computation

MDPI

Review

Mathematical Optimization in Machine Learning for Computational Chemistry

Ana Zekić

Department of Mathematical Sciences, Faculty of Technology and Metallurgy, University of Belgrade, 11000 Belgrade, Serbia; azekovic@tmf.bg.ac.rs

Abstract

Machine learning (ML) is transforming computational chemistry by accelerating molecular simulations, property prediction, and inverse design. Central to this transformation is mathematical optimization, which underpins nearly every stage of model development, from training neural networks and tuning hyperparameters to navigating chemical space for molecular discovery. This review presents a structured overview of optimization techniques used in ML for computational chemistry, including gradient-based methods (e.g., SGD and Adam), probabilistic approaches (e.g., Monte Carlo sampling and Bayesian optimization), and spectral methods. We classify optimization targets into model parameter optimization, hyperparameter selection, and molecular optimization and analyze their application across supervised, unsupervised, and reinforcement learning frameworks. Additionally, we examine key challenges such as data scarcity, limited generalization, and computational cost, outlining how mathematical strategies like active learning, meta-learning, and hybrid physics-informed models can address these issues. By bridging optimization methodology with domain-specific challenges, this review highlights how tailored optimization strategies enhance the accuracy, efficiency, and scalability of ML models in computational chemistry.

Keywords: mathematical optimization; machine learning in chemistry; computational chemistry; Bayesian optimization

check for updates

Academic Editors: Aleksey E. Kuznetsov, Alexander S. Novikov and Felipe Fantuzzi

Received: 6 June 2025
Revised: 21 June 2025
Accepted: 9 July 2025
Published: 11 July 2025

Citation: Zekić, A. Mathematical Optimization in Machine Learning for Computational Chemistry. *Computation* **2025**, *13*, 169. https://doi.org/10.3390/computation13070169

Copyright: © 2025 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Machine learning (ML) has become a cornerstone of computational chemistry, enabling the prediction of molecular properties, materials discovery, and reaction modeling with unprecedented speed and accuracy. However, the performance of ML models critically depends on mathematical optimization techniques.

Optimization plays a central role at multiple levels of the ML pipeline. It is used to minimize loss functions, fine-tune hyperparameters, select data points in active learning, and ensure stable training of deep architectures such as graph neural networks (GNNs). These tasks are especially important in chemistry, where datasets are often high-dimensional, noisy, and computationally expensive to generate.

In this review, we examine how mathematical optimization supports diverse ML tasks in computational chemistry. Rather than limiting the discussion to a single application domain, we illustrate optimization's versatility across a range of representative challenges, from general property prediction to quantum-level modeling tasks such as learning interatomic potentials. This approach enables us to highlight both methodological depth and domain-specific relevance.

We review core optimization methods widely used in chemical ML workflows, including gradient-based algorithms like stochastic gradient descent (SGD) and Adam, global optimization methods like Bayesian optimization and Monte Carlo techniques, and spectral methods applied in graph-based models. Each method is discussed in terms of its mathematical foundation, implementation strategy, and relevance to applications such as quantum chemistry, molecular design, and supervised or unsupervised learning.

Throughout this review, we use the term optimization in a broad sense, encompassing (1) model parameter learning, (2) hyperparameter tuning, and (3) search over molecular or latent input space. Each of these targets presents distinct challenges and is addressed using different mathematical approaches, which we clarify throughout the following sections.

Finally, we highlight ongoing challenges, such as data scarcity, transferability, and computational trade-offs, and explore how optimization frameworks can help address these limitations. Our goal is to provide both a conceptual and practical guide for researchers applying optimization in chemical machine learning.

2. Optimization Methods in Machine Learning for Chemistry

Machine learning models rely on an objective function known as the loss function, which quantifies the error between the model's predictions and the true values. The goal of mathematical optimization in machine learning is to minimize the loss function by iteratively adjusting the model parameters. Different optimization techniques are applied to efficiently navigate the parameter space and improve machine learning models. In the context of computational chemistry, these techniques ensure convergence, enhance accuracy, and reduce computational costs. Below, we discuss key optimization methods employed in machine learning for chemical applications.

2.1. Optimization Targets and Learning Settings

In the context of machine learning applied to chemistry, the term "optimization" can refer to several distinct processes, each targeting a different component of the modeling pipeline. These include the following:

- Model parameter optimization: This refers to the adjustment of internal model weights during training to minimize a predefined loss function. Common methods include stochastic gradient descent (SGD), Adam, and other gradient-based optimizers. This process is central to supervised learning tasks such as molecular property prediction or spectroscopic signal modeling.
- Hyperparameter optimization: Hyperparameters, such as the learning rate, number of layers, and regularization coefficients, are not learned during training and must be selected externally. Methods such as grid search, random search, and Bayesian optimization are commonly used to identify optimal hyperparameter configurations that maximize model performance on validation sets.
- Molecular optimization: In generative tasks or molecular design, the optimization target is not the model itself but rather the molecular input or its latent representation. The goal is to discover new chemical structures that maximize or minimize desired properties, such as solubility or reactivity. This type of molecular optimization is typically approached via Bayesian optimization, reinforcement learning, or differentiable surrogate models.

Although these forms of optimization share algorithmic foundations, they differ significantly in their objectives, evaluation criteria, and constraints. Throughout this review, we highlight how mathematical optimization contributes to each of these settings.

In addition to the optimization targets, chemical machine learning tasks also differ in learning settings. In supervised learning, models are trained using labeled data, such as

molecular structures paired with known properties (e.g., energy, solubility, and toxicity). This setting is dominant in property prediction tasks. In contrast, unsupervised learning focuses on extracting patterns from unlabeled data, often used for clustering molecular fingerprints or learning low-dimensional latent representations. Reinforcement learning, though less common, is increasingly applied in molecular optimization and generative design, where the model interacts with an environment (e.g., chemical space) and is rewarded for producing molecules with desired properties.

Optimization strategies in machine learning differ not only in algorithmic formulation but also in search behavior and scope. Some methods are designed for local optimization, refining solutions by following local gradients toward nearby minima. Others are formulated for global optimization, aiming to search more broadly and escape local traps by evaluating diverse regions of the solution space.

These behaviors map onto the concepts of exploitation, leveraging known high-performing regions of the search space, and exploration, which prioritizes information gain from uncertain or underexplored regions. The balance between these two objectives is central to designing effective optimization routines, particularly in chemical applications where objective functions may be multi-modal, non-convex, or expensive to evaluate.

2.2. Stochastic Gradient Descent (SGD)

Stochastic gradient descent (SGD) is a foundational optimization algorithm widely used in training machine learning models, particularly deep neural networks. It belongs to the family of first-order methods and operates by iteratively updating model parameters in the direction that minimizes a given loss function. Unlike full-batch gradient descent, which computes the gradient using the entire dataset, SGD estimates the gradient using a single randomly selected sample or a small mini-batch. This approach introduces stochasticity into the learning process and reduces the computational cost per iteration [1].

To better understand its mechanics and relevance to chemical learning tasks, we next present the mathematical formulation of SGD and its key variants. The update rule for SGD is provided by the following:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla L(\boldsymbol{\theta}_t; \boldsymbol{x}_i, y_i). \tag{1}$$

We use bold symbols (e.g., $\boldsymbol{\theta}_t$, $\boldsymbol{x}_i$) to denote vectors and regular font for scalar quantities. Here, $\boldsymbol{\theta}_t$ represents the model parameters at iteration t, η is the learning rate, and $\nabla L(\boldsymbol{\theta}_t; \boldsymbol{x}_i, y_i)$ is the gradient of the loss function with respect to the model parameters, computed using input $\boldsymbol{x}_i$ and the true label y_i. In the context of chemical machine learning, $\boldsymbol{x}_i$ could represent molecular descriptors or graph embeddings, while y_i could be a quantum chemical property such as dipole moment, energy gap, or solvation energy.

While SGD is fundamentally a local optimization method, relying on gradient information at each step, its stochasticity introduces small-scale exploration, which can help the model avoid sharp local minima without providing true global search capabilities [1]. However, it also introduces noise, which may destabilize convergence if not properly controlled.

To improve performance, several enhanced variants of SGD have been proposed:

- Momentum-based SGD incorporates an exponentially weighted average of past gradients to smooth updates and accelerate convergence, particularly in ravine-shaped loss landscapes.
- Nesterov accelerated gradient (NAG) improves upon classical momentum by computing the gradient not at the current position but at the anticipated future position of the parameters. This often leads to faster convergence in practice [2].
- Mini-batch SGD uses batches of 16–256 samples to strike a balance between noisy single-sample updates and slow full-batch updates.

These variants mitigate the effects of noisy gradients and help stabilize training, especially in deep architectures used in quantum chemistry or molecular design tasks.

A representative application in chemical machine learning is the work of Rupp et al. [3], who trained neural networks using mini-batch SGD to predict molecular atomization energies in the QM7 dataset based on Coulomb matrix descriptors. This approach demonstrated that SGD could efficiently scale to chemically diverse datasets while maintaining predictive accuracy.

While these techniques improve the basic behavior of SGD, its performance still depends heavily on the choice of hyperparameters, such as the learning rate and batch size.

