PROPERTY PREDICTION OF STACKED BILAYER MATERIALS: A MULTIMODAL LEARNING APPROACH

An Vuong*, **Minh-Hao Van***, **Chen Zhao**†, **Xintao Wu***
***University of Arkansas** †**Baylor University**

ABSTRACT

AI for materials science is a critical topic within AI for science, aiming to accelerate materials discovery and produce accurate property predictions. Bilayer 2D material stacking is essential for exploring new materials with novel functions and inherent phenomena, enabling the creation of new 2D bilayers for diverse real-world applications. Research on bilayer vdWs materials has made significant progress from experimental and computational perspectives. Various bilayer materials have been successfully synthesized experimentally and the increasing utilization of high-throughput computing technology has constructed several computational two-dimensional materials databases. However, the use of AI to model bilayer stacking and predict new properties remains underexplored, necessitating further research studies. In this work, we propose a novel multimodal learning approach to study the interfaces between dissimilar materials that jointly enable new or multiple functions, and to predict new properties arising from the vertical integration (stacking) of different functional material layers under given configurations. Comprehensive experiments demonstrate the effectiveness and efficiency of our approach compared to baseline methods. Our code is available at `https://tinyurl.com/bimat-ml-code`.

1 INTRODUCTION

Bilayer materials are two-dimensional structures composed of two individual layers of 2D materials that are stacked on top of each other. The layers are held together by weak van der Waals forces, but the way they are stacked can be manipulated, a process that allows for fine-tuning of the material's properties. Stacking layers can create new electronic, optical, and mechanical properties that are not present in the single-layer (monolayer) versions of the materials. This is especially true when the layers are twisted relative to each other, creating a moiré superlattice. For example, twisted bilayers (TBL) of 2D materials have emerged as a versatile platform for novel interface phenomena (Li et al., 2010). The two layers that form these twisted structures when isolated do not exhibit ferroelectricity, indicating that the effect arises from interlayer coupling, charge redistribution, and moiré-induced asymmetry rather than lattice displacements.

There has been increasing necessity of considering various stacking modes in 2D vdWs bilayer structures, including patterns and sequences, which significantly affect the material properties. A key task is to predict the properties of stacked bilayer materials from their stacking configuration. Computational simulation such as density functional theory (DFT) has been a popular method for calculating the intrinsic properties of materials on the basis of electron density. When stacking 2D materials, the high degree of freedom of stacking results in a large material space. The high computational cost of DFT makes it difficult to study stacked 2D materials. Graph neural networks (GNNs) have been introduced into materials science for processing molecular/crystal data with non-Euclidean structures and many works (Xie & Grossman, 2018b; Choudhary & DeCost, 2021) demonstrate high performance because they effectively represent and capture graph structures from crystallographic information files (CIFs) of materials. However, for stacked two dimensional materials, inter-layer atomic interactions are bound by weak van der Waals forces and intra-layer atomic interactions are bound by chemical bounds. Directly applying GNN models on the stacked materials cannot differentiate between intralayer and interlayer interactions, thus being unable to achieve accurate property predictions of the properties of stacked 2D materials.

In this paper, we propose a unified multimodal learning framework for modeling bilayer materials, namely BiMat-ML, that jointly models monolayer material structures, stacking configurations, and intrinsic monolayer properties to predict target properties of stacked bilayer materials. By explicitly integrating information across these complementary modalities, our approach captures both intra-layer chemistry and inter-layer stacking effects within a single predictive model. Our proposed framework is model-agnostic, applicable to diverse stacking configurations of both homobilayer and heterobilayer settings. Experimental results demonstrate that our method achieves prediction accuracy comparable to density functional theory (DFT) calculations while providing orders-of-magnitude improvements in computational efficiency. These results highlight the potential of multimodal learning as a general and scalable paradigm for rapid screening and inverse design of stacked 2D materials.

2 Related Work

Bilayer materials are atomically thin structures consisting of two stacked layers of 2D materials, such as graphene or transition metal dichalcogenides, held together by van der Waals (vdWs) forces. These stacked layers have unique electronic, optical, and mechanical properties that can be tuned by controlling the stacking pattern between the layers. The different stacking patterns can be created through rotations around the vertical axis or horizontal layer sliding (Wang et al., 2021). Research on bilayer vdWs materials has made significant progress from experimental and computational perspectives. Various bilayer materials have been successfully synthesized experimentally and the increasing utilization of high-throughput computing technology has constructed several computational two-dimensional materials databases such as C2DB (Gjerding et al., 2021), MC2D (Mounet et al., 2018), 2DMatPedia (Zhou et al., 2019), BMDS (Barik & Woods, 2023), BiDB (Pakdel et al., 2024), and HetDB (Sauer et al., 2025). These databases provide a rich theoretical foundation for materials research and significantly accelerate the research and development of 2D materials. To simply the high-throughtput computational process, Zhang et al. (2025) developed a specialized Python package, PyHTStack2D, designed for the efficient High-Throughput Stacking of 2D materials, including both homo- and hetero- structures. The package assists in generating input files and shell scripts for batch computation submissions to the Vienna Ab initio Simulation Package (VASP).

Advanced AI models are catalyzing a transformative shift in materials science. Many AI algorithms have been developed to support various discovery tasks such as atomistic simulation, property prediction, materials structure design and discovery, process planning and optimization (Van et al., 2025; Merchant et al., 2023). Unlike 3D materials, the AI development for bilayer 2D materials received very little study. Chen et al. (2024) developed a structural embedding method for property prediction of stacked bilayer materials. The developed structure-embedded PAINN (SE-PAINN) independently considers intra-layer and inter-layer neighbors when determining the connectivity between nodes (atoms) and uses two symmetric neural networks to handle intra-layer interactions and inter-layer interactions separately. They showed SE-PAINN can approximately reproduce the DFT calculation results of predicting the binding energy and band gap of twisted stacked 2D materials with significantly reduced computational cost. However, this method still requires the access of CIFs of stacked bilayer materials in both training and inference, thus being unable to support the bilayer property prediction task when CIFs of only monolayer materials are available.

