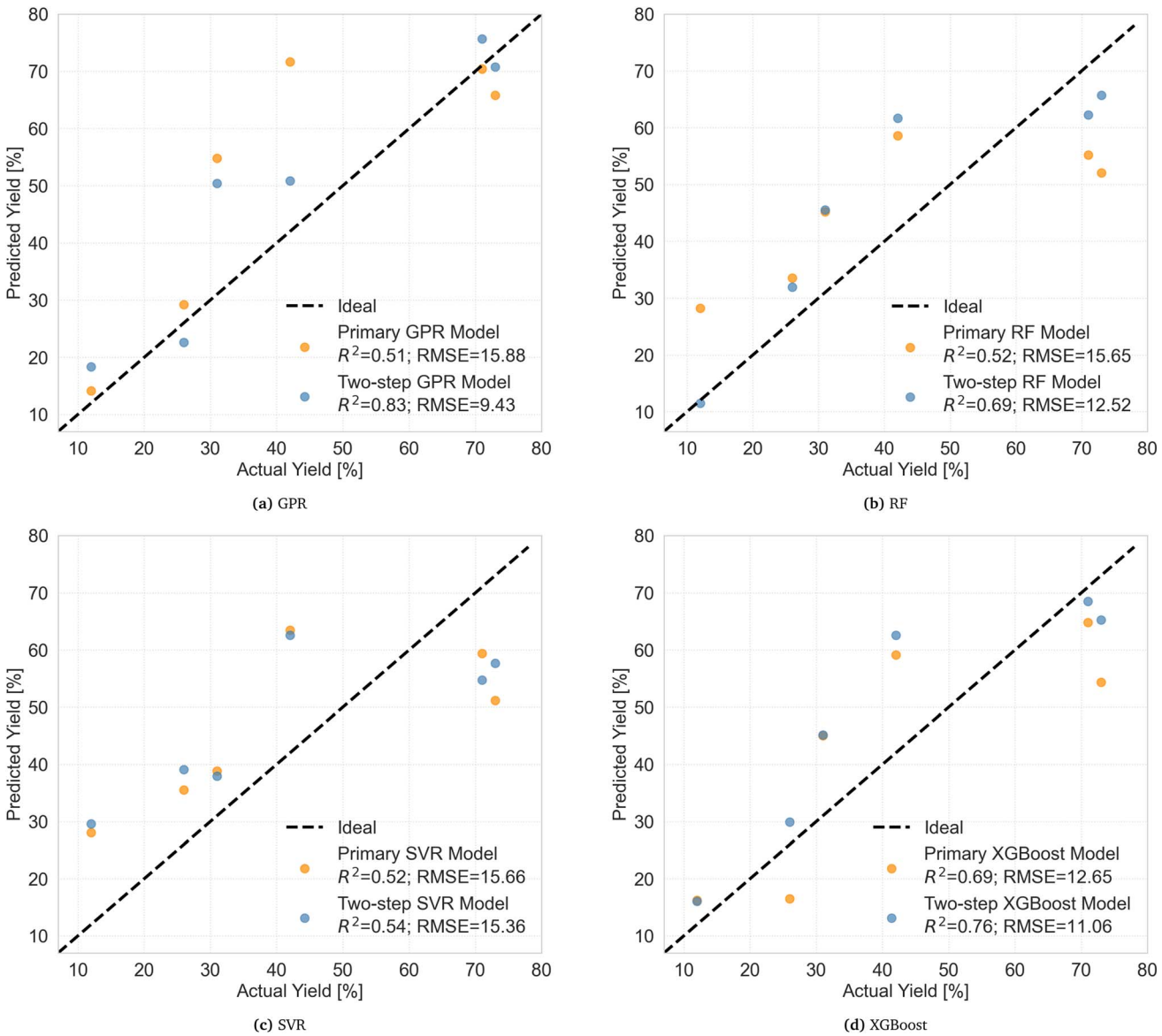


Fig. 2 Predictions on the test set from primary and two-step variants of each regressor: (a) GPR, (b) RF, (c) SVR and (d) XGBoost.