This sensitivity has motivated the development of optimizers such as RMSprop [4], which adapt the learning rate using exponentially decaying averages of squared gradients. By adjusting the step size dynamically, these methods improve convergence in noisy or curved loss landscapes often encountered in chemical datasets.

The limitations of both SGD and RMSprop have led to further developments, such as the Adam optimizer, which combines ideas from both momentum and adaptive learning rates. This method will be discussed in the following section.

2.3. Adam Optimizer

Adam (adaptive moment estimation) is an extension of SGD incorporating adaptive learning rates for better convergence. It combines the benefits of momentum-based acceleration and adaptive learning rates to improve convergence. Introduced by Kingma and Ba [5], Adam dynamically adjusts learning rates based on first and second moment estimates of the gradients, making it robust to noisy updates and effective across a wide range of machine learning applications. These two estimates are denoted by $\boldsymbol{m_t}$ and $\boldsymbol{v_t}$, respectively, and are used to scale the update step for each parameter individually.

The Adam update rule is provided by the following:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\hat{\boldsymbol{m}}_t}{\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon}, \tag{2}$$

where

- $\hat{\boldsymbol{m}}_t = \frac{\boldsymbol{m_t}}{1-\beta_1^t}$, $\hat{\boldsymbol{v}}_t = \frac{\boldsymbol{v_t}}{1-\beta_2^t}$ are bias-corrected estimates;
- β_1 and β_2 are hyperparameters controlling the decay rates of the moment estimates (commonly set to 0.9 and 0.999, respectively);
- ϵ is a small constant added to avoid division by zero;
- η is the learning rate.

The model takes into account previous steps, which helps achieve faster and more stable convergence. The first moment helps reduce oscillations, allowing the optimization process to move more consistently toward the global minimum rather than varying unpredictably in different directions. Despite this improved stability, it is important to note that Adam remains a local optimization method. Its adaptive update mechanism enables smoother convergence within the local loss landscape, but it does not perform global search or incorporate mechanisms for broader exploration, unlike methods such as Bayesian optimization.

One important class of problems in computational chemistry involves learning quantum-level properties from data, including total energies, electron densities, and molecular potential energy surfaces. These quantities, typically derived from first-principles methods like density functional theory (DFT), are computationally intensive to calculate, motivating the use of machine learning models trained to approximate them.

Schütt et al. (2017) [6] developed a neural network-based approach for approximating DFT calculations and predicting electronic structure properties. Their model was trained using the Adam optimizer on molecular datasets to learn relationships between electron density distributions, total energies, and molecular potential energy surfaces. In particular, their approach approximates wavefunctions and potential energy surfaces, enabling efficient predictions of electronic properties. Their study evaluates the capability of machine learning models to reproduce DFT-derived quantities while analyzing the trade-off between computational cost and predictive accuracy.

Later, Wu et al. (2018) [7] introduced MoleculeNet, a benchmarking platform designed to evaluate machine learning models for predicting molecular and biophysical properties. They applied graph neural networks (GNNs) trained with the Adam optimizer to predict chemical properties such as dipole moments, polarizability, atomic partial charges, and reaction energies. Their results demonstrated that GNNs when optimized appropriately, can effectively capture molecular representations and achieve high accuracy across diverse chemical datasets.

The Adam optimizer combines the benefits of SGD with momentum and RMSprop, making it popular for training deep learning models. It is more suitable than SGD for problems with sparse gradients and noisy datasets by adapting the learning rate for each parameter. Additionally, it requires less hyperparameter tuning compared to standard gradient descent methods. However, Adam may lead to suboptimal convergence in some cases, as its adaptive updates can result in overly aggressive parameter changes that prevent the model from settling into a good local minimum. To mitigate these issues, several variants of Adam have been proposed. AMSGrad [8] introduces a more stable second-moment estimate to avoid rapid oscillations. AdaBelief [9] modifies the update rule to adapt based on the difference between predicted and actual gradients, improving generalization. Other methods like QHM (quasi-hyperbolic momentum) balance fast learning with better convergence stability [10]. While SGD is often preferred for large-scale datasets due to its simplicity and strong theoretical guarantees, Adam is particularly useful in computational chemistry applications where optimization landscapes are complex and adaptive learning rates improve the stability of training quantum and molecular models.

In addition to widely used first-order optimizers such as SGD and Adam, several quasi-Newton methods based on second-order approximations are gaining traction in scientific machine learning, particularly in physics-informed neural networks (PINNs). Among them, L-BFGS (limited-memory Broyden–Fletcher–Goldfarb–Shanno) provides a memory-efficient approximation of the inverse Hessian matrix, improving convergence near optimality without incurring the full cost of second-order derivatives. It is frequently used in PINNs for fine-tuning, often in tandem with Adam, which handles the early stages of training. This hybrid strategy improves both training stability and final model accuracy [11].

A more recent development is the self-scaled Broyden (SSBroyden) method, a symmetric quasi-Newton algorithm that uses rescaled gradient information to achieve faster convergence and greater numerical stability. It has shown promising results in physics-informed training regimes, often outperforming traditional L-BFGS in terms of convergence speed and robustness.

These quasi-Newton methods enrich the optimization toolkit in computational chemistry ML, particularly for problems with stiff gradients, complex physical constraints, or highly non-linear loss surfaces, conditions under which purely gradient-based optimizers like Adam may converge slowly or become unstable. Recent comparative studies of optimizer performance in physics-informed and scientific ML models have highlighted several concrete limitations of Adam. Notably, its reliance on historical gradient magnitudes to

scale updates can lead to poor alignment between the update direction and the true gradient, especially in regions of high curvature or stiff dynamics. This misalignment often results in oscillatory behavior, instability, or premature convergence to suboptimal solutions [12]. In contrast, quasi-Newton approaches such as L-BFGS and SSBroyden incorporate curvature information more effectively, enabling faster and more reliable convergence in such challenging settings.

In the following section, we describe the backpropagation algorithm, which serves as the computational backbone for gradient calculation in optimizers such as Adam and SGD.

2.4. Backpropagation Algorithm

The backpropagation algorithm originally introduced in [13] is an essential component of training deep neural networks, providing a way to compute gradients required for local optimization. It is primarily used in conjunction with gradient-based optimization techniques, such as SGD and its variants. Backpropagation enables neural networks to adjust their parameters by propagating errors from the output layer back through the network, ensuring that updates are performed efficiently across multiple layers.

Backpropagation operates by computing the gradient of the loss function $L(\boldsymbol{\theta}; \boldsymbol{x}_i, y_i)$ with respect to the model parameters $\boldsymbol{\theta}$ using the chain rule. The process consists of two main steps:

- Forward Pass: Computes the output of the neural network by applying a series of weighted transformations and activation functions to the input $\boldsymbol{x_i}$.
- Backward Pass (Backpropagation Step): Computes the gradient of the loss function by propagating the error signal backward through the network using the chain rule.

The backpropagation update rule is provided by the following:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_\theta L(\boldsymbol{\theta}_t; \boldsymbol{x}_i, y_i), \tag{3}$$

where $\nabla_\theta L(\boldsymbol{\theta}_t; \boldsymbol{x}_i, y_i)$ represents the gradient of the loss function with respect to the model parameters $\boldsymbol{\theta}$, computed using the chain rule in a multi-layer neural network. Unlike standard optimization methods, backpropagation propagates the gradient backward through each layer of the network, allowing all parameters to be updated efficiently.

The gradient computation follows the general rule below:

$$\nabla_\theta L(\boldsymbol{\theta}; \boldsymbol{x}_i, y_i) = \frac{\partial L(\boldsymbol{\theta}; \boldsymbol{x}_i, y_i)}{\partial \boldsymbol{\theta}}. \tag{4}$$

Backpropagation itself does not perform optimization but provides the gradients needed for optimizers like SGD or Adam. The loss gradient is first computed at the output layer and then propagated backward through the hidden layers using the chain rule. This ensures that each parameter update accounts for its contribution to the total loss. This ability to efficiently compute gradients supports a wide range of learning tasks in chemical modeling. One notable application is in quantum chemistry, where backpropagation has been used to learn representations of complex physical quantities, such as energy functionals and electron density distributions.

Deep learning models trained with backpropagation have also been applied to quantum chemistry tasks that involve approximating complex energy terms, such as exchange-correlation functionals in DFT. These applications highlight how optimization enables learning of quantum-level structure–property relationships.

Snyder et al. (2012) [14] employed backpropagation in training neural networks to approximate exchange-correlation functionals in DFT, using gradient-based optimization to iteratively refine model parameters. They applied SGD to minimize the loss function, allowing the neural network to approximate exchange-correlation functionals from training

data and apply them accurately to new molecular systems not included in the training set. This is a typical supervised learning setting, where the model learns from input–output pairs (e.g., molecular configurations and known DFT-calculated energies), and backpropagation is used to update the model based on prediction errors.

As discussed in previous sections, GNNs have also been applied to molecular property prediction, as demonstrated in [7]. Here, backpropagation was essential for optimizing the network weights to improve predictive accuracy for molecular descriptors. Furthermore, deep learning models trained using backpropagation have been successfully applied to refine wavefunction-based methods and accelerate energy evaluations, as described in [6].