3 Property Prediction of Stacked Bilayer Materials via Multimodal Learning

Bilayer materials are created by stacking two monolayers according to a specified configuration. In typical bilayer construction workflows, one monolayer is assigned as the bottom layer and the other as the top layer. Each monolayer is described by a crystallographic information file (CIF), which stores its crystal structure data. The objective of this work is to predict the target properties of the resulting bilayer material using the stacking configuration, the CIF files and the known properties of individual monolayers. In this section, we first present our framework BiMat-ML and then describe in detail each component including the graph encoder used for learning representation of each monolayer, and stacking configuration autoencoder used for learning configuration representation.

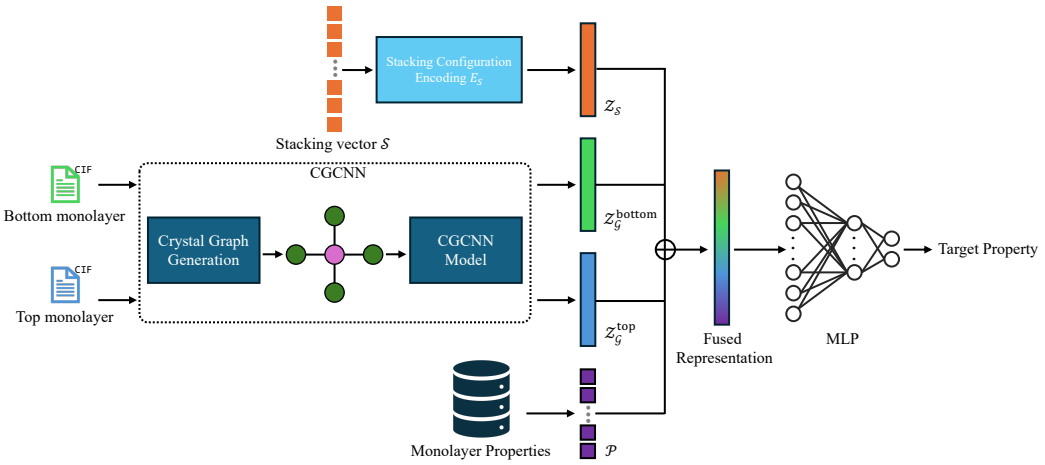

Figure 1: Architecture of property prediction of stacked bilayer material via multimodal learning (BiMat-ML)

3.1 Framework

We propose a unified multimodal learning framework that jointly models monolayer material structures, stacking configuration, and properties of monolayer materials, to predict target properties of bilayer materials. Formally, the bilayer material dataset is represented as $\mathcal{D} = \{(\mathcal{G}^{\text{bottom}}, \mathcal{P}^{\text{bottom}}, \mathcal{G}^{\text{top}}, \mathcal{P}^{\text{top}}, \mathcal{S}), \mathcal{Y}\}$, where $\mathcal{G}^{\text{bottom}}$ ($\mathcal{G}^{\text{top}}$) denotes the crystal graph structure of the bottom (top) monolayer, $\mathcal{P}^{\text{bottom}}$ ($\mathcal{P}^{\text{top}}$) represents known properties of the bottom (top) monolayer, $\mathcal{S}$ denotes the stacking configuration information, and $\mathcal{Y}$ is the target property of the bilayer material. The objective is to learn a mapping function

$$f_\theta : (\mathcal{G}^{\text{bottom}}, \mathcal{P}^{\text{bottom}}, \mathcal{G}^{\text{top}}, \mathcal{P}^{\text{top}}, \mathcal{S}) \rightarrow \mathcal{Y}, \tag{1}$$

where θ denotes the set of learnable parameters. Figure 1 illustrates the architecture of BiMat-ML that consists of three main components: (i) a graph encoder $E_\mathcal{G}$ that encodes monolayer crystal graphs into graph representations, (ii) a stacking configuration encoder $E_\mathcal{S}$ that encodes stacking configurations into latent representations, and (iii) a multimodal fusion module that integrates structural, stacking, and property information into a joint embedding for property prediction.

Algorithm 1 Property Prediction of Stacked Bilayer Material via Multimodal Learning (BiMat-ML)

Input: Bottom and top monolayer CIF $\mathcal{C}^{\text{bottom}}, \mathcal{C}^{\text{top}}$; stacking configurations $\mathcal{S}$; monolayer properties $\mathcal{P}^{\text{bottom}}, \mathcal{P}^{\text{top}}$

Output: Predicted bilayer property $\hat{\mathcal{Y}}$

$\mathcal{Z}_\mathcal{G}^{\text{bottom}} \leftarrow E_\mathcal{G}(\mathcal{C}^{\text{bottom}})$
$\mathcal{Z}_\mathcal{G}^{\text{top}} \leftarrow E_\mathcal{G}(\mathcal{C}^{\text{top}})$
$\mathcal{Z}_\mathcal{S} \leftarrow E_\mathcal{S}(\mathcal{S})$
$\mathcal{Z} \leftarrow \mathcal{Z}_\mathcal{G}^{\text{bottom}} \oplus \mathcal{P}^{\text{bottom}} \oplus \mathcal{Z}_\mathcal{G}^{\text{top}} \oplus \mathcal{P}^{\text{top}} \oplus \mathcal{Z}_\mathcal{S}$
$\hat{\mathcal{Y}} \leftarrow \text{MLP}(\mathcal{Z})$
$\mathcal{L} \leftarrow \|\hat{\mathcal{Y}} - \mathcal{Y}\|$
Update all parameters by minimizing $\mathcal{L}$
return $\hat{\mathcal{Y}}$

Algorithm 1 shows the pseudo code of BiMat-ML training algorithm. Specifically, we use CGCNN encoder (denoted as $E_\mathcal{G}$) to learn graph-level embeddings of each monolayer. $E_\mathcal{G}$ first constructs monolayer crystal graphs ($\mathcal{G}^{\text{bottom}}$,$\mathcal{G}^{\text{top}}$) from monolayer CIFs ($\mathcal{C}^{\text{bottom}}$, $\mathcal{C}^{\text{top}}$) and then maps crystal graphs to graph-level embeddings ($\mathcal{Z}_\mathcal{G}^{\text{bottom}}$, $\mathcal{Z}_\mathcal{G}^{\text{top}}$). We than use the stacking configuration encoder $E_\mathcal{S}$ to map stacking configurations $\mathcal{S}$ to stacking embeddings. These representations, together with the monolayer material properties $\mathcal{P}$, are subsequently fused to form a unified multimodal representation.

