UC Berkeley
UC Berkeley Previously Published Works

Title

Machine-Learned Leftmost Hessian Eigenvectors for Robust Transition State Finding

Permalink

https://escholarship.org/uc/item/3b60j9mt

Journal

Journal of Chemical Theory and Computation, 22(13)

ISSN

1549-9618

Authors

Wu, Guanchen
Yuan, Eric C-Y
Hegazy, Kareem
et al.

Publication Date

2026-07-14

DOI

10.1021/acs.jctc.6c00554

Copyright Information

This work is made available under the terms of a Creative Commons Attribution-NonCommercial-ShareAlike License, available at https://creativecommons.org/licenses/by-nc-sa/4.0/

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

arXiv:2603.21323v1 [physics.chem-ph] 22 Mar 2026

Machine-Learned Leftmost Hessian Eigenvectors for Robust Transition State Finding

Guanchen Wu[1], Chung-Yueh Yuan[1,5], Kareem Hegazy[4,7], Samuel M. Blau[6], and Teresa Head-Gordon[1−3,5]

[1]Kenneth S. Pitzer Theory Center and Department of Chemistry, [2]Department of Bioengineering, [3]Department of Chemical and Biomolecular Engineering, [4]Department of Statistics, University of California, Berkeley, CA, 94720 USA
[5]Chemical Sciences Division and [6]Energy Technologies Area, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720 USA
[7]International Computer Science Institute, Berkeley, CA 94704 USA
corresponding author: thg@berkeley.edu

Abstract

The reliable determination of transition states (TSs) benefits from second-order information for robust convergence and validation, but the computational expense of Hessians prohibits their routine use in TS optimization. Here, we present a machine-learning-driven TS optimizer that directly predicts the leftmost Hessian eigenvector (LMHE), the critical mode that locally approximates the reaction coordinate encompassing the TS. We demonstrate that our LMHE optimizer recovers TS solutions at the same rate as full Hessian optimizers, and more robustly from degraded initial guess geometries, thereby eliminating the excessively long wall times characteristic of full-Hessian approaches and reducing total gradient evaluations compared to standard quasi-Newton methods. We further improve accuracy and robustness using uncertainty quantification for identifying occasional LMHE prediction failures, that then falls back to a full Hessian update from the machine learned potential at that optimization step, avoiding expensive active learning. Overall our methodology and semi-automated workflow delivers second-order stability at first-order computational expense to provide a highly efficient engine for high-throughput reaction discovery.

1 Introduction

Understanding reaction chemistry and predicting kinetics requires a detailed knowledge of the potential energy surface (PES), especially the location and characterization of transition states (TSs). Mathematically defined as first-order saddle points on the PES, TSs represent the highest energy point along the minimum energy path connecting reactants to products of an elementary reaction. They are characterized by having exactly one negative eigenvalue in the Hessian matrix, corresponding to a single imaginary vibrational frequency. Unlike equilibrium structures which can be determined by optimizing along descent directions, locating saddle

points is a more complex optimization as it requires maximizing the energy along a single specific mode, the reaction coordinate, while simultaneously minimizing it along all other degrees of freedom.

The central challenge in TS optimization is correctly identifying this single ascent direction. The success of a TS optimization often depends crucially on the quality of the curvature information from the Hessian matrix, as it provides the necessary geometric guidance to distinguish the reaction coordinate from orthogonal conformational or non-reactive modes. Second-order methods utilize the full Hessian matrix to explicitly identify the mode with the most negative curvature as the local approximation of the reaction coordinate. While robust, calculating the exact Hessian at every optimization step using *ab initio* methods such as density functional theory (DFT) is prohibitively expensive for most practical applications.

Consequently, modern geometry optimization on the *ab initio* PES typically relies on quasi-Newton (QN) methods to construct cheaper approximate Hessians using only gradient information. While the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is arguably the most widely used approach for minimization, its standard formulation enforces a positive-definite Hessian, making it inherently unsuitable for locating saddle points which require indefinite Hessians[1–4]. To address this, other update schemes capable of handling indefinite curvature have been developed, such as Powell-symmetric Broyden (PSB)[5,6], symmetric rank one (SR1)[7], Murtagh-Sargent-Powell (MSP)[8,9] and TS-BFGS[10,11] methods[12]. These QN methods attempt to recover curvature information solely from the optimization trajectory history (i.e. coordinate and gradient changes). However, this reconstruction is often inaccurate, particularly at the beginning of the optimization or in flat regions, leading to convergence to an unintended TS rather than the target reaction-product pair, or failure to find the saddle point at all.

To mitigate this, more advanced QN strategies have been developed to better recover the transition mode. Some approaches calculate an exact full Hessian at the first step to ensure a good starting point, though this information decays over time. Others periodically employ iterative diagonalization (e.g. Jacobi-Davidson) to actively locate the leftmost eigenvector during the run[13–17]. While this strategy represents a significant improvement over history-only methods, robustness remains a critical challenge. Furthermore, iterative diagonalization introduces a new computational cost: determining the eigenvector via finite difference requires multiple auxiliary gradient evaluations. This cost prevents applying diagonalization at every optimization step such that standard optimizers apply diagonalization only intermittently and rely on history-based updates in between, limiting robust and rapid convergence. The reliance on these fragile TS optimizations also limits the efficacy of broader reaction path search workflows as well. Even when employing established double- and single-ended interpolation methods such as the Nudged Elastic Band (NEB)[18,19], Quadratic Synchronous Transit (QST)[20], or the Growing String Method (GSM)[21], a subsequent local optimization is typically required to pinpoint the exact saddle point. As a result, despite the availability of these global search strategies, the final location of the TS often remains a bottleneck, requiring significant user involvement and trial-and-error when Hessian information is absent.

Recent advances in geometric deep learning have presented a potential solution to TS optimization. Machine learning interatomic potentials (MLIPs) have demonstrated the ability to achieve near quantum-chemical accuracy, and as demonstrated by Yuan et al., the analytical full Hessian can be obtained from these potentials via automatic differentiation, offering a significant computational speedup over *ab initio* methods[22]. Nevertheless, this second-order MLIP approach is not without computational bottlenecks. The calculation of exact second-order derivatives via automatic differentiation necessitates the construction and retention of

extensive computational graphs. While feasible, this process remains significantly more computationally intensive than simple gradient evaluations, imposing a substantial memory and time overhead if performed at every optimization step.

In this work, we propose a novel ML architecture and TS optimization strategy to solve this key dilemma between robustness and computational efficiency. Our approach leverages the fundamental insight that the critical curvature information required for efficient TS optimization is encapsulated primarily in the leftmost Hessian eigenvector (LMHE), the vector corresponding to the most negative eigenvalue, which locally approximates the reaction coordinate and defines the ascent direction for TS optimization. We train a MLIP model to predict this vector directly from coordinates. But because TS modes often involve concerted, non-local atomic motions, they pose a unique challenge for learning the LMHE using standard local message-passing neural networks (MPNNs). To capture these motional dependencies efficiently, we introduce a novel $E(3)$-equivariant architecture that couples a message passing encoder with an induced global attention decoder that avoids the prohibitive quadratic scaling of standard full attention. The resulting learned LMHE is used to guide the geometry optimization within a restricted-step partitioned rational function optimization (RS-PRFO) framework[23–25], and failure modes identified through uncertainty quantification provides an occasional correction using a full Hessian update to improve success rates. Our LMHE approach is implemented in the user-friendly Sella optimization software package[26], and is tested across 240 organic combustion from the Sella benchmark set[26]. We show our method achieves the robustness of second-order methods at a fraction of the computational cost, thereby replacing expensive full Hessian calculations or iterative diagonalization at each optimization step, aided by a workflow that semi-automates the success in TS convergence.

2 Methods

2.1 Data Preparation for energies, gradients, Hessians, and LMHE

All energy and gradient evaluations were conducted on the PES defined by the NewtonNet MPNN model[27] trained on conformations from the original Transition1x (T1x) dataset[28], augmented with one million structures from the ANI-1x dataset[29], and described in previous work[22]. This choice of MLIP is motivated by two factors. First, the NewtonNet model is fine-tuned on an augmented T1x dataset which contains extensive reactivity data, explicitly sampling reaction pathways and saddle points rather than just equilibrium geometries. As demonstrated by Yuan et al., this specialized training regime accurately reproduces the topography of the reference DFT PES around TSs, making the NewtonNet MLIP model highly effective for TS finding[22]. Second, the differentiable nature of the NewtonNet MLIP enables the calculation of exact Hessians via automatic differentiation, and allows us to include a rigorous full Hessian baseline in our benchmarks, which would otherwise be prohibitively expensive to compute with *ab initio* methods such as DFT.

The dataset for training the LMHE predictors was generated using the fine-tuned NewtonNet model[22,27]. For each structure in the training data, the full Hessian was computed via automatic differentiation[22]. The six (or five for linear molecules) trivial eigenvectors corresponding to overall translation and rotation were removed through projection. The LMHE was then identified from the remaining non-trivial modes. To prevent data leakage, the dataset was partitioned into training, validation, and test sets based on chemical composition, strictly

following the partitioning strategy used for the NewtonNet fine-tuning[22] for consistency.

2.2 Eigenvector-Informed Hessian Update Scheme

To integrate the machine-learned LMHE into the TS optimization, a modified QN Hessian updating scheme was developed. This approach leverages the predicted eigenvector to construct a more accurate approximate Hessian at each optimization step. The geometry updates were then performed within a RS-PRFO framework[23–25], as implemented in the Sella optimizer[17].

In the notation that follows, all vectors are assumed to be column vectors. At optimization step k, the approximate Hessian for the subsequent step, $\mathbf{H}^{(k+1)}$, is constructed by incorporating the predicted leftmost eigenvector, $\mathbf{v}_1^{(k+1)}$. The Hessian is decomposed into components parallel ($\|$) and perpendicular ($\perp$) to this eigenvector:

$$\mathbf{H}^{(k+1)} = \mathbf{H}_{\|}^{(k+1)} + \mathbf{H}_{\perp}^{(k+1)}. \tag{1}$$

The parallel component is constructed from the eigenvector and its corresponding eigenvalue, $\lambda_1^{(k+1)}$:

$$\mathbf{H}_{\|}^{(k+1)} = \lambda_1^{(k+1)} \frac{\mathbf{v}_1^{(k+1)} \mathbf{v}_1^{(k+1)^{\mathrm{T}}}}{\mathbf{v}_1^{(k+1)^{\mathrm{T}}} \mathbf{v}_1^{(k+1)}}. \tag{2}$$

The leftmost eigenvalue, $\lambda_1^{(k+1)}$, is estimated using the Rayleigh-Ritz quotient:

$$\lambda_1^{(k+1)} = \frac{\mathbf{y}^{\mathrm{T}} \mathbf{v}_1^{(k+1)}}{\mathbf{s}^{\mathrm{T}} \mathbf{v}_1^{(k+1)}}, \tag{3}$$

where $\mathbf{s} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ is a vector representing the change in atomic coordinates and $\mathbf{y} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$ is the corresponding vector representing the change in gradients.

