

## Computer Programs in Physics

## PyHTStack2D: A Python package for high-throughput homo/hetero stacking of 2D materials

Qian Zhang<sup>a</sup>, Jinlong Yang<sup>b</sup>, Wei Hu<sup>a, ID, \*</sup><sup>a</sup> Hefei National Research Center for Physical Sciences at the Microscale, Hefei National Laboratory, University of Science and Technology of China, Hefei, Anhui 230026, China<sup>b</sup> Key Laboratory of Precision and Intelligent Chemistry, Department of Chemical Physics, University of Science and Technology of China, Hefei, Anhui 230026, China

## ARTICLE INFO

The review of this paper was arranged by Prof. Weigel Martin

## Keywords:

High-throughput first-principle calculation  
2D layered structure  
Stacking mode  
Python  
Bash

## ABSTRACT

Two-dimensional (2D) van der Waals (vdWs) structures are the subject of extensive research in materials science, celebrated for their unique physical properties and potential technological applications. However, the diversity of stacking modes in 2D vdWs structures poses a challenge for research. In response to the complexity of the stacking process for these layered structures, we have developed a Python package, PyHTStack2D, specifically designed to support High-Throughput Stacking of 2D materials research. The package provides two primary functionalities: Firstly, it facilitates the batch stacking of homo- and heterostructures, with careful consideration of specific sequences and patterns, such as those observed in the 1T/2H phase transitions of transition metal dichalcogenides; Secondly, it aids in the efficient creation of computational directories and the generation of requisite shell scripts for the batch computation submissions of the stacked structures. By employing this package, we performed high-throughput computational simulations of properties such as electronic energy band structures and magnetic ground states of bilayers composed of 2H-TMDHs. These results have enabled us to identify the types of electronic band structures within these systems, providing critical insights into their potential applications in optoelectronics and photocatalysis. Furthermore, preliminary findings indicate the potential feasibility of generating bipolar magnetic semiconductors via the stacking of magnetic monolayers. The PyHTStack2D package provides an opportunity to perform efficient high-throughput calculations of 2D vdWs homo/heterostructures.

## Program Summary

**Program title:** PyHTStack2D

**CPC Library link to program files:** <https://doi.org/10.17632/n94n2zcxdr.1>

**Developer's repository link:** <https://github.com/QianZhang21/pyhtstack2d-master>

**Licensing provisions:** BSD 3-clause

**Programming language:** Python 3, Bash (shell)

**Nature of problem:** Implement the theoretical calculations of 2D layered materials, focusing on the challenges posed by the diversity of stacking modes. While stacking these layers to form homo/heterostructures can overcome some limitations inherent to monolayers, the variety of possible configurations introduces significant complexity. This complexity must be incorporated into research to enable accurate modeling and facilitate further studies on these materials, ensuring that the unique properties of different stacking arrangements are properly understood and effectively utilized.

**Solution method:** The Python package, PyHTStack2D, is used to perform high-throughput stacking of 2D vdW homo/heterostructures. It accounts for various stacking patterns in specific systems and considers different stacking sequences, such as a Janus-TMD monolayer that is asymmetric in the vertical direction. By flipping the monolayer in the Z-direction before stacking, different bilayer structures can be generated. PyHTStack2D also assists in generating shell scripts for high-throughput first-principles calculations, facilitating large batch computations.

## 1. Introduction

Two-dimensional (2D) monolayer materials, with their distinctive physical and chemical properties, demonstrate considerable potential across various application domains [1–3]. Notably, in fields such as photocatalysis and electronic devices, these materials are highly regarded for their large specific surface area and exceptional electronic properties. Despite their advantages, challenges persist in these appli-

\* Corresponding author.

E-mail address: [whuustc@ustc.edu.cn](mailto:whuustc@ustc.edu.cn) (W. Hu).

<https://doi.org/10.1016/j.cpc.2025.109618>

Received 4 December 2024; Received in revised form 8 March 2025; Accepted 7 April 2025

cations, including narrow absorption spectra and short carrier lifetimes in photocatalysis, as well as low on/off ratios and inadequate stability in electronic devices [4–7]. The construction of van der Waals (vdWs) heterostructures represents an effective strategy to overcome these limitations [4,8–11]. By forming heterostructures with other materials, it is possible to broaden the absorption spectrum, enhance carrier separation efficiency, and optimize the electronic transport performance and stability of electronic devices. Furthermore, vdWs heterostructures formed through the vertical stacking of two-dimensional materials typically do not rely on chemical bonds and are unaffected by lattice matching constraints [8,10,12]. This flexibility in assembly provides extensive opportunities for property tuning in new material design, thereby opening new prospects for research and application.

With the advancement of technology, research on bilayer vdWs materials has made significant progress. Experimentally, various bilayer materials such as bilayer  $\text{MoS}_2$  [13,14], h-BN/graphene [15], metal transition metal dichalcogenides (m-TMD)/semiconductor TMD (s-TMD) heterostructures [16], and  $\text{MoS}_2$ /graphene heterostructures [17] have been successfully synthesized. These achievements have stimulated further interest in the study of 2D vdWs materials. However, experimental research often faces challenges due to technical limitations, environmental conditions, and human factors, etc. Fortunately, theoretical computational predictions have emerged as one of the powerful tools in materials research, significantly accelerating the research cycle [18,19]. In recent years, the increasing utilization of high-throughput computing technology has made the construction of large-scale materials databases possible [20–23]. These databases provide a rich theoretical foundation for materials research. By integrating experimental studies with computational simulations [24–28], researchers can achieve a deeper understanding and a more comprehensive grasp of material properties and behaviors.

High-throughput computational studies of 2D bilayer vdW structural systems have also been reported, including homostructure databases developed by R. K. Barik et al. [29] and S. Pakde et al. [30], as well as a heterostructure database for TMDs with the chemical formula  $\text{XY}_2$  by R. Dong et al. [31]. These studies illustrate the necessity of considering various stacking modes in 2D vdW bilayer structures, including patterns and sequences, which significantly affect the material properties. This point is also confirmed by many other reports [32–34], such as the research conducted by C. Zhang et al. [35], who theoretically and experimentally investigated  $\text{MoS}_2/\text{WSe}_2$  hetero-bilayer superlattices in AA, AB, and various other stacking patterns, revealing a direct correlation between stacking patterns and local electronic structures. The different stacking patterns can be created through rotations around the vertical axis or horizontal layer sliding [36,37]. It is worth noting that X. Zhang et al. [38] successfully synthesized large-size bilayer  $\text{MoS}_2$  with AA and AB stacking by employing reverse-flow chemical vapor epitaxy, precisely controlling the growth temperature during the key step.

The stacking patterns and sequences in heterostructures generally exhibit greater complexity compared to those in homostructures. For instance, while stacking hexagonal  $\text{MoS}_2$  homostructures involves fewer configurations, heterostructures such as  $\text{MoSSe}/\text{WSTe}$  in identical stacking patterns necessitate consideration of four additional stacking sequences. Due to this complexity, few software tools are available to facilitate the high-throughput stacking of bilayers, especially for heterostructures. Furthermore, high-throughput computations generate large datasets, emphasizing the need for efficient tools to rapidly identify materials with desirable properties for specific applications, thereby accelerating the materials discovery and design process. For example, direct bandgap materials are essential for optoelectronic devices [39,40], Type II band alignment heterojunctions are advantageous for photocatalytic applications [31,41,42], and for bipolar magnetic semiconductors (BMS), it is crucial that the spin polarization of the conduction band minimum (CBM) and valence band maximum (VBM) are opposite [43,44]. To address these challenges, we have developed a specialized Python package, PyHTStack2D, designed for the rapid

and efficient High-Throughput Stacking of 2D materials, including both homo- and heterostructures. This package enables stacking of a broader range of systems and considers various high-symmetry stacking patterns or sequences, freeing up the manual stacking process. Additionally, the package assists in generating input files and shell scripts for batch computation submissions, with specific support for high-throughput first-principles calculations using the Vienna Ab initio Simulation Package (VASP) [45], thereby simplifying the high-throughput computational process. The package also provides tools for batch extraction and preliminary analysis of results, aiding in the efficient identification of materials of interest. The main classes and functions of PyHTStack2D, along with its overall workflow, are illustrated in Fig. 1, providing a visual guide to its structure and functionality.

The remainder of this manuscript is organized as follows: Section 2 briefly introduces some 2D monolayer structures and their stacking modes as considered in PyHTStack2D, outlining the specific stacking sequences and patterns achievable with the package. Section 3 provides a comprehensive summary of the main capabilities of the modules implemented in PyHTStack2D, highlighting their functional benefits and potential applications. Section 4 showcases a series of example calculations, demonstrating the utility of PyHTStack2D in practical scenarios. Finally, Section 5 consolidates our findings and discusses potential future developments and enhancements to the PyHTStack2D package.