As the learned graph representations $\mathcal{Z}_{\mathcal{G}}^{\text{bottom}}, \mathcal{Z}_{\mathcal{G}}^{\text{top}} \in \mathbb{R}^{d_{\mathcal{G}}}$, stacking configuration representation $\mathcal{Z}_{\mathcal{S}} \in \mathbb{R}^{d_{\mathcal{S}}}$, and monolayer property $\mathcal{P}^{\text{bottom}}, \mathcal{P}^{\text{top}} \in \mathbb{R}^{d_{\mathcal{P}}}$, all modality-specific representations are concatenated to form a unified embedding

$$\mathcal{Z} = \mathcal{Z}_{\mathcal{G}}^{\text{bottom}} \oplus \mathcal{P}^{\text{bottom}} \oplus \mathcal{Z}_{\mathcal{G}}^{\text{top}} \oplus \mathcal{P}^{\text{top}} \oplus \mathcal{Z}_{\mathcal{S}} \ \in \ \mathbb{R}^{2(d_{\mathcal{G}}+d_{\mathcal{P}})+d_{\mathcal{S}}}.$$

The joint representation $\mathcal{Z}$ is subsequently passed through a multi-layer perceptron (MLP) to predict the target bilayer material property. The entire framework is trained end-to-end by minimizing the mean absolute error loss $\mathcal{L} = \|\hat{\mathcal{Y}} - \mathcal{Y}\|$.

3.2 Monolayer Representation via Graph Encoder

Algorithm 2 CGCNN-based Graph Encoding ($E_{\mathcal{G}}$)

Input: CIF file $\mathcal{C}$; number of graph convolution layers T
Output: Graph representation $\mathcal{Z}_{\mathcal{G}}$

Parse $\mathcal{C}$ to obtain atomic species and coordinates.
For each atom i, identify neighboring atoms within cutoff radius R and form a local edge set $\mathcal{E}_i$.
Construct crystal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.
Initialize node features $\{v_i^{(0)} \mid i \in \mathcal{V}\}$.
Initialize edge features $\{u_{(i,j)_k} \mid (i,j)_k \in \mathcal{E}_i\}$.
for $t = 0$ **to** $T-1$ **do**
  for each atom $i \in \mathcal{V}$ **do**
    $v_i^{(t+1)} \leftarrow v_i^{(t)}$
    for each $(i,j)_k \in \mathcal{E}_i$ **do**
      $z_{(i,j)_k}^{(t)} \leftarrow v_i^{(t)} \oplus v_j^{(t)} \oplus u_{(i,j)_k}$
      $v_i^{(t+1)} \leftarrow v_i^{(t+1)} + \sigma(z_{(i,j)_k}^{(t)} W_f^{(t)} + b_f^{(t)}) \odot g(z_{(i,j)_k}^{(t)} W_s^{(t)} + b_s^{(t)})$
    end for
  end for
end for
$\mathcal{Z}_{\mathcal{G}} \leftarrow \frac{1}{|\mathcal{V}|} \sum_i v_i^{(T)}$
return $\mathcal{Z}_{\mathcal{G}}$

Algorithm 2 describes steps to use CGCNN to map a crystal structure to a fixed-dimensional representation. The structure information in a CIF file $\mathcal{C}$ is first parsed to extract atomic species, lattice vectors, and atomic coordinates. Based on the atomic coordinates and periodic boundary conditions, interatomic distances are computed. For each atom i, neighboring atoms are identified based on interatomic distances within a cutoff radius R, and up to a maximum of N nearest neighbors are retained to form a local edge set $\mathcal{E}_i$. The crystal graph is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of atoms and $\mathcal{E} = \bigcup_{i=1}^{|\mathcal{V}|} \mathcal{E}_i$. Each atom $i \in \mathcal{V}$ is initialized with an elemental feature vector $v_i^{(0)}$, obtained by mapping its atomic identity to a fixed-length embedding following the CGCNN framework. Due to periodic boundary conditions, multiple edges $(i,j)_k$ may exist between the same atom pair, where $(i,j)_k$ denotes the k-th bond connection between atom i and atom j. For each edge $(i,j)_k$, the corresponding interatomic distance $d_{(i,j)_k}$ is encoded into a fixed-dimensional edge feature vector $u_{(i,j)_k}$ using a Gaussian basis expansion.

The graph encoder applies T graph convolution layers to iteratively update node representations. At convolution layer t, the feature of atom i is updated by aggregating information from its neighboring atoms j connected through edges $(i,j)_k \in \mathcal{E}_i$. For each neighbor interaction, the atom feature $v_i^{(t)}$, neighbor feature $v_j^{(t)}$, and edge feature $u_{(i,j)_k}$ are concatenated to form $z_{(i,j)_k}^{(t)} = v_i^{(t)} \oplus v_j^{(t)} \oplus u_{(i,j)_k}$. A gated convolution operation is then applied, where a sigmoid function $\sigma(\cdot)$ produces a learned gate and $g(\cdot)$ denotes a nonlinear activation function. The update rule is given by

$$v_i^{(t+1)} = v_i^{(t)} + \sum_{(i,j)_k \in \mathcal{E}_i} \sigma\left(z_{(i,j)_k}^{(t)} W_f^{(t)} + b_f^{(t)}\right) \odot g\left(z_{(i,j)_k}^{(t)} W_s^{(t)} + b_s^{(t)}\right),$$

where $W_f^{(t)}$, $W_s^{(t)}$ and $b_f^{(t)}$, $b_s^{(t)}$ are learnable weight matrices and bias vectors at layer t, and $\odot$ denotes element-wise multiplication. After T convolution layers, the final node representations $\{v_i^{(T)}\}_{i\in\mathcal{V}}$ encode the local atomic environments. A mean pooling operation over all atoms yields the graph-level representation

$$\mathcal{Z}_\mathcal{G} = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} v_i^{(T)},$$

where $\mathcal{Z}_\mathcal{G} \in \mathbb{R}^{d_\mathcal{G}}$ is the graph embedding and serves as a compact structural embedding of the crystal.