The perpendicular component of the Hessian is updated in the subspace orthogonal to $\mathbf{v}_1^{(k+1)}$. This is achieved using a projection matrix, $\mathbf{P}_{\perp}$:

$$\mathbf{P}_{\perp} = \mathbf{I} - \frac{\mathbf{v}_1^{(k+1)} \mathbf{v}_1^{(k+1)^{\mathrm{T}}}}{\mathbf{v}_1^{(k+1)^{\mathrm{T}}} \mathbf{v}_1^{(k+1)}}, \tag{4}$$

where $\mathbf{I}$ is the identity matrix. The perpendicular components of the coordinate change ($\mathbf{s}_{\perp}$), gradient change ($\mathbf{y}_{\perp}$), and the previous Hessian ($\mathbf{H}_{\perp}^{(k)}$) are then calculated:

$$\mathbf{s}_{\perp} = \mathbf{P}_{\perp}\mathbf{s}, \quad \mathbf{y}_{\perp} = \mathbf{P}_{\perp}\mathbf{y}, \quad \mathbf{H}_{\perp}^{(k)} = \mathbf{P}_{\perp}\mathbf{H}^{(k)}\mathbf{P}_{\perp}. \tag{5}$$

By explicitly partitioning out the transition mode (the $\mathbf{H}_{\|}$ component), the optimization problem in the remaining $(N-1)$ dimensional subspace is effectively a minimization problem, for which QN updates are well-established and robust. Thus, the perpendicular Hessian component is updated using a standard QN formula:

$$\mathbf{H}_{\perp}^{(k+1)} = \mathrm{QN}(\mathbf{H}_{\perp}^{(k)}, \mathbf{s}_{\perp}, \mathbf{y}_{\perp}). \tag{6}$$

While Eq. 6 is compatible with various QN formulations, this work employs the TS-adapted Broyden-Fletcher-Goldfarb-Shanno method (TS-BFGS)[10,11]. Unlike the standard BFGS update, which requires a positive-definite Hessian, the TS-BFGS formulation is designed to be

compatible with indefinite approximate Hessians. This provides the necessary flexibility for the perpendicular components to possess negative eigenvalues, thereby allowing the full approximate Hessian to temporarily contain multiple negative eigenvalues during optimization. A complete mathematical description of this update is provided in Supplementary Appendix 1.

To ensure that the predicted eigenvector corresponds to the lowest eigenvalue of the composite Hessian $\mathbf{H}^{(k+1)}$, the estimated eigenvalue $\lambda_1^{(k+1)}$ is constrained. If $\lambda_1^{(k+1)}$ is greater than the lowest eigenvalue of the perpendicular Hessian ($\lambda_{\perp 1}^{(k+1)}$), it is adjusted as follows:

$$\lambda_1^{(k+1)} = \lambda_{\perp 1}^{(k+1)} - 1. \tag{7}$$

For the first optimization step, we initialize the Hessian as:

$$\mathbf{H}^{(0)} = \mathbf{I} - 2\frac{\mathbf{v}_1^{(0)}\mathbf{v}_1^{(0)\mathrm{T}}}{\mathbf{v}_1^{(0)\mathrm{T}}\mathbf{v}_1^{(0)}}. \tag{8}$$

This specific construction for the initial Hessian is chosen to precondition the optimizer by immediately incorporating the machine-learned leftmost eigenvector. The resulting $\mathbf{H}^{(0)}$ matrix has an eigenvalue of -1 along the predicted eigenvector $\mathbf{v}_1^{(0)}$ and eigenvalues of $+1$ for all modes in the orthogonal subspace. This initialization provides the RS-PRFO framework with a Hessian that already possesses the correct qualitative structure of a TS (i.e. one negative mode) and aligns this mode with the predicted leftmost eigenvector, offering a significant advantage over a default positive-definite guess (e.g. the identity matrix) which lacks any imaginary frequency.

2.3 Transition State Optimization

The performance of the TS optimizers is evaluated by the Sella benchmark dataset[26]. This dataset contains 500 organic molecules (7 to 25 atoms) in configurations that approximate TS geometries of different reactions, among which 265 are closed-shell. As 25 of these reactions are also present in the T1x dataset, they were excluded from our analysis leaving 240 reactions for analysis.

All TS optimizations and intrinsic reaction coordinate (IRC) calculations for the Sella benchmark dataset were performed in Cartesian coordinates using the Sella optimizer[17], implemented using the Atomic Simulation Environment (ASE)[30]. These optimizations were conducted on the NewtonNet PES, defined by the same model used for the augmented T1x data preparation (Section 2.1). The Sella implementation was modified to allow for the provision of an external Hessian matrix at each optimization step.

The Sella framework involves iterative diagonalization using the Rayleigh–Ritz procedure with a modified Jacobi–Davidson method (JD0, or Olsen's method), a finite difference step size of 10^{-4} Å, and a convergence threshold of 0.1[17]. The trust radius of RS-PRFO is initialized to 0.1 and adjusted based on the ratio between the predicted and actual energy changes; the radius is increased by a factor of 1.15 when this ratio is below 1.035 and decreased by a factor of 0.65 when it exceeds 5.0. The IRC is located by performing energy minimization within a trust radius of 0.1 $\text{Å/amu}^{-1/2}$ in mass-weighted coordinates[31]. A maximum of 1000 steps was imposed for both the TS optimization and the IRC search. To evaluate the optimization results, reactants and products were compared by testing for graph isomorphism. Molecular connectivity graphs, including atom indexing, were generated using Open Babel[32] and compared using the VF2 algorithm[33].

2.4 Ensemble Consistency Check

We employ an ensemble-based consistency check to govern a fallback mechanism to the exact Hessian from automatic differentiation when prediction uncertainty is high. To facilitate this ensemble approach, we trained five independent instances of the 10.1 M parameter model with our proposed architecture, initialized with distinct random seeds, while the single-model optimizer utilizes one of these instances. The specific training details and hyper-parameters are provided in Supplementary Table S1.

At each optimization step, the LMHE is predicted by all $N = 5$ models in the ensemble, denoted as $\{\mathbf{v}_1^{(1)}, \ldots, \mathbf{v}_1^{(N)}\}$. Each predicted vector is normalized to unit length. The final predicted eigenvector, $\bar{\mathbf{v}}_1$, is computed as the normalized mean of these ensemble members. Due to the sign ambiguity of eigenvectors (i.e. $\mathbf{v}$ and $-\mathbf{v}$ are equivalent), we quantify consensus using a sign-invariant metric based on the mean outer product matrix $\bar{\mathbf{Q}}$:

$$\bar{\mathbf{Q}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{v}_1^{(i)} \mathbf{v}_1^{(i)^{\mathrm{T}}}. \tag{9}$$

The prediction uncertainty, σ, is defined as:

$$\sigma = 1 - \lambda_{max}(\bar{\mathbf{Q}}), \tag{10}$$

where $\lambda_{max}(\bar{\mathbf{Q}}$ is the largest eigenvalue of $\bar{\mathbf{Q}}$. This formulation naturally accounts for the sign ambiguity, as the outer product $\mathbf{v}\mathbf{v}^{\mathrm{T}}$ is invariant to the transformation $\mathbf{v} \mapsto -\mathbf{v}$. The uncertainty metric σ ranges from 0 (perfect consensus, where all predictions are collinear) to a value approaching 1 (maximal disagreement).

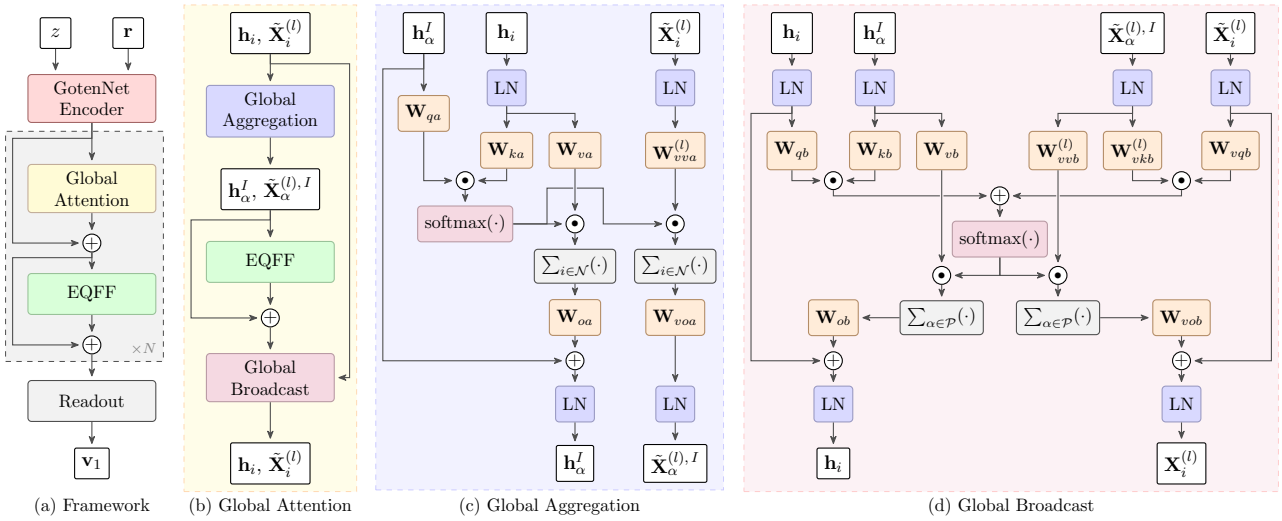
If σ exceeds a predefined threshold τ, the prediction is considered unreliable. In such cases, the optimizer reverts to calculating the exact Hessian via automatic differentiation for that specific step. Conversely, if σ is below τ, the normalized mean vector $\bar{\mathbf{v}}_1$ is utilized as the consensus prediction in the update scheme described in Section 2.2.