## 2. 2D monolayers and their stacking modes

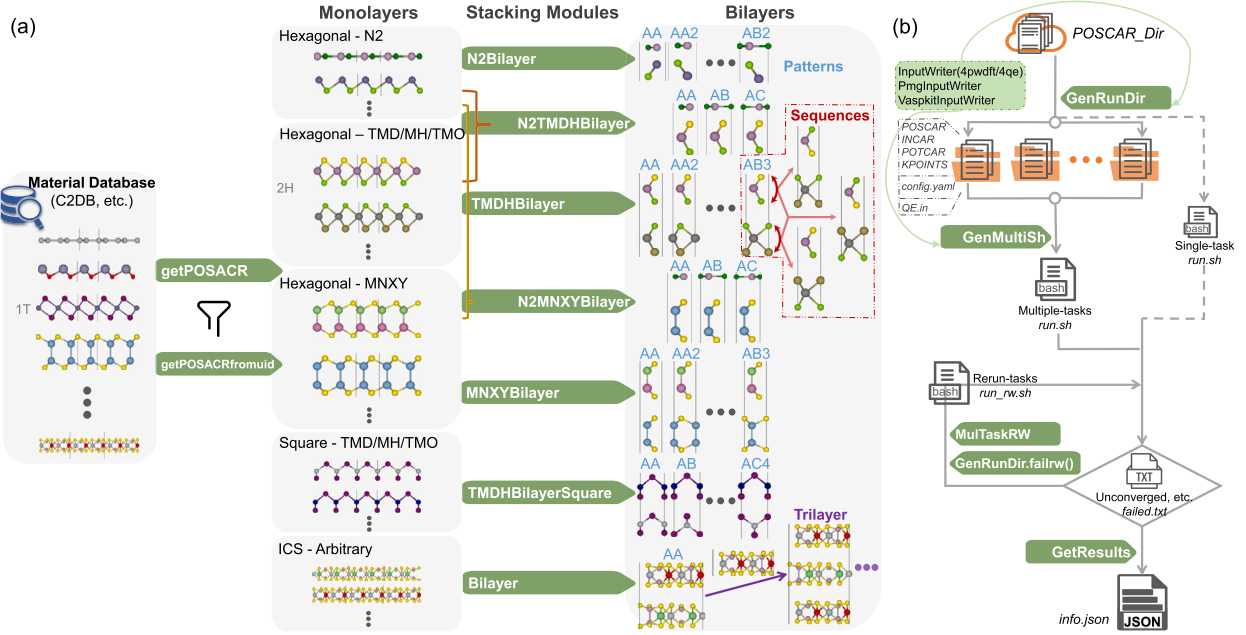
Currently, over 50 monolayers have been synthesized or exfoliated, and computational studies predict that many hundreds of known layered bulk crystals can be exfoliated into monolayers [1,46]. There is a database of 2D materials known as the Computational 2D Materials Database (C2DB) that incorporates a comprehensive workflow for characterizing the stability and properties of hypothetical monolayers [47]. The C2DB already includes over 15,000 entries and will continue expanding over time. Therefore, considering the acquisition of 2D monolayer structures from the C2DB would be a judicious selection. We here present a further discussion on several types of monolayer structures, including their designations in the package and their stacking modes.

### 2.1. N2 monolayer and N2/N2 bilayer

We designate the hexagonal unit cell system, which contains only two atoms, as the “N2” monolayer. In the C2DB database, stable monolayers belonging to this category can be further classified into three types based on their layer group: ‘P6/mmm’ (e.g., graphene), ‘P-6m2’ (e.g., BN), and ‘P3m1’ (e.g., GeSe). For bilayer structures composed solely of N2 monolayers, such as BN/BN, we consider five high-symmetry stacking modes: (1) Eclipsed with B over B and N over N; (2) Eclipsed with B over N; (3) Staggered with B over N; (4) Staggered with B over B; (5) Staggered with N over N. In the study by Gabriel et al. [48], these stacking modes are labeled as AA, AA’, AB, AB’, and A’B, respectively. However, in the PyHTStack2D package, considering that the first two stacking modes have identical atomic positions when the element types are ignored, they are grouped under the AA stacking pattern, while the latter three modes are grouped under the AB stacking pattern. Subsequently, by considering the specific element types and adjusting the stacking sequence of atoms within the same pattern, we can recover the five stacking modes identified in Gabriel et al.’s study. For the N2 monolayer with ‘P3m1’, the height difference between the two atoms along the vertical (Z-) direction sometimes necessitates additional consideration of vertical arrangement when stacking it with another layer. In *pyhtstack2d/Example1*, the bilayer structures formed by stacking BN and GeSe are demonstrated.

### 2.2. TMDH monolayer and TMDH/TMDH bilayer

For unit cell systems containing three atoms, this category predominantly includes well-known transition metal dichalcogenides (TMDs)



**Fig. 1.** The main classes, functions, and workflow of PyHTStack2D. (a) Obtaining the structure files of monolayers from the materials database. For the C2DB database, POSCAR files can be obtained using the `getPOSCAR` or `getPOSCARfromuid` function interfaces. For forming bilayer structures of hexagonal N2, TMDH, MNXY, and square TMDH systems, the classes `N2Bilayer`, `TMDHBilayer`, `MNXYBilayer`, and `TMDHBilayerSquare` are used to perform stacking under various modes and sequences. For mixed systems of N2 with TMDH, and N2 with MNXY, the `N2TMDHBilayer` and `N2MNXYBilayer` classes can be used for stacking. For arbitrary system under the identical crystal system (ICS), stacking can be achieved by providing the POSCAR files of two structures along with the `Bilayer` class, considering the stacking sequence. (b) For a set of POSCAR files stored in `POSCAR_Dir` to be computed, the `GenRunDir` class generates the computation directories and input files for each POSCAR. The generated files can be in VASP, PWDFT, or QE input formats, based on the input parameters. At the same time, for single-task computations, `GenRunDir` also generates the corresponding job submission script. For multi-task computations, the `GenMultiSh` class is used to generate the multi-task job submission script `run.sh`. During computation, failed files due to non-convergence or other factors are recorded in `failed.txt`, which is re-read by `GenRunDir` (for single tasks) or `MultiTaskRW` (for multi-tasks) to generate the re-computation submission script `run_rw.sh`. Upon completion of all computation tasks, the `GetResults` class is used to extract the calculated material properties and store them in the `info.json` file.

and metal halides (MHs), with a smaller subset classified as transition metal oxides (TMOs). We abbreviate these systems as TMDH. The stable monolayer structures of these materials primarily exhibit hexagonal and square lattice types. Among the hexagonal monolayers, the two most common polytypes are octahedral (1T) and trigonal prismatic (2H). S. Wang et al. [37] provided a comprehensive exposition on the fabrication of bilayer heterostructures based on 2H-TMDs and unified the naming for high-symmetry stacking patterns. Here, we take the 2H-MoS<sub>2</sub>/WSe<sub>2</sub> bilayer as an example to list six high-symmetry stacking patterns, where the notation before the slash corresponds to the reference naming, and that after the slash to the PyHTStack2D package: (1) AA/AA, eclipsed with Mo over W and S over Se; (2) AA'/AA2, eclipsed with Mo over Se and staggered with S over W; (3) A'A/AA2-2, staggered with Mo over Se and eclipsed with S over W; (4) AB/AB, eclipsed with Mo over Se and S over W; (5) AB'/AB2, staggered with Mo over W and eclipsed with S over Se; (6) A'B/AB3, eclipsed with Mo over W and staggered with S over Se. Following these eclipsed and staggered arrangements, we have similarly constructed and named the stacking patterns for bilayers formed by the 1T phase and square monolayers in the PyHTStack2D package. Additionally, T-phase and H-phase coupling stackings were also explored, with the high-symmetry stacking patterns under investigation being consistent with those previously reported [49]. It is crucial to note that when one layer in the bilayer is no longer mirror-symmetric, such as in the case of Janus-MoSSe/WSTe in heterojunction stacking, the stacking sequence along the Z-direction can vary within the same stacking pattern. For example, the sequence could be S-Mo-Se-S-W-Te, Se-Mo-S-S-W-Te, S-Mo-Se-Te-W-S, or Se-Mo-S-Te-W-S, and these different sequences may result in variations in electronic properties and other characteristics. Therefore, in some cases, it is necessary to consider the stacking sequence when constructing bilayer structures. Similarly, the bilayer structures formed by the stacking of MoS<sub>2</sub>/WSe<sub>2</sub>, MgI<sub>2</sub>/ZnI<sub>2</sub>,

and CrI<sub>2</sub>/NiI<sub>2</sub> in `pyhtstack2d/Example1` illustrate the implementation for 2H, 1T, and square stacking patterns within the PyHTStack2D package. Additionally, the MoS<sub>2</sub>/WSTe example demonstrates the handling of stacking sequence.