While backpropagation provides an efficient mechanism for computing gradients in differentiable models, other chemical learning tasks require optimization strategies that do not rely on gradient information, such as Monte Carlo methods, which we explore in the following section.

2.5. Monte Carlo Optimization

Monte Carlo optimization methods are used in computational chemistry and machine learning to optimize complex, high-dimensional functions. These methods rely on stochastic sampling to explore the parameter space and find optimal solutions, making them particularly useful when gradient-based techniques struggle due to non-differentiability or highly non-convex function.

The Metropolis Monte Carlo (MMC) algorithm is a stochastic optimization method used to sample from a probability distribution by iteratively updating model parameters. Originally introduced by Metropolis et al. (1953) [15], this algorithm was designed to simulate the equilibrium properties of statistical mechanical systems by generating representative configurations according to the Boltzmann distribution.

The general acceptance criterion for an update from $\boldsymbol{\theta}_t$ to $\boldsymbol{\theta}_{t+1}$ is determined by the following probability:

$$P_{\text{accept}} = \min\left(1, e^{-\frac{L(\boldsymbol{\theta}_{t+1};\boldsymbol{x_i},y_i)-L(\boldsymbol{\theta}_t;\boldsymbol{x_i},y_i)}{T}}\right), \tag{5}$$

where T is a control parameter. By tuning the control parameter T, the MMC algorithm can balance exploration and exploitation, allowing occasional uphill moves and reducing the likelihood of premature convergence to local minima. For this reason, Monte Carlo methods are generally regarded as global optimization techniques. This global behavior makes them complementary to local techniques such as gradient descent.

Self-learning hybrid Monte Carlo (SLHMC), introduced by Nagai et al. (2019) [16], integrates machine learning with Monte Carlo sampling to enhance the efficiency of DFT-based simulations. The algorithm employs an artificial neural network (ANN) to approximate the potential energy surface of a molecular system, generating proposed configurations that are then validated using the Metropolis acceptance criterion. This approach has been applied in quantum chemistry to improve the sampling of molecular conformations and accelerate simulations of reaction mechanisms, reducing the computational cost of first-principles calculations while maintaining high accuracy. These Monte Carlo-based methods operate without requiring labeled data and are, therefore, naturally suited for unsupervised learning tasks, particularly in sampling and exploration of molecular conformational spaces.

More recently, Karandashev et al. (2023) [17] introduced an evolutionary Monte Carlo algorithm, MOSAiCS, designed for molecular optimization in chemical space. This method applies Monte Carlo sampling to explore molecular configurations that optimize properties, such as solvation energy and dipole moment while preserving key energetic characteristics. By leveraging Monte Carlo-based search strategies, MOSAiCS enables efficient molecular

design by iteratively refining candidate structures based on their predicted electronic properties. This approach highlights the versatility of Monte Carlo algorithms in machine learning-driven molecular optimization and their potential for accelerating the discovery of novel functional materials.

2.6. Bayesian Optimization

Bayesian optimization (BO) is a probabilistic optimization technique that models the loss function as a Gaussian process (GP) and selects new evaluation points based on an acquisition function. In machine learning, BO is commonly used for hyperparameter tuning, where the goal is to identify settings such as learning rate, number of layers, or regularization strength that yield the best model performance on a validation set. Unlike model training, which adjusts internal weights via gradients, hyperparameter optimization operates externally and does not involve gradient-based updates. This makes BO especially valuable in chemistry-related tasks where each model evaluation (e.g., with quantum-level accuracy) can be computationally expensive [18]. The acquisition function quantifies the utility of evaluating the loss function at a given point, guiding the search toward promising regions of the parameter space.

The core idea of Bayesian Optimization is to continuously update our probabilistic model of the loss function (posterior distribution) as we gather new data and use this improved model to select the next points for evaluation. Given a prior belief over $L(\boldsymbol{\theta}; \boldsymbol{x_i}, y_i)$, the posterior mean $\mu(\boldsymbol{\theta})$ and variance $\sigma^2(\boldsymbol{\theta})$ are updated at each step. The next candidate, $\boldsymbol{\theta}_{t+1}$, is chosen by maximizing an acquisition function $a(\boldsymbol{\theta})$. The expected improvement (EI) acquisition function selects the next evaluation point by estimating the expected amount by which the new function value $L(\boldsymbol{\theta})$ will improve upon the current best observed value L_{min}:

$$a(\boldsymbol{\theta}) = \mathbb{E}[\max(0, L_{\min} - L(\boldsymbol{\theta}; \boldsymbol{x_i}, y_i))]. \tag{6}$$

Similar to the role of the control parameter T in MMC, BO controls the trade-off between exploration and exploitation through the choice of the acquisition function. Unlike MMC, which allows probabilistic acceptance of suboptimal solutions, BO actively models uncertainty to guide the search towards promising regions in the parameter space. Its ability to explore the entire search space through uncertainty modeling makes it a powerful strategy for identifying globally optimal solutions.

This makes BO especially valuable for chemical tasks that involve expensive simulations or multi-modal design spaces, where efficient global search is essential.

In their 2017 study, Hernández-Lobato et al. [19] explored the application of BO to enhance the accuracy of surrogate models in electronic structure calculations. The authors focused on predicting exchange-correlation functionals within DFT. By employing BO, they optimized the hyperparameters of machine learning models to improve the prediction of these functionals. This approach led to a reduction in computational cost.

Beyond hyperparameter tuning, Bayesian optimization has been applied to a very different type of problem in chemistry: optimizing molecular structures themselves by searching over latent spaces to maximize the desired chemical properties.

A 2018 study by Gómez-Bombarelli et al. [20] introduced a novel approach for converting discrete molecular representations into a multidimensional continuous space. Traditional molecular representations, such as SMILES notation, are inherently discrete and not well-suited for smooth optimization. By mapping molecules into a continuous latent space, optimization algorithms, including BO, can operate more efficiently. In this context, optimization refers not to the model itself but to the input space, which is often a latent vector or molecular representation, with the goal of finding structures that maximize de-

sired chemical properties such as solubility, selectivity, or electronic stability. This form of optimization is widely used in inverse molecular design, where models are trained to suggest new candidate molecules with optimal predicted properties. To achieve this, the authors trained a deep neural network to construct an encoder, a decoder, and a property predictor. The encoder transforms discrete molecular representations into continuous real-valued vectors while the decoder reconstructs these vectors back into valid molecular structures. The model was trained to predict molecular properties such as quantum energy levels and solubility, enabling the generation of new molecules with tailored characteristics.

Bayesian optimization is most commonly used in supervised learning settings, where the objective function is learned from known input–output relationships. In contrast, reinforcement learning offers an alternative optimization framework particularly suited to sequential decision problems such as molecule generation. In this setting, the model explores chemical space by learning from reward signals that reflect property constraints or design goals. For example, Olivecrona et al. [21] trained a recurrent neural network with reinforcement learning to generate valid SMILES strings optimized for drug-likeness scores. Their work illustrates how reward-guided training can steer molecular generation toward chemically desirable outputs.

Case Study: Multi-task Bayesian optimization for C–H Activation in Drug Development

A concrete and well-documented example of how machine learning can accelerate real-world chemical discovery comes from the work of Taylor et al. [22], who applied multi-task Bayesian optimization (MTBO) to the yield optimization of pharmaceutical intermediates involving C–H activation reactions. This study exemplifies the end-to-end integration of machine learning, historical chemical data, and experimental automation to efficiently identify optimal reaction conditions, including catalyst selection, within a drug discovery context.

Problem Formulation: The authors focused on the optimization of reaction conditions for six distinct C–H functionalization reactions, representative of real challenges encountered in medicinal chemistry. Traditional methods would approach each new reaction from scratch, often requiring 20 to 30 experimental iterations to reach optimal yields. To improve efficiency, Taylor et al. reformulated the problem as a Bayesian optimization task, leveraging prior data from 23 previously optimized reactions.

Each reaction was characterized by a set of experimental parameters, including the following:

- Type of catalyst or catalyst precursor;
- Solvent;
- Base;
- Temperature;
- Residence time in a flow reactor.

These were treated as categorical and continuous variables to be optimized simultaneously.

Model Design and Training: To guide this optimization, the authors trained a multi-task GP surrogate model. The multi-task component allowed the model to generalize across chemically related reactions by sharing statistical strength from prior tasks (i.e., reactions). This means that instead of starting from a blank slate, the model was initialized with historical knowledge, giving it an informed prior over the expected reaction outcomes.

The objective function (reaction yield) was treated probabilistically. The GP predicted both the mean yield and variance for each combination of experimental parameters. An EI acquisition function was then used to iteratively propose new experiments that balance exploration (uncertain areas) and exploitation (promising areas).

Experimental Execution: The optimization loop was implemented on an autonomous flow-based reactor platform. For each target reaction, the system began with a small number of randomly selected combinations of experimental parameters (typically 3–5). These combinations were tested in real reactions, and the observed yields were fed back into the model.

The surrogate model then updated its predictions based on the new data and proposed the next batch of experimental conditions by optimizing the EI acquisition function.

This iterative loop continued until convergence was reached, usually defined as no significant improvement in yield across successive rounds.