3 Results

3.1 GotenNet and Global Attention Architecture

The prediction of the LMHE is geometrically analogous to predicting atomic forces, as both tasks require the output to transform equivariantly with the rotation and translation of the molecule. Our method utilizes an $E(3)$-equivariant neural network trained to predict the LMHE directly from atomic coordinates using a sequential encoder-decoder architecture shown in Figure 1.

The encoder, GotenNet[34], serves as an efficient equivariant MPNN to capture spatial information without the computational overhead often associated with irreducible representations connected via Clebsch-Gordon (CG) coefficients to ensure that molecular features rotate correctly with the molecule. While theoretically rigorous, these tensor product operations are computationally intensive and scale poorly, while GotenNet employs an attention mechanism that utilizes inner products of high-degree steerable features to capture angular dependencies and spatial relationships instead of explicit CG transforms. This approach allows the network to learn rich, equivariant representations essential for predicting vector quantities like the LMHE



(a) Framework (b) Global Attention (c) Global Aggregation (d) Global Broadcast

Figure 1: *GotenNet-GA model architecture used for transition state searches with the learned LMHE.* (a) The macroscopic framework, comprising a GotenNet encoder, a global interaction module, and a readout. (b) Architecture of the Global Attention block, illustrating its internal mechanism and explicitly incorporating the encompassing residual connection shown in (a). (c) The Global Aggregation block which maps features from the physical graph nodes to the latent inducing points. (d) The Global Broadcast block, which propagates the updated global context from inducing points back to the physical nodes. In the figure, + denotes addition, · denotes dot product, $\sum$ denotes aggregation with summation, $\mathcal{N}$ denotes the set of all real nodes in the graph, $\mathcal{P}$ denotes the set of all inducing points, and LN denotes layer normalization.

while maintaining a small computational overhead. This efficiency is particularly critical for our iterative TS optimization workflow, where the model must be inferred repeatedly to guide the geometry updates.

While atomic forces are largely determined by local environments, the LMHE often describes a collective motion of atoms across the entire molecule and is therefore an inherently non-local feature. Mathematically, this non-locality is a consequence of eigen-decomposition. Standard MPNN-based MLIPs, which rely on the assumption of locality to update atomic features, are constrained in capturing the long-range dependencies necessary to describe this concerted motion. To bridge this gap, we developed an architecture that couples the GotenNet encoder with a specialized $E(3)$-equivariant global attention decoder (Figure 1). Unlike standard MPNNs that diffuse information slowly through local neighbors, this decoder leverages an induced set attention mechanism to aggregate local atomic features into a set of global inducing points, allowing the model to effectively capture the context in the entire molecule before broadcasting it back to the atomic level[35]. This architecture ensures that the prediction is informed by global context without incurring the quadratic computational cost of conventional full attention mechanisms while ensuring that the entire architecture is $E(3)$-equivariant. The complete mathematical formulation of this decoder architecture is provided in Supplementary Appendix 2, and a formal proof demonstrating its $E(3)$-equivariance is presented in Supplementary Appendix 3.

3.2 Machine-Learned Leftmost Hessian Eigenvectors

We trained two variants of our GotenNet-GA model with approximately 2.1 M and 10.1 M parameters on an augmented T1x[28] dataset described in Methods Section 2.1. To evaluate the efficacy of the proposed architecture, we also trained two baseline models using the standard GotenNet architecture, explicitly constructed to match these parameter counts. The specific training details and hyper-parameters are provided in Supplementary Table S1. This controlled comparison allows us to isolate performance gains attributable to the global attention decoder, distinct from general scaling effects. We evaluated the errors of the models using the root mean square (RMS) sine value between the predicted and ground truth eigenvectors. Sine values are chosen to account for the sign ambiguity of eigenvectors, where 0 indicates perfect alignment and 1 indicates orthogonality (the worst possible alignment).

Table 1: *Root mean square sine value for leftmost Hessian eigenvectors on the augmented T1x dataset.* The comparison includes baseline GotenNet models and our proposed architecture with global attention (GotenNet-GA) at both Small (S, ∼2.1 M parameters) and Medium (M, ∼10.1 M parameters) parameter scales.

Subsets	GotenNet_S	GotenNet_M	$\text{GotenNet-GA}_\text{S}$	$\text{GotenNet-GA}_\text{M}$
Train	<u>0.33</u>	**0.26**	0.37	0.29
Validation	0.52	<u>0.48</u>	0.50	**0.46**
Test	0.53	<u>0.49</u>	0.51	**0.47**

The quantitative benefit of explicitly modeling global context is evident in Table 1. At the small scale (∼2.1 M parameters), the $\text{GotenNet-GA}_\text{S}$ architecture reduces the test error, and this performance advantage persists as model capacity increases at the medium scale, $\text{GotenNet-GA}_\text{M}$ with 10.1 M parameters, achieving the lowest RMS error of 0.47. Notably, the gap between training and test performance is consistently narrower for our proposed GotenNet-GA architecture, supporting the idea that explicit modeling of global context facilitates more robust generalization to unseen reactions.

3.3 Transition State Optimizations using LMHE

We integrated the learned LMHE into the Sella optimizer[17] using the RS-PRFO framework[23–25]. Within the PRFO framework the leftmost eigenvector of the Hessian $\mathbf{v}_1$ locally approximating the reaction coordinate serves as the ascending direction along the PES for the TS search. This predicted vector $\mathbf{v}_1$ is utilized into a specialized Hessian update scheme (detailed in Section 2.2), providing a rigorous curvature guidance at every step of optimization.

To systematically evaluate the LMHE approach, we benchmarked the TS optimizer on 240 reactions from the Sella benchmark set[26]. The initial TS geometries for these reactions were regenerated using KinBot with reaction templates[36], which also define the intended reactant and product states. After each optimization, we followed the intrinsic reaction coordinate (IRC) from the resulting TS to determine the minimum energy path connecting the reactant and product wells. Outcomes are classified as intended (the IRC connects the exact KinBot target states), partially intended (only one of the two target state is matched), unintended (a valid TS is found, but connects to an alternative pathway), no reaction (optimization collapsed to a local minimum), or TS error (failure to converge).

We compare our single inference LMHE method against two established baselines: the standard algorithm implemented in Sella[17], which serves as a state-of-the-art QN method using iterative diagonalization, and a Full Hessian reference method where the exact Hessian is computed at every step via automatic differentiation of the MLIP. Our single inference LMHE approach demonstrates performance for intended and partially intended reactions as good as the Hessian and the QN baseline TS optimzers as seen in Figure 2. However, the single-inference model introduces a specific vulnerability: when the optimization trajectory enters regions of the PES that differ significantly from the training data, the predicted eigenvector may become inaccurate, leading to incorrect geometry updates. This leads to a higher rate of convergence failures compared to the baselines, as the single model lacks the ability to self-diagnose unreliable LMHE predictions (Figure 2).

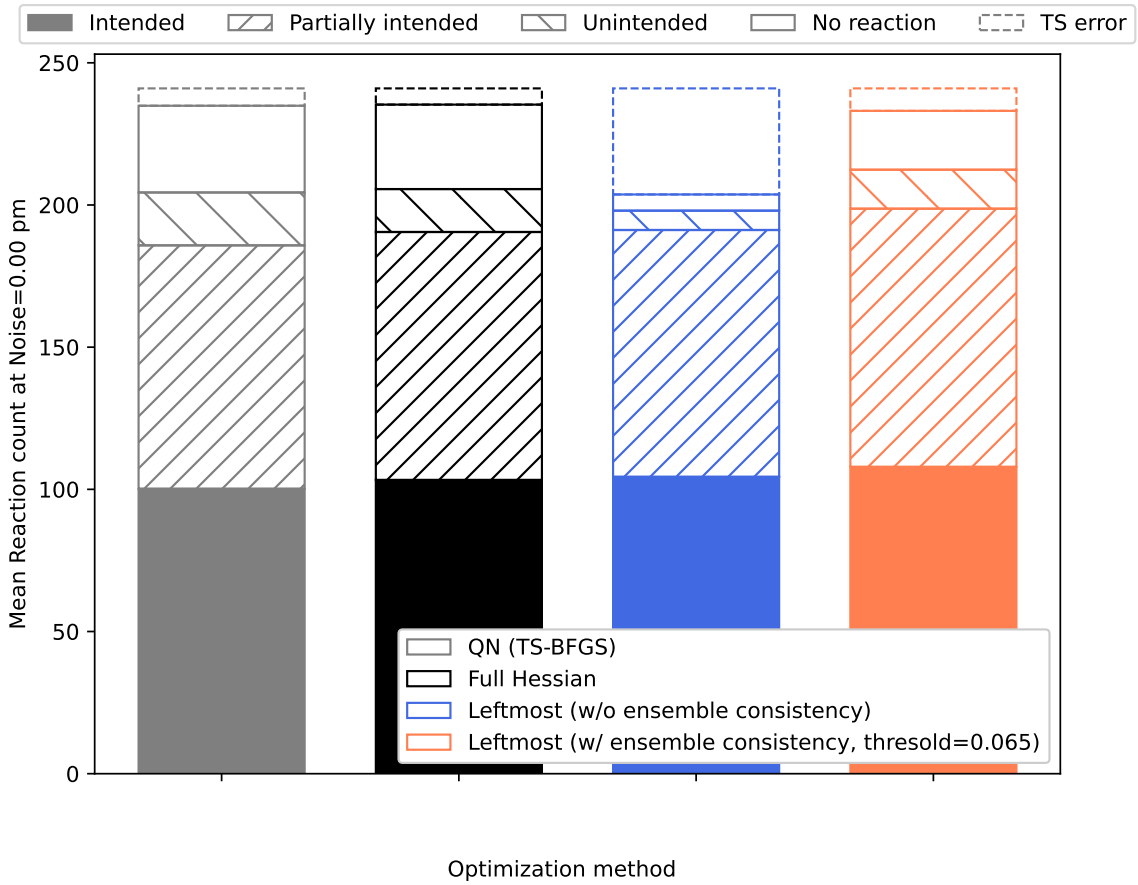


Figure 2: *Comparison of predictions of different TS optimizers against 240 Sella benchmark reactions.* The performance of the LMHE approach against the general QN baseline (TS-BFGS) and full Hessian TS optimization methods is quantified by comparing the predicted reaction paths obtained through IRC calculations against the exact reactant and product states defined by KinBot templates. Outcomes are classified as intended (the IRC connects the exact target states), partially intended (only one target state is matched), unintended (a valid TS is found, but connects to an alternative pathway), no reaction (optimization collapsed to a local minimum), or TS error (failure to converge). The single model LMHE strategy exhibits a higher failure rate due to inaccurate LMHE predictions. The LMHE ensemble consistency check identifies unreliable predictions, triggering a fallback to exact Hessian calculations and reducing failure rates to levels that are competitive with the performance of the full Hessian optimization, and significantly outperforming the standard QN baseline.