3.3 Stacking Configuration Representation via AutoEncoder

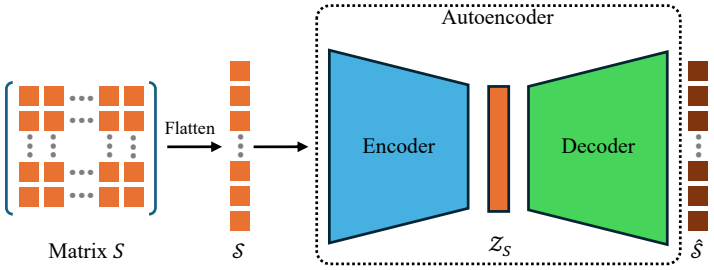

Figure 2: Architecture of extracting stacking configuration representation via Autoencoder ($E_\mathcal{S}$).

Stacking two-dimensional materials provides a powerful strategy for engineering novel properties beyond those of the individual layers. By vertically assembling atomically thin monolayers into bilayer heterostructures, we can tune electronic, optical, and mechanical behavior through interlayer coupling, relative orientation, and stacking order. Variations in stacking configuration, such as layer sequence, interlayer distance, and twist angle, can significantly affect the atomic coordinates and the in-plane lattice vectors. As a result, stacked 2D materials may display emergent phenomena not present in isolated monolayers. Stacking configurations is often represented as matrices and mapped to embeddings. Figure 2 illustrates the construction of stacking configuration embedding via AutoEncoder.

Algorithm 3 describes the stacking configuration encoder used to obtain a latent representation of a stacking configuration. Each stacking configuration is represented as a matrix $S \in \mathbb{R}^{n\times m}$ and is flattened into a vector $\mathcal{S} \in \mathbb{R}^{nm}$. The encoder maps the flattened stacking vector $\mathcal{S}$ into a latent space through a feed-forward network. Specifically, a linear transformation followed by a ReLU activation maps $\mathcal{S}$ to a hidden vector $h_1 \in \mathbb{R}^{d_S/2}$, which is then linearly projected to a latent stacking representation $\mathcal{Z}_S \in \mathbb{R}^{d_S}$. The dimensionality d_S is chosen to be consistent with the graph embedding dimension, enabling fusion with structural graph representations. A symmetric decoder maps the latent vector $\mathcal{Z}_S$ to a hidden vector $h_2 \in \mathbb{R}^{d_S/2}$ using a linear layer followed by a ReLU activation, and then linearly projects h_2 to reconstruct the stacking configuration vector $\hat{\mathcal{S}}$. The encoder–decoder network is trained by minimizing the reconstruction loss $\mathcal{L}_\mathcal{S} = \|\hat{\mathcal{S}} - \mathcal{S}\|^2$. After training, the latent vector $\mathcal{Z}_S$ is used as the stacking configuration representation for subsequent multimodal fusion.

4 Stacking Settings of Bilayer Materials

Bilayer materials can be broadly classified into homobilayers and heterobilayers based on the composition of their constituent layers. Homobilayer materials are formed by stacking two identical monolayers, where differences in physical properties arise primarily from variations in stacking order, relative translation, or twist angle between the layers. In contrast, heterobilayer materials consist of two distinct monolayers with different chemical compositions or crystal structures, enabling more diverse interlayer interactions and band alignments.

Algorithm 3 Stacking Configuration Encoding ($E_{\mathcal{S}}$)

Input: Stacking configuration matrix $S \in \mathbb{R}^{n\times m}$
Output: Latent stacking representation $\mathcal{Z}_S \in \mathbb{R}^{d_S}$
1: $\mathcal{S} \leftarrow \mathrm{Flatten}(S), \quad \mathcal{S} \in \mathbb{R}^{nm}$
2: $h_1 \leftarrow \mathrm{ReLU}(W_1\mathcal{S} + b_1), \quad h_1 \in \mathbb{R}^{d_S/2}$
3: $\mathcal{Z}_S \leftarrow W_2 h_1 + b_2, \quad \mathcal{Z}_S \in \mathbb{R}^{d_S}$
4: $h_2 \leftarrow \mathrm{ReLU}(W_3\mathcal{Z}_S + b_3), \quad h_2 \in \mathbb{R}^{d_S/2}$
5: $\hat{\mathcal{S}} \leftarrow W_4 h_2 + b_4, \quad \hat{\mathcal{S}} \in \mathbb{R}^{nm}$
6: $\mathcal{L}_{\mathcal{S}} \leftarrow \|\hat{\mathcal{S}} - \mathcal{S}\|^2$
7: Update parameters to minimize $\mathcal{L}_{\mathcal{S}}$
8: **return** $\mathcal{Z}_S$

4.1 HOMOBILAYER MATERIALS

Homogeneous bilayers keep one monolayer fixed as a reference. Different stacking configurations are obtained by applying a layer-specific transformation of the form $t \circ R \circ F$ to the other monolayer, where t denotes an in-plane translation, R denotes an in-plane rotation, F denotes an optional operation. Specifically, the stacking configuration can be described as a single affine transformation matrix

$$A = \begin{bmatrix} p_1 & p_3 & 0 \\ p_2 & p_4 & 0 \\ 0 & 0 & p_5 \\ p_6 & p_7 & \delta z \end{bmatrix},$$

where p_1–p_4 encode the in-plane rotation R, p_6–p_7 specify the in-plane translation t, and $p_5 \in \{+1, -1\}$ represents the optional flip transformation F. The out-of-plane shift δz is defined as the fractional displacement along the z axis that positions the transformed top monolayer relative to the fixed bottom monolayer, $\delta z = z^{\mathrm{top}} - p_5\, z^{\mathrm{bottom}}$, where z^{bottom} and z^{top} denote the fractional z coordinates of corresponding atoms in the bottom and top monolayers, respectively. In BiDB, the stacked bilayer structure is first optimized using a z-scan procedure to determine the optimal interlayer distance, after which the resulting $\delta z \in [0, 1]$ is computed from the optimized atomic coordinates. The value of δz varies across different bilayers but remains identical for all corresponding atom pairs within the same bilayer up to numerical precision. Since the optimized δz is only available after the z-scan procedure, it cannot be used as an input parameter when constructing stacked bilayers from monolayers. Therefore, in our experiments, the stacking configuration matrix A is constructed using the parameters p_1–p_7, with the out-of-plane shift δz fixed as a constant initial value.

