Colour me shocked: Exact Molecular Hessians from local MLIPs in O(N) time using sparse differentiation!

Luca Thiede*
Department of Computer Science
University of Toronto
Vector Institute for Artificial Intelligence
Toronto, Canada
luca.thiede@yahoo.com

Andreas Burger*
Department of Computer Science
University of Toronto
Vector Institute for Artificial Intelligence
Toronto, Canada

Alán Aspuru-Guzik
Department of Computer Science, University of Toronto
Vector Institute for Artificial Intelligence
Acceleration Consortium
Senior Fellow, Canadian Institute for Advanced Research (CIFAR)
NVIDIA
Toronto, Canada

Abstract

The Hessian of the energy with respect to the nuclear positions is indispensable in atomistic modelling. However, constructing this matrix requires $O(N)$ Hessian vector products, traditionally limiting high-accuracy Hessians to small systems. Machine learning interatomic potentials (MLIPs) have accelerated atomistic modelling by providing highly accurate energies and forces at $O(N)$ cost, yet the resulting $O(N^2)$ cost of Hessians remains a practical bottleneck for large systems. Based on the insight that we can derive the sparsity pattern for an MLIP's Hessians in closed form, we show in this paper how to use techniques from sparse automatic differentiation to reduce the cost of a local MLIP's Hessians to a system-size-independent number of Hessian-vector products, yielding overall $O(N)$ total cost without any approximations. We benchmark our approach on a variety of systems ranging from alkane chains to water clusters to Aβ40 conformers. Depending on the MLIP configuration, we achieve the linear scaling regime already on relatively small systems, resulting in large runtime reductions between $2\times$-$15\times$ for these systems. This opens up the possibility of scaling high-accuracy MLIP Hessians to very large systems, such as proteins that were previously inaccessible.

1 Introduction

The Hessian is a basic ingredient in atomistic modelling, entering, for example, vibrational analysis, normal-mode calculations, transition-state characterization, and second-order geometry optimization. Yet building the full Hessian is far more expensive than evaluating the underlying energy or forces, typically scaling by a factor of $O(N)$ worse.

Machine learning interatomic potentials built on architectures such as SchNet, MACE, eSEN, AimNet, and TensorNet and trained on large databases of electronic structure data, such as OMOL25 Levine et al. [2025], now provide reliable surrogate potentials Schütt et al. [2017], Batatia et al. [2022], Fu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

et al. [2025], Zubatyuk et al. [2019], Simeon and De Fabritiis [2023], Ko et al. [2025] with accuracy comparable to high level density functional theory (DFT) at a scaling of $O(N)$ due to the enforcement of cutoff radii. Forces are trivially available because they follow by differentiating the learned scalar energy with respect to Cartesian coordinates once, which roughly doubles the cost but introduces no additional asymptotic scaling compared to the energy. By contrast, constructing the full $3N \times 3N$ Cartesian Hessian requires $3N$ second-derivative probes, and $O(N^2)$ storage, thereby introducing an additional factor in system size in time and memory and leading to a total time complexity of $O(N^2)$, see section 2.

Prior work in chemistry has repeatedly exploited locality and effective sparsity to reduce the cost of second derivatives. Representative examples include reduced-scaling analytic Hessians based on localized density-matrix response, sparse numerical Hessians that evaluate only important couplings, block- or fragment-reduced vibrational models for large systems, or by using Hessian information from a cheaper method to accelerate the Hessian computation of a more expensive method Kussmann et al. [2015], Yang et al. [2024], Ghysels et al. [2011], Nakata and Fedorov [2021], Sanders et al. [2015]. A different route is taken in recent work that achieves $O(N)$ Hessian prediction by learning second-order information directly, rather than by differentiating through the force field Burger et al. [2025]. However, all these methods have in common that they are only approximately correct. In parallel, the field of automatic differentiation developed methods to exploit sparsity in derivatives for arbitrary black-box computer programs. In contrast to the previously listed approaches, sparse automatic differentiation is guaranteed to be exact Curtis et al. [1974], Newsam and Ramsdell [1983], Coleman and Moré [1984], Powell and Toint [1979], Coleman and Cai [1986], Gebremedhin et al. [2005]. The central practical limitation of sparse automatic differentiation is that it relies on knowing the sparsity pattern of the Hessian Hill et al. [2025]. In general-purpose settings, obtaining that pattern requires a sparsity-detection pass, which can be costly. Here we show, however, that for a broad class of modern local message-passing interatomic potentials, the Hessian sparsity pattern follows trivially from how information propagates through a finite number of message-passing layers within a radius of neighbours.

Once that pattern is known, the classical sparse-derivative pipeline constructs a seed matrix from an admissible colouring, evaluates compressed Hessian-vector products, and recovers the full Hessian exactly. This graph-derived sparsity removes the sparsity pattern detection pass and makes exact sparse Hessians efficient even for single Hessian evaluations of relatively small systems. By defining heuristic sparsity patterns based on a cutoff radius, we can further extend the idea more broadly to any ab-initio method as long as the Hessian is well behaved and we accept an approximation error. This leads to an efficient black-box sparse Hessian algorithm without the need to change any of the underlying quantum chemistry code.

We summarize the contributions of this paper as follows:

1. We derive the sparsity pattern for Cartesian Hessians of the most common forms of MLIPs in closed form.
2. We introduce `ColPackPy`, an easy to use Python interface for sparse Hessian computations, including the graph-derived sparsity construction for MLIPs, colouring via ColPack Gebremedhin et al. [2013], compressed probing, and differentiable recovery in PyTorch.
3. We benchmark the resulting pipeline across representative molecular systems and MLIPs and quantify the resulting speedups across favourable and unfavourable compression regimes.

2 Background

The standard way to obtain second-order information in major automatic differentiation frameworks such as Jax and PyTorch is through Hessian-vector products (HVPs) Pearlmutter [1994], Griewank and Walther [2008], Baydin et al. [2018]. For a black box scalar function $E : \mathbb{R}^{3N} \to \mathbb{R}$ (in our case this will of course be the MLIP mapping the atomic positions to the energy), evaluated at coordinates $\mathbf{R}$, let $\mathbf{g}(\mathbf{R}) = \nabla_{\mathbf{R}} E(\mathbf{R})$. Then for any probing vector $\mathbf{v} \in \mathbb{R}^{3N}$,

$$\mathbf{H}(\mathbf{R})\mathbf{v} = \frac{d}{d\epsilon}\, \mathbf{g}(\mathbf{R} + \epsilon \mathbf{v})\Big|_{\epsilon=0}, \tag{1}$$

that is, the directional derivative of the gradient, equivalently the Jacobian-vector product of $\mathbf{g}$ in the direction $\mathbf{v}$ Pearlmutter [1994], Griewank and Walther [2008]. In settings where an exact HVP

operator is unavailable, the same directional probe can be approximated by the finite difference

$$\mathbf{H}(\mathbf{R})\mathbf{v} \approx \frac{\mathbf{g}(\mathbf{R} + \epsilon\mathbf{v}) - \mathbf{g}(\mathbf{R})}{\epsilon}, \tag{2}$$

Importantly, AD frameworks expose these products without explicit formation of $\mathbf{H}$ Pearlmutter [1994], Baydin et al. [2018]. In fact, to get the full dense Hessian, we usually apply HVPs with the standard basis vectors $\{\mathbf{e}_i\}_{i=1}^{3N}$ to recover the explicit Hessian column by column,

$$\mathbf{H}(\mathbf{R})\mathbf{e}_i = \mathbf{H}_{:,i}(\mathbf{R}), \qquad \mathbf{H}_{\text{explicit}} = \mathbf{H}_{\text{implicit}}\,\mathbf{I}, \tag{3}$$

In contrast, sparse-Hessian methods reduce the number of probes by replacing the identity matrix with a compressed seed matrix derived from a colouring of the sparsity pattern, as we will describe in the next section Coleman and Moré [1984], Gebremedhin et al. [2005, 2009]; see also Fig. 1.