To mitigate convergence failures caused by inaccurate eigenvector guidance, we implemented an ensemble consistency check which effectively neutralizes this failure mode. As illustrated in Supplementary Figure S1, predictions exhibiting higher ensemble variance σ among five independently trained models are statistically more likely to be inaccurate. By utilizing this variance as a real-time proxy for uncertainty, the optimizer can dynamically detect unreliable predictions. When the ensemble disagreement σ exceeds a predefined threshold, the optimizer triggers a fallback to an exact Hessian calculation with automatic differentiation at that step only. As observed in Supplementary Figure S2, the ensemble check is largely insensitive to the value of the variance threshold, and hence we utilize an intermediate value of 0.065. As seen in Figure 2, the LMHE-ensemble approach now outperforms the baselines in reducing failure rates while retaining the high success count of TS optimizations using the single inference LMHE approach.

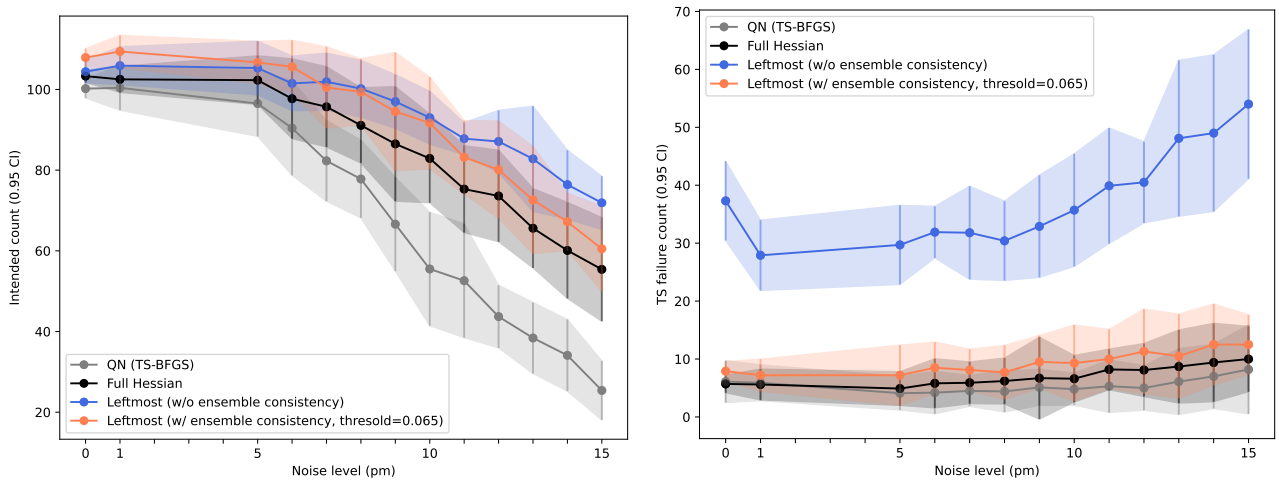


Figure 3: *Comparison of the Robustness of TS Optimizers for 240 test combustion reactions.* (a) The count of intended TS optimizations is plotted against the amplitude of random Gaussian noise applied to the initial geometries. The LMHE optimizer exhibits significantly more intended counts compared to the standard TS-BFGS QN method, maintaining high success rates comparable to the full Hessian approach even at high noise levels. (b) The number of TS optimization runs failing to converge is plotted against noise level. The single model LMHE strategy exhibits a higher failure rate due to inaccurate LMHE predictions. The LMHE ensemble consistency check identifies unreliable predictions, triggering a fallback to exact Hessian calculations and reducing failure rates to levels competitive with the full Hessian and QN baselines. Shaded regions represent the 95% confidence interval derived from 10 independent noise realizations.

To investigate TS optimizer robustness, we systematically degraded the initial guess TS structures by introducing random Gaussian noise ranging from 0 to 15 pm to the Cartesian atomic positions to test the optimizers' capability to recover the intended TS from degraded starting geometries. Figure 3a shows that both the single inference and ensemble LMHE methods demonstrate robustness against structural noise comparable to the expensive full Hessian method for intended TS recovery rates. As the initial geometry degrades, the standard QN optimizer struggles to recover the correct ascent direction even with iterative diagonalization, leading to a rapid decline in the recovery rate for the intended TS. Even so Figure 3b shows that the single inference LMHE model still has higher failure rates, while the ensemble-LMHE approach is the best tradeoff for robustness and accuracy. As observed in Supplementary Figure

S3, the ensemble check is largely insensitive to the value of the variance threshold, and hence again we utilize an intermediate value of 0.065.

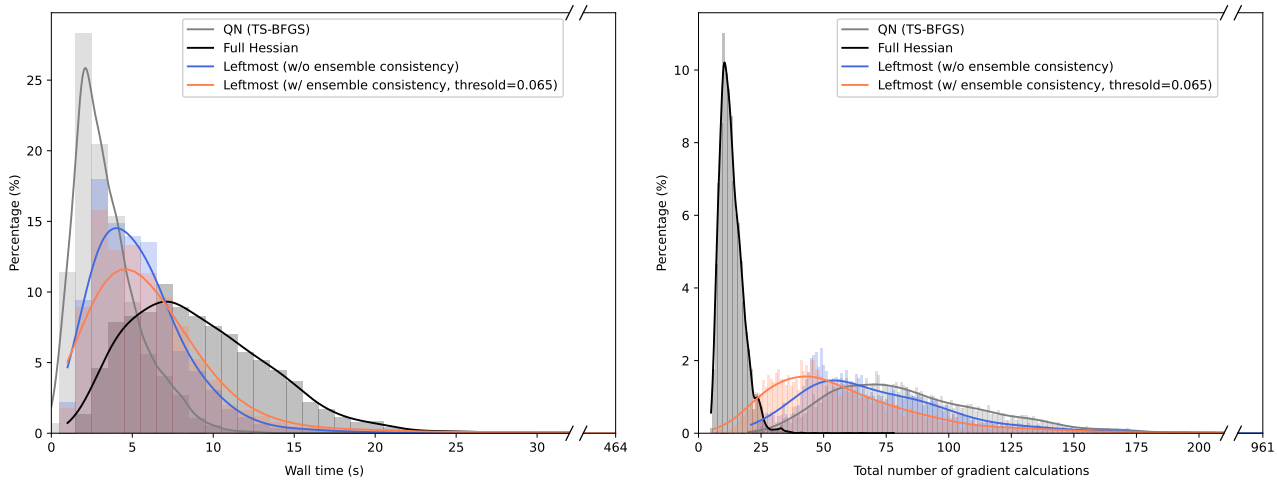


Figure 4: *Wall time and number of gradient evaluation comparisons between TS optimizers.* (a) The distribution of wall times required for converged optimizations across all noise levels. While the standard QN method is the fastest, the full Hessian method suffers from a heavy tail due to the high cost of exact second-derivative calculations. The LMHE achieves a favorable balance, significantly reducing the computational cost relative to the full Hessian approach while providing the curvature information necessary for robust convergence compared to the QN baseline. (b) The distribution of total gradient calculations required for converged optimizations. While the full Hessian method requires the fewest evaluations due to exact curvature information, among the approximate methods both LMHE approaches require fewer total gradient evaluations than the standard QN method. To ensure comparability, the data includes only those optimization where all optimizers successfully recovered the intended TS from the same initial geometry.

While the single inference and ensemble-LMHE matches the full Hessian's robustness, it does so at a fraction of the computational cost. Figure 4a illustrates the wall-time distribution for converged optimizations. The full Hessian method suffers from a "heavy tail," reflecting the expense of calculating second derivatives for every step. Both of our LMHE TS optimizers eliminates this tail, shifting the distribution significantly toward lower wall time values. The single-inference strategy defines the efficiency ceiling of our method, yielding the most compact time distribution by relying exclusively on rapid model predictions. Our ensemble-checked method effectively matches this performance, closely tracking the Single-Inference wall-time profile. This confirms that the ensemble-checked strategy does not simply recreate the full Hessian method via frequent fallbacks. Instead, it retains the high speed of the single-inference base for the vast majority of the trajectory, limiting expensive exact curvature calculations to rare, high-uncertainty regions. Finally, our method demonstrates a superior scalability regarding gradient evaluations, consistently reducing the total count of gradient evaluations compared to the QN baseline by eliminating the need for iterative diagonalization (Figure 4b). Although the standard QN method yields faster wall times, this occurs because the computational cost of the inference of our leftmost eigenvector predictor exceeds that of the inexpensive gradient evaluations on the MLIP PES used here. For completeness, Supplementary Figure S4, compares the computational cost for the ensemble check approach for various values of the variance threshold.

4 Conclusion

We have presented a novel strategy for TS optimization that effectively bridges the gap between the robustness of second-order methods and the computational efficiency of first-order approaches. By directly predicting the LMHE which serves as the local approximation of the reaction coordinate, we demonstrated that a machine-learning model can provide the critical curvature information required for saddle point location without the prohibitive cost of DFT calculations nor the expense of automatic differentiation of the MLIP to create Hessians at each optimization step.

The central contribution of this study is the development of a robust optimizer that effectively integrates geometric deep learning with PRFO algorithms. Crucial to this success was addressing the geometric distinction between local atomic forces and the non-local nature of the LMHE. We showed that standard equivariant MPNNs, which rely on local diffusion of information, are insufficient for capturing the collective atomic motions characteristic of transition modes. The introduction of a scalable $E(3)$-equivariant global attention decoder was found to reduce both the prediction error and generalization gap by enabling the model to aggregate and broadcast information across the entire molecular graph.