The stacking transformation is applied by multiplying the homogeneous atomic coordinates $(x_i, y_i, z_i, 1)$ of each atom in the bottom reference monolayer with the affine transformation matrix A, yielding the corresponding atomic coordinates of the top monolayer. Periodic boundary conditions in the in-plane directions are enforced by wrapping the resulting fractional coordinates back into the reference unit cell, which is equivalent to applying a modulo-1 operation. Under the row-vector convention adopted in this work, the atomic coordinates of the bottom reference monolayer are collected row-wise in the matrix X^{bottom}, and the stacking configuration is represented by the affine transformation matrix A with the homogeneous column omitted. The stacking transformation is applied by direct matrix multiplication between the bottom-layer atomic coordinate matrix and the stacking configuration matrix $X^{\mathrm{top}} = \mathcal{M}(X^{\mathrm{bottom}} A)$, where $\mathcal{M}$ is a function to perform element-wise modulo-by-1 operation. In Appendix A, we show an example CIF file and configuration matrix of the bilayer material Al_4S_4 that is stacked by two monolayer materials of Al_2S_2.

For homobilayer materials, since both layers share the same crystal structure, only a monolayer graph is built from its CIF file and then is encoded using the graph encoder $E_{\mathcal{G}}(\cdot)$ to obtain a monolayer-level representation $\mathcal{Z}_{\mathcal{G}} \in \mathbb{R}^{d_{\mathcal{G}}}$. The stacking configuration of the bilayer is independently encoded by the stacking configuration encoder $E_{\mathcal{S}}(\cdot)$ to obtain a stacking representation $\mathcal{Z}_{\mathcal{S}} \in \mathbb{R}^{d_{\mathcal{S}}}$. In addition, monolayer properties are represented by $\mathcal{P} \in \mathbb{R}^{d_{\mathcal{P}}}$. These representations are concatenated to form a homogeneous bilayer embedding

$$\mathcal{Z}_{\mathrm{homo}} := \mathcal{Z}_{\mathcal{G}} \oplus \mathcal{Z}_{\mathcal{S}} \oplus \mathcal{P} \ \in \ \mathbb{R}^{d_{\mathcal{G}}+d_{\mathcal{S}}+d_{\mathcal{P}}}.$$

4.2 HETEROBILAYER MATERIALS

A heterobilayer consists of two distinct 2D monolayers stacked via van der Waals interactions. The stacking patterns and sequences in heterostructures generally exhibit greater complexity compared to those in homostructures. The stacking configuration of heterobilayer is often defined by in-plane relative translation between the two layers, relative rotation (twist angle), interlayer distance, and relative orientation of sublattices and atomic species. For hexagonal lattices such as transition-metal dichalcogenides and graphene-based systems, several high-symmetry stacking configurations are commonly considered, including AA stacking, where identical atoms are vertically aligned, and AB or BA stacking, where one layer is shifted so that different atomic species overlap. In heterobilayers, unlike homobilayers, these configurations are generally inequivalent due to broken inversion symmetry, resulting in distinct total energies and electronic structures. The relative stability of different stackings arises from the interplay of electrostatic interactions, interlayer orbital hybridization, and local atomic registry, and while one configuration corresponds to the global energy minimum, metastable stackings may coexist and form experimentally observed stacking domains.

When stacking heterogeneous bilayers, primitive unit cells of two different monolayers usually cannot be stacked directly due to large lattice mismatch. To resolve this, lattice transformation matrices are applied to each monolayer to construct supercells, and the lattice mismatch between the two supercell monolayers is quantified by the induced in-plane strain. For each monolayer pair, multiple supercell pairs are first generated and then filtered by requiring the in-plane strain of both monolayers to remain below a prescribed threshold, together with additional constraints such as a limit on the total number of atoms in the supercell. Among all supercell pairs that satisfy the prescribed criteria, the optimal pair is selected and stacked to form a bilayer structure, which is then subjected to further optimization steps, including interlayer distance optimization and structural relaxation, in order to limit computational cost.

5 EXPERIMENTS

5.1 DATASETS

Homobilayer materials. We use the BiDB dataset, which contains homogeneous bilayers derived from monolayers in the C2DB database. After removing 250 bilayers associated with 10 monolayers lacking CIF files, we obtain 10,899 valid bilayer structures. Eliminating duplicate stacking configurations yields 3,902 unique configurations, which are used to train the autoencoder. We choose bandgap as our target prediction property and exclude samples without bandgap labels, resulting in a final dataset of 6,683 bilayer materials formed from 940 unique monolayers. Since a single monolayer can generate multiple homogeneous bilayers under different stacking configurations, we perform 4-fold cross-validation by splitting the data at the monolayer level rather than the bilayer level. This strategy ensures that bilayers derived from the same monolayer do not appear in both training and test sets, thereby preventing data leakage. In our experiments, the stacking configuration is given by the affine matrix A with parameters p_1–p_7 and constant δz as described in Section 4.1.

Heterobilayer materials. We use the HetDB dataset, which comprises 336 heterogeneous bilayer materials constructed from 38 distinct monolayers. Each bilayer in HetDB is formed by stacking two different monolayers, and no duplicate bilayers are generated from the same monolayer pair under different stacking configurations. Accordingly, we employ 4-fold cross-validation, with data splits performed at the bilayer level, to evaluate model performance on HetDB. We use the twisted angle between the two monolayers provided directly in HetDB as the stacking configuration for each bilayer. The graph encoder $E_{\mathcal{G}}(\cdot)$ is applied independently to the bottom and top monolayers to obtain $\mathcal{Z}_{\mathcal{G}}^{bottom}, \mathcal{Z}_{\mathcal{G}}^{top} \in \mathbb{R}^{d_{\mathcal{G}}}$. These embeddings are then combined with the corresponding monolayer property vectors $\mathcal{P}^{bottom}, \mathcal{P}^{top} \in \mathbb{R}^{d_{\mathcal{P}}}$ and stacking configuration embedding $\mathcal{Z}_{\mathcal{S}} \in \mathbb{R}^{d_{\mathcal{S}}}$ to form the final representation.