### 2.3. MNXY monolayer and MNXY/MNXY bilayer

For 2D materials with four atoms in the unit cell, we focus on hexagonal systems, particularly a class of materials resembling the TMD structure, with the chemical formula MNXY, where M and N are metal elements, and X and Y are chalcogen or halogen elements. The MNXY monolayer also has the two most common polymorphs, referred to as the 1T and 2H phases in the PyHTStack2D package. B. Sa et al. [50] constructed Janus 2D III-VI bilayer heterostructures based on the 2H-MNXY structure, in which the high-symmetry stacking patterns are essentially similar to those of 2H-TMDH. The main distinction is that, in the former, each monolayer consists of two chalcogen or halogen layers sandwiching two metal atoms aligned in-plane, rather than a single transition metal or metal element. In B. Sa et al.'s study, the stacking patterns are differentiated by rotation angles: 0°, 60°, 120°, 180°, 240°, and 300°, corresponding to the stacking configurations named AA, AA2-2, AA2, AB, AB3, and AB2 in the PyHTStack2D package. The examples in `pyhtstack2d/Example1` demonstrate the construction of bilayer MNXY structures in both polymorphs, using GaInS<sub>2</sub>/GaAlSe<sub>2</sub> and In<sub>2</sub>Se<sub>2</sub>/Ga<sub>2</sub>Te<sub>2</sub> as cases. Similarly, when either layer lacks mirror symmetry, different stacking sequences will be considered.

### 2.4. N2/TMDH and N2/MNXY bilayer

In addition to the aforementioned special high-symmetry stacking configurations, we also identified the high-symmetry stacking patterns

for N2/TMDH and N2/MNXY, focusing solely on the 2H phase of TMDH or MNXY. Under these conditions, we primarily consider three types of high-symmetry stacking patterns. Using BP/MoS<sub>2</sub> from *pyhtstack2d/Example1* as an example: (1) AA, B eclipsed with over Mo and P over S; (2) AB, B eclipsed with over Mo and staggered with P over S; and (3) AC, B eclipsed with over S and staggered with P over Mo. In this system, the stacking sequence is considered by altering the atomic arrangement of the BP layer. For instance, modifying the atomic sequence in the AA stacking pattern leads to a new bilayer structure where B is eclipsed over S and P over Mo.

### 3. Capabilities of modules

In this section, we provide a concise overview of the five primary capabilities of the PyHTStack2D package, which include:

- Retrieving POSCAR files for 2D monolayer materials from the C2DB database.
- Constructing bilayer structures through stacking.
- Automating the batch generation of input files for first-principles calculations.
- Creating shell scripts for efficient batch job submission.
- Batch extracting essential information and conducting preliminary data analysis.

#### 3.1. Retrieving POSCAR files from C2DB

To conveniently and efficiently batch retrieve POSCAR files for 2D monolayer structures that meet specific criteria from the C2DB [47], one can use the `getPOSCAR` or `getPOSCARfromuid` function interfaces within the `queryDB` module, as shown in Listing 1. It should be emphasized that both interfaces require the complete dataset file *c2db.db* as input, which can be obtained by contacting Professor Kristian S. Thygesen. Furthermore, the `DBuildPOSCAR` function in the `getPOSCARurl` module enables the retrieval of POSCAR files based on the provided uid (Unique ID in C2DB) by accessing the web page created by their team at <https://c2db.fysik.dtu.dk/>. It is also worth noting that obtaining POSCAR files for 2D monolayer structures from other databases [1,51] is encouraged, as long as they comply with the standard POSCAR file format, allowing them to be used for subsequent high-throughput bilayer stacking.

Listing 1: Obtain the POSCAR files from C2DB.

```
from pyhtstack2d.tools.queryDB import *

# Selection criteria:
# Only 3 atoms, and layer group of P3m1.
criteria = 'natoms=3, layergroup=p3m1'
getPOSCAR("c2db.db", criteria)

# uid is the Unique ID in C2DB
uid="1MoSse-1"
getPOSCARfromuid("c2db.db", uid)

from pyhtstack2d.tools.getPOSCARurl import *

# uid is the Unique ID in C2DB
uid="2YSe3-1"
DBuildPOSCAR(uid, print_url=True)
```

#### 3.2. Stacking bilayers

In Section 2, we introduce five bilayer systems: N2/N2, TMDH/TMDN, MNXY/MNXY, N2/TMDH, and N2/MNXY. These systems can be stacked using the `stackBilayer` module through the classes `N2Bilayer`, `TMDHBilayer`, `MNXYBilayer`, `N2TMDHBilayer`, and

`N2MNXYBilayer`, respectively. For each system, at least three high-symmetry stacking patterns and their possible stacking sequences will be considered. Additionally, the `Bilayer` class can be applied to stack any two monolayer structures with the identical crystal system (ICS). Although this class does not account for specific stacking patterns, it allows the inversion of one layer to achieve different stacking sequences if one of the layers lacks mirror symmetry. The `Bilayer` class retains the in-plane atomic coordinates from the input monolayer structure files, making it possible to explore multiple stacking patterns by manually rotating the atomic coordinates of one layer before input. Moreover, the generation of layered structures with more than two layers can be achieved using the `Bilayer` class by providing a non-monolayer POSCAR file as input. Listing 2 demonstrates the use of the `WritePOSCAR` function of `Bilayer` class to generate MoS<sub>2</sub>/WSe<sub>2</sub> bilayers in AA and AB stacking patterns, as well as a MoS<sub>2</sub>/WSe<sub>2</sub>/MoS<sub>2</sub> trilayer structure. Notably, the lattice constant of the newly generated heterostructure is the average of the lattice constants of the two provided monolayer structures.

Listing 2: Using the `Bilayer` class to generate multilayer structures.

```
from pyhtstack2d.buildbilayer.\
    stackBilayer import Bilayer

# Element, lattice and position of MoS2
elem1 = ['Mo', 'S', 'S']
la1 = [[3.184, 0.0, 0.0],
       [-1.592, 2.757, 0.0],
       [0.0, 0.0, 18.127]]
pos1 = [[0.0, 0.0, 0.50],
        [2/3, 1/3, 0.59],
        [2/3, 1/3, 0.41]]

# Element, lattice and position of WSe2
elem2 = ['W', 'Se', 'Se']
la2 = [[3.319, 0.0, 0.0],
       [-1.659, 2.874, 0.0],
       [0.0, 0.0, 18.358]]
pos2 = [[0.0, 0.0, 0.50],
        [2/3, 1/3, 0.59],
        [2/3, 1/3, 0.41]]

# Construct the AA stacking of MoS2/WSe2
Bilayer(elem1, elem2,
        la1=la1, la2=la2,
        position1=pos1,
        position2=pos2).WritePOSCAR()

# Position of WSe2
pos2 = [[2/3, 1/3, 0.50],
        [0.0, 0.0, 0.59],
        [0.0, 0.0, 0.41]]

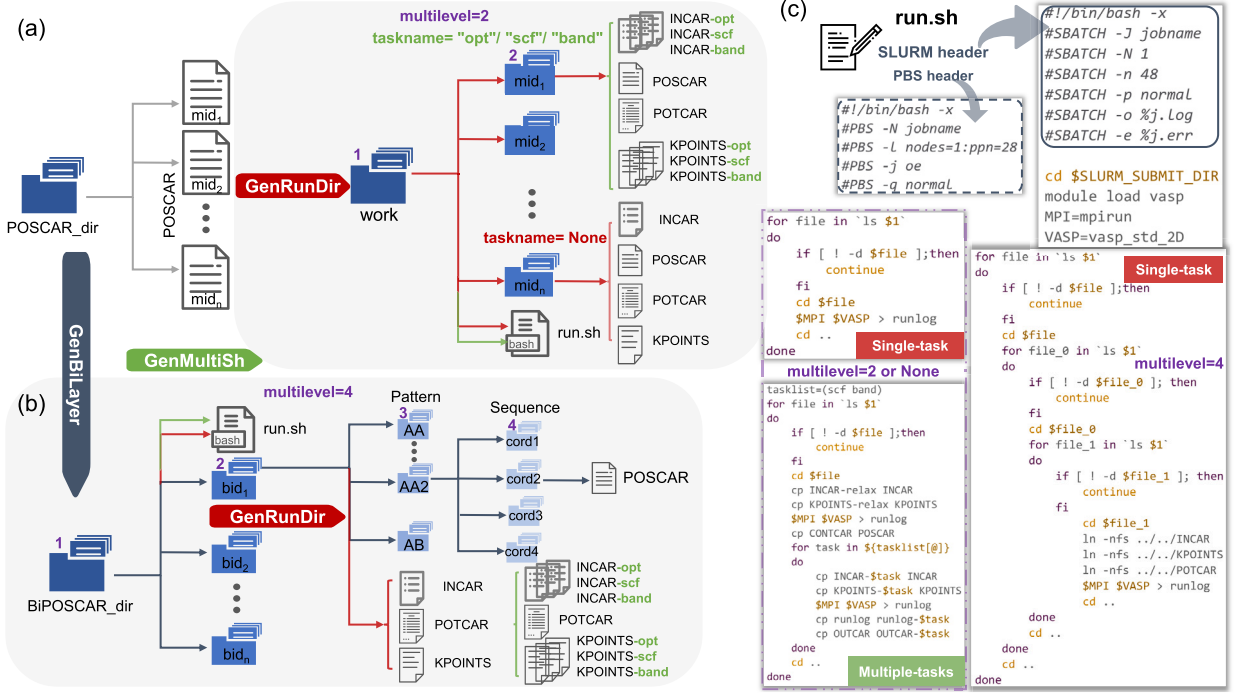
# Construct the AB stacking of MoS2/WSe2
Bilayer(elem1, elem2,
        la1=la1, la2=la2,
        position1=pos1, position2=pos2,
        stackmode="AB").WritePOSCAR()

# The path of the POSCAR file of MoS2/WSe2
# S2MoSe2W_4.15%_3 divided into 3 parts by '_'
# S2MoSe2W: formula
# 4.15%: lattice mismatch
# 3: the number of atoms in the first layer
# AA: stacking pattern
# cord1: first stacking sequence
MoS2WSe2_file = "BiPOSCAR_dir/S2MoSe2W" \
                "_4.15%_3/AA/cord1/POSCAR"

# Construct the three-layer of MoS2/WSe2/MoS2
Bilayer(elem1, MoS2WSe2_file, la1=la1,
        position1=pos1).WritePOSCAR()
```