Benchmarks on the Sella dataset reveal that LMHE optimization scheme is highly resilient to structural noise. Our approach recovers intended TS from degraded initial guesses with a success rate comparable to exact Hessian methods, yet does so with a wall-time distribution significantly faster than full Hessian methods. Furthermore, the implementation of an ensemble consistency check addresses the "black box" reliability issue common in machine learning. By using ensemble variance as a proxy for uncertainty, the optimizer dynamically identifies unreliable predictions and selectively falls back to the higher fidelity second order method, ensuring convergence stability. Ultimately, this methodology offers a scalable pathway for high-throughput reaction discovery, minimizing the need for human intervention in refining TS guesses.

References

[1] C. G. Broyden. The convergence of a class of double-rank minimization algorithms 1. general considerations. IMA Journal of Applied Mathematics, 6(1):76–90, 03 1970.

[2] R. Fletcher. A new approach to variable metric algorithms. Comput. J., 13:317–322, 1970.

[3] Donald Goldfarb. A family of variable-metric methods derived by variational means. Mathematics of computation, 24(109):23–26, 1970.

[4] David F Shanno. Conditioning of quasi-newton methods for function minimization. Mathematics of computation, 24(111):647–656, 1970.

[5] Michael JD Powell. A new algorithm for unconstrained optimization. In Nonlinear programming, pages 31–65. Elsevier, 1970.

[6] Michael James David Powell. Convergence properties of a class of minimization algorithms. In Nonlinear programming 2, pages 1–27. Elsevier, 1975.

[7] Bruce A Murtagh and Roger WH Sargent. Computational experience with quadratically convergent minimisation methods. The Computer Journal, 13(2):185–194, 1970.

[8] Josep Maria Bofill. Updated hessian matrix and the restricted step method for locating transition structures. Journal of Computational Chemistry, 15(1):1–11, 1994.

[9] Josep Maria Bofill and Móanica Comajuan. Analysis of the updated hessian matrices for locating transition structures. Journal of computational chemistry, 16(11):1326–1338, 1995.

[10] Josep Maria Bofill. Remarks on the updated hessian matrix methods. International journal of quantum chemistry, 94(6):324–332, 2003.

[11] Josep Maria Anglada and Josep Maria Bofill. How good is a broyden–fletcher–goldfarb–shanno-like update hessian formula to locate transition structures? specific reformulation of broyden–fletcher–goldfarb–shanno for optimizing saddle points. Journal of computational chemistry, 19(3):349–362, 1998.

[12] John E Dennis Jr and Robert B Schnabel. Numerical methods for unconstrained optimization and nonlinear equations. SIAM, 1996.

[13] Gerard LG Sleijpen and Henk A Van der Vorst. A jacobi–davidson iteration method for linear eigenvalue problems. SIAM review, 42(2):267–293, 2000.

[14] Jeppe Olsen, Poul Jørgensen, and Jack Simons. Passing the one-billion limit in full configuration-interaction (fci) calculations. Chemical Physics Letters, 169(6):463–472, 1990.

[15] Andreas Stathopoulos, Yousef Saad, and Charlotte F Fischer. Robust preconditioning of large, sparse, symmetric eigenvalue problems. Journal of computational and applied mathematics, 64(3):197–215, 1995.

[16] Gerard LG Sleijpen and Henk A Van der Vorst. The jacobi-davidson method for eigenvalue problems as an accelerated inexact newton scheme. In IMACS Conference proceedings, 1995.

[17] Eric D Hermes, Khachik Sargsyan, Habib N Najm, and Judit Zádor. Accelerated saddle point refinement through full exploitation of partial hessian diagonalization. Journal of chemical theory and computation, 15(11):6536–6549, 2019.

[18] Hannes Jónsson, Greg Mills, and Karsten W Jacobsen. Nudged elastic band method for finding minimum energy paths of transitions. In Classical and quantum dynamics in condensed phase simulations, pages 385–404. World Scientific, 1998.

[19] Graeme Henkelman, Blas P Uberuaga, and Hannes Jónsson. A climbing image nudged elastic band method for finding saddle points and minimum energy paths. The Journal of chemical physics, 113(22):9901–9904, 2000.

[20] Niranjan Govind, Max Petersen, George Fitzgerald, Dominic King-Smith, and Jan Andzelm. A generalized synchronous transit method for transition state location. Computational materials science, 28(2):250–258, 2003.

[21] Baron Peters, Andreas Heyden, Alexis T Bell, and Arup Chakraborty. A growing string method for determining transition states: Comparison to the nudged elastic band and string methods. The Journal of chemical physics, 120(17):7877–7886, 2004.

[22] Eric C. Y. Yuan, Anup Kumar, Xingyi Guan, Eric D. Hermes, Andrew S. Rosen, Judit Zádor, Teresa Head-Gordon, and Samuel M. Blau. Analytical ab initio hessian from a deep learning potential for transition state optimization. Nature Communications, 15(1):8865, 2024.

[23] Ajit Banerjee, Noah Adams, Jack Simons, and Ron Shepard. Search for stationary points on surfaces. The Journal of Physical Chemistry, 89(1):52–57, 1985.

[24] Josep Maria Anglada and Josep Maria Bofill. A reduced-restricted-quasi-newton–raphson method for locating and optimizing energy crossing points between two potential energy surfaces. Journal of computational chemistry, 18(8):992–1003, 1997.

[25] Emili Besalú and Josep Maria Bofill. On the automatic restricted-step rational-function-optimization method. Theoretical Chemistry Accounts, 100(5):265–274, 1998.

[26] Eric D Hermes, Khachik Sargsyan, Habib N Najm, and Judit Zádor. Sella, an open-source automation-friendly molecular saddle point optimizer. Journal of Chemical Theory and Computation, 18(11):6974–6988, 2022.

[27] Mojtaba Haghighatlari, Jie Li, Xingyi Guan, Oufan Zhang, Akshaya Das, Christopher J. Stein, Farnaz Heidar-Zadeh, Meili Liu, Martin Head-Gordon, Luke Bertels, Hongxia Hao, Itai Leven, and Teresa Head-Gordon. Newtonnet: a newtonian message passing network for deep learning of interatomic potentials and forces. Digital Discovery, 1:333–343, 2022.

[28] Mathias Schreiner, Arghya Bhowmik, Tejs Vegge, Jonas Busk, and Ole Winther. Transition1x - a dataset for building generalizable reactive machine learning potentials. Scientific Data, 9(1):779, 2022.

[29] Justin S. Smith, Roman Zubatyuk, Benjamin Nebgen, Nicholas Lubbers, Kipton Barros, Adrian E. Roitberg, Olexandr Isayev, and Sergei Tretiak. The ani-1ccx and ani-1x data sets, coupled-cluster and density functional theory properties for molecules. Scientific Data, 7(1):134, 2020.

[30] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, et al. The atomic simulation environment—a python library for working with atoms. Journal of Physics: Condensed Matter, 29(27):273002, 2017.

[31] Klaus Müller and Leo D Brown. Location of saddle points and minimum energy paths by a constrained simplex optimization procedure. Theoretica chimica acta, 53(1):75–93, 1979.

[32] Noel M. O'Boyle, Michael Banck, Craig A. James, Chris Morley, Tim Vandermeersch, and Geoffrey R. Hutchison. Open babel: An open chemical toolbox. Journal of Cheminformatics, 3(1):33, 2011.

[33] Pasquale Foggia, Carlo Sansone, and Mario Vento. An improved algorithm for matching large graphs. In 3rd IAPR-TC15 workshop on graph-based representations in pattern recognition, 01 2001.

[34] Sarp Aykent and Tian Xia. Gotennet: Rethinking efficient 3d equivariant graph neural networks. In The Thirteenth International Conference on Learning Representations, 2025.

[35] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 3744–3753. PMLR, 09–15 Jun 2019.

[36] Ruben Van de Vijver and Judit Zádor. Kinbot: Automated stationary point search on potential energy surfaces. Computer Physics Communications, 248:106947, 2020.

[37] Guanchen Wu, Eric Yuan, Kareem Hegazy, Samuel Blau, and Teresa Head-Gordon. Data for machine-learned leftmost hessian eigenvectors for robust transition state finding, Mar 2026.

[38] Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks, 2022.

Data Availability

All data[37] including the training, validation, and test datasets for the LMHE predictors, as well as the initial transition state guess structures, ground truth reactants and products, optimized transition states, and corresponding predicted reactants and products with their coordinates, are available at `https://doi.org/10.6084/m9.figshare.31791964`. Source data for Figs. 2 to 4 is available with this manuscript. Source data are provided in this paper.

Code Availability

The codebase is comprised of several publicly available packages and tools that contribute to the project. The core framework developed for this study, including the implementation and training scripts for the GotenNet and GotenNet-GA models, model checkpoints, TS optimization scripts, and the Jupyter notebooks for evaluating ensemble consistency and analyzing optimization results, is accessible at `https://github.com/THGLab/LMHE-TSopt`. The TS optimizations were driven by the Sella optimization framework[17], which is available with comprehensive documentation at `https://github.com/zadorlab/sella`. NewtonNet[27], another integral part of the project, is also publicly available. Specifically, the HessianCalculator branch (`https://github.com/THGLab/NewtonNet/tree/HessianCalculator`) was utilized to ensure compatibility with the find-tuned models established in our previous work[22]. The parameters for this fine-tuned NewtonNet model[22] are publicly available at `https://github.com/THGLab/MLHessian-TSopt`. Finally, our primary repository directly includes modified components from GotenNet[34] with original code accessible at `https://github.com/sarpaykent/GotenNet`, as well as the spherical harmonics implementation extracted from the e3nn library[38] (`https://github.com/e3nn/e3nn`).

Acknowledgment

E.C.-Y.Y. and T.H-G. thank the CPIMS program, Office of Science, Office of Basic Energy Sciences, Chemical Sciences Division of the U.S. Department of Energy under Contract DE-AC02-05CH11231 for support. This work used computational resources provided by the National

Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract DE-AC02-05CH11231, and the Lawrencium computational cluster resource provided by the IT Division at the Lawrence Berkeley National Laboratory (Supported by the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231).

Author contributions

G.W. and T.H.G. designed the LMHE method and wrote the manuscript. G.W. carried out all training and implemented and executed all workflows. All authors discussed the results and provided edits to the manuscript.

Competing interests

The authors declare no competing interests.