5.2 EXPERIMENT SETTINGS

BiMat-ML. In the homobilayer setting, our BiMat-ML fuses the CGCNN-derived monolayer representation, the autoencoder-derived stacking configuration embedding, and the monolayer band gap. The autoencoder is trained for 100 epochs using Adam (learning rate 0.001, batch size 16) to obtain

latent stacking representations. The CGCNN model consists of three convolutional layers ($T = 3$). Atomic neighbors are determined within a cutoff radius $R = 8$ Å, with a maximum of $N = 12$ neighbors per atom. The model uses 64-dimensional atomic features and 128 hidden features, and is trained for 500 epochs using stochastic gradient descent (SGD) with a learning rate of 0.001 and a batch size of 128. For heterobilayers, we adopt the same CGCNN architecture and training setup. In HetDB, we additionally include the conduction band minimum (CBM) of both monolayers as input features, since the band gap is defined by the CBM–VBM difference and combining the band gap with either CBM or VBM is sufficient.

Baselines. For comparison, we evaluate several baseline models, including the original CGCNN (referred to as Direct in our experiments) (Xie & Grossman, 2018a), SE-CGCNN (Chen et al., 2024), SE-MEGNET (Chen et al., 2024) and SE-PAINN (Chen et al., 2024). For both the BiDB and HetDB datasets, these baselines are trained using the CIFs of stacked bilayer materials and also require access to bilayer CIFs during the inference. Unfortunately, CIFs of stacked bilayers can only be obtained through costly DFT calculations. In contrast, our proposed BiMat-ML does not rely on this strong assumption and instead predicts bilayer properties using only the CIFs and properties of the monolayers. All experiments are conducted on NVIDIA V100 GPU with 32GB RAM.

5.3 Experiment Results

5.3.1 Performance on Homobilayer Materials

Table 1: Performance comparison of BiMat-ML and baseline models for bandgap prediction on the BiDB.

Model	MAE ↓	MSE ↓	RMSE ↓	R^2 ↑
BiMat-ML	**0.13 ± 0.02**	**0.07 ± 0.02**	**0.26 ± 0.05**	**0.94 ± 0.02**
BiMat-ML w/o $\mathcal{P}$	0.35 ± 0.05	0.36 ± 0.11	0.59 ± 0.09	0.68 ± 0.06
Direct	0.38 ± 0.02	0.38 ± 0.02	0.61 ± 0.04	0.66 ± 0.02
SE-CGCNN	0.38 ± 0.05	0.49 ± 0.19	0.69 ± 0.12	0.57 ± 0.12
SE-MEGNET	0.36 ± 0.05	0.37 ± 0.13	0.60 ± 0.11	0.67 ± 0.06
SE-PAINN	0.36 ± 0.04	0.40 ± 0.08	0.63 ± 0.07	0.64 ± 0.06

Table 1 compares the bandgap prediction performance of BiMat-ML and baseline models on the BiDB dataset. BiMat-ML achieves the best overall performance, exhibiting the lowest MAE (0.13), MSE (0.07), and RMSE (0.26), as well as the highest coefficient of determination ($R^2 = 0.94$). Conventional single-CIF baselines, including Direct, SE-CGCNN and SE-PAINN (all without $\mathcal{P}$), show comparable but inferior performance, with MAE values of around 0.38.

Removing the monolayer property component (BiMat-ML without $\mathcal{P}$) results in a substantial degradation in performance relative to the full BiMat-ML model, with MAE and RMSE increasing to 0.35 and 0.59, respectively, and R^2 decreasing to 0.68. Nevertheless, even under this ablation, BiMat-ML outperforms the other baselines across most metrics, including MAE, MSE, and RMSE. These results highlight the critical role of incorporating monolayer property information for accurate bilayer bandgap prediction.

Overall, these results demonstrate that BiMat-ML not only significantly outperforms existing baselines but also benefits strongly from explicitly incorporating monolayer property information, enabling more accurate and robust bilayer bandgap prediction without requiring bilayer CIFs.

5.3.2 Performance on Heterobilyaer Materials

Table 2 reports the band gap prediction performance of BiMat-ML and baseline models on the HetDB dataset. In contrast to the BiDB results, conventional single-CIF baselines achieve stronger performance on HetDB, with SE-PAINN and SE-CGCNN yielding the lowest errors and highest predictive accuracy. In particular, SE-PAINN achieves the best overall performance, with an MAE of 0.11, an RMSE of 0.17, and an R^2 of 0.91, closely followed by SE-CGCNN with comparable accuracy. The original CGCNN model (Direct) also performs competitively, achieving an MAE of 0.14 and an R^2 of 0.85. BiMat-ML attains strong performance without access to bilayer CIFs, achieving an MAE of 0.13, an RMSE of 0.21, and an R^2 of 0.88, outperforming the Direct baseline

Table 2: Performance comparison of BiMat-ML and baseline models for bandgap prediction on the HetDB.

Model	MAE ↓	MSE ↓	RMSE ↓	R^2 ↑
BiMat-ML	0.13 ± 0.02	0.04 ± 0.01	0.21 ± 0.03	0.88 ± 0.04
BiMat-ML w/o $\mathcal{P}$	0.16 ± 0.01	0.09 ± 0.04	0.29 ± 0.06	0.76 ± 0.12
BiMat-ML w/o $\mathcal{S}$	0.14 ± 0.02	0.05 ± 0.02	0.22 ± 0.04	0.86 ± 0.05
BiMat-ML w/o $\mathcal{S}$ & $\mathcal{P}$	0.16 ± 0.02	0.09 ± 0.03	0.30 ± 0.04	0.73 ± 0.09
Direct	0.14 ± 0.01	0.05 ± 0.02	0.22 ± 0.05	0.85 ± 0.07
SE-CGCNN	0.12 ± 0.02	0.03 ± 0.02	0.18 ± 0.02	0.91 ± 0.02
SE-MEGNET	0.19 ± 0.02	0.09 ± 0.03	0.29 ± 0.06	0.76 ± 0.08
SE-PAINN	$\mathbf{0.11 \pm 0.02}$	$\mathbf{0.03 \pm 0.02}$	$\mathbf{0.17 \pm 0.02}$	$\mathbf{0.91 \pm 0.02}$

in terms of MAE and R^2. These findings indicate that, for heterogeneous bilayers, direct access to bilayer structures offers a predictive advantage by explicitly capturing complex interlayer interactions and stacking diversity. Nevertheless, BiMat-ML remains competitive without requiring bilayer CIFs derived from computationally expensive DFT calculations, making it a practical and scalable alternative for large-scale bilayer screening.