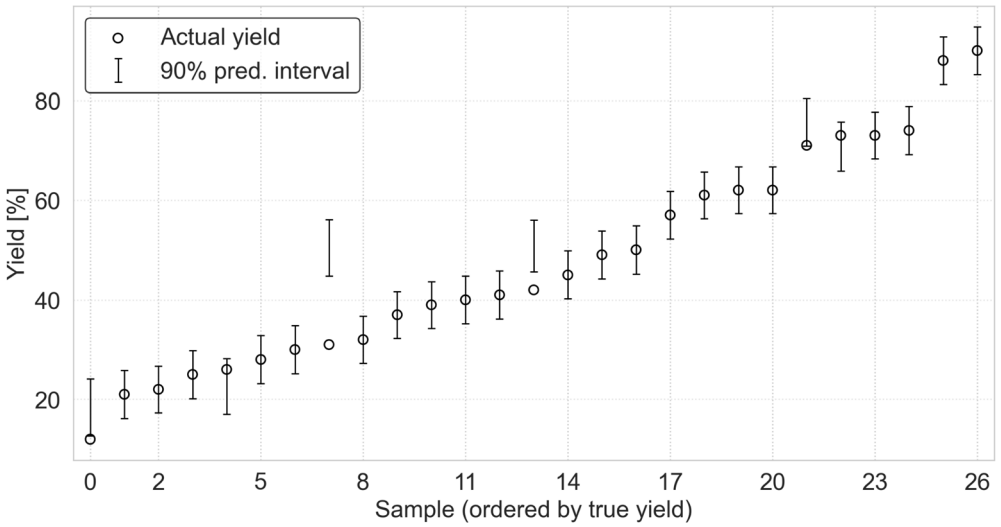
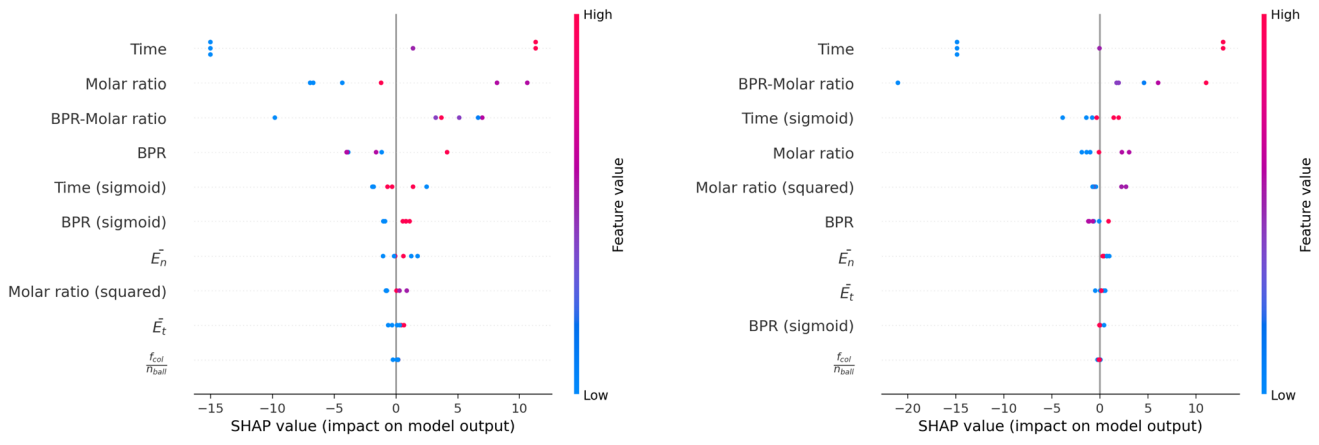


Fig. 3 Actual yields and predictions from the two-step GPR model. Points show the actual yield and error bars show 90% predictive intervals.

predictions (see Fig. 3). Here, the effect of the weighted objective is evident; cases above 70% yield exhibit closer agreement with the model, indicating that errors at high yield were effectively down-weighted during training. This error scale is valuable for guiding future experiments and assessing prediction reliability, especially when dealing with limited or costly experimental data, a common scenario in developing fields like mechanochemistry. The adaptability of its kernel functions also allows for encoding prior knowledge about the process, if available. The superior performance of the two-step GPR suggests that by first isolating the primary time trend, the GPR framework could more effectively model the subtle, potentially non-linear interactions among the remaining process parameters through its covariance structure.

To further delve into the interpretability of the more complex non-linear models, particularly the tree-based ensembles, SHAP (SHapley Additive exPlanations) analysis was employed for the two-step variants of RF and XGBoost. SHAP values provide a unified measure of feature importance by attributing to each feature the change in the expected model prediction when conditioning on that feature. A SHAP summary plot visualizes these attributions: each point represents a SHAP value for a feature and an instance, where the position on the x-axis indicates the impact on the model output (positive or negative), and the color represents the feature's value (high or low). Features are ranked by the sum of absolute SHAP values across all samples.