Results and Impact: The results demonstrated a dramatic reduction in experimental effort:

- For each new reaction, the optimal yield was typically reached in 6–8 experiments, compared to 20+ in conventional practice.
- The system also demonstrated strong transfer learning capabilities, performing better on reactions that were chemically related to those in the historical dataset.
- Cost savings were realized through reduced use of reagents and instrument time, and optimization times were cut by more than half in most cases.

Moreover, the MTBO model correctly identified unusual or non-intuitive reaction conditions that outperformed human intuition, highlighting the added value of model-driven discovery.

Conclusion: This case study demonstrates how multi-task Bayesian optimization can solve a real and recurring problem in catalyst and condition selection for C–H activation. By integrating machine learning with historical data and automated experimentation, researchers achieved highly efficient, data-driven optimization with significant practical benefits [22].

While Bayesian optimization focuses on global search in continuous parameter spaces, many chemical problems also require structured representations of molecules, such as graphs, motivating the use of graph-based optimization methods explored in the next section.

2.7. Optimization in Graph-Based Models

Graph-based models [23], particularly graph neural networks (GNNs), have gained significant attention in machine learning applications for chemistry, where molecular structures and electronic properties can be naturally represented as graphs. Optimization techniques play a crucial role in improving the performance and efficiency of GNNs by enhancing stability, reducing computational complexity, and ensuring smooth propagation of information across the graph structure.

Two key optimization approaches in graph-based models are graph Laplacian optimization, which leverages the graph Laplacian matrix to enforce smoothness and improve generalization, and spectral methods, which utilize eigendecomposition and graph Fourier transforms to reduce computational overhead and enhance learning efficiency. These methods are particularly useful in tasks such as charge density prediction, molecular property estimation, and quantum chemistry simulations. These will be discussed in the following subsections.

2.7.1. Graph Laplacian Optimization

Graph Laplacian optimization is used in graph-based machine learning models, particularly in GNNs, where it improves stability and generalization by encouraging neighboring nodes to share similar feature representations. This is achieved by incorporating the graph

Laplacian matrix [24], which acts as a regularization mechanism to smooth node features and control information propagation.

The graph Laplacian matrix is defined as follows:

$$\mathcal{L} = D - A, \tag{7}$$

where A is the adjacency matrix of the graph, encoding connections between nodes, and D is the diagonal degree matrix, representing the number of connections for each node. The Laplacian matrix plays a key role in controlling how information spreads across the network, ensuring that connected nodes influence each other's feature representations.

In GNNs, regularization using the graph Laplacian ensures that neighboring nodes maintain similar feature values, reducing overfitting and improving generalization [25]. The node feature matrix X represents the numerical attributes of each node, such as atomic properties in molecular graphs. The smoothness constraint imposed by the Laplacian is expressed as follows:

$$\mathcal{L}_{\text{smooth}} = \text{Tr}(X^T \mathcal{L} X), \tag{8}$$

where $\text{Tr}(\cdot)$ denotes the trace of a matrix. This penalty function minimizes differences between connected nodes, leading to more stable and coherent learned representations.

Graph Laplacian optimization is a pivotal technique in computational chemistry, particularly when employing GNNs to model complex molecular structures. In tasks such as charge density prediction, it's essential to ensure smooth transitions of electronic properties across bonded atoms for accurate modeling. Regularization through the Laplacian matrix enhances the numerical stability of GNNs in electronic structure problems. These applications illustrate how chemical problems involving spatial or topological structure, such as force field parameterization, naturally motivate the use of Laplacian-based regularization.

Recent work has extended graph optimization techniques to the prediction of physical interaction parameters in force fields, connecting learned graph structures with classical molecular mechanics. While graph Laplacian optimization operates on structured representations, the parameter tuning involved is typically local, adjusting force-field or interaction parameters based on local loss feedback, which is a typical feature of local optimization. However, the graph topology itself often reflects global molecular structure, indirectly guiding the model toward more globally coherent solutions, thereby introducing a topology-driven form of global optimization.

In their study, Thürlemann et al. (2022) [26] proposed a novel approach to parameterize force fields (FFs) by integrating machine learning with gradient-descent optimization while maintaining a physics-based functional form. They employed graph neural networks (GNNs) to predict FF parameters from potential energy surfaces, focusing on intramolecular interactions. Their method enables the GNN to learn and predict parameters such as bond lengths, angles, and dihedral angles, which are crucial for accurately modeling molecular geometries and dynamics. This approach enabled flexible, data-driven simulations by removing the need for manually defined functional forms. To encourage physically meaningful parameter variation, their method also leverages smoothness penalties, such as Laplacian-based regularization, which promote coherence across the molecular graph topology. This application represents a form of molecular optimization where model-driven learning is used to fine-tune physical parameters that define molecular structure and behavior.

These findings motivate further exploration of spectral techniques, which analyze graph structure through eigenvalue decomposition to enhance model performance, a topic we discuss in the following section.

2.7.2. Spectral Methods

Spectral methods are a class of optimization techniques that utilize the eigenvalues and eigenvectors of matrices associated with graphs to improve the efficiency and accuracy of machine learning models. These methods are particularly useful in GNNs, where they enable a more efficient representation of graph structures and facilitate computationally efficient learning. Unlike Laplacian regularization, which promotes local smoothness across neighboring nodes, spectral methods capture global structural patterns by analyzing the overall connectivity encoded in the Laplacian graph.

A key concept in spectral methods is the spectral decomposition of the graph Laplacian $\mathcal{L}$, defined as follows:

$$\mathcal{L} = U\Lambda U^T, \tag{9}$$

where U is the matrix of eigenvectors, and Λ is the diagonal matrix containing the corresponding eigenvalues. This decomposition provides a basis for spectral analysis of graph structures and allows for transformations such as spectral filtering.

The spectral decomposition enables the definition of the graph Fourier transform (GFT), which maps signals on a graph into the spectral domain:

$$\hat{X} = U^T X. \tag{10}$$

Here, X represents the node feature matrix, and $\hat{X}$ denotes the transformed representation in the spectral domain. This transformation is fundamental for spectral filtering, where specific frequency components can be enhanced or suppressed to optimize learning. This type of filtering is particularly useful in molecular graphs for reducing noise, highlighting relevant structural patterns, and improving generalization in downstream tasks.

Spectral methods are particularly effective in tasks that require dimensionality reduction or structural clustering, such as visualizing large chemical libraries or uncovering latent molecular features. Spectral clustering has been applied to group molecules based on structural similarity [27], while Laplacian Eigenmaps preserve local relationships among molecular descriptors. Reutlinger et al. used this approach to improve the visualization of chemical libraries [28], and Gill et al. applied unsupervised learning with spectral embeddings to support emergent property prediction from SMILES representations [29].

Although spectral methods are not optimization algorithms in the traditional sense, they enable global structural analysis by capturing the overall topology of molecular graphs. This facilitates the exploration of chemical space and supports downstream optimization tasks by improving latent representations and clustering structure. Unsupervised learning methods such as spectral clustering and dimensionality reduction are particularly valuable in chemical ML when labeled data are scarce. These methods help uncover latent structures in molecular datasets, organize chemical space, and facilitate the pre-training of models that are later fine-tuned in supervised tasks.

Extending beyond unsupervised learning, spectral methods have been integrated into ML frameworks for quantum chemistry, where they support both molecular property prediction and quantum interaction modeling. Their integration into GNNs has shown notable improvements in predicting electronic structure and chemical reactivity by leveraging the spectral properties of molecular graph Laplacians [30]. These approaches include message-passing neural networks [31], spectral graph convolutions, and models tailored to learning quantum chemical interactions, all contributing to more accurate and efficient molecular simulations.

Notable examples include SchNet, DimeNet, and PaiNN [32–34], with the latter belonging to the class of E(3)-equivariant models specifically designed to preserve rotational and reflection symmetries in molecular systems. Such equivariant architectures enable

more accurate predictions of tensorial properties, such as dipole moments, vibrational modes, and molecular spectra [35].

These advancements demonstrate how spectral techniques bridge the gap between structural representation and quantum-level prediction, making them a powerful complement to other optimization strategies in chemical machine learning.

2.8. Summary of Optimization Strategies and Their Applications

The previous subsections discussed various mathematical optimization techniques used in machine learning for computational chemistry. These methods differ in their roles within the learning process, their scope (local vs. global), and the learning paradigms they support. The table 1 provides an integrative overview, categorizing each method by its optimization target, associated learning setting and search scope.

Table 1. Summary of optimization strategies and their applications in chemical ML.

Optimization Method	Optimization Target(s)	Learning Setting(s)	Search Scope
Stochastic Gradient Descent (SGD)	Model parameter optimization	Supervised learning	Local optimization
Adam	Model parameter optimization	Supervised learning	Local optimization
Backpropagation	Gradient computation (for model training)	Supervised learning	Supports local optimization (via optimizers)
Monte Carlo (MMC, SLHMC)	Molecular optimization	Reinforcement learning, Unsupervised	Global optimization
Bayesian Optimization	Hyperparameter, Molecular optimization	Supervised, Reinforcement	Global optimization
Graph-based Models	Molecular optimisation	Supervised, Unsupervised	Local (parameter tuning) and global (topology-driven)

As shown in the table, each optimization method plays a distinct role depending on the stage of the learning pipeline and the specific demands of chemical modeling tasks. While these methods are powerful enablers of accurate and scalable ML, their effectiveness is often constrained by real-world challenges such as limited data availability, model transferability, and computational cost. These limitations are addressed in the following section.

