Modified framework for deep learning representation of density functional theory Hamiltonian

Yashas S. B.
2022113001
IIIT Hyderabad
yashas.b@research.iiit.ac.in

Shreyas Sinha
2021113010
IIIT Hyderabad
shreyas.sinha@research.iiit.ac.in

Abstract

The combination of deep learning and ab initio calculation has shown great promise in revolutionizing future scientific research, but how to design neural network models incorporating a priori knowledge and symmetry requirements is a key challenging subject. In our project we modify an already proposed deep-learning framework to represent density functional theory (DFT) Hamiltonian as a function of material structure. We try to improve upon the already submitted DEEPH model by using physics informed input representations and by adding components to the model that improves its performance. We report a 10x improvement in the loss, compared to the DEEPH baseline in our results, thus verifying the success of our project.

1 Introduction

The accurate prediction of electronic properties in materials science is pivotal for the design and discovery of novel materials. Density Functional Theory (DFT) has long been the cornerstone for such predictions, providing a quantum mechanical description of the electronic structure of many-body systems. However, the computational intensity of DFT, especially for large or complex systems, poses significant challenges. To address this, recent advancements have explored the integration of deep learning techniques with DFT, aiming to retain accuracy while enhancing computational efficiency.

One such advancement is the development of DeepH, a deep-learning framework designed to predict the DFT Hamiltonian of crystalline materials. By leveraging message-passing neural networks (MPNNs), DeepH captures the local chemical environment of atoms, enabling the prediction of Hamiltonian matrices without the need for self-consistent field iterations inherent in traditional DFT calculations. This approach not only accelerates computations but also extends the applicability of DFT to larger and more diverse material systems .

In our work, we focus on enhancing the DeepH framework by incorporating the AGNI fingerprint representation as input features. The AGNI fingerprints effectively encode the local atomic environments, capturing both geometric and chemical information, which is crucial for accurate Hamiltonian predictions. Furthermore, we integrate attention mechanisms into the MPNN architecture, employing both global self-attention and attention over message parsing. These mechanisms allow the model to weigh the significance of different atomic interactions dynamically, leading to improved accuracy in capturing complex electronic behaviors.

The integration of AGNI fingerprints and attention mechanisms within the DeepH framework represents a significant step toward more efficient and accurate electronic structure predictions. By reducing computational costs and maintaining high fidelity in predictions, this approach holds promise for accelerating materials discovery and expanding the horizons of computational materials science.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1.1 The DeepH network

DeepH was introduced by Li et al in [1]. It is a Message Parsing Graph Neural Network, which stores atoms embedded as vertices. The more interesting part is how the edges are stored. The edges are initially embedded as the interatomic distance expanded with the Gaussian basis, centered at different points. These edges are made only for atom pairs which are closer than a certain cutoff distance.

Two neural networks are made. One updates the atom state using the previous state of itself, and all of the other atoms connected to it along with the edges connecting them. The other updates the edges using the current state of the edge and the state of the two atoms/vertices it connects. The two neural nets implemented are both fairly simple feedforward neural networks.

The Message Passing Neural Network performs the update in 5+1 layers. There are 5 layers which use the MP Layer neural network path, and 1 layer which used the LCMP Layer neural network path. The LCMP layer exists as a result of introducing equivariance into the network. Each layer acts as a time step in spreading the information from a nearby node. More details can be found in the paper [1].

- **Input:** The input to the DeepH model is the Cartesian coordinates of the system. These coordinates are then converted to a graph. Each node of the gtaph corresponds to an atom in the system. The atom's element type is encoded as the node.
- **Output:** In the last stage, the edges are converted to the required Hamiltonian Matrix for each edge. The Hamiltonian Matrix (in the Tight Binding format) is represented for each edge (where an edge is made only for pairs that are within a cutoff distance), and for each edge it is a 13×13 matrix (since the authors use 13 non orthogonal basis functions).

2 Dataset Description

The dataset utilized in this work is publicly available on Zenodo under DOI 10.5281/zenodo.6555484. It was published on May 17, 2022, (version v1) and last modified on March 10, 2023.

2.1 Content and Scope

This dataset comprises DFT-calculated atomic structures and corresponding Hamiltonian matrices for monolayer graphene. Various structures were generated through Molecular Dynamics methods using the VASP package. All DFT results were generated using the OpenMX package, ensuring consistency in basis-set and overlap matrix calculations. Further orbital information required for overlap computations can be found in the original Nature Computational Science article by Li *et al.* [1] (2022).