Ablation results further highlight the contributions of different model components. Removing monolayer property information ($\mathcal{P}$) leads to a noticeable degradation in performance, with the MAE increasing to 0.16 and the R^2 score dropping to 0.76. In contrast, removing stacking descriptors ($\mathcal{S}$) results in a smaller but consistent decline in accuracy (MAE $= 0.14$, $R^2 = 0.86$). Eliminating both components yields the poorest performance among all BiMat-ML variants, confirming that monolayer properties and stacking information provide complementary predictive value for heterogeneous bilayers. However, compared with the ablation results observed on BiDB, both monolayer property and stacking configuration information are less critical for the HetDB dataset. For HetDB, the correlations between the top and bottom monolayer band gaps and the bilayer band gap are substantially weaker ($r \approx 0.42$ and 0.54, respectively). In contrast, for BiDB, the monolayer and bilayer band gaps are much more strongly correlated (Pearson $r \approx 0.94$).

5.3.3 Running Time

We compare the training and inference runtimes of BiMat-ML and baseline models on both datasets. On BiDB, BiMat-ML and CGCNN-Direct have similar training times (900 s), while SE-PAINN remains comparable (923 s), SE-CGCNN is slower (1501 s), and SE-MEGNET is the slowest (2725 s). On HetDB, BiMat-ML is the most efficient, requiring 100 s for training versus 212 s for CGCNN-Direct, 236 s for SE-CGCNN, 375 s for SE-MEGNET and 110 s for SE-PAINN. All models exhibit identical per-sample inference times (0.5 s). In contrast, density functional theory calculations using VASP (Kresse & Furthmüller, 1996) take approximately 4.9 hours on an Intel Xeon Gold 6130H CPU (Chen et al., 2024). Overall, BiMat-ML achieves competitive or superior efficiency while avoiding costly DFT-based structure calculation.

6 Conclusion

In this work, we proposed a multimodal learning framework that can effectively capture the relationships between the structures and properties of stacked two-dimensional materials. Our BiMat-ML framework does not require the access of CIFs of stacked bilayer materials as baselines and instead uses two CGCNNs to process CIFs of bottom and top layer materials respectively anf effectively fuses them with configuration representation learned by one autoeconder. We emphasize that CIFs of stacked bilayer materials are less available due to very high DFT computational cost. We evaluated our method with BiDB for homogeneous bilayers and HetDB for heterogeneous bilayers. Experimental results evaluated with BIDB for homogeneous bilayers and HetDB for heterogeneous bilayers demonstrate effectiveness and efficiency of our framework. Our BiMat-ML framework is model-agnostic and readily applicable to a range of graph neural network architectures. In future, we will study other GNN architectures such as MEGNET (Chen et al., 2019) and PAINN (Schütt et al., 2023). We also plan to develop algorithms for determining stacking configurations that potentially produce emergent properties in stacked bilayer materials.

ACKNOWLEDGMENTS

This work was supported in part by the National Institute of General Medical Sciences of National Institutes of Health under award P20GM139768, and the Arkansas Integrative Metabolic Research Center at the University of Arkansas.

REFERENCES

Ranjan Kumar Barik and Lilia M Woods. High throughput calculations for a dataset of bilayer materials. *Scientific Data*, 10(1):232, 2023.

Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31(9):3564–3572, 2019.

Xinyu Chen, Guoliang Ru, and Weihong Qi. Structural embedding methods for machine learning models accelerate research on stacked 2d materials. *The Journal of Physical Chemistry C*, 128 (37):15512–15521, 2024.

Kamal Choudhary and Brian DeCost. Atomistic line graph neural network for improved materials property predictions. *npj Computational Materials*, 7(1):185, 2021.

M. N. Gjerding, A. Taghizadeh, A. Rasmussen, S. Ali, F. Bertoldo, T. Deilmann, N. R. Knøsgaard, M. Kruse, A. H. Larsen, S. Manti, T. G. Pedersen, U. Petralanda, T. Skovhus, M. K. Svendsen, J. J. Mortensen, T. Olsen, and K. S. Thygesen. Recent progress of the computational 2d materials database (c2db). *2D Materials*, 8(4):044002, 2021. doi: 10.1088/2053-1583/ac1059. URL https://doi.org/10.1088/2053-1583/ac1059.

Georg Kresse and Jürgen Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical review B*, 54(16):11169, 1996.

Guohong Li, A Luican, JMB Lopes dos Santos, AH Castro Neto, A Reina, J Kong, and EY Andrei. Observation of van hove singularities in twisted graphene layers. *Nature physics*, 6(2):109–113, 2010.

Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

Nicolas Mounet, Marco Gibertini, Philippe Schwaller, Davide Campi, Andrius Merkys, Antimo Marrazzo, Thibault Sohier, Ivano Eligio Castelli, Andrea Cepellotti, Giovanni Pizzi, et al. Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds. *Nature nanotechnology*, 13(3):246–252, 2018.

Sahar Pakdel, Asbjørn Rasmussen, Alireza Taghizadeh, Mads Kruse, Thomas Olsen, and Kristian S Thygesen. High-throughput computational stacking reveals emergent properties in natural van der waals bilayers. *Nature Communications*, 15(1):932, 2024.

M. O. Sauer, P. M. Lyngby, and K. S. Thygesen. Dispersion-corrected machine learning potentials for 2d van der waals materials. *Physical Review Materials*, 9(7), 2025. doi: 10.1103/cl8c-8f1f. URL https://doi.org/10.1103/cl8c-8f1f.

Kristof T Schütt, Oliver T Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. 2021. *URL https://arxiv. org/abs/2102.03150*, 2023.

Minh-Hao Van, Prateek Verma, Chen Zhao, and Xintao Wu. A survey of ai for materials science: Foundation models, llm agents, datasets, and tools. *arXiv preprint arXiv:2506.20743*, 2025.

Shixuan Wang, Xuehao Cui, Chang'e Jian, Haowei Cheng, Mengmeng Niu, Jia Yu, Jiaxu Yan, and Wei Huang. Stacking-engineered heterostructures in transition metal dichalcogenides. *Advanced Materials*, 33(16):2005735, 2021.

Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical Review Letters*, 120(14):145301, 2018a. doi: 10.1103/PhysRevLett.120.145301. URL https://doi.org/10.1103/PhysRevLett.120.145301.