Supplementary Information: Machine-Learned Leftmost Hessian Eigenvectors for Robust Transition State Finding

Guanchen Wu[1], Chung-Yueh Yuan[1,5], Kareem Hegazy[4,7], Samuel M. Blau[6], and Teresa Head-Gordon[1−3,5]

[1]Kenneth S. Pitzer Theory Center and Department of Chemistry, [2]Department of Bioengineering, [3]Department of Chemical and Biomolecular Engineering, [4]Department of Statistics, University of California, Berkeley, CA, 94720 USA
[5]Chemical Sciences Division and [6]Energy Technologies Area, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720 USA
[7]International Computer Science Institute, Berkeley, CA 94704 USA
corresponding author: thg@berkeley.edu

Figures

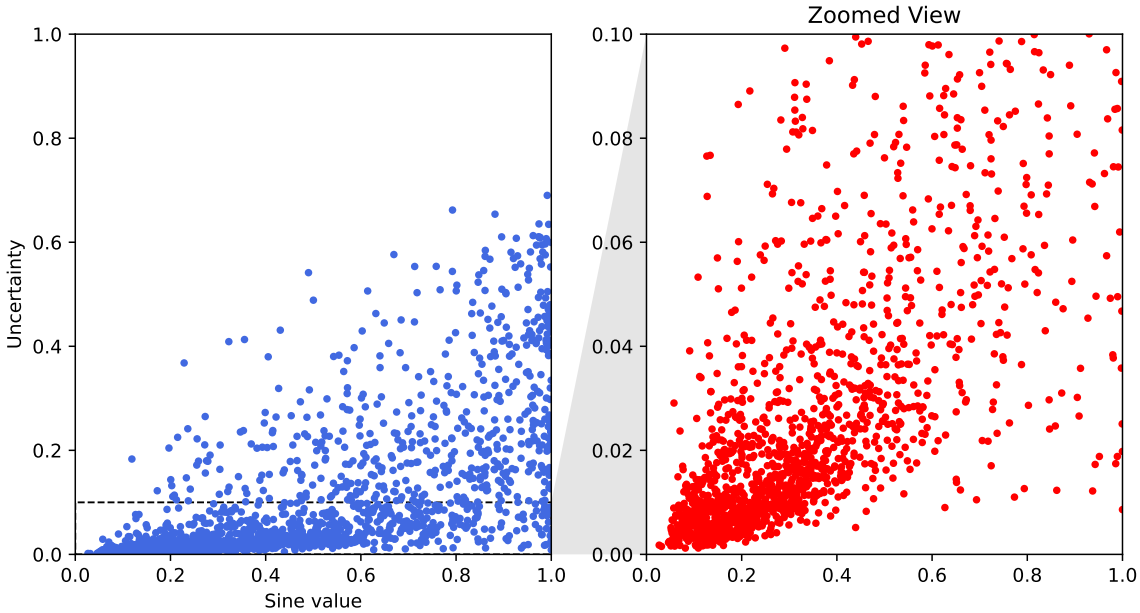


Figure S1: *Correlation between uncertainty and prediction error.* A scatter plot of the ensemble uncertainty metric versus the sine value (error) of the predicted leftmost eigenvector relative to the ground truth. To maintain visual clarity, the results of 2,000 configurations were randomly sampled from the full augmented T1x test set. The positive correlation demonstrates that higher variance among the ensemble members serves as a reliable indicator of inaccurate predictions.

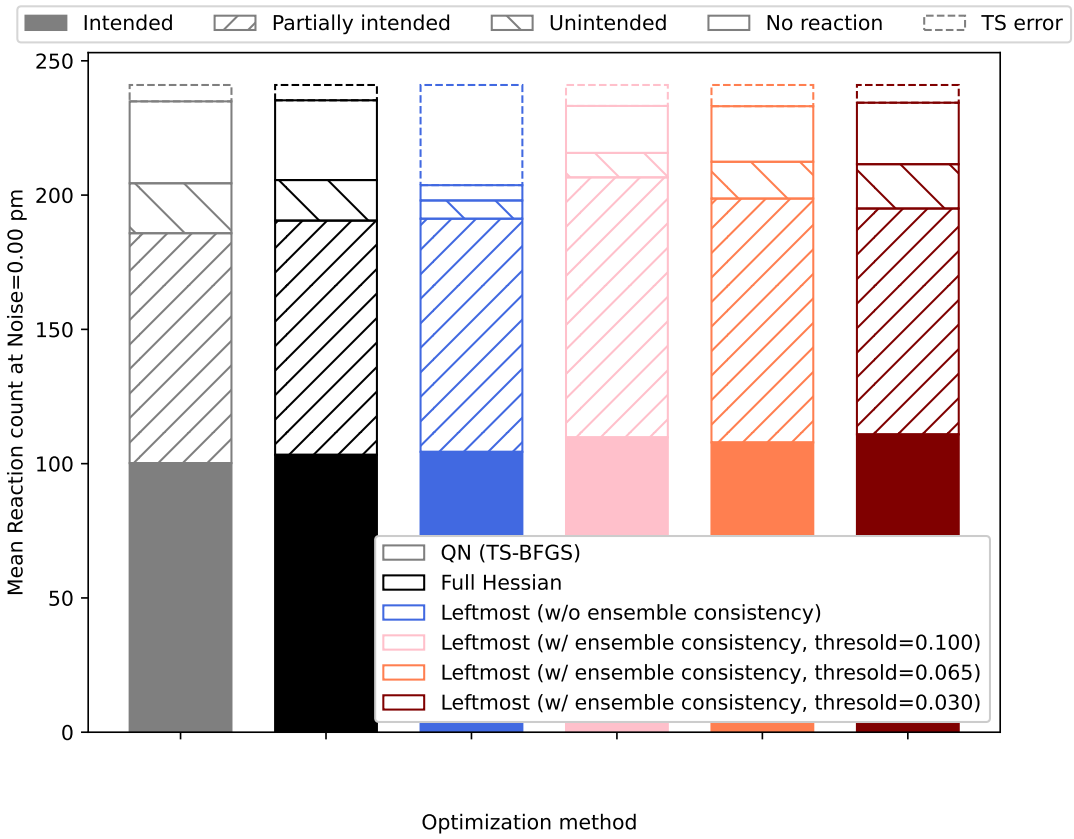


Figure S2: *Ensemble check variation in variance threshold.* The single model LMHE strategy exhibits a higher failure rate due to inaccurate LMHE predictions. The LMHE ensemble consistency check identifies unreliable predictions, triggering a fallback to exact Hessian calculations and reducing failure rates to levels that are competitive with the performance of the full Hessian optimization, and significantly outperforming the standard QN baseline. The ensemble consistency approach is largely insensitive to values of the threshold.

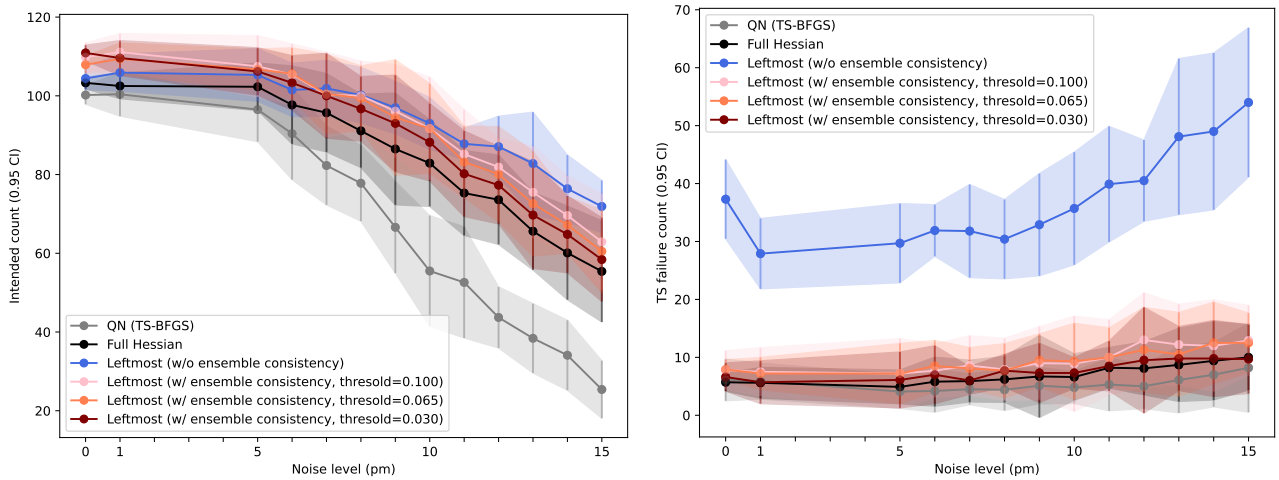


Figure S3: *Comparison of the Robustness of TS Optimizers for 240 test combustion reactions.* (a) The count of intended TS optimizations is plotted against the amplitude of random Gaussian noise applied to the initial geometries. The LMHE optimizer exhibits significantly more intended counts compared to the standard TS-BFGS QN method, maintaining high success rates comparable to the full Hessian approach even at high noise levels. (b) The number of TS optimization runs failing to converge is plotted against noise level. The single model LMHE strategy exhibits a higher failure rate due to inaccurate LMHE predictions. The LMHE ensemble consistency check identifies unreliable predictions, triggering a fallback to exact Hessian calculations and reducing failure rates to levels competitive with the full Hessian and QN baselines. Shaded regions represent the 95% confidence interval derived from 10 independent noise realizations.