3 Methods

The DeepH implementation realizes the Deep-learning DFT Hamiltonian (DeepH) approach via an equivariant Message-Passing Neural Network (MPNN), exploiting locality and gauge covariance to predict Hamiltonian matrix elements directly from atomic structures. We made changes to the model by using a different form of input representation - AGNI fingerprints. AGNI fingerprints embed known interatomic-environment patterns (Gaussian expansions, cutoff functions, chemical weighting) so the model doesn't have to re-discover basic radial/chemical relationships from raw data. So, we expect that, with stronger initial features, you often need fewer training samples to reach a given accuracy. Also, we added made several types of attention-based changes to the architecture to try and improve its performance.

3.1 AGNI fingerprints

We compute atom-centered AGNI fingerprints using a custom Python script built on ASE's neighbor-list utilities and NumPy for efficient numerical operations. The overall workflow follows the procedure outlined in del Rio *et al.* [2] for constructing invariant fingerprints via Gaussian basis expansions and a smooth cutoff function.

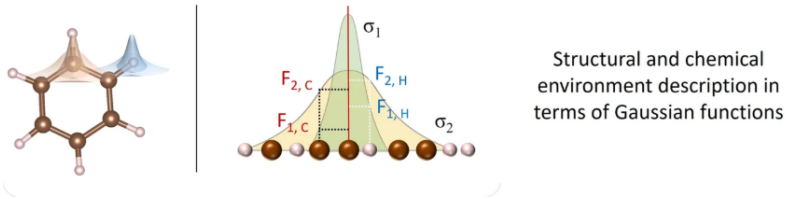


Figure 1: Fingerprinting

Hyperparameters We set a global cutoff of $d_c = 6.0$Å and Gaussian width $\sigma = 0.5$Å. The directional index $\alpha = 2$ (i.e. z–component) was chosen to generate vector-type features. We use $D = 8$ centers $\{a_k\}$ equally spaced in $[0, d_c]$. Two strategies are supported:

- **Weighted**: fingerprints of size $2D$, combining constant- and electronegativity-weighted sums.
- **Augmented**: fingerprints of size $D \times N_{\text{types}}$, summing over each element type separately.

Weights are taken as Pauling electronegativities by default.

Neighbor List and Geometry For each structure, atomic symbols and positions are loaded via ASE's `Atoms` object. Periodic boundary conditions and a cutoff of d_c Å are passed to `ase.neighborlist.neighbor_list` to obtain arrays $(i, j, R_{ij}, \mathbf{r}_{ij})$ of neighbor pairs, distances, and displacement vectors.

Cutoff Function We employ a smooth cosine cutoff function:

$$f_c(R) = \begin{cases} \frac{1}{2}[\cos(\pi R/d_c) + 1] & \text{if } 0 \le R \le d_c \\ 0 & \text{if } R > d_c \end{cases} \tag{1}$$

which ensures that both the fingerprint values and their derivatives decay continuously to zero at the cutoff distance d_c.

Component Calculation For each neighbor j of atom i, the kth fingerprint component is computed as:

$$\phi^{\alpha}_{k,ij} = w_j \, \frac{r^{\alpha}_{ij}}{R_{ij}} \, \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{(R_{ij} - a_k)^2}{2\sigma^2}\right) \, f_c(R_{ij}), \tag{2}$$

where

- $R_{ij} = \|\mathbf{r}_j - \mathbf{r}_i\|$ is the interatomic distance,
- r^{α}_{ij} is the αth Cartesian component of the displacement $\mathbf{r}_j - \mathbf{r}_i$,
- σ is the Gaussian width,
- a_k is the kth center in the Gaussian expansion, and
- w_j is the weight for atom j (either unity or its Pauling electronegativity).

Summing $\phi^{\alpha}_{k,ij}$ over all neighbors j yields the kth descriptor value for atom i.

This implementation yields a rotation-, translation-, and permutation-invariant representation of local atomic environments, suitable as input features to our modified DeepH-pack graph neural network.

3.2 Transformer Encoder Integration

In addition to the message-passing architecture inherited from DeepH-pack, we incorporate a Transformer encoder to enhance long-range interactions among orbital features. After constructing and flattening the per-atom fingerprint-derived feature vectors (of length N_{orb}), we project them into a d_{model}-dimensional embedding space and process them through L stacked Transformer encoder layers.