3 Sparse Automatic Differentiation

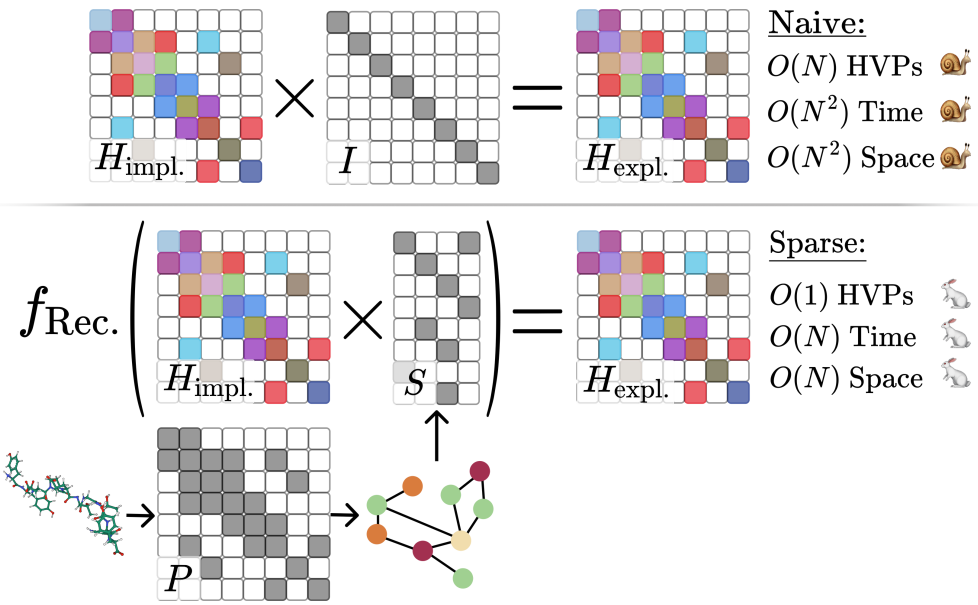


Figure 1: Top: The naive method probes each direction of the implicit Hessian individually, yielding one column of the Hessian per HVP. This requires N HVPs, where N is the dimension of the Hessian and can be written as the multiplication of the implicit Hessian with the identity matrix. Bottom: We first derive a sparsity pattern from the molecule and the MLIP cutoff and layer count. Using a graph colouring algorithm, this is mapped into the seed matrix S which is much lower dimensional than I. After multiplication with the implicit Hessian $Y = H_{\text{impl.}}S$ we can use a recovery algorithm $f_{\text{Rec.}}$ to reconstruct the explicit $H_{\text{expl.}}$.

The sparse Hessian procedure employed here comprises four stages, see figure 1:

1. First we derive a sparsity pattern $\mathbf{P}$ from the atomistic graph, section 3.1
2. We then map the sparsity pattern to a graph and use graph colouring algorithms to obtain a seed matrix $\mathbf{S} \in \{0,1\}^{3N\times p}$, whose columns are our probing vectors. Our goal is to get a seed matrix with p as small as possible, while still having enough information to recover the Hessian exactly.
3. We probe the Hessian using the seed matrix to obtain the response $\mathbf{Y} = \mathbf{H}_{\text{implicit}}\mathbf{S}$
4. We use a reconstruction algorithm to get the Hessian $\mathbf{H}_{\text{explicit}} = f_{\text{rec.}}(\mathbf{Y})$

The essential simplification relative to generic sparse-derivative settings is that we can derive the sparsity pattern $\mathbf{P}$ *a priori* from the locality of the message-passing architecture (or get an approximately by a heuristic cutoff radius).

3.1 The Sparsity Pattern of MLIPs

To apply sparse derivative techniques, we need the sparsity pattern $\mathbf{P}$ of the Hessian Coleman and Moré [1984], Gebremedhin et al. [2009],

$$P_{ij} = \begin{cases} 1 & \text{if } \mathbf{H}_{ij} \neq 0, \\ 0 & \text{if } \mathbf{H}_{ij} = 0. \end{cases} \tag{4}$$

which indicates whether the Hessian entry coupling i and j can be nonzero. Detecting this sparsity pattern is often complicated and expensive Gebremedhin et al. [2009] for black box programs. Therefore, using sparse automatic differentiation gives the most benefits if we either reuse the same sparsity pattern across multiple calculations with different inputs, or we can use knowledge about the problem to determine the pattern efficiently. Our approach falls into the latter category.

Most modern ML interatomic potentials are local message-passing models and can be written as

$$E_{\text{tot}} = \sum_{a=1}^{N} E_a\big(\{\mathbf{R}_i\}_{i \in \mathcal{R}(a)}\big), \tag{5}$$

where $\mathcal{R}(a)$ is the receptive field of atom a. This form is standard for atom-centered and message-passing models for molecules and materials Gilmer et al. [2017], Schütt et al. [2017], Batatia et al. [2022]. If $\mathbf{A}$ is the adjacency matrix of the interaction graph and the model consists of L message-passing layers, then

$$\mathcal{R}(a) = \left\{ i \;\middle|\; (\tilde{\mathbf{A}}^{\,L})_{ai} \neq 0 \right\}, \qquad \tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}, \tag{6}$$

where the identity is included so that each atom remains in its own receptive field. This is the standard graph-theoretic interpretation of L rounds of message passing: Information propagates along paths of length of at most L Gilmer et al. [2017].

We define the receptive-field incidence matrix

$$\mathbf{M} = \mathbf{1}\left[\tilde{\mathbf{A}}^{\,L} \neq 0\right], \tag{7}$$

so that $M_{ai} = 1$ exactly when atom i lies in the receptive field of center a. Then a Hessian block (i, j) can be nonzero only if there exists some atomic term E_a that depends on both $\mathbf{R}_i$ and $\mathbf{R}_j$, i.e.

$$\boxed{P^{\text{atom}}_{ij} = \mathbf{1}[\exists a \text{ such that } M_{ai} M_{aj} = 1] = \mathbf{1}\big[(\mathbf{M}^\top \mathbf{M})_{ij} \neq 0\big]} \tag{8}$$

This gives a graph-based upper bound on the atomwise Hessian sparsity pattern induced by the local decomposition of the energy Schütt et al. [2017], Batatia et al. [2022]. Extensions to sparsity patterns of higher order derivatives are trivial by adding more M_{ak} to the product.

So far, we have only considered nodes, but each node is embedded in Cartesian space, and therefore, the full Cartesian Hessian sparsity pattern is

$$\boxed{\mathbf{P} = \mathbf{P}^{\text{atom}} \otimes \mathbf{1}_{3\times 3}} \tag{9}$$

Sparsity is therefore inferred from the graph topology, avoiding the computation from the sparsity-detection pass Gebremedhin et al. [2009].

In some cases, the model does not permit any exact sparsity. This can occur when message-passing cutoffs are too large compared to the system size, when the network has many layers or non-local interactions such as charge equilibration, or when one seeks to apply the same sparsity logic to electronic-structure methods such as density functional theory. In these cases, we can instead use a heuristic cutoff-based sparsity pattern $\mathbf{P}_{\text{approx}}$. There are several reasonable choices. The simplest one is to set $L = 1$ together with a distance cutoff. However, we can also use other, more sophisticated patterns like choosing $L > 1$ and a smaller cutoff to capture long-range responses mediated by covalent bonds, or include coupling terms between formally charged groups and other prior chemical domain knowledge. A more blackbox approach is to run a cheap method like xtb Grimme et al. [2017] or lower rung DFT with a large or no cutoff, threshold the Hessian to obtain a sparsity pattern, and use it for a more expensive method like higher rung DFT.

Next, we use $\mathbf{P}$ to construct a compressed set of probing vectors for the Hessian, so that the Hessian can be reconstructed exactly from as few directional probes as permitted by the chosen colouring and recovery scheme Coleman and Moré [1984], Gebremedhin et al. [2005, 2009].