When a series of monolayer structure files from consistent crystal systems is provided, the directory path containing these files is used as the input parameter. Batch stacking of structures can be performed using the `GenBiLayer` class in the `batchStackBilayer` module. During this process, a threshold for lattice mismatch is applied to control the number of stacked bilayer structures. The lattice mismatch is defined as





**Fig. 2.** Generate input files and high-throughput job submission scripts, using VASP calculations as an example. (a) All structure files are stored in the *POSCAR\_dir* folder with material ID (*mid*) filenames. The **GenRunDir** class is employed to create the corresponding calculation directories and input files for each structure. If the *taskname* parameter is set to *None*, the system defaults to single-task calculations, and the job submission script *run.sh* is automatically generated. However, if *taskname* is set to a task-specific string, the *run.sh* script is not created; instead, input files are renamed with appropriate suffixes in preparation for multi-task calculations. The job submission script for multi-task calculations is generated using the **GenMultiSh** class. (b) The **GenBiLayer** class is utilized to stack pairs of structure files stored in the *POSCAR\_dir* folder, forming bilayer structures while adhering to the maximum lattice mismatch tolerance. These bilayer structure files are stored in nested directories. The *multilevel* parameter in the **GenRunDir** or **GenMultiSh** class controls the generation of input files and the *run.sh* script for this hierarchical storage format. (c) Basic templates for job submission scripts (*run.sh*) are presented, consisting of header and body parts. The header specifies details such as the number of nodes, cores, queue name, and the loading of first-principles software modules. The body outlines the computational workflow. When the *multilevel* parameter is set to 2 or *None*, the *run.sh* script includes commands to traverse the calculation directories once. In contrast, when the *multilevel* parameter is set to 4, the script performs three loops to traverse the directories. For multi-task calculations, additional commands to rename files are also incorporated into the *run.sh* script.

$2|lc_1 - lc_2| / (lc_1 + lc_2) \times 100\%$ , where *lc<sub>1</sub>* and *lc<sub>2</sub>* denote the lattice constants of the first and second layers, respectively. The bilayer structures generated through batch stacking will be stored in a nested directory hierarchy as follows: *BiPOSCAR\_dir*, {formula}\_{lattice mismatch}\_{number of atoms in the first layer}, {stacking pattern}, and {stacking sequence}, as shown in Fig. 2(b). The example implementation in Section 4 will demonstrate the required use of the **GenBiLayer** class.

### 3.3. Generating input files

Before performing first-principles calculations, it is crucial to prepare the necessary input files. For the widely used first-principles software VASP, four essential input files are required for basic property calculations: INCAR, the control file, which specifies key parameters such as calculation accuracy and convergence criteria; POSCAR, the structure file, containing the atomic coordinates of the unit cell; POTCAR, the pseudopotential file, which provides the pseudopotentials for each atomic species used in the calculation, with their order matching the element order in the POSCAR file; and KPOINTS, the k-point sampling file, which defines the Bloch vectors for Brillouin zone sampling during the calculation. Additionally, Plane Wave Density Functional Theory (PWFDT) [52,53] and the Quantum-ESPRESSO (QE) package [54] are also popular software for first-principles calculations. Unlike VASP, which separates structural and calculation parameters across multiple files, PWFDT and QE consolidate this information into a single *config.yaml* or *QE.in* file (with flexible file names, such as *pwsf.in* also being permissible).

When dealing with a large number of structures requiring first-principles property calculations, manually preparing the input files becomes highly inefficient. Therefore, in the *genInput* module, the *genInputfile* function of **GenRunDir** class allows for batch generation of input files based on the provided POSCAR files. By setting the *genmode* parameter, different generation modes can be invoked. As shown in Table 1, *genmode* offers several selectable values. When set to “basic”, “vaspkit”, or “pymg”, VASP input files are generated, with the latter two serving as interfaces for VASPKIT [55] and Pymatgen [56], respectively. Thus, the corresponding software packages must be installed when using these modes. Under the “basic” mode, the generated INCAR file contains user-defined parameters commonly used for calculating 2D materials. For different property calculations, specific parameter values in the INCAR file must be adjusted accordingly. For example, the *ISIF* parameter in INCAR is typically set to 3 for structural optimization and to 2 for self-consistent field (SCF) calculations. Similarly, for non-magnetic and magnetic calculations, the *ISPIN* parameter is usually set to 1 and 2, respectively. To facilitate this, an *inputOpt* parameter interface is provided for generating common INCAR templates. Moreover, the *user\_incar\_settings* parameter allows for flexible modification of INCAR parameters during specific calculations. When the *genmode* is set to “pawdft” or “qe”, the corresponding *config.yaml* and *QE.in* files are generated based on the POSCAR file. Listing 3 and Fig. 2(a) show the batch creation of calculation subfolders and corresponding input files for structure files stored in *POSCAR\_dir*, which will be saved in the *work* directory.

**Table 1**

Selectable values for the `genmode` parameter and their corresponding modules and software.

genmode content	Invoked module	Corresponding software
basic	vaspSetsWriter	Vienna Ab initio Simulation Package (VASP)
vaspkit	vaspVaspkitWriter	VASP using VASPKIT interface
pmg	vaspPmgWriter	VASP using Pymatgen interface
pwdft	pwdftSet	Plane Wave Density Functional Theory
qe	qeSets	Quantum-ESPRESSO

**Listing 3: Generate input files.**

```

from pyhtstack2d.tools.genInput import GenRunDir

# POSCAR file storage directory path
posdir="POSCAR_dir"
# Generate the SCF INCAR template
inputOpt = "scf"
# Modify INCAR template with user_incar_settings
user_incar_settings = {
    "NPAR": 4,
    "EDIFF": 1e-8,
    "LDIPOL": ".TRUE.",
    "IDIPOL": 3,
    "DIPOL": [0.5, 0.5, 0.5],
    "IVDW": 12,
}

# genInputfile function generate input files
GenRunDir(posdir="POSCAR_dir",
           genmode="basic", inputOpt=inputOpt,
           user_incar_settings=user_incar_settings
           ).genInputfile()

# Generation of HSE06 inputs via vaspkit interface
GenRunDir(genmode="vaspkit",
           kmeshrv=0.02,
           Hybridbandkpath=[251, 2, 0.03, 0.04]
           ).genInputfile()

# Generation of inputs via Pymatgen interface
# "MPStatic" helps create a VASP input set
# based on materials project data
# for static calculations
GenRunDir(genmode="pmg",
           workdir="work",
           inputOpt="MPStatic").genInputfile()

# For multi-task
# Add the suffix of INCAR and KPOINTS
taskname = "opt"
GenRunDir(posdir="BiPOSCAR_dir", multilevel=4,
           inputOpt="relax", taskname=taskname
           ).genInputfile()
taskname = "scf"
GenRunDir(posdir="BiPOSCAR_dir", multilevel=4,
           inputOpt="scf", taskname=taskname
           ).genInputfile()
taskname = "band"
GenRunDir(posdir="BiPOSCAR_dir", multilevel=4,
           inputOpt="band", taskname=taskname
           ).genInputfile()

```

In most property predictions, stepwise calculations are required, such as in electronic band structure calculations, which typically involve structure optimization, SCF calculation, and band structure calculation in sequence. We refer to this as multi-task computation. Correspondingly, performing only structure optimization is termed single-task computation. To achieve multi-task computation in a single job submission, the necessary input files for different tasks must be prepared in advance. For this purpose, the `taskname` parameter in the `GenRunDir` class is used to append suffixes to the generated input files. For example, in VASP calculations for band structure tasks, the suffixes “opt”, “scf”, and “band” are added to the respective INCAR files, resulting in INCAR-opt, INCAR-scf, and INCAR-band.