The XGBoost two-step model also demonstrated considerable improvements with the second strategy, emerging as the second-best performing model with a weighted MSE of 139.56 and an R^2 of 0.76. Tree-based ensemble methods like XGBoost are inherently capable of capturing complex non-linear relationships and variable interactions. The SHAP summary plot for this model (Fig. 4(a)) reveals that 'time' remains, as expected, the most influential feature for the overall two-step prediction, with higher time values generally pushing the prediction higher



(a) XGBoost (two-step) (b) Random forest (two-step)

Fig. 4 SHAP summary plots for (a) the two-step XGBoost model and (b) the two-step random-forest model. See main text for interpretation.

(positive SHAP values). Following 'time', features such as 'molar ratio' and the interaction 'BPR-molar ratio' exhibit significant importance, where higher values of these ratios tend to positively influence the predicted yield. These results align with our previous findings.[3] 'BPR' and 'time (sigmoid)' also show discernible impacts. It should be highlighted that, while the use of engineered features can be beneficial for capturing complex relationships and potentially improving model accuracy, it underscores a common trade-off in machine learning: a balance must often be struck between the enhanced predictive power gained from feature engineering and the goal of maintaining straightforward interpretability in terms of original process parameters. Despite this consideration, the two-step approach has effectively enabled the two-step XGBoost model to focus its learning capabilities on variance not explained by time, leading to more accurate predictions.

The Random Forest (RF) two-step model also benefited from the residual strategy, with a weighted MSE of 177.17 and an R^2 of 0.69. Similar to XGBoost, RF models can effectively map non-linearities and interactions. The SHAP summary plot for the RF two-step model (Fig. 4(b)) shows a similar pattern of feature importance. 'Time' is again paramount, with a wide spread of SHAP values. 'BPR-molar ratio' and 'molar ratio' are the next most impactful features, with higher values generally increasing the predicted yield. Other engineered features like 'molar ratio (squared)' and 'time (sigmoid)' also contribute, though to a lesser extent than the top three. For both tree-based models, the features related to energy input ('$\bar{E}_n$', '$\bar{E}_t$') and collision frequency $\left(\frac{f_{col}}{n_{ball}}\right)$ show relatively lower overall SHAP values, suggesting a smaller impact on the output of these two-step models compared to the primary chemical and time-related parameters. However, it is crucial to reiterate that these mechanical milling properties were obtained under a constant fill ratio, and their influence is subject to change as this parameter is varied. Furthermore, this aligns with our previous finding that rotational speed, which has been abstracted into these variables, has a relatively lower relevance compared to the rest of the operational variables when maintaining a constant fill ratio.[3]

For Support Vector Regression (SVR), the two-step approach provided a minimal improvement in weighted MSE (320.32 for two-step *vs.* 346.48 for primary) and R^2 (0.540 *vs.* 0.522 for primary). SVR models, particularly with non-linear kernels like RBF, can be effective in high-dimensional spaces and are less sensitive to the dimensionality of the feature space. Their reliance on support vectors (a subset of training data) can make them memory efficient. However, in this case, the gains from the two-step strategy were less pronounced compared to GPR and the tree-based ensembles, suggesting that the primary SVR model might have already captured much of the structure the two-step approach aimed to resolve, or that its specific way of defining the decision boundary was less amenable to this sequential decomposition. As such, the SVR should not be investigated further until more data can be collected.

In summary, the GPR two-step model is the top-performing model across most key metrics based on the current dataset and evaluation criteria. The two-step modeling approach proved to be highly advantageous, particularly for GPR, XGBoost, and RF, significantly enhancing their predictive accuracy, especially when considering the weighted error. These results emphasize the importance of considering tailored modeling strategies. The distinct characteristics of each algorithm (*i.e.*, GPR's probabilistic outputs, the non-linear mapping capabilities and interpretability *via* SHAP of tree-based ensembles, and SVR's margin-based optimization) offer different strengths beyond raw predictive power. Therefore, the ultimate choice of predictive model in mechanochemical studies, or indeed any application, should not be solely dictated by a narrow focus on performance metrics. For instance, a model that performs slightly worse on a specific metric might be preferred if its intrinsic properties, such as superior interpretability, the ability to quantify uncertainty (as with GPR), or robustness to certain data characteristics, align more closely with the specific goals of the investigation or the practical constraints of its application. Factors such as data availability, the cost of acquiring more data, the need for uncertainty quantification for decision-making, and the desired level of insight into the underlying process mechanisms must be weighed alongside predictive accuracy.