3.2 From Sparsity to Seed Matrix

Given the structural sparsity pattern $\mathbf{P}$ of the Hessian, we form the associated adjacency graph

$$G(\mathbf{P}) = (V, E), \qquad V = \{1, \dots, n\}, \qquad (i,j) \in E \iff i \neq j \text{ and } P_{ij} = 1. \tag{10}$$

This means, the vertices correspond to the columns of the sparsity pattern $\mathbf{P}$ and two vertices are connected by an edge if the off-diagonal entry in sparsity pattern that connects them is non-zero.

A colouring of $G(\mathbf{P})$ partitions these columns into colour classes, which defines a seed matrix $\mathbf{S} \in \{0,1\}^{n \times p}$. If $c : V \to \{1, \dots, p\}$ denotes the colouring, then

$$S_{ik} = \mathbf{1}[c(i) = k]. \tag{11}$$

Each column of $\mathbf{S}$ is therefore a probing direction that simultaneously perturbs all variables assigned the same colour. The compressed Hessian

$$\mathbf{Y} = \mathbf{HS} \tag{12}$$

can then be computed using one Hessian probe per colour class. The objective is to find a colouring, and hence a seed matrix $\mathbf{S}$, with as few colours as possible while still allowing exact reconstruction of $\mathbf{H}$ Gebremedhin et al. [2005, 2009].

While there are several colouring schemes that permit exact recovery, the minimum requirement common to all of them is that the colouring is *proper*:

$$(i,j) \in E \implies c(i) \neq c(j). \tag{13}$$

This ensures that two variables whose coupling may be nonzero are never assigned to the same probing direction. However, proper colouring alone is not sufficient for exact Hessian recovery. Even when adjacent vertices have different colours, the compressed product $\mathbf{Y}$ can still mix multiple unknown Hessian entries in a way that prevents unique separation. Additional structural constraints are therefore needed to guarantee that $\mathbf{Y}$ contains enough information to recover $\mathbf{H}$ exactly. We distinguish between *direct recovery*, where Hessian entries can be read off directly from $\mathbf{Y}$, and *substitution-based recovery*, where some entries are obtained directly from $\mathbf{Y}$ and the remaining ones are recovered sequentially by subtracting contributions from entries that have already been determined Gebremedhin et al. [2005, 2007, 2009], Montoison et al. [2025].

For direct recovery of a symmetric Hessian, the colouring must satisfy the `star colouring` condition, which demands in addition to proper colouring that every path on four vertices uses at least three colours Gebremedhin et al. [2007, 2009].

For substitution-based recovery, one can weaken this to an `acyclic colouring`, which demands, in addition to proper colouring, that every cycle uses at least three colours Gebremedhin et al. [2007, 2009].

Star colouring is more restrictive, but it leads to a simpler reconstruction procedure. Acyclic colouring is less restrictive and therefore typically requires fewer colours, but the reconstruction step is more involved; in practice, indirect (substitution-based) recovery is often faster overall despite the more complicated decompression step. For implementation details on colouring and recovery, please refer to Gebremedhin et al. [2009]

For sparse graphs with maximum degree δ, the coloring is typically inexpensive, with commonly used heuristics scaling on the order of $O(|V|\delta^2)$. For sparse graphs, δ becomes a constant with graph size, and the coloring itself therefore grows linear with system size.

3.3 Scaling

We now make the scaling regime in which the number of Hessian-vector products becomes independent of system size precise: Consider a sequence of systems with fixed MLIP cutoff radius, fixed message-passing depth L, and bounded local density, i.e. a bounded number of at most d atoms inside any cutoff ball, independent of the total system size N. Then each L-hop receptive field has size bounded by a constant depending only on d and L. Consequently, the induced Cartesian Hessian sparsity graph $G(P)$ has maximum degree $\delta(G(P))$ bounded by a function dependent only on d and L, and hence independent of the number of atoms N. Standard sparse-Hessian colouring results then imply that admissible star or acyclic colourings require $p = O_{d,L}(1)$ colours, corresponding to a system-size-independent number of HVPs Gebremedhin et al. [2005]. Since each HVP through a local MLIP costs $O(N)$, the sparse Hessian can be constructed in $O(N)$ time and stored in $O(N)$ space, using a sparse tensor format.

3.4 ColPackPy

We package the full sparse-Hessian workflow in a lightweight Python interface using Python bindings of the C++ ColPack library Gebremedhin et al. [2013] for the graph-colouring algorithms, reimplementations of the recovery algorithms in PyTorch, and utility functions to derive the sparsity pattern from the connectivity graph and number of layers. The library supports sparse return types for scalable Hessians of very large systems. For the jax ecosystem, asdex Hill and Dalle [2026] provides similar functionality, although no acyclic coloring is supported as of now. Concurrent work Langer et al. [2026] used asdex to sparsely calculate MLIP Hessian similarly to our work.

A typical usage pattern is shown below:

```python
from colpackpy.torch_utils import compute_sparse_hessian_fn

def energy_fn(x_flat: torch.Tensor) -> torch.Tensor:
    pos = x_flat.view(-1, 3)
    return mlip_energy(pos)  # scalar torch.Tensor from any MLIP

sparsity_pattern, n = create_sparsity_pattern_from_graph(g, # "dgl.DGLGraph" or torch_geometric.data.Data
    num_layers # int
    )

H, stats = compute_sparse_hessian_fn(
    energy_fn,
    positions, # flattened Cartesian coordinates, shape (3 * n_atoms,)
    sparsity_pattern,
    n=n,
    coloring_algorithm="ACYCLIC",    # or "STAR"
    return_stats=True,
)
```

Because the compressed probing and recovery steps are implemented in PyTorch, the resulting sparse-Hessian calculation remains differentiable.

4 Experiments

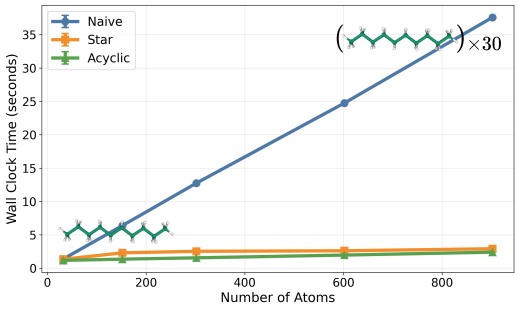
The results presented here address five questions.

1. How much does sparse colouring reduce the number of HVPs?
2. How strongly does that reduction translate into wall-clock savings?
3. How do the savings vary across the systems studied?
4. Which colouring algorithm performs best?
5. How large is the error stemming from an approximate sparsity pattern?

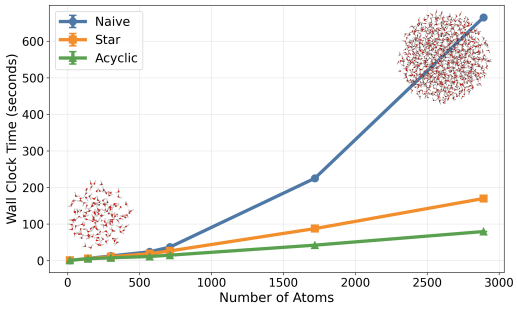
We compare the (`naive`) Hessian to the sparse Hessian via direct recovery (`star`), and substitution-based recovery (`acyclic`). All reported runtimes are timed on a Quadro RTX 5000 GPU, and all MLIP implementations and checkpoints came from the MatGL library Ko et al. [2025]. The maximal difference between the naive and the sparsely recovered Hessians over all elements is on the order of 10^{-8} for float32, so within floating point precision as expected.