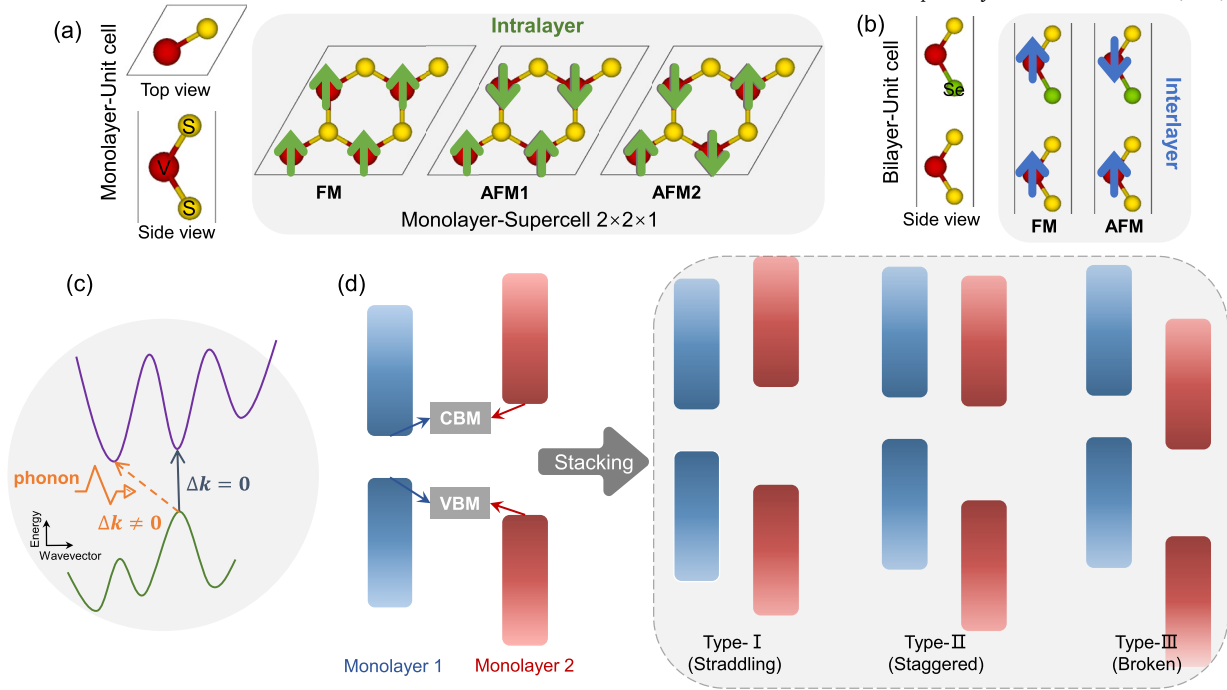
The current parameter configuration is primarily tailored for cases where all POSCAR files are stored in a single directory (e.g., POSCAR

files named by their chemical formulas within the `POSCAR_dir` folder). Typically, this setup corresponds to one structure per chemical formula. However, when considering bilayer structures formed through different stacking modes, multiple structure files may exist for the same chemical formula. As discussed in Section 3.2, bilayer structures generated using the PyHTStack2D package are stored in nested directories, with each bottom-level directory containing a single POSCAR file. In such cases, the bottom-level directory can be considered as the computation directory. It is important to note that the PyHTStack2D package enforces a consistent element order across POSCAR files for different stacking modes, enabling the reuse of input files (except for the POSCAR file) across all stacking modes of the same chemical formula in VASP calculations. To support the creation of input files for POSCAR files stored in nested directories, the `GenRunDir` class features a multi-level parameter. The value of this parameter is based on the number of directory levels. For instance, if the POSCAR file is stored in `Bi-POSCAR_dir/BNBN_0.00%_2/AA/cord1`, `multilevel=4`.

For magnetic systems, the most fundamental calculations involve determining the magnetic ground state (MGS). Due to the relative complexity of these calculations, which often involve supercell treatment and the identification of magnetic atoms, we developed the `vasp-MagSetsWriter` module specifically to generate the necessary input files. When performing calculations with VASP, it is important to set `ISPIN=2` and `LORBIT=11` to ensure that the OUTCAR file includes the magnetization of each atom along the x-direction, which can then be extracted to identify the magnetic atoms within the system. If only one magnetic atom is detected, the unit cell must be expanded to form a supercell, thereby increasing the number of magnetic atoms, as illustrated in Fig. 3 (a). By adjusting both the initial values and orientations of MAGMOM in the INCAR file, antiferromagnetic/ferrimagnetic calculations can be carried out. These procedures are implemented in the `MonoMagState` and `BiMagState` classes of the `vaspMagSetsWriter` module, with the `getEnergy` function being used to automatically compare the energies of ferromagnetic (FM) and antiferromagnetic (AFM)/ferrimagnetic (FIM) configurations, enabling the determination of the MGS, while a more detailed relevant elaboration is provided in the last part of this section. The key distinction between the `MonoMagState` and `BiMagState` classes is that the latter is tailored for bilayer structures, where the MGSs of each monolayer are typically required before stacking. This allows the MGS to be computed by focusing solely on interlayer coupling—either FM or AFM/FIM—as demonstrated in Fig. 3 (b), significantly reducing computational costs.

### 3.4. Creating submission scripts

When computational cluster resources are available, it is often advisable to use the Simple Linux Utility for Resource Management (SLURM) [57] or Portable Batch System (PBS) [58] for resource management and job scheduling. These systems execute computations by submitting BASH (shell) scripts through `sbatch` and `qsub`, respectively. Typically, the structure of job submission scripts is quite similar. To avoid the repetitive manual creation of these scripts, for single-task computations, the `genInputfile` function within the `GenRunDir` class generates both the input files and the `run.sh` file for job submission. For multi-task computations, the `GenMultiSh` class within the `gen-Input` module is required to generate the `run.sh` script. Job submission



**Fig. 3.** Magnetic order, bandgap type, and band alignment. (a) For a monolayer containing only one magnetic atom, such as in  $\text{VS}_2$  with V as the magnetic atom, calculation of the MGS requires cell expansion first; the figure illustrates a  $2 \times 2 \times 1$  supercell expansion, where each structure contains four V magnetic atoms. The intralayer magnetic order can exhibit FM, AFM1, and AFM2 configurations. (b) When stacking two FM monolayers, interlayer magnetic coupling can also be either FM or AFM/FIM. Green and blue arrows indicate magnetic directions. (c) Direct (blue-gray arrow) and indirect bandgaps (orange dashed arrow), with the former having  $\Delta k = 0$  (where  $k$  is the wave vector) and the latter requiring phonon participation during transitions as  $\Delta k \neq 0$ . (d) Following the stacking of two semiconductor monolayers to form a bilayer structure, three possible types of band alignment may occur: Type I (straddling gap), Type II (staggered gap), and Type III (broken gap). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

scripts generally consist of two main parts: the header, which specifies parameters such as node count, core count, and queue name, and usually requires customization based on the user's specific cluster configuration; and the body part, which defines the computational workflow. Fig. 2(c) presents several simplified *run.sh* template versions. For tasks where POSCAR files are stored in nested directories, the body part is modified by adding traversal loop commands for multiple directory levels. For multi-task computations, the script should rename input files by removing suffixes at each computational step, particularly for VASP calculations, as VASP does not recognize filenames with suffixes (e.g., renaming INCAR-opt by removing "-opt").

In multi-task computations, it is often necessary to rename output files to avoid overwriting, as different tasks may generate output files with identical names. Alternatively, a more straightforward approach is to create separate task directories and retain all associated files within them. Each method has its advantages: the former requires less storage space, while the latter preserves more comprehensive results. The choice between these two methods is controlled by the boolean parameter *saveall*. When this parameter is set to *True*, the *run.sh* script includes commands to create separate task folders within each bottom-level calculation directory.

It is important to note that both single-task and multi-task computations may encounter issues such as non-converging calculations or task interruptions. To handle such scenarios, the *run.sh* script includes default check commands, such as verifying whether the VASP output file (OUTCAR) contains the keywords "reached" and "Voluntary", whether the QE output file contains "JOB DONE", or whether a *statfile* is generated in PWDF calculations. The paths of structures requiring recalculation are recorded in the *failed.txt* file. Subsequently, the *failrw* function of the *GenRunDir* class, or the *genRunsh* function of the *MultTaskRW* class, can be used in conjunction with the *failed.txt* file to generate a new submission script, *run\_rw.sh*, for recalculating these failed tasks. The use case of these functions is illustrated in Listing 4.

During recalculations, it may be necessary to adjust the input parameters to improve convergence or slightly modify the initial material structure. In cases where only a minimal number of calculations fail, users may also consider excluding these cases to maintain consistency across all computations performed with uniform input parameters. Furthermore, *Custodian* [56] can be employed to manage computational tasks. As a powerful toolkit, it administers a series of jobs with an extensive list of error handlers, addressing nearly all common issues that may arise in VASP calculations. When encountering computational errors, *Custodian* automatically adjusts the INCAR parameters until the calculation is successfully completed or the predefined maximum number of correction attempts is reached. This approach requires only a simple modification to the job submission script, where *python custodian\_vasp.py* replaces the direct execution of *mpirun vasp\_std*. The corresponding script is available in the *pyhtstack2d/Script/handlebycustodian* directory.