Embedding and Model Dimension Let N_{orb} be the number of orbital-pair blocks to predict. We set the Transformer model dimension

$$d_{\text{model}} = \begin{cases} \texttt{transformer\_dim}, & \text{if provided,} \\ N_{\text{orb}}, & \text{otherwise.} \end{cases}$$

An initial linear layer maps the flattened fingerprint vector $\mathbf{x} \in \mathbb{R}^{N_{\text{orb}}}$ to $\mathbb{R}^{d_{\text{model}}}$.

Encoder Layer Configuration Each Transformer encoder layer uses the following hyperparameters:

- **Number of heads** (`nhead`): `transformer_heads`,
- **Feed-forward dimension**: $4 \times d_{\text{model}}$,
- **Dropout rate**: 0.1,
- **Activation function**: ReLU.

Formally, an encoder layer computes:

$$\mathbf{z}' = \text{LayerNorm}\big(\mathbf{z} + \text{MultiHeadSelfAttention}(\mathbf{z})\big), \quad \mathbf{z}'' = \text{LayerNorm}\big(\mathbf{z}' + \text{FFN}(\mathbf{z}')\big),$$

where $\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2$ with $W_1, W_2 \in \mathbb{R}^{d_{\text{model}} \times 4d_{\text{model}}}$.

Stacking and Output We stack $L = \texttt{transformer\_layers}$ identical encoder layers to form the Transformer block:

$$\text{Transformer}(\mathbf{X}) = \underbrace{\text{EncLayer}\big(\dots \text{EncLayer}(\mathbf{X}) \dots\big)}_{L \text{ times}}.$$

The final output sequence is average-pooled across the sequence dimension and passed to a linear readout head to produce the predicted Hamiltonian elements.

This Transformer integration enables the model to capture non-local correlations among orbital features beyond the fixed-radius graph, improving predictive performance on long-range electronic interactions.

3.3 Attention over Message-Passing steps

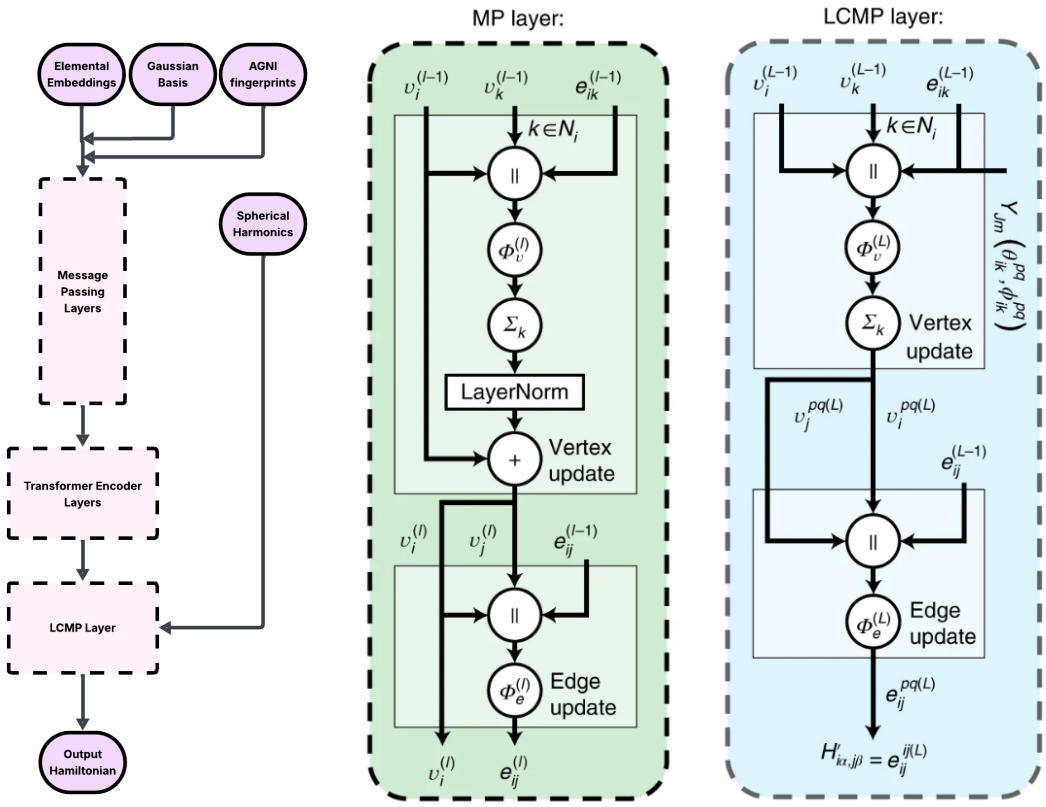
"Attention over message parsing steps" refers to the incorporation of attention mechanisms that dynamically weigh the contributions of messages received at different propagation steps. Instead of treating each message passing iteration equally, the model learns to assign varying levels of importance to the information aggregated at each step. This approach enables the network to focus on the most relevant information across different stages of message propagation, enhancing its ability to capture complex dependencies and patterns within the graph.