4.1 Exact recovery

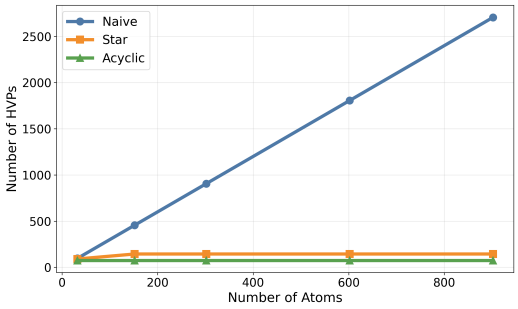
We characterize the speedup of exact sparse differentiation across three systems and two MLIP settings: We use linear alkane chains to test the best case scenario, dense water clusters Gijón and Hernández [2022] for the worst case, and Aβ40, a 40-residue protein in differently folded states for intermediate cases taken from the protein ensemble database Protein Ensemble Database, Lazar et al. [2021]. For the MLIPs, we use TensorNet Simeon and De Fabritiis [2023] with one layer and a



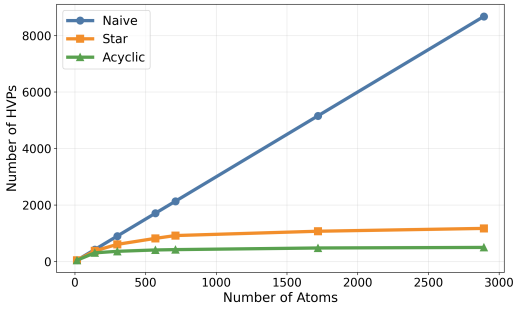
(a) Runtime scaling for a one layer TensorNet on alkane chains.



(b) Runtime scaling for a one layer TensorNet on water clusters.



(c) HVP count scaling for a one layer TensorNet on Alkane chains.



(d) HVP count scaling for a one layer TensorNet on water clusters.

Figure 2: Runtime and HVP count scaling of a one layer TensorNet on increasing alkane chains and water clusters.

cutoff of 5 Å, similar to what is used in simulations of very large systems such as Wang et al. [2025], Song et al. [2024], Willman et al. [2024], Liu et al. [2026] and TensorNet with two layers and a cutoff of 5 Å, similar to some foundation models such as MACE-OFF Kovács et al. [2023]. For each system, we count the number of HVPs and the total runtime. We additionally tested MACE-OFF23 (medium) which has the same sparsity pattern as the larger tested TensorNet. Sparsity therefore saves the same number of HVPs; however, each MACE HVP is 38x more expensive than TensorNet (48ms vs 1852ms on the C300 Alkane chain on our system). The share of the total runtime of the coloring overhead therefore shrinks from 26% for TensorNet to 0.8% for MACE, and the speedup for MACE becomes even larger.

Alkanes and Water Cluster We plot the timings of the one-layer MLIP in 2a and 2b. Plots of the HVP counts as well as the timings of the two-layer MLIPs are in appendix 6 and 7. The HVP count for the `naive` approach grows linearly, while it tapers and becomes constant for the sparse approaches as expected. This translates into linear scaling timings even in the unfavourable dense water cluster case. Especially `acyclic` leads to consistent and large runtime savings of up to $15\times$ for alkanes and $8\times$ for water cluster, which is why we recommend it as the default choice. The `star` algorithm still reduces the HVP count but generally performs worse than the`acyclic` coloring and should therefore only be understood as a reference.

Aβ40 Protein Conformer We further plot the HVP count and timings with a breakdown in colouring, HVPs, and recovery in 3 for the one-layer MLIP, and in the appendix 8 for the two-layer MLIP. We see a similar story as in the previous section, with consistent and significant savings in HVP and runtime with up to $6\times$ runtime savings.

The breakdown also reveals that the colouring, while small, is not completely negligible. The reason for this is that the HVPs are running on GPU while the colouring is currently implemented on CPU, which could be addressed in the future with GPU based colouring algorithms.

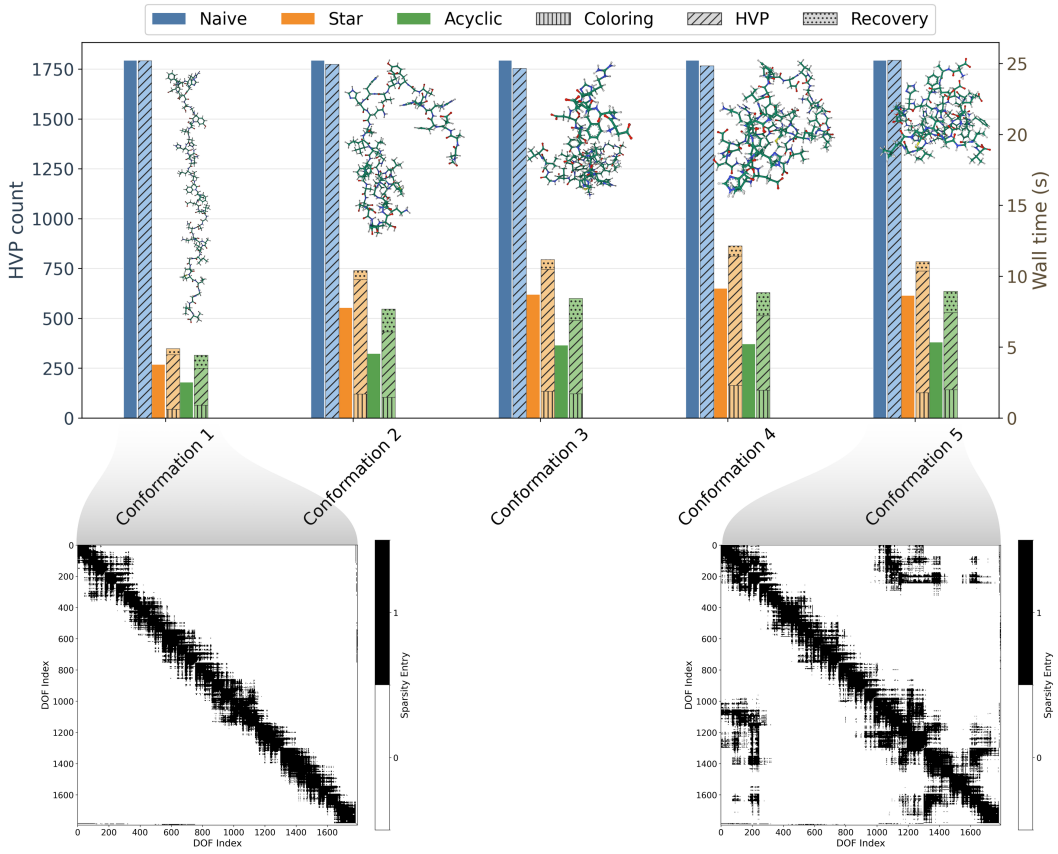


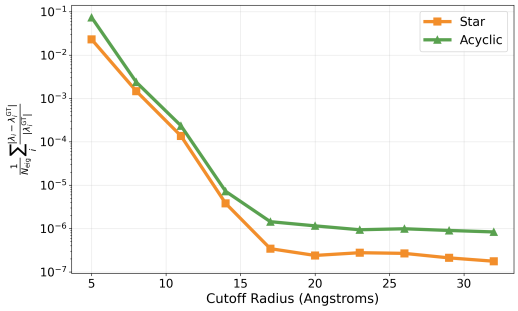
Figure 3: Runtime for computing the Hessian of a one layer TensorNet on different Aβ40 conformations with a runtime breakdown into the colouring, HVP and recovery step. The corresponding sparsity pattern induced by TensorNet is shown for two examples on the bottom.