Listing 4: Generate job submission scripts.

```
from pyhtstack2d.tools.genInput import GenMultiSh

# For multi-task calculation
tasklist=["opt", "scf", "band"]
# Generate run.sh for VASP calculations
# saveall=True: Create separate task folders
# e.g., add "mkdir opt" command to create folder
GenMultiSh(tasklist=tasklist,
            saveall=True).genRunsh()

from pyhtstack2d.tools.genInput import GenMultiQESh

# Generate run.sh for QE calculations
GenMultiQESh(tasklist=tasklist).genRunsh()

from pyhtstack2d.tools.genInput import GenRunDir
```

```
# run_rw.sh for recalculating failed tasks
# Generate run_rw.sh for single-task computation
GenRunDir().failrw()
# Generate run_rw.sh for nested directories
GenRunDir(multilevel=4).failrw()

from pyhtstack2d.tools.genInput import MulTaskRW

# The current path must contain failed.txt
# Generate run_rw.sh for multi-task computation
MulTaskRW(saveall=True).genRunsh()
```

### 3.5. Extracting and analysing results

After completing high-throughput calculations, a large volume of data is generated, making the automated extraction of essential information particularly important. The `GetResults` class of `extractResults` module systematically traverses all computational structures within the working directory to gather fundamental data. This data is stored in JSON format, using key-value pairs, which facilitates future reference and retrieval of material properties. The extracted data includes key information such as structural details, energy, and space group, among others. For bilayer materials, unlike monolayers, the formation energy is calculated using  $E_f = E_{\text{bilayer}} - E_{\text{ML1}} - E_{\text{ML2}}$ , where  $E_{\text{bilayer}}$ ,  $E_{\text{ML1}}$ , and  $E_{\text{ML2}}$  represent the energies of the bilayer and the two individual monolayers, respectively, which reflects thermodynamic stability to some extent. Furthermore, electronic structure information is routinely computed to determine characteristics such as metallicity, bandgap values, and the nature of the bandgap—whether direct or indirect. The type of bandgap is critically important in the field of photovoltaics; indirect bandgap semiconductors require the participation of phonons, leading to reduced efficiency. Consequently, direct bandgap materials are preferred in optoelectronic devices due to their higher operational efficiency. Fig. 3(c) illustrates the types of bandgaps identified. Another critical piece of information for 2D bilayer materials is band alignment. Structures formed by stacking two semiconductor layers can exhibit three distinct types of band alignment: Type I (straddling gap), Type II (staggered gap), and Type III (broken gap), as illustrated in Fig. 3(d). Type I alignment confines both electrons and holes within the same region, facilitating effective carrier recombination which is highly beneficial in optical devices. Type II alignment promotes effective separation of electrons and holes across different layers, offering potential applications in photocatalysis and solar cells. Type III alignment enhances band-to-band tunneling effects, making it advantageous for applications in tunnel field-effect transistors and photodetectors [31,41]. However, this information is not readily available in the raw computational results.

To address this, we determine the type of band alignment by comparing the contributions of atoms from different layers to the VBM and CBM. This method requires the VASP band structure calculation to include the PROCAR file, which contains the projection information of various atomic orbitals onto electronic states. This file can be generated by enabling the `LORBIT=11` setting in VASP. Additionally, the package automatically identifies and labels atoms from different layers based on structural information, eliminating the need for manual indexing of layer atoms. This enables the automatic calculation of layer contributions at each band point, allowing for the generation of layer-projected band structure plots, which provide a more intuitive and visual representation of the electronic structure.

Earlier, we discussed that for magnetic materials, determining the MGS is one of the most fundamental calculations. To support this, we specifically developed the `vaspMagSetsWriter` module, which automates the creation of input files and facilitates the extraction and comparison of energy data. It is important to highlight that the initially provided magnetic ordering configuration may change after optimization calculations. Therefore, it is necessary to re-identify the magnetic

atoms and their magnetic moments (both magnitude and direction) from the output files to adjust the actual magnetic configuration at this stage. Consequently, at least two rounds of magnetic moment extraction and analysis are conducted for each structure. The first round takes place after a spin-polarized calculation without manually presetting the magnetic ordering configuration, which allows for the identification of magnetic atoms in the system and provides the basis for setting the initial MAGMOM values. In the case of bilayer systems, if the MGS of individual layers is provided in advance, this information can be used to significantly reduce the number of magnetic configurations to be considered, limiting the focus to interlayer magnetic coupling. The second round occurs after performing calculations on different magnetic ordering configurations for the same structure. By comparing the energies, the magnetic moments of the lowest-energy configuration and the magnetic moments of each atom in that configuration are extracted and analyzed, leading to the determination of the true MGS obtained through first-principles convergence. These processes have been implemented in the PyHTStack package, which automates the entire workflow for MGS calculations. The results of the MGS calculations are also stored in JSON format for easy retrieval and further analysis. In addition to identifying the MGS of each material, this module integrates the magnetic properties of bilayer structures with different stacking modes under the same chemical formula, allowing for the rapid identification of bilayer materials that exhibit magnetic tunability through rotation or sliding.

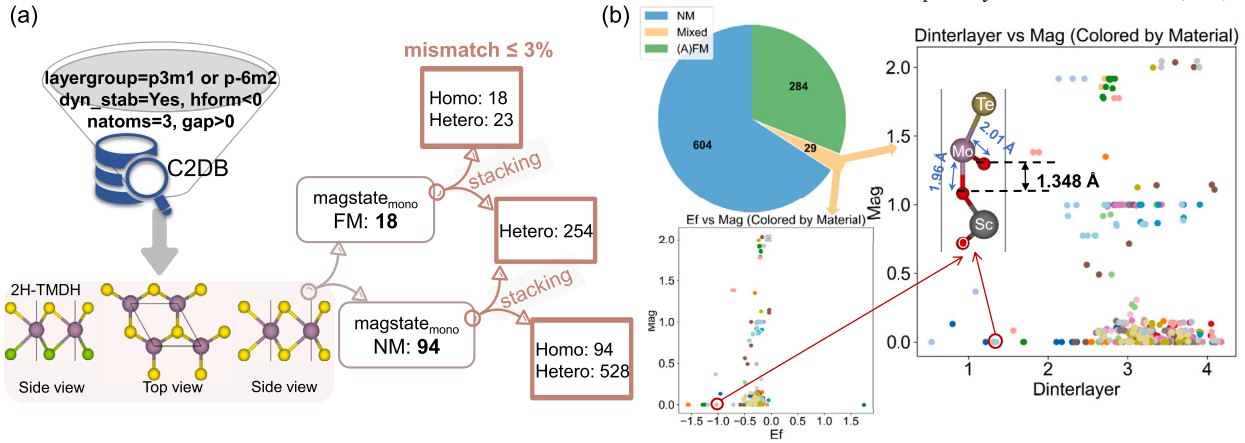
Automated batch result extraction significantly reduces manual effort and accelerates the identification of promising materials. In the next section, we will present practical examples of using the PyHTStack2D package to showcase its capabilities in a more intuitive and comprehensive way.

## 4. Illustrative examples

In this section, we use the PyHTStack2D package to stack 2H phase TMDH monolayers selected from the C2DB database to form bilayer structures, performing high-throughput calculations on the basic properties of these homojunctions and heterojunctions. The code required to execute the bilayer stacking process is located in `pyhtstack2d/Example2`. The bilayer stacking process for 2H-phase TMDH monolayers will exclusively use the `TMDHBilayer` module. For other 2D systems, such as graphene, the execution code can follow the example in `pyhtstack2d/Example2`, with the corresponding stacking modules adjusted accordingly. A detailed list of material systems and their associated modules can be found in Sections 2 and 3.2.

As illustrated in Fig. 4(a), we access the C2DB database through functions in the `queryDB` module, selecting monolayers with 3 atoms, dynamic and thermodynamic stability, layer groups 'P3m1' or 'P-6m2,' and a bandgap greater than zero. This screening identifies stable 2H-TMDH semiconducting monolayers, including 18 magnetic and 94 non-magnetic types. Stacking these monolayers yields 112 homo-bilayers; for heterojunctions, applying a lattice mismatch limit of less than 3% results in 23 magnetic heterobilayers and 528 non-magnetic ones. Additionally, combinations of magnetic and non-magnetic monolayers yield 254 heterojunction bilayers. Stacking layers with distinct magnetic properties is crucial for research in spintronics and valleytronics, such as controlling valley excitons through magnetic proximity effects in  $\text{CrI}_3/\text{WSe}_2$  heterostructures [59], and holds promise for creating magnetic tunnel junctions [60]. To comprehensively explore these combinations, we use the `batchStackBilayer` module to generate bilayer structures with high-symmetry stacking patterns across 917 component pairs. Simultaneously, the `genInput` module's `GenRunDir` class prepares input files and batch job scripts for over 10,000 structures. Structural optimizations are performed using the VASP first-principles calculation software with spin polarization enabled (`ISPIN=2`) to observe potential magnetic transitions induced by stacking, with detailed computational parameters provided in Section 4.1. The PyHTStack2D package extracts and saves all property calculation results as JSON files,





**Fig. 4.** High-throughput computation. (a) Workflow involves the selection of stable 2H phase TMDH semiconducting monolayers from the C2DB database, comprising 94 non-magnetic (NM) and 18 ferromagnetic (FM) monolayers. Considering a lattice mismatch of less than 3%, 112 homojunctions and 805 heterojunctions components are obtained. High-throughput stacking of these components is conducted under various stacking modes. (b) Structural optimizations of 917 bilayer structures display no significant magnetism under any high-symmetry stacking configurations, 284 structures exhibit magnetic behavior across all configurations, and 29 materials demonstrate variable magnetic properties depending on the stacking arrangement. By comparing magnetic moments with interlayer distances or formation energies, some structurally implausible configurations are rapidly identified and excluded (e.g., red circles highlight data related to the MoTeO/ScO<sub>2</sub> heterostructure in the AB stacking pattern, where the black double arrow represents the interlayer distance, and the blue double arrows represent the Mo-O bond lengths), with the anomalous properties attributed to unreasonable structures. The different colors indicate different stacking components.

which, along with their analysis and visualizations, are stored in *pyht-stack2d/Example3*.