3. Challenges in Machine Learning for Computational Chemistry

3.1. Data Scarcity and Quality

Machine learning models for density functional theory and molecular mechanics heavily rely on high-quality datasets. However, obtaining sufficient labeled data is often computationally expensive and limited by the accuracy of quantum chemical calculations [36,37]. The lack of sufficient data presents several key challenges:

Overfitting: Small datasets lead to overfitting, where models memorize training data instead of learning generalizable patterns. This is particularly problematic in computational chemistry, where molecular diversity is high, but the cost of generating labeled examples severely limits dataset size. As a result, models may capture noise or dataset-specific artifacts, performing well on seen molecules but failing to generalize to new chemical

structures. This undermines the predictive reliability in downstream tasks such as drug discovery, molecular screening, or reaction modeling.

Computational Cost: High-fidelity quantum chemical calculations, such as coupled cluster or DFT-based simulations, demand substantial computational resources, sometimes requiring hours or days per molecule. This bottleneck restricts not only dataset size but also the diversity and complexity of chemical systems that can be explored. As a result, optimization workflows that depend on repeated model evaluations, such as hyperparameter tuning or active molecule selection, become prohibitively expensive.

Data Bias: Limited datasets often undersample specific regions of chemical space, leading to biased models that overfit common scaffolds or molecular features. Such bias reduces model robustness and hampers generalization across compound classes, physical conditions, or functional groups. This is particularly concerning in applications like materials discovery or catalyst design, where extrapolation is essential. For instance, in molecular property prediction tasks, even a modest bias in training data, such as the overrepresentation of small molecules, has been shown to reduce predictive accuracy by up to 15–20% when tested on larger, more complex compounds [38].

These challenges significantly impact the development of robust machine learning models in computational chemistry, limiting both the predictive accuracy and the effectiveness of optimization strategies. Addressing them requires carefully designed mathematical methods, which are discussed in the following sections.

Mathematical Models for Improving Data Quality

To address challenges associated with data scarcity and quality in computational chemistry, various mathematical techniques have been developed.

Regularization Techniques: Regularization methods, such as L2 regularization (ridge regression), add a penalty for large coefficients in the model, leading to simpler and more generalizable solutions. In the context of neural networks used for chemical property prediction, regularization reduces model variance and discourages overfitting to noisy or small datasets. This is particularly valuable when training on sparse quantum chemical datasets, where small perturbations in data could lead to unstable or biased models. The application of L2 regularization has been shown to improve the generalization ability of chemical models by promoting smoother parameter updates and preventing overfitting [39,40].

Data Augmentation in Chemical Space: Generating synthetic molecular data using generative models, such as variational autoencoders (VAEs [41]) or graph neural networks (GNNs), increases the diversity and volume of available data. These models sample from a learned latent space to create novel yet chemically valid structures, effectively enriching the training distribution. For example, the materials graph network (MEGNet) model has demonstrated improved performance in predicting molecular properties by leveraging graph-based representations to augment training data [42]. Augmentation not only mitigates data scarcity but also helps models learn more robust features, reducing their sensitivity to dataset-specific biases.

Active Learning: This strategy enables models to identify and select the most informative molecules for annotation, optimizing resource utilization and improving training efficiency. Instead of randomly choosing training samples, active learning prioritizes the most uncertain data points, where the model exhibits the highest prediction uncertainty. By focusing on examples where the model struggles the most, such as molecules with conflicting property predictions or high entropy in the output distribution, active learning significantly accelerates learning while reducing the number of required annotated samples. Implementing this approach in chemistry has led to notable improve-

ments in model accuracy, particularly in molecular property prediction and reaction optimization [43,44]. This trade-off between exploiting known regions of chemical space and exploring uncertain ones parallels strategies found in global optimization algorithms.

Bayesian Optimization for Data Selection: Utilizing Bayesian optimization to identify the most valuable data points enables minimization of computational costs. Wu, Walsh, and Ganose demonstrated how Bayesian optimization can navigate parameter spaces by iteratively selecting experiments to balance exploration with exploitation [45]. This approach has been shown to accelerate the discovery of optimal molecules and materials while reducing the number of required experiments or calculations.

By integrating these mathematical strategies, machine learning models in computational chemistry can achieve higher accuracy and better generalization, even when faced with limited or low-quality data.

3.2. Transferability and Generalization

Machine learning models trained on a specific set of molecular systems often struggle to generalize to new, unseen chemical environments due to variations in molecular representation, data distribution, and quantum mechanical effects [46]. The ability to transfer knowledge from one dataset to another is a crucial requirement for developing robust models. Generalization ensures that models can make accurate predictions across diverse chemical domains, but several key challenges hinder this capability:

Limited Extrapolation Capabilities: Neural networks often fail to make accurate predictions for out-of-distribution molecules, particularly those with significantly different atomic compositions or electronic properties compared to the training set [47]. This is because most models are inherently designed to interpolate within the training domain rather than extrapolate beyond it. As a result, when confronted with molecules that exhibit novel bonding patterns, unusual charge distributions, or rare functional groups, these models tend to produce unreliable outputs. The inability to extrapolate limits the practical application of ML in scenarios such as the discovery of new materials or drugs, where generalization to unexplored chemical space is essential.

Molecular Representation Bias: The choice of molecular descriptors and embeddings (e.g., graph-based representations and SMILES encoding) heavily influences how well a model generalizes [48]. Different representations emphasize different chemical features; for instance, some may better capture local bonding environments, while others are more effective at preserving global topology. A model trained on one representation may not effectively interpret or generalize to molecules encoded differently, introducing a form of structural bias. Moreover, some representations may obscure key quantum or spatial information, which further reduces model transferability across tasks or datasets.

Long-Range Interactions and Quantum Effects: Machine learning models frequently struggle to capture long-range electron correlation and nonlocal quantum effects, which are essential for accurately predicting chemical reactivity, excited-state dynamics, and intermolecular interactions [49]. Many models rely on local information or limited neighborhood aggregation, which may omit subtle but significant quantum phenomena such as polarization or dispersion forces. These limitations are especially problematic in larger or more flexible molecular systems, where the interplay of distant atomic interactions significantly influences energy landscapes and reactivity profiles.

Domain Shift in Experimental vs. Simulated Data: Many ML models in chemistry are trained on simulated datasets, such as DFT-calculated properties, while real-world applications often require extrapolation to experimental data. This discrepancy, known as domain shift, can cause significant performance degradation when models transition from theoretical calculations to experimental validation [50]. Experimental datasets

may include noise, measurement error, or artifacts not present in clean simulation data, which introduces a mismatch in data distributions. Consequently, models may appear highly accurate in silico but fail to replicate their performance when tested under real experimental conditions.

Taken together, these limitations underscore a fundamental obstacle in the development of reliable chemical machine learning models: the inability to consistently generalize beyond the training domain. To mitigate these challenges, the following section surveys established mathematical approaches aimed at enhancing generalization across heterogeneous chemical spaces and data regimes.

Mathematical Strategies for Improving Generalization

To enhance the generalizability of machine learning models in chemistry, researchers have explored various mathematical approaches. These strategies address the underlying causes of poor transferability, such as distribution shifts, representation limitations, and the lack of physical constraints.

Bayesian Optimization: This method is employed to identify optimal hyperparameters and neural architecture configurations in a principled, data-efficient manner. By modeling the uncertainty in the objective function, Bayesian optimization enables efficient exploration of hyperparameter space, which contributes to improved robustness across diverse chemical datasets and molecular structures [45].

Adversarial Training and Domain Adaptation: These techniques introduce deliberate perturbations to molecular inputs during training, forcing the model to generalize better under slight variations. Adversarial examples simulate domain shift scenarios, while domain adaptation methods explicitly align the feature distributions between source (training) and target (test) domains. Together, they enhance resilience to out-of-distribution chemical data and improve real-world applicability [51,52].

Meta-Learning and Few-Shot Learning: These approaches are designed to enable rapid adaptation to new chemical tasks or molecular families using only a small number of labeled examples. By learning to generalize from limited data, such models are particularly useful in low-resource domains where collecting new quantum chemical data is expensive or impractical. Meta-learning frameworks help models internalize inductive biases that are transferable across related chemical tasks [53].

Hybrid ML–Physics Models: Combining machine learning with physics-based constraints improves extrapolation to molecules beyond the training set. These hybrid models enhance predictive reliability by incorporating domain knowledge, such as symmetry, conservation laws, or electronic structure principles, helping ensure that outputs remain physically meaningful even in out-of-distribution scenarios [54].

However, integrating physical constraints into ML architectures is far from trivial and presents several implementation challenges. One major difficulty lies in encoding domain-specific physical laws, such as energy conservation, permutation invariance, or force fields, into differentiable neural network structures. This often requires extending existing equivariant neural network architectures to ensure they respect additional symmetry constraints specific to the task (e.g., energy conservation or force matching), crafting problem-specific loss functions, or incorporating symbolic terms to preserve quantum mechanical priors. These approaches demand extensive domain expertise, complicate model implementation, and hinder transferability to other chemical systems with different physical characteristics [55]. Furthermore, enforcing these constraints strictly (e.g., through hard architectural rules) can make optimization more brittle and sensitive to initialization, while softer regularization-based approaches may compromise physical fidelity in favor of smoother convergence [56]. These trade-offs between accuracy, generalizability, and

tractability are still an active area of research in physics-informed ML, particularly in applications involving quantum chemistry and molecular dynamics.