4.2 Cutoff-based approximate recovery

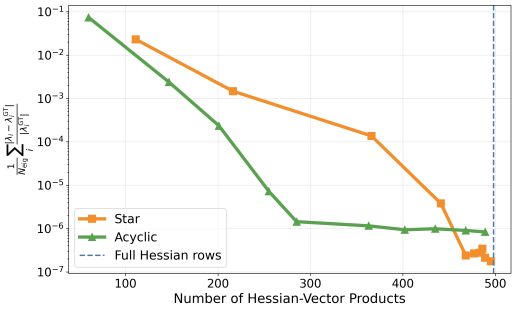
In cases where the receptive field of the model becomes too large compared to the system, such that there is no exactly exploitable sparsity, we can use a heuristic sparsity pattern as discussed in section 3.1. The simplest of such heuristics is to build it from a simple atom pair cutoff radius. We test this on the Chignolin protein Wang et al. [2023] in a difficult non-equilibrium conformation with maximal span of about 32Å, by sweeping over cutoff radii from 5 to 32 Åand record the average relative eigenvalue error $\frac{1}{3N}\sum_i^{3N}\frac{|\lambda_i^{\text{approx}}-\lambda_i^{\text{GT}}|}{|\lambda_i^{\text{GT}}|+1e^{-5}}$ after Eckart projection Eckart [1935] and the number of HVPs. In the appendix A.3 we additionally show the maximum error over Hessian elements. We test two models:

CHGNet We use a CHGNet model Deng et al. [2023] with 5 layers and 6 Å, resulting in a receptive field of 30 Å, almost entirely covering the protein. The results are shown in figure 4a and 4b. We see that both `acyclic` and `star` get, for this system, close to numerical accuracy with an error between $10^{-6} - 10^{-7}$ after about 17 Å. However, the `acyclic` method clearly saves more HVPs at the 17 Å cutoff.

DFT We further test the approach with DFT at the PBE level of theory on the same Chignolin protein using ORCA Neese [2012]. As ORCA does not expose their HVP API we emulate the HVPs. For DFT and HVP details, see appendix A.3. The results are shown in figure 5a and 5b. Again, we drop to very low errors of about 0.1% average eigenvalue error after about 17 Å. However, compared to CHGNet the error drops slower, indicating that there are slightly more long-range dependencies present in DFT Hessians.

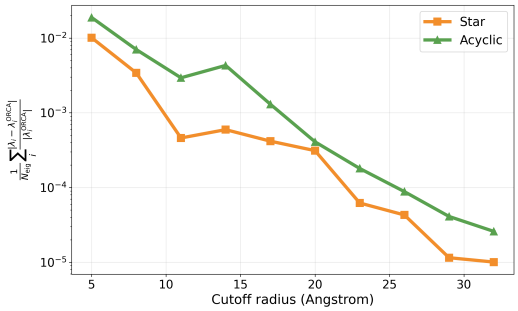


(a) Average relative eigenvalue error after Eckart projection on Chignolin, dependent on the cutoff radius when used for the CHGNet MLIP with a receptive field covering the entire molecule.

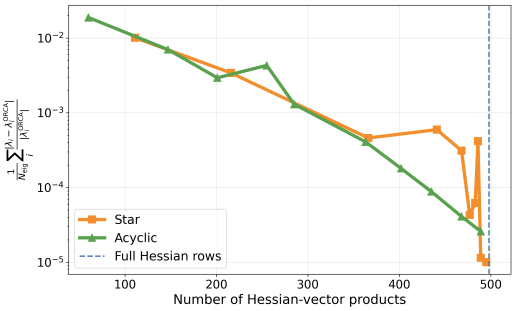


(b) Average relative eigenvalue error after Eckart projection on Chignolin, dependent on the number of HVP we perform as a result of the cutoff radius implied sparsity pattern.

Figure 4: Approximation error induced by using an approximate cutoff-based sparsity pattern with the CHGNet MLIP with a receptive field covering the entire protein.



(a) Average relative eigenvalue error after Eckart projection on Chignolin, dependent on the cutoff radius when used with DFT at PBE level of theory.



(b) Average relative eigenvalue error after Eckart projection on Chignolin with DFT at PBE level of theory, dependent on the number of HVP we perform as a result of the cutoff radius implied sparsity pattern.

Figure 5: Approximation error induced by using an approximate cutoff-based sparsity pattern with DFT at the PBE level of theory.

5 Extensions and future work

Currently, the colouring cost is usually small, but not completely negligible, as it runs on CPU. In the future, it might be possible to further reduce the runtime by adapting advances in GPU accelerated colouring algorithms to star and acyclic colouring Chen et al. [2017].

The framework can also be extended to higher derivatives like the anharmonicity Kaser et al. [2021]. Our sparsity pattern derivation extends trivially to higher orders. The coloring can be achieved either by chaining lower-order derivatives or using tensor colorings Deussen and Naumann [2019]. Sparse higher-order derivatives are useful, for example, to scale anharmonic vibrational corrections like VPT2 to large systems.

Another application of our work is batched parallel computation of Hessians of different non-interacting systems. Naively, this requires $3N_{\text{tot}}$ HVPs with a total number of atoms N_{tot}. However, the total Hessian is block-diagonal, and using our framework, the number of HVPs can therefore be reduced to at most $3N_{\text{max}}$, with the maximum number of atoms over the systems in the batch N_{max}, even if each subsystem is dense.

Finally, it might be possible to supervise MLIP second derivatives in the compressed subspace spanned by the seed matrix, instead of random HVPs as is currently done Rodriguez et al. [2026], Koker et al. [2026].

6 Conclusions

By deriving the sparsity pattern for MLIPs analytically without the need for expensive detection algorithms, this paper makes sparse second-order derivatives practical, even for relatively small systems, yielding linear scaling exact Hessians while leaving the underlying force-field architecture unchanged. To make these techniques as accessible as possible, we introduce the easy to use ColPackPy library. Across various systems, the method reduces both probe counts and runtimes considerably. Real production biomolecular systems can be orders of magnitude larger than the considered systems. For example, a typical protein can have on the order of 1000 residues, compared to the 40 of our Aβ40 test case. Since our approach scales better, the savings for more typical systems will be even larger.

References

Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in neural information processing systems*, 35:11423–11436, 2022. 1, 3.1, 3.1

Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018. URL https://jmlr.org/papers/v18/17-468.html. 2, 2

Andreas Burger, Luca Thiede, Nikolaj Rønne, Varinia Bernales, Nandita Vijaykumar, Tejs Vegge, Arghya Bhowmik, and Alan Aspuru-Guzik. Hip: Hessian interatomic potentials without derivatives. *arXiv preprint arXiv:2509.21624*, 2025. URL https://arxiv.org/abs/2509.21624. 1

Xuhao Chen, Pingfan Li, Jianbin Fang, Tao Tang, Zhiying Wang, and Canqun Yang. Efficient and high-quality sparse graph coloring on gpus. *Concurrency and Computation: Practice and Experience*, 29(10):e4064, 2017. 5

Thomas F. Coleman and J. Y. Cai. The cyclic coloring problem and estimation of sparse hessian matrices. *SIAM Journal on Algebraic and Discrete Methods*, 7(2):221–235, 1986. doi: 10.1137/0607026. URL https://doi.org/10.1137/0607026. 1

Thomas F. Coleman and Jorge J. Moré. Estimation of sparse hessian matrices and graph coloring problems. *Mathematical Programming*, 28(3):243–270, 1984. doi: 10.1007/BF02612334. URL https://doi.org/10.1007/BF02612334. 1, 2, 3.1, 3.1

A. R. Curtis, M. J. D. Powell, and J. K. Reid. On the estimation of sparse jacobian matrices. *IMA Journal of Applied Mathematics*, 13(1):117–119, 1974. doi: 10.1093/imamat/13.1.117. URL https://doi.org/10.1093/imamat/13.1.117. 1

Bowen Deng, Peichen Zhong, KyuJung Jun, Janosh Riebesell, Kevin Han, Christopher J. Bartel, and Gerbrand Ceder. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, 9 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00716-3. URL https://doi.org/10.1038/s42256-023-00716-3. 4.2

Jens Deussen and Uwe Naumann. Efficient computation of sparse higher derivative tensors. In *International Conference on Computational Science*, pages 3–17. Springer, 2019. 5

Carl Eckart. Some studies concerning rotating axes and polyatomic molecules. *Physical Review*, 47 (7):552, 1935. 4.2

Xiang Fu, Brandon M Wood, Luis Barroso-Luque, Daniel S Levine, Meng Gao, Misko Dzamba, and C Lawrence Zitnick. Learning smooth and expressive interatomic potentials for physical property prediction. In *Forty-second International Conference on Machine Learning*, 2025. 1