When high-throughput calculations are performed for functional material design, computational resources and memory constraints are unavoidable. System size, structural complexity, first-principles software, and output file size all affect feasibility. Based on our studies, we recommend starting with low-cost functionals (e.g., LDA, PBE) for large or complex structures to minimize computational overhead. Subsequently, a systematic selection strategy can be employed to identify promising candidates for high-cost, high-accuracy calculations. Managing large output files is crucial to avoiding excessive memory use, and symbolic links for referencing duplicate files across different computational tasks can also be an effective solution. When dealing with complex computational systems that require high accuracy, it is crucial to select appropriate first-principles software carefully. High-performance first-principles software incorporating acceleration algorithms, such as PWDFT [53], can be considered. Another promising approach to reducing the computational cost of large-scale high-throughput calculations is leveraging machine learning for 2D material property prediction [61,62]. Our team is developing deep learning-based 2D potential energy models to efficiently evaluate tens of thousands of 2D homo/heterostructures. In the future, we plan to integrate these models with the PyHTStack2D package to further enhance research efficiency and support related studies.

#### 4.1. Computational details

First-principles high-throughput calculations are performed using density functional theory (DFT) in the Vienna Ab initio Simulation Package (VASP) [45], with the projector augmented wave (PAW) method [63] and the Perdew-Burke-Ernzerhof generalized gradient approximation (GGA-PBE) [64]. The plane-wave cutoff energy is set to 500 eV, and the convergence criteria for energy and force are  $10^{-6}$  eV and 0.01 eV/Å, respectively. A  $13 \times 13 \times 1$  k-point mesh is used. The vacuum along the z-direction is 30 Å for bilayers, and the DFT-D3 method [65] describes the van der Waals (vdW) interactions. For certain magnetic materials containing transition metal elements, we tested the effect of a Hubbard U correction with a value of 4.0 eV on the calculation of their properties.

#### 4.2. Fundamental properties

After structural optimization, the properties of the systems were initially assessed by analyzing the magnetic moments, with the results displayed in Fig. 4(b). The results of these calculations can be extracted using the *GetResults* class in the *extractResults* module to obtain the *info.json* file. See Listing 5 for the code illustrating how to execute this extraction, where the parameter *mag* is set to enable the retrieval of magnetic moment information. In structures formed by stacking purely non-magnetic monolayers, 604 retained their non-magnetic characteristics across all stacking configurations. Conversely, 283 structures composed either entirely of magnetic monolayers or of combinations of magnetic and non-magnetic monolayers exhibited significant magnetic moments. Additionally, 29 stacking combinations demonstrated mixed properties. Further investigations into these combinations analyzed the relationships between magnetic moments and either formation energies or interlayer distances across various stacking configurations, with the interlayer distance defined as the vertical separation between the bottom atom of the upper layer and the top atom of the lower layer. It was observed that certain bilayer structures presented anomalies, such as excessively reduced interlayer distances and abnormally high or low formation energies, which often diverged from the properties exhibited under most stacking modes. These anomalies are attributed primarily to structural implausibilities rather than to the stacking itself. For instance, the MoTeO/ScO<sub>2</sub> heterostructure exhibits non-magnetic behavior exclusively in the AB stacking pattern, with the sequence Te-Mo-O-O-Mo-O. In contrast, all other high-symmetry stacking modes considered display clear magnetic properties. In most stacking modes, the optimized interlayer distance is around 2.5 Å, with formation energies close to -0.3 eV. However, the nonmagnetic AB-stacked structure shows a significant deviation, with an interlayer distance of only 1.35 Å and a formation energy of -1.20 eV. Additionally, in this structure, the Mo-O bond length in the upper layer is 2.01 Å, while the bond length between the Mo atom in the upper layer and the O atom in the lower layer is only 1.96 Å, indicating that this is clearly not a well-formed bilayer heterostructure. Consequently, implausible models can be swiftly ruled out by simply examining the interlayer distances and formation energies, allowing them to be excluded from further property calculations, thereby reducing computational costs.

Listing 5: Extraction and analysis of results.

```

from pyhtstack2d.analysis.\
    extractResults import GetResults

# Extract total magnetic moment
# and optimized structural information
GetResults(mag=True)
# Calculations in a multilevel directory structure
GetResults(mag=True, multilevel=4)
# Calculations stored in the task folder
GetResults(scf="opt", mag=True, multilevel=4)
# Bilayer system band information extraction
# Set nlayer=2 and specify band="band" or ""
# Prior information on monolayers required
# monoinfo.json: the information of monolayer
GetResults(band="band", nlayer=2, multilevel=4).\
    getInfoBi(monodict="monoinfo.json")

from pyhtstack2d.tools.\
    genInput import GetMagEntropy

# Separate computational paths
# based on different magnetic properties
GetMagEntropy().separateMag(magtol=0.1)
# In task and multilevel directories
GetMagEntropy(magpath="opt", multilevel=4).\
    .separateMag(magtol=0.1)

```

Due to the possibility of both magnetic and non-magnetic stacking configurations existing within bilayer structures of the same chemical composition, it is necessary to distinguish between these configurations before proceeding with further property calculations. This analysis can be conducted using the `separateMag` function of the `GetMagEntropy` class within the `genInput` module, where the total magnetic moment is compared against the `magtol` parameter to check if it exceeds a specified threshold, thus classifying the material as magnetic. The relevant execution code is also shown in the last two commands in Listing 5. Executing this process will generate two files, `maglist.txt` and `nomaglist.txt`, containing the paths for magnetic and non-magnetic materials, respectively. Since this setup does not require traversing all folders, the job submission scripts must be adjusted to iterate only through paths specified in the designated TXT files. See `PyHTStack2D/Script/runScript` for the relevant scripts. It is worth noting that in our example, the separation and slight script adjustments simulate a scenario where the magnetic properties of the system are unknown. For magnetic materials, the MGS must first be determined, whereas for non-magnetic materials, spin-polarized calculations can be omitted, thus making this separation beneficial. However, if the calculation steps are consistent across all systems, this step can be skipped, and the scripts generated by the `GenMultiSh` class can be run directly.

#### 4.3. Electronic properties

Constructing bilayer structures is an effective approach for modulating the electronic properties of materials, enabling adjustments such as tuning bandgap size, achieving indirect-to-direct bandgap transitions, and establishing various band alignments tailored to specific applications. For materials exhibiting negligible magnetic moments after structural optimization, spin-polarized calculations are omitted to enhance the efficiency of electronic band structure computations. Following the extraction of relevant band information, we analyze and categorize bandgap types and band alignments. These insights provide a foundation for exploring potential applications in optoelectronics, semiconductor devices, and related fields. It is important to emphasize that for accurate information extraction in bilayer structures, prior knowledge of the monolayer's energy and its metallic or non-metallic nature is essential. For instance, if a bilayer structure exhibits metallic characteristics, it can only be classified as a Type III band alignment if the original monolayer was a semiconductor; otherwise, the metallic nature may stem from the monolayer itself, precluding the formation of a broken gap. The specific implementation involves setting the parameter `nlayer` in