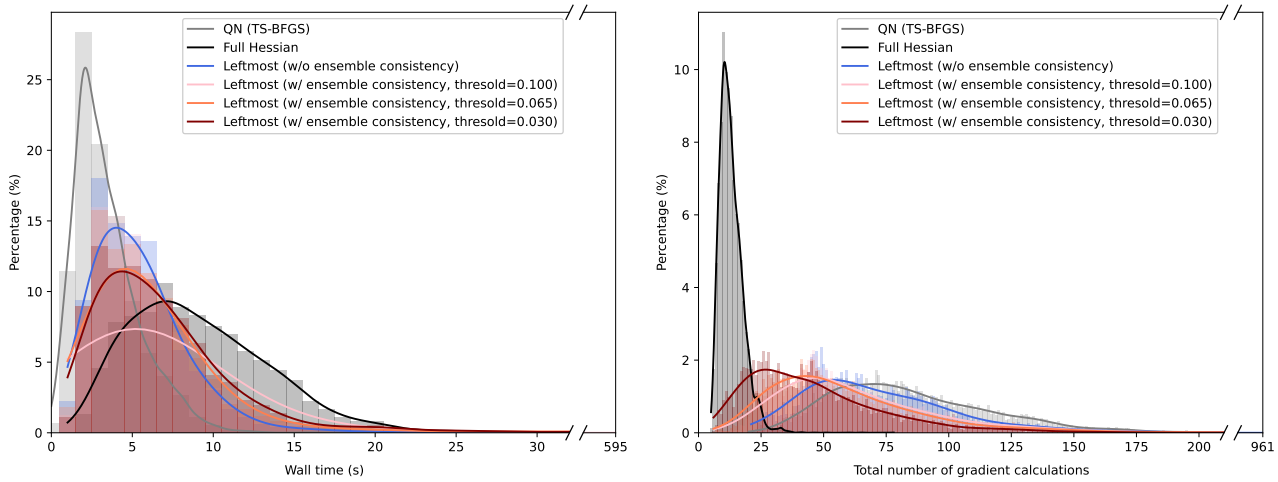


Figure S4: *Wall time and number of gradient evaluation comparisons between TS optimizers.* (a) The distribution of wall times required for converged optimizations across all noise levels. While the standard QN method is the fastest, the full Hessian method suffers from a heavy tail due to the high cost of exact second-derivative calculations. The LMHE achieves a favorable balance, significantly reducing the computational cost relative to the full Hessian approach while providing the curvature information necessary for robust convergence compared to the QN baseline. (b) The distribution of total gradient calculations required for converged optimizations. While the full Hessian method requires the fewest evaluations due to exact curvature information, among the approximate methods both LMHE approaches require fewer total gradient evaluations than the standard QN method. To ensure comparability, the data includes only those optimization where all optimizers successfully recovered the intended TS from the same initial geometry.

Tables

Table S1: *Hyper-parameters for the augmented T1x dataset.* The specific hyper-parameters used to train our encoder-decoder architecture (GotenNet-GA) and the baseline (GotenNet) models at two scales: Small (S, ∼2.1 M parameters) and Medium (M, ∼10.1 M parameters). The five independent models trained for the ensemble-based TS optimizer, including the model used for the single-model optimizer, all utilize the same hyper-parameters as the GotenNet-GA$_{\mathrm{M}}$ model.

Hyper-parameters	GotenNet-GA$_{\mathrm{S}}$	GotenNet-GA$_{\mathrm{M}}$	GotenNet$_{\mathrm{S}}$	GotenNet$_{\mathrm{M}}$
Optimizer	RAdamW	RAdamW	RAdamW	RAdamW
Learning rate scheduling	Reduce on plateau			
Maximum learning rate	1e−4	1e−4	1e−4	1e−4
Learning rate decay	0.8	0.8	0.8	0.7
Learning rate patience	5	5	5	5
Loss function	$1 - (\text{Cosine Similarity})^2$			
Gradient clipping	5.0	5.0	5.0	5.0
Batch size	64	64	64	64
Early stopping	Minimizing validation loss			
Early stopping patience	25	25	25	25
Maximum number of epochs	1000	1000	1000	1000
Dropout rate	0.1	0.1	0.1	0.1
Node dimension (d_{ne})	128	256	128	256
GotenNet edge dimension	128	256	128	256
GotenNet edge refinement dimension	128	256	128	256
Maximum degree (L_{max})	2	2	2	2
Number of GotenNet layers	2	4	4	5
Number of global attention layers	2	1	–	–
Number of radial basis functions	32	32	32	32
Number of inducing points (M)	16	16	–	–
Number of Attention Heads	8	8	8	8
Cutoff radius	5.0	5.0	5.0	5.0

Appendices

Appendix 1. TS-BFGS Formulation

As introduced in Section 4.3 in the main text, the perpendicular component of the Hessian, $\mathbf{H}_{\perp}^{(k+1)}$, is updated using a standard QN formula. Generally, the QN update function takes the form $\mathbf{H}^{(k+1)} = \mathrm{QN}(\mathbf{H}^{(k)}, \mathbf{s}, \mathbf{y})$., where $\mathbf{s} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$ and $\mathbf{y} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$. In our method, this update is applied in the projected subspace, i.e. $\mathbf{H}_{\perp}^{(k+1)} = \mathrm{QN}(\mathbf{H}_{\perp}^{(k)}, \mathbf{s}_{\perp}, \mathbf{y}_{\perp})$. For clarity in the descriptions of the standard algorithms in Appendix , the $\perp$ subscript is omitted.

This work utilizes the TS-BFGS method[1,2]. The formulation adopted in this work is given by:

$$|\mathbf{H}^{(k)}| = \sum_{i}^{d} |\lambda_i^{(k)}| \mathbf{v}_i^{(k)} \mathbf{v}_i^{(k)\mathrm{T}}, \tag{1}$$

$$\mathbf{M} = \mathbf{y}\mathbf{y}^{\mathrm{T}} + |\mathbf{H}^{(k)}| \mathbf{s}\mathbf{s}^{\mathrm{T}} |\mathbf{H}^{(k)}|, \tag{2}$$

$$\mathbf{j} = \mathbf{y} - \mathbf{H}^{(k)}\mathbf{s}, \tag{3}$$

$$\mathbf{u} = \frac{\mathbf{M}\mathbf{s}}{\mathbf{s}^{\mathrm{T}}\mathbf{M}\mathbf{s}}, \tag{4}$$

$$\mathbf{E} = \mathbf{u}\mathbf{j}^{\mathrm{T}} + \mathbf{j}\mathbf{u}^{\mathrm{T}} - \mathbf{u}\mathbf{j}^{\mathrm{T}}\mathbf{s}\mathbf{u}^{\mathrm{T}}, \tag{5}$$

$$\mathbf{H}^{(k+1)} = \mathbf{H}^{(k)} + \mathbf{E}, \tag{6}$$

where d is the number of dimensions in $\mathbf{H}^{(k)}$, and $\lambda_i^{(k)}$ and $\mathbf{v}_i^{(k)}$ are the eigenvalues and eigenvectors of $\mathbf{H}^{(k)}$.

Appendix 2. Global Attention Decoder Architecture

As introduced in the main text, the decoder is composed of global attention layers designed to capture context in the entire molecule, built upon the induced set attention mechanism[3]. As is done in GotenNet, we denote the tensors on each node into two types of features: scalar features $\mathbf{h} \in \mathbb{R}^{d_{ne}}$ invariant under transformations, and high-degree steerable features $\tilde{\mathbf{X}}^{(l)} \in \mathbb{R}^{d_{ne} \times (2l+1)}$ whose transformations depend on their degree l, where d_{ne} denotes node embedding dimension.

In each induced set attention layer, there are M inducing points, where M is a hyperparameter. Each inducing point consists of trainable scalar parameters $\mathbf{h}^I \in \mathbb{R}^{d_{ne}}$. First we aggregate information from graph nodes into inducing points through attention. For inducing point α and graph node i, we compute the query ($\mathbf{q}$) and key ($\mathbf{k}$) representations:

$$\mathbf{q}_{\alpha} = \mathbf{W}_{qa}\mathbf{h}_{\alpha}^{I}, \quad \mathbf{k}_i = \mathbf{W}_{ka}\mathbf{h}_i, \tag{7}$$

where $\mathbf{W}_{qa}, \mathbf{W}_{ka} \in \mathbb{R}^{d_{ne} \times d_{ne}}$ are learnable matrices. A softmax attention is subsequently applied:

$$a_{\alpha i} = \frac{\exp(\mathbf{q}_{\alpha}^{\mathrm{T}}\mathbf{k}_i / \sqrt{d_{ne}})}{\sum_{j \in \mathcal{N}} \exp(\mathbf{q}_{\alpha}^{\mathrm{T}}\mathbf{k}_j / \sqrt{d_{ne}})}, \tag{8}$$

where $\mathcal{N}$ is the set of all nodes in the graph. We compute the value ($\mathbf{v}$ and $\tilde{\mathbf{V}}^{(l)}$) representation:

$$\mathbf{v}_i = \mathbf{W}_{va}\mathbf{h}_i, \quad \tilde{\mathbf{V}}_i^{(l)} = \mathbf{W}_{vva}^{(l)}\tilde{\mathbf{X}}_i^{(l)}, \tag{9}$$

where $\mathbf{W}_{va}, \mathbf{W}_{vva}^{(l)} \in \mathbb{R}^{d_{ne} \times d_{ne}}$ are learnable matrices. Here, $\mathbf{W}_{vva}^{(l)}$ is degree-specific, and uniform weights are applied across spatial dimensions under the same degree to preserve equivariance.

The attention outputs are then computed:

$$\mathbf{h}_\alpha^{attn} = \sum_{i\in\mathcal{N}} a_{\alpha i}\mathbf{v}_i, \quad \tilde{\mathbf{X}}_\alpha^{(l),\,attn} = \sum_{i\in\mathcal{N}} a_{\alpha i}\tilde{\mathbf{V}}_i^{(l)}. \tag{10}$$

Subsequently, the features of the inducing points are updated:

$$\mathbf{h}_\alpha^I \leftarrow \mathbf{h}_\alpha^I + \mathbf{W}_{oa}\mathbf{h}_\alpha^{attn}, \quad \tilde{\mathbf{X}}_\alpha^{(l),\,I} \leftarrow \mathbf{W}_{voa}\tilde{\mathbf{X}}_\alpha^{(l),\,attn}, \tag{11}$$

where $\mathbf{W}_{oa}, \mathbf{W}_{voa} \in \mathbb{R}^{d_{ne}\times d_{ne}}$ are learnable matrices, and $\mathbf{W}_{voa}$ is shared across degrees. The updated features are then passed through an equivariant feed-forward (EQFF) layer[4]:

$$\mathbf{h}_\alpha^I,\, \tilde{\mathbf{X}}_\alpha^{(l),\,I} \leftarrow \mathrm{EQFF}(\mathbf{h}_\alpha^I,\, \tilde{\mathbf{X}}_\alpha^{(l),\,I}). \tag{12}$$