3.3 Model generalization under mechanical regime change

To probe prospective generalization beyond the training distribution, we evaluated the two best-performing models on two new milling conditions (Table 7). In the original dataset (Table 1), the distribution of mechanical stressing conditions was effectively held constant, with a dissipation ratio of … Therefore, we expose the model to an unseen mechanical regime where the dissipation ratio is tuned to increase the dominance of tangential dissipation $\bar{E}_t/\bar{E}_n = 3.26$. This is practically achieved by reducing the fill ratio in the Emax ball mill.

We scaled the features with the training scaler and obtained predictions from the two-step GPR and XGBoost without refitting. Table 8 reports point predictions and absolute errors relative to the measured yields.

In case 27, the fill ratio (6%) departs from the training domain. The two-step models under-predict by ≈11 percentage points, which is broadly consistent with their held-out RMSE (≈9.43) and indicates that a modest shift toward a more

Table 7 Out-of-sample milling conditions used for the generalization check. $\bar{E}_n$ and $\bar{E}_t$ are the mean normal and tangential energy dissipated per collision from DEM; f_{col}/n_{ball} is the specific collision frequency per ball. For these two cases the fill ratio was 6%, and the dissipation ratio increased from $\bar{E}_t/\bar{E}_n = 2.27$ in Table 1 to 3.26 here, *i.e.*, a more tangential-dominated stressing regime. Rotational speeds were 600 rpm (case 27) and 788 rpm (case 28)

Case	BPR	Mol ratio	Time [h]	$\bar{E}_n$ [μJ]	$\bar{E}_t$ [μJ]	f_{col}/n_{ball} [s^{-1}]	Yield [%]
27	30	10	12.5	73	238	700	84
28	30	10	12.5	126	411	920	94

Table 8 Predictions on the two out-of-sample cases. Absolute errors are in percentage points of yield

Case	$Yield_{true}$ [%]	$GPR_{two\text{-}step}$ [%]	$XGB_{two\text{-}step}$ [%]	$\lvert GPR_{two\text{-}step} - true \rvert$	$\lvert XGB_{two\text{-}step} - true \rvert$
27	84.0	73.14	72.42	10.86	11.58
28	94.0	74.09	72.42	19.91	21.58

tangentially dominated stressing state can be tolerated when other operating factors remain consistent. However, in case 28, the fill ratio and rotational speed are shifted simultaneously, which further increases tangential stressing and the specific collision frequency. Errors rise to 20–22 percentage points, showing that the tangential mechanical regime alters the influence of the rotational speed on yield in a way that the models have not yet learned, leading to systematic under-prediction. As more variables move outside the training distribution, errors compound because of unseen nonlinear interactions between operational variables. This underscores the value of an expandable dataset design: targeted additions will expose these interactions and enable refitting for reliable use under regime changes. Because the DEM descriptors are mill-agnostic, different groups can explore the variables and ranges most relevant to them, and the pooled data will steadily improve accuracy and generalization.

4 Conclusions

In this study, we have demonstrated the effectiveness and applicability of various machine learning algorithms to link mechanical and chemical parameters of mechanochemistry to predict conversion yield in the regeneration of $NaBH_4$ from a system of $NaBO_2 \cdot 4H_2O$ and MgH_2. We have evaluated two distinct modeling strategies designed to account for the scarcity of data and the dominant influence of milling time in the process. Our findings indicate that carefully selected and configured ML models can provide valuable predictive capabilities, offering a pathway to optimize experimental efforts at a fraction of the time compared to the classic 'trial and error' approach, and gain deeper insights into the complex interplay of parameters in mechanochemistry.

The most compelling predictive performance was achieved by the Gaussian Process Regression (GPR) two-step model, which consistently outperformed all other evaluated algorithms across key metrics, including the lowest weighted MSE (93.37) and the highest R^2 (0.83). Following GPR, the two-step XGBoost and Random Forest models also delivered strong results. Beyond mere predictive accuracy, the choice of an appropriate ML model should also be guided by the specific objectives of the research and the intrinsic characteristics of the algorithms.