Tian Xie and Jeffrey C Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14):145301, 2018b.

Qian Zhang, Jinlong Yang, and Wei Hu. Pyhtstack2d: A python package for high-throughput homo/hetero stacking of 2d materials. *Computer Physics Communications*, 312:109618, 2025.

Jun Zhou, Lei Shen, Miguel Dias Costa, Kristin A Persson, Shyue Ping Ong, Patrick Huck, Yunhao Lu, Xiaoyang Ma, Yiming Chen, Hanmei Tang, et al. 2dmatpedia, an open computational database of two-dimensional materials from top-down and bottom-up approaches. *Scientific data*, 6(1):86, 2019.

A Example of Stacking Configuration Construction

```
data_image0
_chemical_formula_structural       Al2S2Al2S2
_chemical_formula_sum              "Al4 S4"
_cell_length_a       3.584005211768305
_cell_length_b       3.5840052117683046
_cell_length_c       30.47305759801167
_cell_angle_alpha    90.0
_cell_angle_beta     90.0
_cell_angle_gamma    120.00000000000001

_space_group_name_H-M_alt    "P 1"
_space_group_IT_number       1

loop_
  _space_group_symop_operation_xyz
  'x, y, z'

loop_
  _atom_site_type_symbol
  _atom_site_label
  _atom_site_symmetry_multiplicity
  _atom_site_fract_x
  _atom_site_fract_y
  _atom_site_fract_z
  _atom_site_occupancy
  Al  Al1       1.0  0.6666666683151299  0.33333333433347323  0.2812051466262817  1.0000
  Al  Al2       1.0  0.6666666683151299  0.33333333433347323  0.3664662836698296  1.0000
  S   S1        1.0  0.3333333327225652  0.6666666654451305  0.24611905043914484  1.0000
  S   S2        1.0  0.3333333327225652  0.6666666654451305  0.40155237985696646  1.0000
  Al  Al3       1.0  0.6666666683151299  0.33333333433347323  0.629471080094424  1.0000
  Al  Al4       1.0  0.6666666683151299  0.33333333433347323  0.5442099427227175  1.0000
  S   S3        1.0  0.0  0.0  0.6645571759534022  1.0000
  S   S4        1.0  0.0  0.0  0.5091238465355805  1.0000
```

Figure 3: CIF file of an Al_4S_4 bilayer showing the fractional atomic coordinates.

We illustrate the stacking configuration construction using the bilayer Al_4S_4 shown in Figure 3. The corresponding BiDB stacking descriptor is `2AlS-2-2-1_0_0_-1-Iz-0.33_-0.33` which specifies a stacking configuration that includes a flip transformation (`Iz`, $p_5 = -1$). Inspection of the associated CIF file shows that the first four atoms belong to the bottom monolayer, while the remaining four atoms correspond to the top monolayer.

We first determine the out-of-plane shift δz from the CIF. For the Al atom pair (Al_1, Al_3), the fractional z coordinates are $z^{\text{bottom}} = 0.28$ and $z^{\text{top}} = 0.63$. Using the definition $\delta z = z^{\text{top}} - p_5\, z^{\text{bottom}}$ with $p_5 = -1$, we obtain $\delta z = 0.63 + 0.28 = 0.91$. The same value is obtained for all other corresponding atom pairs (e.g., Al_2/Al_4, S_1/S_3, S_2/S_4), confirming that the interlayer shift is uniform across the

bilayer. In practice, δz can therefore be computed from any corresponding atom pair and used as a consistency check for the stacking configuration.

Next, we construct the stacking transformation from the descriptor parameters. The in-plane rotation parameters are $p_1 = -1$, $p_2 = 0$, $p_3 = 0$, and $p_4 = -1$; the flip parameter is $p_5 = -1$; the in-plane translation is $(p_6, p_7) = (0.33, -0.33)$; and the interlayer shift is $\delta z = 0.91$. Extracted from the CIF file shown in Figure 3, we have atomic information in Table 3.

Table 3: Atomic information extracted from CIF.

Mono.	Type	Label	Multi.	x	y	z	Occu.
Bottom	Al	Al_1	1.00	0.67	0.33	0.28	1.00
	Al	Al_2	1.00	0.67	0.33	0.37	1.00
	S	S_1	1.00	0.33	0.67	0.25	1.00
	S	S_2	1.00	0.33	0.67	0.40	1.00
Top	Al	Al_3	1.00	0.67	0.33	0.63	1.00
	Al	Al_4	1.00	0.67	0.33	0.54	1.00
	S	S_3	1.00	0.00	0.00	0.66	1.00
	S	S_4	1.00	0.00	0.00	0.51	1.00

For better mathematical presentation, we re-denote X^{top} as X^{t} and X^{bottom} as X^{b}. From that, let X^{b} denote the fractional coordinates of the bottom monolayer atoms in homogeneous coordinates, and let A denote the stacking transformation matrix:

$$X_{\text{b}} = \begin{bmatrix} 0.67 & 0.33 & 0.28 & 1 \\ 0.67 & 0.33 & 0.37 & 1 \\ 0.33 & 0.67 & 0.25 & 1 \\ 0.33 & 0.67 & 0.40 & 1 \end{bmatrix}, A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0.33 & -0.33 & 0.91 \end{bmatrix}.$$

The stacking transformation is applied via matrix multiplication and we can calculate atomic position of top monolayer X^{t} using inputs from X^{b} and A:

$$\begin{aligned} X^{\text{t}} &= \mathcal{M}(X^{\text{b}} A) \\ &= \mathcal{M}\left(\begin{bmatrix} -0.33 & -0.67 & 0.63 \\ -0.33 & -0.67 & 0.54 \\ 0 & -1.00 & 0.66 \\ 0 & -1.00 & 0.51 \end{bmatrix} \right) = \begin{bmatrix} 0.67 & 0.33 & 0.63 \\ 0.67 & 0.33 & 0.54 \\ 0.00 & 0.00 & 0.66 \\ 0.00 & 0.00 & 0.51 \end{bmatrix} \end{aligned}$$

where $\mathcal{M}$ applies a modulo-1 operation to the in-plane coordinates to enforce periodic boundary conditions. This yields the final fractional atomic coordinates of the top monolayer.