Second, we update the features on graph nodes with the aggregated information on inducing points through another attention layer. The query, key and value representations of graph node i and inducing point α are computed:

$$\begin{aligned} \mathbf{q}_i = \mathbf{W}_{qb}\mathbf{h}_i, \quad \mathbf{k}_\alpha =& \mathbf{W}_{kb}\mathbf{h}_\alpha^I, \quad \mathbf{v}_\alpha = \mathbf{W}_{vb}\mathbf{h}_\alpha^I, \\ \tilde{\mathbf{Q}}_i^{(l)} = \mathbf{W}_{vqb}\tilde{\mathbf{X}}_i^{(l)}, \quad \tilde{\mathbf{K}}_\alpha^{(l)} =& \mathbf{W}_{vkb}^{(l)}\tilde{\mathbf{X}}_\alpha^{(l),\,I}, \quad \tilde{\mathbf{V}}_\alpha^{(l)} = \mathbf{W}_{vvb}^{(l)}\tilde{\mathbf{X}}_\alpha^{(l),\,I}, \end{aligned} \tag{13}$$

where $\mathbf{W}_{qb}, \mathbf{W}_{kb}, \mathbf{W}_{vb}, \mathbf{W}_{vqb}, \mathbf{W}_{vkb}^{(l)}, \mathbf{W}_{vvb}^{(l)} \in \mathbb{R}^{d_{ne}\times d_{ne}}$ are learnable matrices. $\mathbf{W}_{vqb}$ is shared across degrees while $\mathbf{W}_{vkb}^{(l)}$ and $\mathbf{W}_{vvb}^{(l)}$ are degree-specific. The softmax attention is subsequently computed:

$$\begin{aligned} w_{i\alpha} &= \mathbf{q}_i^{\mathrm{T}}\mathbf{k}_\alpha + \sum_{l=1}^{L_{max}} \tilde{\mathbf{Q}}_i^{(l)^{\mathrm{T}}}\tilde{\mathbf{K}}_\alpha^{(l)}, \\ a_{i\alpha} &= \frac{\exp(w_{i\alpha}/\sqrt{(L_{max}+1)\cdot d_{ne}})}{\sum_{\beta\in\mathcal{P}}\exp(w_{i\beta}/\sqrt{(L_{max}+1)\cdot d_{ne}})}, \end{aligned} \tag{14}$$

where $\tilde{\mathbf{Q}}_i^{(l)}$ and $\tilde{\mathbf{K}}_\alpha^{(l)}$ are flattened into one dimensional vectors, L_{max} is a hyper-parameter defining the maximum degree we include in the model, and $\mathcal{P}$ is the set of all inducing points. The attention output is then computed:

$$\mathbf{h}_i^{attn} = \sum_{\alpha\in\mathcal{P}} a_{i\alpha}\mathbf{v}_\alpha, \quad \tilde{\mathbf{X}}_i^{(l),\,attn} = \sum_{\alpha\in\mathcal{P}} a_{i\alpha}\tilde{\mathbf{V}}_\alpha^{(l)}. \tag{15}$$

Subsequently, the features of the graph nodes:

$$\mathbf{h}_i \leftarrow \mathbf{h}_i + \mathbf{W}_{ob}\mathbf{h}_i^{attn}, \quad \tilde{\mathbf{X}}_i^{(l)} \leftarrow \tilde{\mathbf{X}}_i^{(l)} + \mathbf{W}_{vob}\tilde{\mathbf{X}}_i^{(l),\,attn}, \tag{16}$$

where $\mathbf{W}_{ob}, \mathbf{W}_{vob} \in \mathbb{R}^{d_{ne}\times d_{ne}}$ are learnable matrices, and $\mathbf{W}_{vob}$ is shared across degrees. The updated features are then passed through an equivariant feed-forward (EQFF) layer[4]:

$$\mathbf{h}_i,\, \tilde{\mathbf{X}}_i^{(l)} \leftarrow \mathrm{EQFF}(\mathbf{h}_i,\, \tilde{\mathbf{X}}_i^{(l)}). \tag{17}$$

Appendix 3. Equivariance Proof of GotenNet-GA Model

We demonstrate that the proposed architecture is equivariant under the Euclidean group $E(3)$. The goal is to show that the model's output, the LMHE $\mathbf{v}_1$ transforms physically under the

group actions, being invariant to translation and equivariant to orthogonal transformations (rotations and reflections). Since the GotenNet encoder is inherently $E(3)$-equivariant, we focus this proof on demonstrating that the global attention decoder preserves these symmetry properties. The global attention decoder is constructed exclusively from four operations: linear projections, attention mechanisms, feature aggregation, and EQFF blocks. The EQFF block is adopted from GotenNet and its equivariance has been rigorously established in prior work[4]. Therefore, it suffices to demonstrate the equivariance of the remaining three operations: linear projections, attention, and aggregation.

Translation Invariance. The GotenNet encoder generates node features $\mathbf{h}_i$ and $\tilde{\mathbf{X}}_i^{(l)}$ derived exclusively from relative coordinate vectors $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and atomic numbers[4], both remaining invariant under a global translation. Consequently, the input features to the decoder are invariant with respect to global translation. Since the decoder operations operate solely on these internal features without re-introducing absolute coordinates, the outputs of the decoder inherits this translation invariance.

Orthogonal Equivariance. We next prove equivariance under the orthogonal group $O(3)$. Let $g \in O(3)$ be an arbitrary orthogonal transformation. The scalar features $\mathbf{h}$ can be considered as zero-degree features $\tilde{\mathbf{X}}^{(0)}$. For the features $\tilde{\mathbf{X}}^{(l)}$, we employ a flattened column vector notation $\mathbf{x}^{(l)} \in \mathbb{R}^{d_{ne}(2l+1)}$ to facilitate the analysis. The group action on this flattened vector is defined by the block-diagonal matrix $\mathcal{D}^{(l)}(g)$:

$$\mathcal{D}^{(l)}(g) = \mathbf{I}_{d_{ne}} \otimes \mathbf{D}^{(l)}(g), \tag{18}$$

where $\mathbf{I}_{d_{ne}}$ is the identity matrix of dimension d_{ne} and $\mathbf{D}^{(l)}(g)$ is the Wigner D-matrix of degree l for the transformation g (accounting for parity if g includes reflection). The transformed feature vector is $\mathbf{x}^{(l)\prime} = \mathcal{D}^{(l)}(g)\mathbf{x}^{(l)}$.

Linear Projections: The decoder employs degree-specific linear projections parameterized by weights $\mathbf{W} \in \mathbb{R}^{d_{ne} \times d_{ne}}$. In our vectorized formalism, this operation is expressed as $\mathcal{W} = \mathbf{W} \otimes \mathbf{I}_{2l+1}$. We verify that this operation commutes with the group action:

$$\begin{aligned}
\mathcal{D}^{(l)}(g)\mathcal{W} &= (\mathbf{I}_{d_{ne}} \otimes \mathbf{D}^{(l)}(g))(\mathbf{W} \otimes \mathbf{I}_{2l+1}) \\
&= (\mathbf{I}_{d_{ne}}\mathbf{W}) \otimes (\mathbf{D}^{(l)}(g)\mathbf{I}_{2l+1}) \\
&= (\mathbf{W}\mathbf{I}_{d_{ne}}) \otimes (\mathbf{I}_{2l+1}\mathbf{D}^{(l)}(g)) \\
&= (\mathbf{W} \otimes \mathbf{I}_{2l+1})(\mathbf{I}_{d_{ne}} \otimes \mathbf{D}^{(l)}(g)) \\
&= \mathcal{W}\mathcal{D}^{(l)}(g).
\end{aligned} \tag{19}$$

Since the operators commute, the projected query ($\mathbf{q}$), key ($\mathbf{k}$), and value ($\mathbf{v}$) vectors transform equivariantly according to $\mathcal{D}^{(l)}(g)$.

Attention Mechanism: The attention logits (Eq. 14) utilize the inner product of flattened feature vectors of each degree $\mathbf{q}^{(l)\mathrm{T}}\mathbf{k}^{(l)}$. Under the transformation g:

$$\begin{aligned}
(\mathbf{q}^{(l)\prime})^{\mathrm{T}}\mathbf{k}^{(l)\prime} &= (\mathcal{D}^{(l)}(g)\mathbf{q}^{(l)})^{\mathrm{T}}(\mathcal{D}^{(l)}(g)\mathbf{k}^{(l)}) \\
&= \mathbf{q}^{(l)\mathrm{T}}\mathcal{D}^{(l)}(g)^{\mathrm{T}}\mathcal{D}^{(l)}(g)\mathbf{k}^{(l)},
\end{aligned} \tag{20}$$

where matrix $\mathcal{D}^{(l)}(g)$ is orthogonal:

$$\mathcal{D}^{(l)}(g)^{\mathrm{T}}\mathcal{D}^{(l)}(g) = \mathbf{I}_{d_{ne}} \otimes (\mathbf{D}^{(l)}(g)^{\mathrm{T}}\mathbf{D}^{(l)}(g)) = \mathbf{I}_{d_{ne}} \otimes \mathbf{I}_{2l+1} = \mathbf{I}. \tag{21}$$

Thus, the inner product is preserved, ensuring that the attention weights a remain invariant scalars.

Feature Aggregation: The attention output is a linear combination of value vectors weighted by invariant attention weights. The aggregated output under transformation g is:

$$\mathbf{x}^{(l),attn'} = \sum a(\mathcal{D}^{(l)}(g)\mathbf{v}^{(l)}) = \mathcal{D}^{(l)}(g)\left(\sum a\mathbf{v}^{(l)}\right) = \mathcal{D}^{(l)}(g)\mathbf{x}^{(l),attn}. \tag{22}$$

This confirms that the output features of the global attention preserve $O(3)$-equivariance.

References

[1] Josep Maria Bofill. Remarks on the updated hessian matrix methods. International journal of quantum chemistry, 94(6):324–332, 2003.

[2] Josep Maria Anglada and Josep Maria Bofill. How good is a broyden–fletcher–goldfarb–shanno-like update hessian formula to locate transition structures? specific reformulation of broyden–fletcher–goldfarb–shanno for optimizing saddle points. Journal of computational chemistry, 19(3):349–362, 1998.

[3] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 3744–3753. PMLR, 09–15 Jun 2019.

[4] Sarp Aykent and Tian Xia. Gotennet: Rethinking efficient 3d equivariant graph neural networks. In The Thirteenth International Conference on Learning Representations, 2025.