The practical implications of this work are significant. By developing reliable predictive models, researchers can substantially reduce the number of exploratory experiments, leading to considerable savings in time, materials, and energy. This is particularly pertinent given the current state of mechanochemistry, where experiments can be resource-intensive and exploratory. More fundamentally, this study aimed to showcase a methodology for employing machine learning to bridge the elusive gap between the mechanical and chemical parameters of a mechanochemical process and the resulting yield outcomes. While the current data set provided a strong starting point, further exploration with more variability in mechanical conditions will enhance this linkage. Currently, performance degrades when multiple variables move outside the training domain, showcasing regime-dependent, nonlinear interactions. To facilitate this, the invariant mechanical characterization used in the dataset makes it readily expandable by independent experiments. Future work should focus on incorporating such expanded datasets, potentially exploring additional feature engineering techniques and advanced deep learning architectures once data volume permits.

Author contributions

Santiago Garrido Nuñez: writing – review & editing, writing – original draft, visualization, validation, methodology, investigation, formal analysis, data curation, conceptualization. Dingena L. Schott: writing – review & editing, supervision. Johan T. Padding: writing – review & editing, funding acquisition, supervision, conceptualization.

Conflicts of interest

There are no conflicts to declare.

Data availability

Data for this article is available at 4TU.ResearchData at **https://doi.org/10.4121/19639371-6f45-4f4b-882e-68aacc6a53a5**.

Acknowledgements

This project has received funding from the Ministry of Economic Affairs and Climate Policy, RDM regulation, carried out by the Netherlands Enterprise Agency (RvO). This work was supported by the project SH2IPDRIVE: Sustainable Hydrogen Integrated Propulsion Drives, funded by the RVO under grant MOB21013.

References

1 J. Batteas and T. Friščić, *RSC Mechanochem.*, 2025, **2**, 175–177.
2 V. Martinez, T. Stolar, B. Karadeniz, I. Brekalo and K. Užarević, *Nat. Rev. Chem.*, 2023, **7**, 51–65.