In the context of DeepH, applying attention over message parsing steps could allow the model to emphasize information from specific propagation iterations that are most pertinent to predicting the DFT Hamiltonian. We have applied Attention over Message Parsing on DeepH and noted an improvement in loss value.

3.4 Global Self-Attention

Global self-attention is a mechanism in GNNs that enables each node to attend to every other node in the graph, regardless of direct connectivity. This approach allows the model to capture long-range dependencies and global context, which are often challenging for traditional message-passing GNNs that primarily focus on local neighborhoods - like DeepH. Although "locality" in DeepH is a deliberate measure to force the principle of locality onto the system while also limitting the amount of calculation required,

Incorporating global self-attention into GNNs facilitates the modeling of complex interactions across the entire graph. In the context of DeepH, integrating global self-attention mechanisms enables the model to effectively capture intricate electronic interactions within a material's structure. By considering the entire graph, we hypothesize that DeepH could produce more accurate predictions of the DFT Hamiltonian, as it accounts for both local and global atomic environments.



(a) Overall model flow (b) Message-passing layers (c) LCMP layer

Figure 2: Architecture overview: (a) High-level model pipeline; (b) Graph-based message-passing network; (c) Final (LCMP) layer of the network. Without this layer, the network might not learn equivariant features very well.

4 Results

We run the model for 1800 epochs, as we notice that the loss for our baseline (ran using code from the DeepH codebase) stagnates at that point. For the baseline, DeepH code was run with default settings. We note that by default the DeepH code selects a GAT for the atom update network. The paper discusses a feed forward network. This might be because the codebase is constantly being updated by the researchers and new things keep coming up. We note however that we are not confident that are results should be compared to the results from the original paper. More work is required to accurately represent the paper's model. But we consider the results produced by the default settings to be an adequate baseline.

We find that the models performs better than the baseline using all the mechanisms. All four methods give improvements an order of magnitude better than the baseline. However, the best performance is given by Attention over Message Parsing. This is summarized in table 1.

We also analyzed the error per element of the predicted Hamiltonian Matrix. This showed results similar to that of the original paper [1]. We observe that the error peaks for the 4th and 5th (0 indexed) orbitals. The authors of the original paper do not go into much detail on why this could be the case, but we hypothesize that this might be because of the choice of basis functions used. The authors went with a non orthogonal Pseudo Atomic Orbital type basis set, with orbitals that were similar to 2 s orbitals, 2 p orbitals and 1 d orbital (a total of $1 + 1 + 3 + 3 + 5 = 13$ orbitals), giving this result. Since the dataset was made using these orbitals, we went with the same. However, we hypothesize that this might improve upon using a different basis set. Experiments using an orthogonal basis might also produce some interesting results.