Assefaw H. Gebremedhin, Arijit Tarafdar, Fredrik Manne, and Alex Pothen. New acyclic and star coloring algorithms with application to computing hessians. *SIAM Journal on Scientific Computing*, 29(3):1042–1072, 2007. doi: 10.1137/050639879. URL https://epubs.siam.org/doi/10.1137/050639879. 3.2

Assefaw H Gebremedhin, Arijit Tarafdar, Alex Pothen, and Andrea Walther. Efficient computation of sparse hessians using coloring and automatic differentiation. *INFORMS Journal on Computing*, 21 (2):209–223, 2009. 2, 3.1, 3.1, 3.1, 3.2, 3.2

Assefaw H. Gebremedhin, Duc Nguyen, Md. Mostofa Ali Patwary, and Alex Pothen. Colpack: Software for graph coloring and related problems in scientific computing. *ACM Trans. Math. Softw.*, 40(1), October 2013. ISSN 0098-3500. doi: 10.1145/2513109.2513110. URL https://doi.org/10.1145/2513109.2513110. 2, 3.4, A.3

Assefaw Hadish Gebremedhin, Fredrik Manne, and Alex Pothen. What color is your jacobian? graph coloring for computing derivatives. *SIAM Review*, 47(4):629–705, 2005. doi: 10.1137/S0036144504444711. URL https://epubs.siam.org/doi/10.1137/S0036144504444711. 1, 2, 3.1, 3.2, 3.2, 3.3

An Ghysels, H Lee Woodcock III, Joseph D Larkin, Benjamin T Miller, Yihan Shao, Jing Kong, Dimitri Van Neck, Veronique Van Speybroeck, Michel Waroquier, and Bernard R Brooks. Efficient calculation of qm/mm frequencies with the mobile block hessian. *Journal of Chemical Theory and Computation*, 7(2):496–514, 2011. 1

A. Gijón and E. R. Hernández. Quantum simulations of neutral water clusters and singly-charged water cluster anions. *Phys. Chem. Chem. Phys.*, 24:14440–14451, 2022. doi: 10.1039/D2CP01088G. URL http://dx.doi.org/10.1039/D2CP01088G. 4.1

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. Pmlr, 2017. 3.1, 3.1

Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008. 2, 2

Stefan Grimme, Christoph Bannwarth, and Philip Shushkov. A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (z= 1–86). *Journal of chemical theory and computation*, 13(5):1989–2009, 2017. 3.1

Adrian Hill and Guillaume Dalle. asdex: Automatic sparse differentiation in jax, 2026. URL https://arxiv.org/abs/2610.12336. 3.4

Adrian Hill, Guillaume Dalle, and Alexis Montoison. An illustrated guide to automatic sparse differentiation. In *ICLR Blogposts 2025*, 2025. 1

Silvan Kaser, Eric D Boittier, Meenu Upadhyay, and Markus Meuwly. Transfer learning to ccsd (t): Accurate anharmonic frequencies from machine learning models. *Journal of Chemical Theory and Computation*, 17(6):3687–3699, 2021. 5

Tsz Wai Ko, Bowen Deng, Marcel Nassar, Luis Barroso-Luque, Runze Liu, Ji Qi, Atul C. Thakur, Adesh Rohan Mishra, Elliott Liu, Gerbrand Ceder, Santiago Miret, and Shyue Ping Ong. Materials graph library (matgl), an open-source graph deep learning library for materials science and chemistry. *npj Computational Materials*, 11:253, 2025. doi: 10.1038/s41524-025-01742-y. URL https://doi.org/10.1038/s41524-025-01742-y. 1, 4, A.3

Teddy Koker, Abhijeet Gangan, Mit Kotak, Jaime Marian, and Tess Smidt. Pft: Phonon fine-tuning for machine learned interatomic potentials. *arXiv preprint arXiv:2601.07742*, 2026. 5

Dávid Péter Kovács, J Harry Moore, Nicholas J Browning, Ilyes Batatia, Joshua T Horton, Yixuan Pu, Venkat Kapil, William C Witt, Ioan-Bogdan Magdău, Daniel J Cole, et al. Mace-off: Transferable short range machine learning force fields for organic molecules. *arXiv preprint arXiv:2312.15211*, 2023. 4.1

Jörg Kussmann, Arne Luenser, Matthias Beer, and Christian Ochsenfeld. A reduced-scaling density matrix-based method for the computation of the vibrational hessian matrix at the self-consistent field level. *The Journal of Chemical Physics*, 142:094101, 2015. doi: 10.1063/1.4908131. URL https://doi.org/10.1063/1.4908131. 1

Marcel F Langer, Adrian Hill, and Michele Ceriotti. Truncated automatic sparse differentiation for machine learning interatomic potentials. *arXiv preprint arXiv:2609.20510*, 2026. 3.4

Tamas Lazar, Elizabeth Martínez-Pérez, Federica Quaglia, András Hatos, Lucía B Chemes, Javier A Iserte, Nicolás A Méndez, Nicolás A Garrone, Tadeo E Saldaño, Julia Marchetti, et al. Ped in 2021: a major update of the protein ensemble database for intrinsically disordered proteins. *Nucleic acids research*, 49(D1):D404–D411, 2021. 4.1

Daniel S Levine, Muhammed Shuaibi, Evan Walter Clark Spotte-Smith, Michael G Taylor, Muhammad R Hasyim, Kyle Michel, Ilyes Batatia, Gábor Csányi, Misko Dzamba, Peter Eastman, et al. The open molecules 2025 (omol25) dataset, evaluations, and models. *arXiv preprint arXiv:2505.08762*, 2025. 1

Xianglin Liu, Kai Yang, Fanli Zhou, Yongxiang Liu, Hao Chen, Yijia Zhang, Dengdong Fan, Wenbo Li, Bingqiang Wang, Shixun Zhang, et al. Smc-ai: Scaling monte carlo simulation to four trillion atoms with ai accelerators. *arXiv preprint arXiv:2604.08250*, 2026. 4.1

Alexis Montoison, Guillaume Dalle, and Assefaw Gebremedhin. Revisiting sparse matrix coloring and bicoloring, May 2025. URL https://arxiv.org/abs/2505.07308. 3.2

Hiroya Nakata and Dmitri G. Fedorov. Development of the analytic second derivatives for the fragment molecular orbital method. In *Recent Advances of the Fragment Molecular Orbital Method*, pages 425–458. Springer Nature Singapore, 2021. doi: 10.1007/978-981-15-9235-5_22. URL https://doi.org/10.1007/978-981-15-9235-5_22. 1

F. Neese. The orca program system. *WIRES Comput. Molec. Sci.*, 2(1):73–78, 2012. doi: 10.1002/wcms.81. 4.2, A.3

Garry N. Newsam and John D. Ramsdell. Estimation of sparse jacobian matrices. *SIAM Journal on Algebraic and Discrete Methods*, 4(3):404–418, 1983. doi: 10.1137/0604041. URL https://doi.org/10.1137/0604041. 1

Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6(1):147–160, 1994. doi: 10.1162/neco.1994.6.1.147. URL https://doi.org/10.1162/neco.1994.6.1.147. 2, 2, 2

M. J. D. Powell and Ph. L. Toint. On the estimation of sparse hessian matrices. *SIAM Journal on Numerical Analysis*, 16(6):1060–1074, 1979. doi: 10.1137/0716078. URL https://epubs.siam.org/doi/10.1137/0716078. 1

Protein Ensemble Database. Structural ensemble of the amyloid beta (aβ40) protein [c36m (reweighted)]. https://proteinensemble.org/entries/PED00531. Accessed: 2026-04-10. 4.1

Austin Rodriguez, Justin S Smith, Sakib Matin, Nicholas Lubbers, Kipton Barros, and Jose L Mendoza-Cortes. Projected hessian learning: Fast curvature supervision for accurate machine-learning interatomic potentials. *arXiv preprint arXiv:2603.04523*, 2026. 5