3 S. Garrido Nuñez, D. L. Schott and J. T. Padding, *Int. J. Hydrogen Energy*, 2025, **97**, 640–648.
4 A. H. Hergesell, C. L. Seitzinger, J. Burg, R. J. Baarslag and I. Vollmer, *RSC Mechanochem.*, 2025, **2**, 263–272.
5 F. Basoccu, P. Caboni and A. Porcheddu, *ChemSusChem*, 2025, e202402547.
6 A. Y. Ibrahim, R. T. Forbes and N. Blagden, *CrystEngComm*, 2011, **13**, 1141–1152.
7 A. Stolle, R. Schmidt and K. Jacob, *Faraday Discuss.*, 2014, **170**, 267–286.
8 S. Garrido Nuñez, D. L. Schott and J. T. Padding, *Powder Technol.*, 2025, **457**, 120919.
9 O. F. Jafter, S. Lee, J. Park, C. Cabanetos and D. Lungerich, *Angew. Chem., Int. Ed.*, 2024, **63**, e202409731.
10 J. F. Reynes, V. Isoni and F. García, *Angew. Chem., Int. Ed.*, 2023, **62**, e202300819.
11 M. Senna, *Powders*, 2023, **2**, 659–677.
12 J. Batteas, K. G. Blank, E. Colacino, F. Emmerling, T. Friščić, J. Mack, J. Moore, M. E. Rivas and W. Tysoe, *RSC Mechanochem.*, 2025, **2**, 10–19.
13 E. Angloua, Y. Chang, W. Bradley, C. Sievers and F. Boukouvala, *ACS Sustain. Chem. Eng.*, 2024, **12**, 9003–9017.
14 J. Yu, K. Raju, S.-H. Jin, Y. Lee and H.-K. Lee, *Int. J. Adv. Des. Manuf. Technol.*, 2022, **123**, 3451–3462.
15 Y. Li, J. Bao, T. Chen, A. Yu and R. Yang, *Powder Technol.*, 2022, **403**, 117409.
16 M. Boyer, D. Tabor, T.-H. Chao and D. Williams, *Center for the Mechanical Control of Chemistry Database: Interface*, 2025, DOI: **10.5281/zenodo.14827611**.
17 D. Tan and F. García, *Chem. Soc. Rev.*, 2019, **48**, 2274–2292.
18 A. A. L. Michalchuk, I. A. Tumanov, S. Konar, S. A. J. Kimber, C. R. Pulham and E. V. Boldyreva, *Advanced Science*, 2017, **4**, 1700132.
19 T. Jarg, J. Tamm, E. Suut-Tuule, K.-M. Lootus, D. Kananovich and R. Aav, *RSC Mechanochem.*, 2025, **2**, 507–515.
20 C. Weidenthaler, *Crystals*, 2022, **12**, 345.
21 S. M. Mennen, C. Alhambra, C. L. Allen, M. Barberis, S. Berritt, T. A. Brandt, A. D. Campbell, J. Castañón, A. H. Cherney, M. Christensen, D. B. Damon, J. Eugenio de Diego, S. García-Cerrada, P. García-Losada, R. Haro, J. Janey, D. C. Leitch, L. Li, F. Liu, P. C. Lobben, D. W. C. MacMillan, J. Magano, E. McInturff, S. Monfette, R. J. Post, D. Schultz, B. J. Sitter, J. M. Stevens, I. I. Strambeanu, J. Twilton, K. Wang and M. A. Zajac, *Org. Process Res. Dev.*, 2019, **23**, 1213–1242.
22 X. Caldentey and E. Romero, *Chem.: Methods*, 2023, **3**, e202200059.
23 C. B. Santiago, J.-Y. Guo and M. S. Sigman, *Chem. Sci.*, 2018, **9**, 2398–2412.
24 L. M. Roch, F. Häse, C. Kreisbeck, T. Tamayo-Mendoza, L. P. E. Yunker, J. E. Hein and A. Aspuru-Guzik, *PLoS One*, 2020, **15**, 1–18.
25 D. Reker, E. A. Hoyt, G. J. Bernardes and T. Rodrigues, *Cell Rep. Phys. Sci.*, 2020, **1**, 100247.
26 N. H. Angello, V. Rathore, W. Beker, A. Wołos, E. R. Jira, R. Roszak, T. C. Wu, C. M. Schroeder, A. Aspuru-Guzik, B. A. Grzybowski and M. D. Burke, *Science*, 2022, **378**, 399–405.
27 A. I. Leonov, A. J. S. Hammer, S. Lach, S. H. M. Mehr, D. Caramelli, D. Angelone, A. Khan, S. O'Sullivan, M. Craven, L. Wilbraham and L. Cronin, *Nat. Commun.*, 2024, **15**, 1240.
28 N. H. Angello, D. M. Friday, C. Hwang, S. Yi, A. H. Cheng, T. C. Torres-Flores, E. R. Jira, W. Wang, A. Aspuru-Guzik, M. D. Burke, C. M. Schroeder, Y. Diao and N. E. Jackson, *Nature*, 2024, **633**, 351–358.
29 M. Kessler and R. Rinaldi, *Front. Chem.*, 2022, **9**, 816553.
30 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.
31 R. Mantovani, A. Rossi, J. Vanschoren and A. Carvalho, *International Workshop on Meta-Learning and Algorithm Selection*, 2015, pp. 80–92.
32 M. S. Sigman, K. C. Harper, E. N. Bess and A. Milo, *Acc. Chem. Res.*, 2016, **49**, 1292–1301.
33 T. T. Cai and P. Hall, *Ann. Math. Stat.*, 2006, **34**, 2159–2179.
34 M. Hashemitaheri, S. M. R. Mekarthy and H. Cherukuri, *Procedia Manufacturing*, 2020, **48**, 1000–1008.
35 V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti and G. Csányi, *Chem. Rev.*, 2021, **121**, 10073–10141.
36 B. M. Abraham, M. V. Jyothirmai, P. Sinha, F. Viñes, J. K. Singh and F. Illas, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2024, **14**, e1730.
37 C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006.
38 L. Breiman, *Mach. Learn.*, 2001, **45**, 5–32.
39 Y.-C. Lo, S. E. Rensi, W. Torng and R. B. Altman, *Drug Discovery Today*, 2018, **23**, 1538–1546.
40 S. K. Niazi and Z. Mariam, *Int. J. Mol. Sci.*, 2023, **24**, 14.
41 G. Biau and E. Scornet, *TEST*, 2016, **25**, 197–227.
42 A. J. Smola and B. Schölkopf, *Statistics and Computing*, 2004, **14**, 199–222.
43 H. Li, Y. Liang and Q. Xu, *Chemom. Intell. Lab. Syst.*, 2009, **95**, 188–198.
44 R. G. Brereton and G. R. Lloyd, *Analyst*, 2010, **135**, 230–267.
45 T. Chen and C. Guestrin, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.
46 J. H. Friedman, *Ann. Stat.*, 2001, **29**, 1189–1232.
47 D. Boldini, F. Grisoni, D. Kuhn, L. Friedrich and S. A. Sieber, *J. Cheminf.*, 2023, **15**, 73.