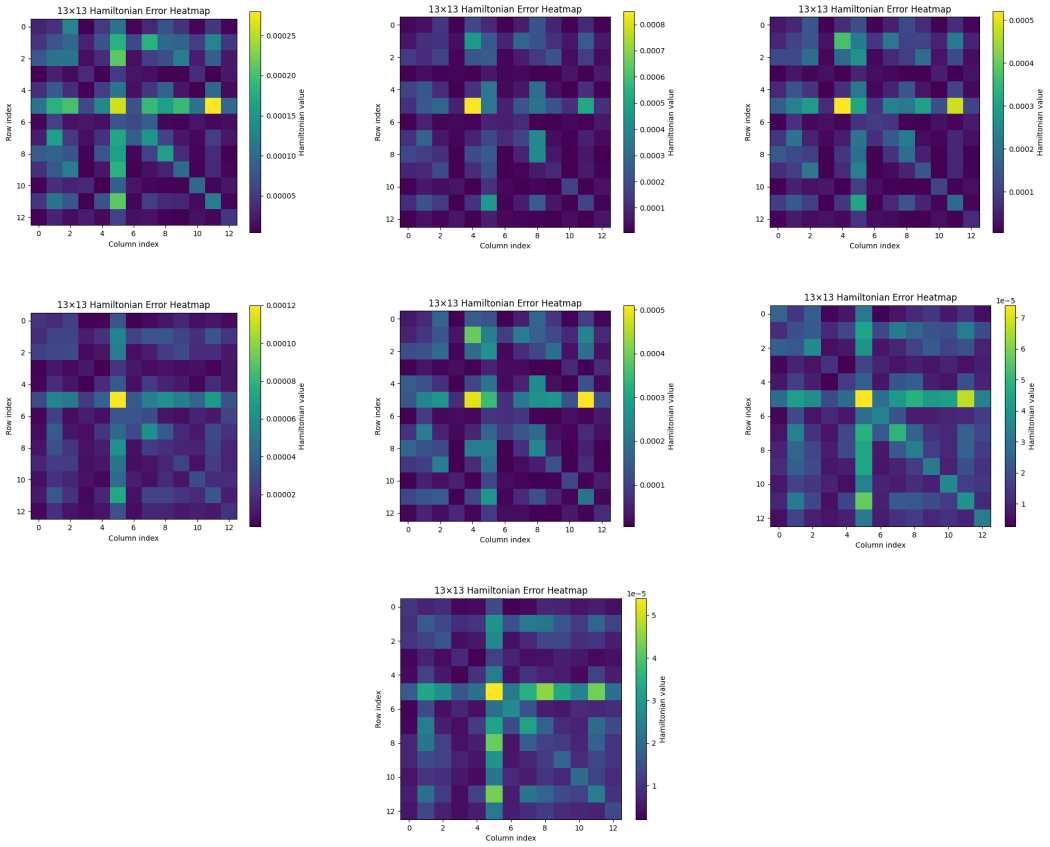


Figure 3: **(a–g)** Seven plots illustrating the Hamiltonian error.
(a–c) Only AGNI, Only Transformer & AGNI + Transformer models; **(d–f)** Only GSA, GSA + AGNI + Transformer & GSA + MPAtt models; **(g)** MPAttention model

4.1 Future work

There is a lot of scope for this project to be expanded. We expect interesting results from the following variations:

- A network with AGNI fingerprints as vertices. Since the fingerprints store much of the local environment, we expect this model to perform quite well despite the lack of edge parameters.
- Hyperparameter tuning could be done on Transformers to get better results.
- ChargE3 net (or other networks) could be used to augment the outputs of this network, as we were planning to do with the original scope of this project.

4.2 Note on change of scope of project

Initially the scope of was to forward the hamiltonian generated by DeepH to function that could get the energy, use this energy as a loss function and use the training process of the network as a surrogate for the DFT Self Consistent Field method. We planned to use AGNI fingerprints to try and enhance the ability of this method to predict the DFT calcuated energy. We also planned to compare this with using charge density instead of hamiltonians.

However, due to the complexities encountered in getting the overlap matrix of the basis orbitals used (which is necessary for getting the energy from the Hamiltonian), we decided to switch the scope of the project to improving the DeepH model using AGNI and attention.

Table 1: Losses (GSA - Global Self Attention, MPA - Attention over Message Parsing)

Model	Test Loss (MSE)
Baseline	1.8×10^{-4}
Transformer Encoder	8×10^{-5}
Using AGNI	5×10^{-5}
GSA	2×10^{-5}
GSA and MPA	1.8×10^{-5}
MPA	1.5×10^{-5}

5 Code Availability

Code available on github

References

[1] He Li et al. "Deep-learning density functional theory Hamiltonian for efficient ab initio electronic-structure calculation". In: *Nature Computational Science* 2 (2022), pp. 367–377. DOI: 10.1038/s43588-022-00265-6. URL: https://doi.org/10.1038/s43588-022-00265-6.

[2] B. G. del Rio, B. Phan, and R. Ramprasad. "A deep learning framework to emulate density functional theory". In: *npj Computational Materials* 9 (2023), p. 158. DOI: 10.1038/s41524-023-01115-3. URL: https://doi.org/10.1038/s41524-023-01115-3.