Jacob N. Sanders, Xavier Andrade, and Alán Aspuru-Guzik. Compressed sensing for the fast computation of matrices: Application to molecular vibrations. *ACS Central Science*, 1(1):24–32, 2015. doi: 10.1021/oc5000404. URL https://doi.org/10.1021/oc5000404. 1

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017. 1, 3.1, 3.1

Guillem Simeon and Gianni De Fabritiis. Tensornet: Cartesian tensor representations for efficient learning of molecular potentials. In *Advances in Neural Information Processing Systems 36*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/75c2ec5f98d7b2f50ad68033d2c07086-Abstract-Conference.html. 1, 4.1

Keke Song, Rui Zhao, Jiahui Liu, Yanzhou Wang, Eric Lindgren, Yong Wang, Shunda Chen, Ke Xu, Ting Liang, Penghua Ying, et al. General-purpose machine-learned potential for 16 elemental metals and their alloys. *Nature Communications*, 15(1):10208, 2024. 4.1

Tong Wang, Xinheng He, Mingyu Li, Bin Shao, and Tie-Yan Liu. AIMD-Chig: exploring the conformational space of 166-atom protein Chignolin with ab initio molecular dynamics. 5 2023. doi: 10.6084/m9.figshare.22786730.v4. URL https://figshare.com/articles/dataset/_strong_AIMD-Chig_exploring_the_conformational_space_of_166-atom_protein_strong_em_strong_Chignolin_strong_em_strong_with_strong_em_strong_ab_initio_strong_em_strong_molecular_dynamics_strong_/22786730. 4.2

Xun Wang, Xiangyu Meng, Zhuoqiang Guo, Mingzhen Li, Lijun Liu, Mingfan Li, Qian Xiao, Tong Zhao, Ninghui Sun, Guangming Tan, et al. 29-billion atoms molecular dynamics simulation with ab initio accuracy on 35 million cores of new sunway supercomputer. *IEEE Transactions on Computers*, 2025. 4.1

Jonathan T Willman, Joseph M Gonzalez, Kien Nguyen-Cong, Sebastien Hamel, Vincenzo Lordi, and Ivan I Oleynik. Accuracy, transferability, and computational efficiency of interatomic potentials for simulations of carbon under extreme conditions. *The Journal of chemical physics*, 161(8), 2024. 4.1

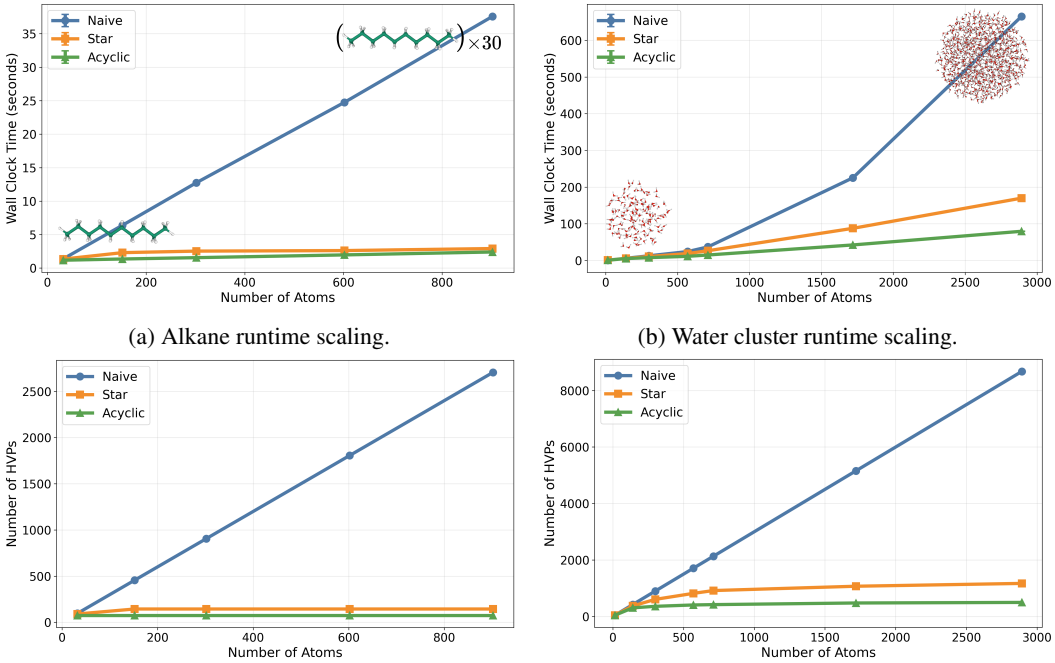
Xingyu Yang, Haitao Ma, Qing Lu, and Wensheng Bian. Efficient method for numerical calculations of molecular vibrational frequencies by exploiting sparseness of hessian matrix. *The Journal of Physical Chemistry A*, 128:3024–3032, 2024. doi: 10.1021/acs.jpca.3c07645. URL https://doi.org/10.1021/acs.jpca.3c07645. 1

Roman Zubatyuk, Justin S Smith, Jerzy Leszczynski, and Olexandr Isayev. Accurate and transferable multitask prediction of chemical properties with an atoms-in-molecules neural network. *Science advances*, 5(8):eaav6490, 2019. 1

A Additional results and details

We provide additional results with the 2 layer TensorNet architecture on the Alkane Chains, Water Cluster and Aβ40 protein. We can see that similar to the 1 layer model, we achieve linear scaling, although somewhat later as the two layer model covers more of the systems and therefore has less sparsity to exploit.

A.1 Alkanes and Water Cluster



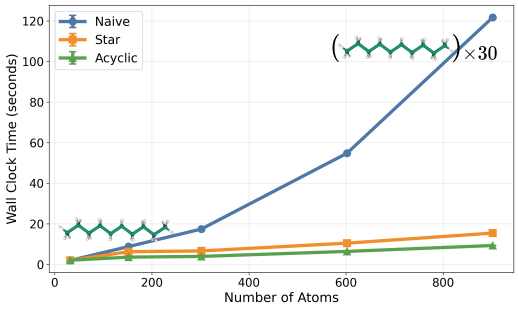
(a) Alkane runtime scaling.

(b) Water cluster runtime scaling.

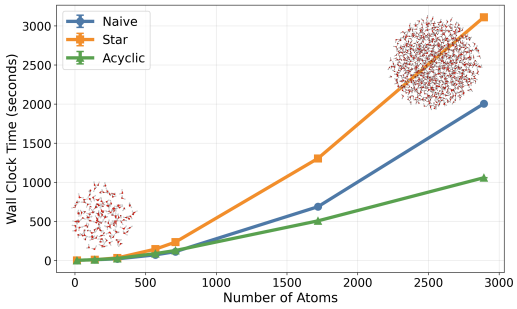
(c) Alkane HVP count scaling.

(d) Water cluster HVP count scaling.

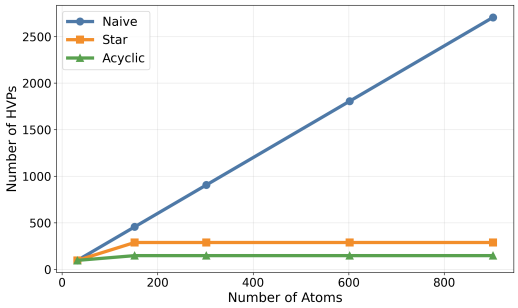
Figure 6: The runtime and HVP count of a 1 Layer TensorNet for increasingly larger alkane chains and water clusters.



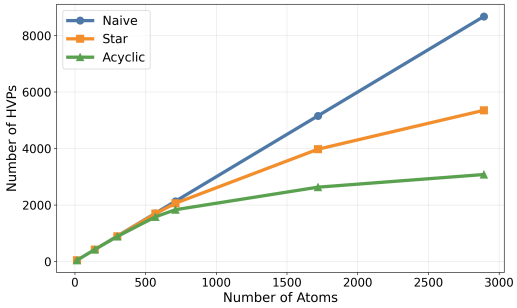
(a) Alkane runtime scaling.