3.3. Computational Cost vs. Accuracy Trade-Offs

One of the key challenges in ML-driven computational chemistry is balancing high predictive accuracy with affordable computational cost. Many high-fidelity quantum chemistry calculations, such as coupled cluster (CCSD(T)) and DFT, require substantial computational resources, often scaling poorly with system size. As a result, large-scale applications, such as molecular screening or material discovery, become impractical [57].

The integration of machine learning models aims to alleviate this burden, yet the models themselves must be carefully designed and trained to avoid excessive computational overhead. Deep neural networks, while powerful, may involve millions of parameters and require extensive training data and resources, especially when tailored to quantum-level accuracy. Furthermore, increasing model complexity does not always yield proportionally better performance and may even lead to diminishing returns.

This creates a fundamental trade-off: achieving high chemical accuracy often comes at the cost of computational scalability. Understanding and navigating this trade-off is essential for the practical deployment of ML in chemistry. It involves careful choices in model architecture, training protocols, and data selection, which are topics addressed in the following section.

Optimization Strategies for Cost-Accuracy Trade-Offs

To address the challenge of balancing computational cost and accuracy in machine learning for computational chemistry, researchers have developed several optimization strategies that make model training and inference more efficient without significantly sacrificing performance:

Grid search vs. Bayesian optimization: Selecting appropriate hyperparameters, such as learning rate, batch size, or network depth, is crucial for controlling both model complexity and resource usage. While traditional grid search exhaustively evaluates combinations of parameters, this approach quickly becomes infeasible in high-dimensional spaces. In contrast, Bayesian optimization models the performance landscape and intelligently selects the most promising configurations to evaluate next. This reduces the number of costly evaluations required to find high-performing configurations and is particularly advantageous when each model training cycle is computationally expensive [58].

While Bayesian optimization has proven advantageous over grid search in many low- to moderate-dimensional tasks, its performance deteriorates in high-dimensional hyperparameter spaces, a common setting in deep learning for computational chemistry. Although BO is valued for its sample efficiency, it suffers from exponentially increased search complexity and poor surrogate model fidelity as dimensionality grows. For instance, simulations have shown that when the dimensionality increases from 2D to 3D, the number of evaluations required to converge to an optimal solution can increase by several hundred iterations, even in synthetic settings for materials synthesis [59]. This combinatorial explosion is primarily due to the difficulty of accurately modeling acquisition functions in high-dimensional regimes, which tend to become nearly flat with few distinguishable optima, hindering effective exploration [60]. As a result, BO may converge to suboptimal hyperparameters or require prohibitively expensive evaluations, especially when each function call involves time-consuming simulations or quantum chemistry calculations.

Adaptive learning rate techniques: Gradient-based optimizers like Adam, RMSprop, and AdaGrad adaptively adjust the learning rate during training by incorporating information about gradient magnitude and variance over time. These methods accelerate

convergence, prevent oscillations in flat regions of the loss surface, and reduce the number of iterations needed for model training [5]. In computational chemistry, where each training iteration might involve large molecular datasets or simulation-derived features, reducing redundant updates contributes significantly to cost efficiency.

Active learning: Beyond improving data quality, active learning offers a powerful mechanism to reduce the number of expensive quantum chemistry calculations. By querying only the most informative or uncertain samples, typically those with high prediction uncertainty, active learning frameworks concentrate resources on cases where model improvement is most needed. In reaction screening tasks, this approach has been shown to reduce the number of DFT evaluations by up to 70% without compromising predictive performance [61]. Similar efficiency gains have been observed in other domains, such as potential energy surface estimation and reaction outcome prediction, where active learning significantly reduces dataset size while preserving target accuracy [44].

While active learning has proven effective in reducing annotation costs by strategically selecting informative samples, it introduces nontrivial practical overheads. After each acquisition step, the model must be retrained from scratch or incrementally updated, which becomes increasingly expensive for deep learning models or ensembles. In computational chemistry, this cost is compounded by the fact that labeling often requires quantum mechanical simulations or DFT calculations, which are time-consuming and resource-intensive. Another significant challenge is the design and selection of acquisition functions. Common strategies such as uncertainty sampling or expected model change may not effectively capture the diversity and complexity of chemical space. If the acquisition function over-focuses on uncertain but chemically redundant regions or conversely ignores rare but important motifs, the result can be suboptimal coverage and wasted computational effort. Tuning these acquisition functions to reflect both predictive uncertainty and chemical diversity remains an open and application-specific problem [62].

Low-Fidelity Approximations with Correction Models: Instead of relying solely on computationally intensive first-principle methods like CCSD(T) or DFT, researchers increasingly adopt multi-fidelity approaches that combine low-cost approximations with learned correction functions. For instance, semi-empirical methods or low-tier DFT can be used to generate large datasets at low cost, and models such as Δ-learning or transfer-learning are trained to learn systematic corrections that map these outputs to high-fidelity results [63]. This enables scalable exploration of chemical space while preserving the accuracy of high-level quantum mechanical methods.

Together, these strategies exemplify how mathematical and algorithmic optimization can directly impact the scalability and feasibility of machine learning in chemistry. By intelligently allocating computational resources and leveraging principled approximations, researchers can extend the applicability of ML models to larger, more diverse molecular systems without prohibitive cost.

Generative modeling frameworks: Generative models, such as variational autoencoders, generative adversarial networks, and diffusion-based architectures, have shown great potential for automated molecular design. However, their integration into practical computational chemistry pipelines is hampered by several technical and domain-specific challenges. First, generated molecules often fall outside the domain of chemical validity, producing invalid SMILES strings, unstable molecular graphs, or compounds that violate basic valence rules. Second, even when chemically valid, many generated candidates are synthetically infeasible, lacking accessible reaction pathways or requiring costly multistep synthesis [64]. Third, the objectives optimized during generation, such as latent space similarity or predicted property scores, may not align with experimentally meaningful endpoints, leading to unrealistic or non-functional molecules [65]. Finally, evaluating

each generated structure for target properties (e.g., binding affinity and HOMO-LUMO gap) typically requires computationally intensive post hoc validation, often involving quantum mechanical methods like DFT [66]. These bottlenecks make it difficult to scale generative approaches or incorporate them directly into iterative design–test cycles in chemistry workflows.

4. Future Directions

Ongoing advances in quantum computing, machine-learned potentials, and large-scale pretraining signal transformative possibilities for optimization in computational chemistry. The future development of these technologies promises to address many of the challenges outlined in this review, including generalization, data scarcity, and computational cost.

Quantum-Classical Hybrid Optimization: Quantum algorithms have already demonstrated potential in simulating molecular systems, yet their integration with classical ML workflows remains at an early stage. Future research may focus on hybrid quantum–classical optimization approaches, where quantum subroutines (e.g., variational quantum eigensolvers or quantum annealing) are embedded into classical ML pipelines. Such methods could enable scalable optimization in high-dimensional chemical spaces, improve sampling, and accelerate hyperparameter tuning for complex models [67].

Machine-Learned Potentials and Transferable Force Fields: Another promising direction is the development of machine-learned interatomic potentials to improve molecular simulations. Current ML-driven FFs are limited by training data and accuracy trade-offs, making it essential to explore transferable and self-improving potential models. Further development of ML-based energy functions could improve the accuracy and efficiency of molecular simulations, enabling faster exploration of chemical space while reducing the reliance on costly quantum chemistry calculations [68].

Optimization for Molecular Discovery and Catalysis: Optimization strategies will continue to play a central role in inverse design tasks. Bayesian optimization, combined with graph-based and spectral methods, offers promising avenues for accelerating ligand design, reaction prediction, and catalyst discovery. The integration of ML with enhanced sampling techniques, such as metadynamics, may further improve the exploration of reaction pathways and rare event dynamics [69].

Interpretability and Explainable Models: To foster greater trust and adoption of ML in scientific domains, models must become more interpretable. Explainability techniques, such as saliency maps, gradient-based attribution (e.g., Grad-CAM), and attention visualization, can help identify the molecular features most responsible for a model's predictions, particularly in tasks like molecular property prediction or drug-likeness evaluation.

Foundation Models and Transfer Learning: Finally, the emergence of large pre-trained chemical foundation models (e.g., ChemBERTa and OC20) opens the door to flexible, data-efficient modeling across tasks. These models, trained on extensive chemical corpora, enable fine-tuning in low-data regimes and enhance cross-domain generalization. Future work may focus on combining foundation models with optimization pipelines to build modular and adaptable frameworks for chemical discovery [70,71].

Together, these future directions suggest a path toward more integrated, scalable, and interpretable frameworks for chemical machine learning. As optimization methods evolve, their potential synergy with quantum computing, foundation models, and physical constraints may play a pivotal role in shaping the next generation of predictive tools in computational chemistry.