(b) Water cluster runtime scaling.



(c) Alkane HVP count scaling.



(d) Water cluster HVP count scaling.

Figure 7: The runtime and HVP count of a 2 Layer TensorNet for increasingly larger alkane chains and water clusters.

A.2 Protein Conformer

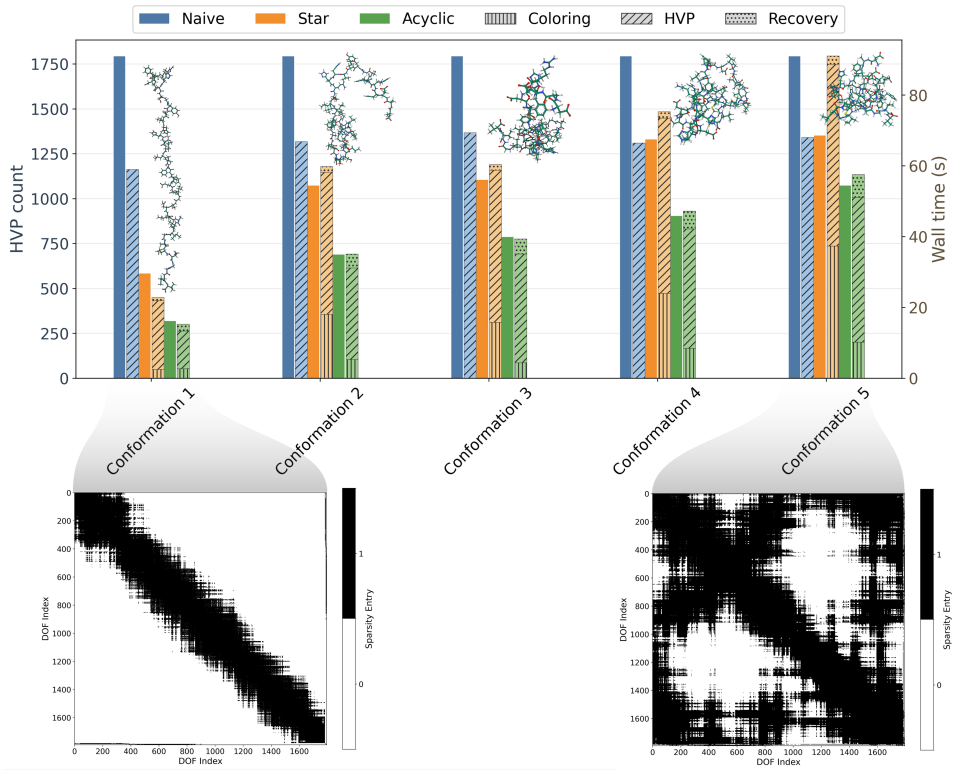


Figure 8: Conformation-dependent sparse Hessian performance for five Aβ40 structures. Variation across conformations arises from differences in the induced graph sparsity pattern.

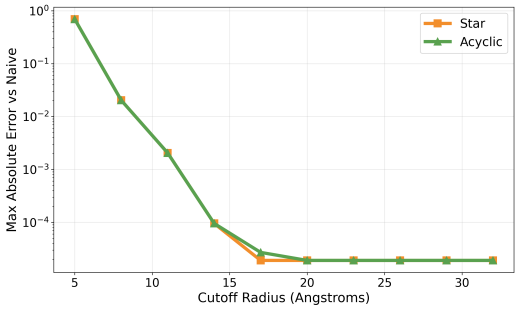
A.3 Cutoff experiments

To use our sparse differentiation in conjunction with DFT and non-local MLIP Hessians, we have to use an approximate sparsity pattern based on atom pair distance cutoffs. For our DFT experiments, we are generating the Hessian with ORCA Neese [2012]. We use the PBE/sto-3g level of theory with the tight convergence criterion. Since the Chignolin geometry is far from equilibrium, convergence of the SCF is tricky, requiring second-order SCF and multiple restarts even in the sto-3g basis set. We used ORCA's tight convergence settings with a total compute time of 3 hours. Since ORCA is closed source and doesn't expose HVPs in their API, we emulate them using the full Hessian. In principle, CPKS allows for HVPs, though. Alternatively, we can implement HVPs using the finite difference approximation 2.

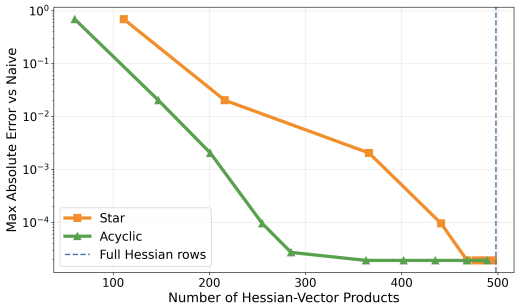
In addition to the average relative eigenvalue spectrum error from figure 4a - 5b we show maximal Hessian errors $\max_{ij}|H_{ij}^{\text{sparse}} - H_{ij}^{\text{naive}}|$ in figure 9a to 9d.

Supporting Information

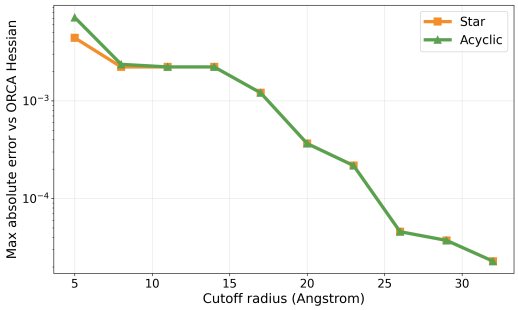
The code is available at `https://github.com/the-matter-lab/sparse_hessian_ff`. The code includes our Python bindings for the ColPack C++ library Gebremedhin et al. [2013] and the integration with the Materials Graph Library (MatGL) Ko et al. [2025], as well as the xyz files for the benchmarked systems.



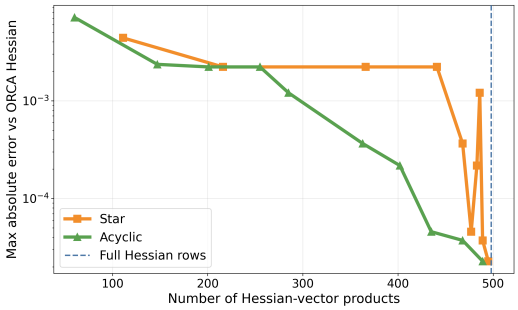
(a) Maximum error vs cutoff radius on Chignolin with the large receptive field CHGNet model.



(b) Maximum error vs HVP count on Chignolin with the large receptive field CHGNet model.



(c) Maximum error vs cutoff radius on Chignolin with DFT Hessians.



(d) Maximum error vs HVP count on Chignolin with DFT Hessians.

Figure 9: The maximum Hessian entry error of the CHGNet model (top) and DFT (bottom) resulting from using an approximate sparsity pattern.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We show linear scaling Hessian on several systems. The Hessian are exact up to floating point errors

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the need for a local receptive field, and that non local modules break this assumption and require heuristic cutoff sparsity pattern

   Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Our core theoretical claim about the sparsity pattern is derived and mathmetically justified.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All steps are clearly layed out and use standard algorithms

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide a zip with all the code to run the experiments. The exception is the DFT cutoff experiments as it requires the commercial ORCA software package. DFT settings are provided for the DFT calculations though.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: No training is performed as we use pretrained checkpoints, however configurations of all MLIPs and DFT calculations are stated

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: The algorithm is deterministic, therefore no error bars are reported

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Timings are provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Faster Hessian calculations does not pose any immediate risks

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Paper does not pose such risks

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets, ColPack, MatGL and datasets are cited

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: ColPackPy library snippet documents its usage

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: No LLMs involved

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.