Despite these promising directions, several critical challenges remain unresolved in the field of optimization for chemical machine learning. Key issues include improving generalization across chemical space in the presence of biased or sparse data, incorporating

physical constraints into ML architectures without introducing excessive complexity, and designing active learning strategies that are both efficient and practical to implement. Furthermore, optimization algorithms such as Bayesian optimization continue to face scalability barriers in high-dimensional parameter spaces.

We believe that addressing these challenges will require tighter integration between theoretical model development and experimental validation, as well as hybrid approaches that combine domain knowledge with data-driven flexibility. In particular, establishing benchmark datasets and reproducible evaluation frameworks will be essential for comparing methods and driving progress in the field.

5. Conclusions

Mathematical optimization plays a critical role in enhancing machine learning models for computational chemistry, enabling more accurate, efficient, and scalable predictions of molecular properties. This review explored key optimization techniques, including stochastic gradient descent, Bayesian optimization, Monte Carlo methods, and spectral approaches, emphasizing their impact on improving ML performance in chemistry applications.

Despite significant progress, challenges such as data scarcity, model generalization, and computational cost remain central issues. Advanced mathematical techniques, including active learning, meta-learning, and hybrid quantum–classical methods, are emerging as promising solutions to overcome these limitations. Additionally, the integration of machine learning with traditional computational chemistry approaches holds great potential for accelerating chemical discovery.

By framing optimization along three principal targets (model training, hyperparameter tuning, and molecular design) this review provides a unified perspective that connects general ML methodology with chemistry-specific needs, such as quantum property prediction and force field development. As optimization techniques evolve, aligning methods with their appropriate targets will be critical to building more robust, transferable, and interpretable chemical ML models. Achieving this will require sustained collaboration across machine learning, quantum chemistry, and materials science, bringing us closer to practical, scalable applications in drug discovery, catalyst design, and materials engineering.

Funding: This research was supported by the Ministry of Science, Technological Development, and Innovation of the Republic of Serbia (Contract No. 451-03-136/2025-03/200135).

Acknowledgments: The author gratefully acknowledges Bojana Nedić Vasiljević for her valuable insights and conceptual input that helped shape the direction of this work. Warm thanks are also extended to Zoran Hadžibabić for thoughtful stylistic suggestions on an earlier draft, as well as for his kind support and encouragement throughout the writing process. Finally, the author is grateful to the referee for their careful reading and insightful comments, which significantly improved the quality of the paper.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Bottou, L. Large-Scale Machine Learning with Stochastic Gradient Descent. In Proceedings of the COMPSTAT'2010, Paris, France, 22–27 August 2010. [CrossRef]
2. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the importance of initialization and momentum in deep learning. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 17–19 June 2013; Volume 28, pp. 1139–1147. Available online: https://proceedings.mlr.press/v28/sutskever13.html (accessed on 9 July 2025).
3. Rupp, M.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O.A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **2012**, *108*, 058301. [CrossRef] [PubMed]
4. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747. [CrossRef]

5. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015. [CrossRef]
6. Schütt, K.T.; Arbabzadah, F.; Chmiela, S.; Müller, K.R.; Tkatchenko, A. Quantum-Chemical Insights from Deep Tensor Neural Networks. *Nat. Commun.* **2017**, *8*, 13890. [CrossRef] [PubMed]
7. Wu, Z.; Ramsundar, B.; Feinberg, E.N.; Gomes, J.; Geniesse, C.; Pappu, A.S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.* **2018**, *9*, 513–530. [CrossRef]
8. Reddi, S.J.; Kale, S.; Kumar, S. On the Convergence of Adam and Beyond. *arXiv* **2019**, arXiv:1904.09237. [CrossRef]
9. Zhuang, J.; Tang, T.; Ding, Y.; Wang, S.; Liu, Z.; Castro, C.D.; Dvornek, N.; Papademetris, X.; Duncan, J.S. AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients. *arXiv* **2020**, arXiv:2010.07468. [CrossRef]
10. Ma, N.Q.; Yarats, D.; Kapturowski, S. Quasi-Hyperbolic Momentum and Adam for Deep Learning. *arXiv* **2018**, arXiv:1810.06801. [CrossRef]
11. Kollmannsberger, S.; D'Angella, D.; Jokeit, M.; Herrmann, L. *Deep Learning in Computational Mechanics*; Springer: Cham, Switzerland, 2021. [CrossRef]
12. Kiyani, E.; Shukla, K.; Urbán, J.F.; Darbon, J.; Karniadakis, G.E. Which Optimizer Works Best for Physics-Informed Neural Networks and Kolmogorov-Arnold Networks? *arXiv* **2025**, arXiv:2501.16371. https://arxiv.org/abs/2501.16371.
13. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
14. Snyder, J.C.; Rupp, M.; Hansen, K.; Müller, K.R.; Burke, K. Finding Density Functionals with Machine Learning. *Phys. Rev. Lett.* **2012**, *108*, 253002. [CrossRef]
15. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller, A.H.; Teller, E. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092. [CrossRef]
16. Nagai, Y.; Okumura, M.; Kobayashi, K.; Shiga, M. Self-learning hybrid Monte Carlo: A first-principles approach. *Phys. Rev. B* **2020**, *102*, 041124. [CrossRef]
17. Karandashev, K.; Weinreich, J.; Heinen, S.; Arismendi Arrieta, D.J.; von Rudorff, G.F.; Hermansson, K.; von Lilienfeld, O.A. Evolutionary Monte Carlo of QM Properties in Chemical Space: Electrolyte Design. *J. Chem. Theory Comput.* **2023**, *19*, 8861–8870. [CrossRef] [PubMed]
18. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; De Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE* **2016**, *104*, 148–175. [CrossRef]
19. Hernández-Lobato, J.M.; Requeima, J.; Pyzer-Knapp, E.O.; Aspuru-Guzik, A. Parallel and Distributed Thompson Sampling for Large-scale Accelerated Exploration of Chemical Space. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1470–1479. Available online: https://arxiv.org/abs/1706.01825 (accessed on 9 July 2025).
20. Gómez-Bombarelli, R. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4*, 268–276. [CrossRef]
21. Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminform.* **2017**, *9*, 48. [CrossRef]
22. Wigh, D.; Jeraal, M.; Johnson, C.; Taylor, C.; Felton, K.; Chessari, G.; Lapkin, A.; Grainger, R. Accelerated Chemical Reaction Optimization Using Multi-Task Learning. *ACS Cent. Sci.* **2023**, *9*, 957–968. [CrossRef]
23. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv* **2017**, arXiv:1609.02907. [CrossRef]
24. Chung, F.R.K. *Spectral Graph Theory*; CBMS Regional Conference Series in Mathematics; American Mathematical Society: Providence, RI, USA, 1997. Available online: https://bookstore.ams.org/cbms-92 (accessed on 9 July 2025).
25. Zhou, D.; Bousquet, O.; Lal, T.N.; Weston, J.; Schölkopf, B. Learning with Local and Global Consistency. In Proceedings of the 17th International Conference on Neural Information Processing Systems, Whistler, BC, Canada, 9–11 December 2004; Advances in Neural Information Processing Systems.
26. Thürlemann, M.; Böselt, L.; Riniker, S. Regularized by Physics: Graph Neural Network Parametrized Differentiable Force Field Models. *J. Chem. Theory Comput.* **2022**, *18*, 7569–7582. [CrossRef]
27. Ningombam, S.S.; Larson, E.J.L.; Indira, G.; Madhavan, B.L.; Khatri, P. Aerosol classification by application of machine learning spectral clustering algorithm. *Atmos. Pollut. Res.* **2024**, *15*, 102026. [CrossRef]
28. Reutlinger, M.; Schneider, G. Nonlinear dimensionality reduction and mapping of compound libraries for drug discovery. *J. Mol. Graph. Model.* **2012**, *34*, 108–117. [CrossRef] [PubMed]
29. Gill, J.; Chakraborty, R.; Gubba, R.; Liu, A.; Jain, S.; Iyer, C.; Khwaja, O.; Kumar, S. Unsupervised Learning of Molecular Embeddings for Enhanced Clustering and Emergent Properties for Chemical Compounds. *arXiv* **2023**, arXiv:2310.18367. [CrossRef]
30. Yu, S.; Dong, H.; Wang, P.; Wu, C.; Guo, Y. Generative Creativity: Adversarial Learning for Bionic Design. *arXiv* **2018**, arXiv:1805.07615. [CrossRef]

31. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural Message Passing for Quantum Chemistry. *arXiv* **2017**, arXiv:1704.01212. [CrossRef]
32. Gasteiger, J.; Groß, J.; Günnemann, S. Directional Message Passing for Molecular Graphs. *arXiv* **2020**, arXiv:2003.03123. [CrossRef]
33. Schütt, K.T.; Kindermans, P.-J.; Sauceda, H.E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. SchNet—A continuous-filter convolutional neural network for modeling quantum interactions. *J. Chem. Phys.* **2018**, *148*, 241722. Available online: https://proceedings.neurips.cc/paper_files/paper/2017/file/303ed4c69846ab36c2904d3ba8573050-Paper.pdf (accessed on 9 July 2025). [CrossRef]
34. Schütt, K.T.; Unke, O.T.; Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv* **2021**, arXiv:2102.03150. Available online: https://arxiv.org/abs/2102.03150 (accessed on 9 July 2025).
35. Batzner, S.; Musaelian, A.; Sun, L.; Geiger, M.; Mailoa, J.P.; Kornbluth, M.; Molinari, N.; Smidt, T.E.; Kozinsky, B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **2022**, *13*, 2453. [CrossRef]
36. Huang, B.; von Rudorff, G.F.; von Lilienfeld, O.A. The central role of density functional theory in the AI age. *Science* **2023**, *81*, 170–175. [CrossRef]
37. Nandy, A.; Duan, C.; Kulik, H.J. Audacity of huge: Overcoming challenges of data scarcity and data quality for machine learning in computational materials discovery. *arXiv* **2021**, arXiv:2111.01905. [CrossRef]
38. Liu, Y.-Y.; Kashima, H. Chemical property prediction under experimental biases. *Sci. Rep.* **2022**, *12*, 8206. [CrossRef]
39. Demir-Kavuk, O.; Kamada, M.; Akutsu, T.; Knapp, E.W. Prediction using step-wise L1, L2 regularization and feature selection for small data sets with large number of features. *BMC Bioinform.* **2011**, *12*, 412. [CrossRef]
40. Lo Y.C.; Rensi, S.E.; Torng, W.; Altman, R.B. Machine learning in chemoinformatics and drug discovery. *Drug Discov. Today* **2018**, *23*, 1538–1546. [CrossRef] [PubMed]
41. Ochiai, T.; Inukai, T.; Akiyama, M.; Furui, K.; Ohue, M.; Matsumori, N.; Inuki, S.; Uesugi, M.; Sunazuka, T.; Kikuchi, K.; et al. Variational autoencoder-based chemical latent space for large molecular structures with 3D complexity. *Commun. Chem.* **2023**, *6*, 249. [CrossRef] [PubMed]
42. Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; Ong, S.P. Graph Networks as a Universal Machine Learning Framework for Molecules and Crystals. *Chem. Mater.* **2019**, *31*, 3564–3572. [CrossRef]
43. van Tilborg, D.; Grisoni, F. Traversing chemical space with active deep learning for low-data drug discovery. *Nat. Comput. Sci.* **2024**, *4*, 786–796. [CrossRef]
44. Khalak, Y.; Tresadern, G.; Hahn, D.F.; de Groot, B.L.; Gapsys, V. Chemical Space Exploration with Active Learning and Alchemical Free Energies. *J. Chem. Theory Comput.* **2022**, *18*, 6259–6270. [CrossRef]
45. Wu, Y.; Walsh, A.; Ganose, A.M. Race to the bottom: Bayesian optimisation for chemical problems. *Digit. Discov.* **2024**, *3*, 1086–1100. [CrossRef]
46. Bartók, A.P.; De, S.; Poelking, C.; Bernstein, N.; Kermode, J.R.; Csányi, G.; Ceriotti, M. Machine learning unifies the modeling of materials and molecules. *Sci. Adv.* **2017**, *3*. [CrossRef]
47. Tossou, P.; Wognum, C.; Craig, M.; Mary, H.; Noutahi, E. Real-World Molecular Out-of-Distribution: Specification and Investigation. *J. Chem. Inf. Model.* **2024**, *64*, 697–711. [CrossRef]
48. Wigh, D.S.; Goodman, J.M.; Lapkin, A.A. A review of molecular representation in the age of machine learning. *WIREs Comput. Mol. Sci.* **2022**, *12*, e1603. [CrossRef]
49. McDonagh, J.L.; Silva, A.F.; Vincent, M.A.; Popelier, P.L.A. Machine Learning of Dynamic Electron Correlation Energies from Topological Atoms. *J. Chem. Theory Comput.* **2018**, *14*, 216–224. [CrossRef] [PubMed]
50. Han, H.; Choi, S. Transfer Learning from Simulation to Experimental Data: NMR Chemical Shift Predictions. *J. Phys. Chem. Lett.* **2021**, *12*, 3662–3668. [CrossRef] [PubMed]
51. Gani, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; March, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. *arXiv* **2015**, arXiv:1505.07818. [CrossRef]
52. Xie, L.; He, S.; Zhang, Z.; Lin, K.; Bo, X.; Yang, S.; Feng, B.; Wan, K.; Yang, K.; Yang, J.; et al. Domain-adversarial multi-task framework for novel therapeutic property prediction of compounds. *Bioinformatics* **2020**, *36*, 2848–2855. [CrossRef]
53. Qian, X.; Ju, B.; Shen, P.; Yang, K.; Li, L.; Liu, Q. Meta Learning with Attention Based FP-GNNs for Few-Shot Molecular Property Prediction. *ACS Omega* **2024**, *9*, 23940–23948. [CrossRef]
54. Keith, J.A.; Vassilev-Galindo, V.; Cheng, B.; Chmiela, S.; Gastegger, M.; Müller, K.-R.; Tkatchenko, A. Combining Machine Learning and Computational Chemistry for Predictive Insights Into Chemical Systems. *Chem. Rev.* **2021**, *121*, 9816–9872. [CrossRef]
55. Wang, X.; Zhang, M. Graph Neural Network with Local Frame for Molecular Potential Energy Surface. In Proceedings of the First Learning on Graphs Conference, Virtual Event, 9–12 December 2022; Volume 198, pp. 19:1–19:30. Available online: https://proceedings.mlr.press/v198/wang22d.html (accessed on 9 July 2025).
56. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [CrossRef]

57. Bogojeski, M.; Vogt-Maranto, L.; Tuckerman, M.E.; Müller, K.-R.; Burke, K. Quantum chemical accuracy from density functional approximations via machine learning. *Nat. Commun.* **2020**, *11*, 5223. [CrossRef]
58. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. *arXiv* **2012**, arXiv:1206.2944. [CrossRef]
59. Hickman, R.J.; Aldeghi, M.; Häse, F.; Aspuru-Guzik, A. Bayesian optimization with known experimental and design constraints for chemistry applications. *Digit. Discov.* **2022**, *1*, 732–744. [CrossRef]
60. Jin, Y.; Kumar, P.V. Bayesian optimisation for efficient material discovery: A mini review. *Nanoscale* **2023**, *15*, 10975–10984. [CrossRef] [PubMed]
61. Eyke, N.S.; Green, W.H.; Jensen, K.F. Iterative Experimental Design Based on Active Machine Learning Reduces the Experimental Burden Associated with Reaction Screening. *React. Chem. Eng.* **2020**, *5*, 1963–1972. [CrossRef]
62. Kulichenko, M.; Barros, K.; Lubbers, N.; Li, Y.W.; Messerly, R.A.; Tretiak, S.; Smith, J.S.; Nebgen, B. Uncertainty-driven dynamics for active learning of interatomic potentials. *Nat. Comput. Sci.* **2023**, *3*, 148–157. [CrossRef]
63. Ko, T.W.; Ong, S.P. Data-efficient construction of high-fidelity graph deep learning interatomic potentials. *npj Comput. Mater.* **2025**, *11*, 65. [CrossRef]
64. Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; et al. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Nat. Mach. Intell.* **2020**, *2*, 554–562. [CrossRef]
65. Bilodeau, C.; Jin, W.; Barzilay, R.; Jaakkola, T.; Jensen, K. Generative models for molecular discovery: Recent advances and challenges. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2022**, *12*, e1608. [CrossRef]
66. Blanchard, A.E.; Zhang, P.; Bhowmik, D.; Mehta, K.; Gounley, J.; Reeve, S.T.; Irle, S.; Pasini, M.L. Computational Workflow for Accelerated Molecular Design Using Quantum Chemical Simulations and Deep Learning Models. *Commun. Comput. Inf. Sci.* **2023**. [CrossRef]
67. Fiedler, L.; Hoffmann, N.; Mohammed, P.; Popoola, G.A.; Yovell, T.; Oles, V.; Ellis, J.A.; Rajamanickam, S.; Cangi, A. Training-free hyperparameter optimization of neural networks for electronic structures in matter. *arXiv* **2022**, arXiv:2202.09186. [CrossRef]
68. Sivaraman, G.; Krishnamoorthy, A.N.; Baur, M.; Holm, C.; Stan, M.; Csányi, G.; Benmore, C.; Vázquez-Mayagoitia, Á. Machine-learned interatomic potentials by active learning: amorphous and liquid hafnium dioxide. *npj Comput. Mater.* **2020**, *6*, 104. [CrossRef]
69. Abrams, C.; Bussi, G. Enhanced Sampling in Molecular Dynamics Using Metadynamics, Replica-Exchange, and Temperature-Acceleration. *arXiv* **2014**, arXiv:1401.0387. [CrossRef]
70. Ahmad, W.; Simon, E.; Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa-2: Towards Chemical Foundation Models. *arXiv* **2022**, arXiv:2209.01712. [CrossRef].
71. Chanussot, L.; Das, A.; Goyal, S.; Lavril, T.; Shuaibi, M.; Riviere, M.; Tran, K.; Heras-Domingo, J.; Ho, C.; Hu, W.; et al. Open Catalyst 2020 (OC20) Dataset and Community Challenges. *ACS Catal.* **2021**, *11*, 6059–6072. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.