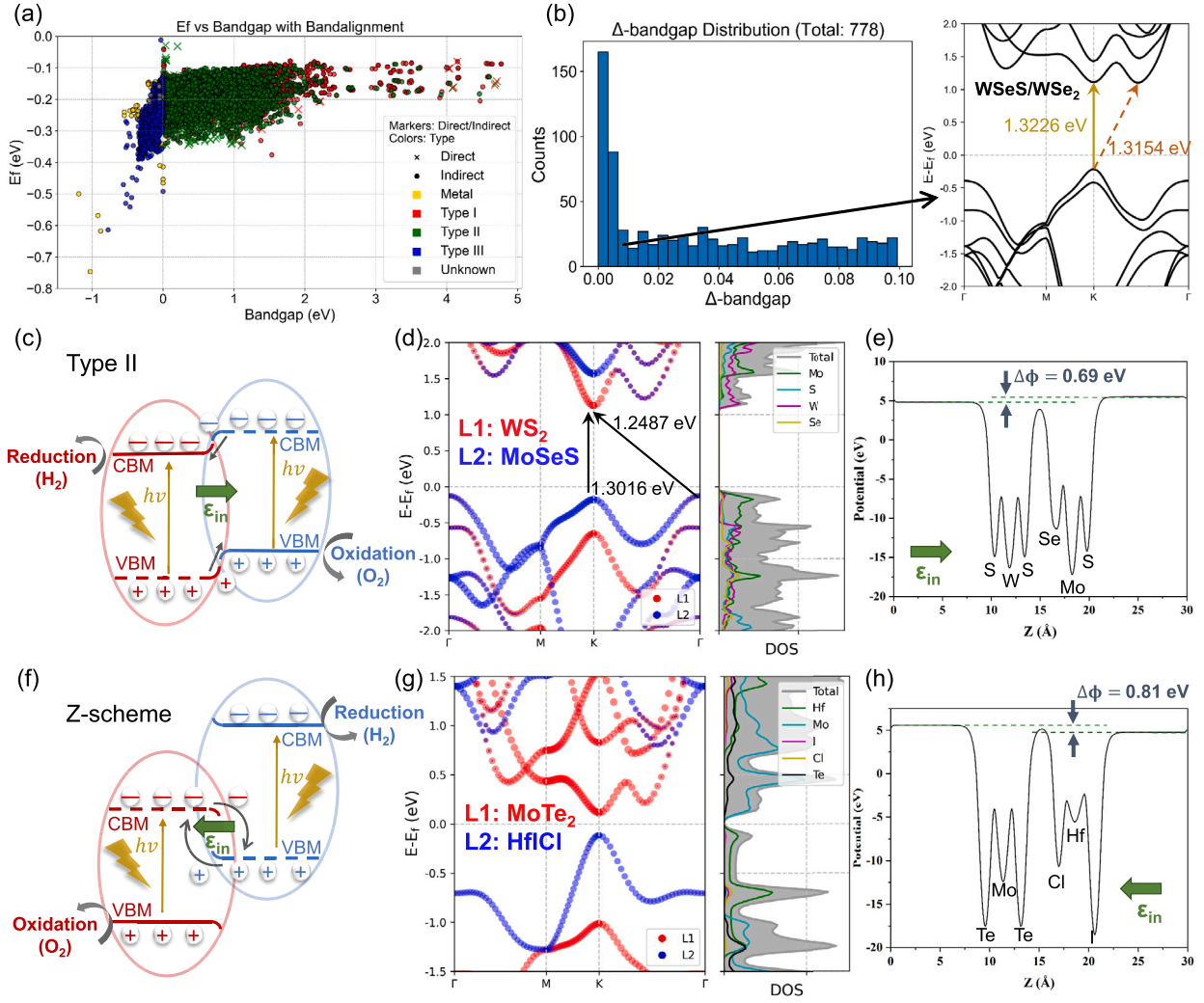
the `GetResults` class to 2, which enables information extraction for bilayer materials. Additionally, calling the `getInfoBi` function calculates the formation energy and classifies the band alignment, with monolayer information provided via the `monodict` parameter. Fig. 5 illustrates the electronic properties of bilayer structures formed through high-symmetry stackings. Fig. 5(a) shows that the majority of these materials exhibit an indirect Type II band alignment. However, over seven hundred structures display a bandgap difference of less than 0.1 eV between indirect and direct configurations, as depicted in Fig. 5(b). Due to this minimal bandgap discrepancy, these materials can effectively be considered near-direct bandgap materials, making them suitable candidates for optoelectronic applications where direct bandgap properties are desirable.

Type II band alignment is highly advantageous in photocatalysis due to its support for effective charge separation. Based on carrier migration mechanisms, Type II alignment can be further classified into conventional Type II and Z-scheme types. In conventional Type II alignment, photogenerated carriers transfer between the band edges, with oxidation and reduction reactions occurring at the heterojunction's VBM and CBM, respectively, as shown in Fig. 5(c). When the direction of the built-in electric field is such that the layer primarily contributing to the CBM points toward the layer contributing primarily to the VBM, carrier recombination is suppressed, enhancing catalytic performance. Figs. 5(d) and (e) display the layer-projected band structure and electrostatic potential of the  $\text{WS}_2/\text{MoSeS}$  heterostructure, which exhibits Type II characteristics. Ma et al. [66] conducted a comprehensive study on this  $\text{WS}_2/\text{MoSeS}$  heterostructure for photocatalytic overall water splitting. Their work included carrier migration analysis via nonadiabatic molecular dynamics (NAMD) simulations, which confirmed that its carrier migration mechanism is consistent with the Type II band alignment. However, due to the specific sites where catalytic reactions occur, Type II alignment generally imposes stricter requirements on the bandgap. In contrast, as illustrated in Fig. 5(f), Z-scheme systems favor recombination of photogenerated holes at the VBM and electrons at the CBM. Consequently, the built-in electric field in Z-scheme structures typically has an orientation opposite to that of conventional Type II, with catalytic reactions occurring at higher reduction potentials and lower oxidation potentials. This configuration has the potential to overcome bandgap limitations, such as the 1.23 eV requirement for overall water splitting. In our previous work [67], we explored design strategies for Z-scheme heterojunctions and conducted an in-depth analysis of the heterojunction formed by  $\text{HfBrCl}/\text{MoS}_2$ . In the present study, as shown in Figs. 5(g) and (h), the  $\text{MoTe}_2/\text{HfCl}$  heterojunction exhibits a relatively small direct bandgap, and the orientation of its built-in electric field indicates potential suitability for Z-scheme.

When conducting similar studies on electronic properties, we recommend several preliminary screening criteria for bilayer stacked structures with potential applications. For optoelectronic applications, materials with a direct bandgap or a minimal difference (less than 0.1 eV, or even smaller) between direct and indirect bandgaps are recommended as effective high-throughput screening conditions. For photocatalytic overall water splitting, in addition to considering the orientation of the built-in electric field, conventional Type II structures with a slightly larger PBE-calculated bandgap (e.g., above 0.8 eV, due to PBE's tendency to underestimate bandgaps) are preferable. In Z-scheme systems, a smaller, direct bandgap is ideal. These criteria can streamline the identification of promising candidate materials, facilitating optimized performance in targeted applications.

#### 4.4. Magnetic properties

For magnetic systems, determining the MGS is one of the most fundamental property calculations. Since, in most cases, the interlayer interactions in vdW layered structures have minimal impact on the intrinsic properties of monolayers, it is recommended to first determine the MGS of each monolayer. If both monolayers exhibit FM ground states,



**Fig. 5.** Electronic properties. (a) The bandgap and band alignment types are displayed, with bandgap values on the x-axis and formation energy on the y-axis. Distinct markers denote direct or indirect bandgaps, while colors differentiate types of band alignment. (b) Counts of near-direct bandgaps, where the  $\Delta$ -bandgap represents the energy difference between direct and indirect bandgaps. WSeS/WSe<sub>2</sub> with AA2-2 stacking pattern and a Se-W-S-Se-W-Se sequence exhibits near-direct characteristics, with a minor  $\Delta$ -bandgap of 0.0073 eV. (c) and (f) Schematic representations of band alignments and the dominant migration mechanisms for photogenerated charge carriers in type II and Z-scheme heterostructures, where each structure shows a built-in electric field oriented in opposite directions. (d) and (g) Layer-resolved band structures of WS<sub>2</sub>/MoSeS and MoTe<sub>2</sub>/HfCl, with the former presenting a larger bandgap and the latter a smaller direct bandgap. (e) and (h) Electrostatic potential profiles with arrows indicating the built-in electric field directions. For the WS<sub>2</sub>/MoSeS heterostructure, the electric field points from WS<sub>2</sub> to MoSeS, while for the MoTe<sub>2</sub>/HfCl heterostructure, the field directs from HfCl towards MoTe<sub>2</sub>.

only two interlayer coupling configurations need to be considered after stacking: FM or AFM/FIM, thereby effectively reducing the number of magnetic order configurations to examine. In the PyHTStack2D package, the `genInputfile` function of `MonoMagState` and `BiMagState` classes in the `vaspMagSetsWriter` module are used to implement various magnetic configurations; the latter is specifically designed for bilayer vdWs structures and requires prior information on the monolayer MGSs, provided in the file `monomagdict.json`. In addition, calling the `genInputfile` function requires providing the default *INCAR-basic* file, which includes basic parameter settings—such as smearing, cutoff energy, and other parameters—applicable to all systems requiring calculation, except for the MAGMOM and U value parameters.

**Listing 6:** Magnetic ground state calculation.

```
from pyhtstack2d.calcSets.\
    vaspMagSetsWriter import MonoMagState
```

```
# Spin-polarized calculations have been performed
MonoMagState().genInputfile()
```

```
# Specify magnetic atoms and U values
magatom = ['V', 'Fe', 'Sc', 'Y']
u = {'V': [2, 3.1, 1],
     'Cr': [2, 3.5, 1]}
MonoMagState(magatomlist=magatom,
             skipmagnetfile=True,
             u_setting=u).genInputfile()

# For bilayer
# 'monomagdict.json' must have been provided
BiMagState().genInputfile()

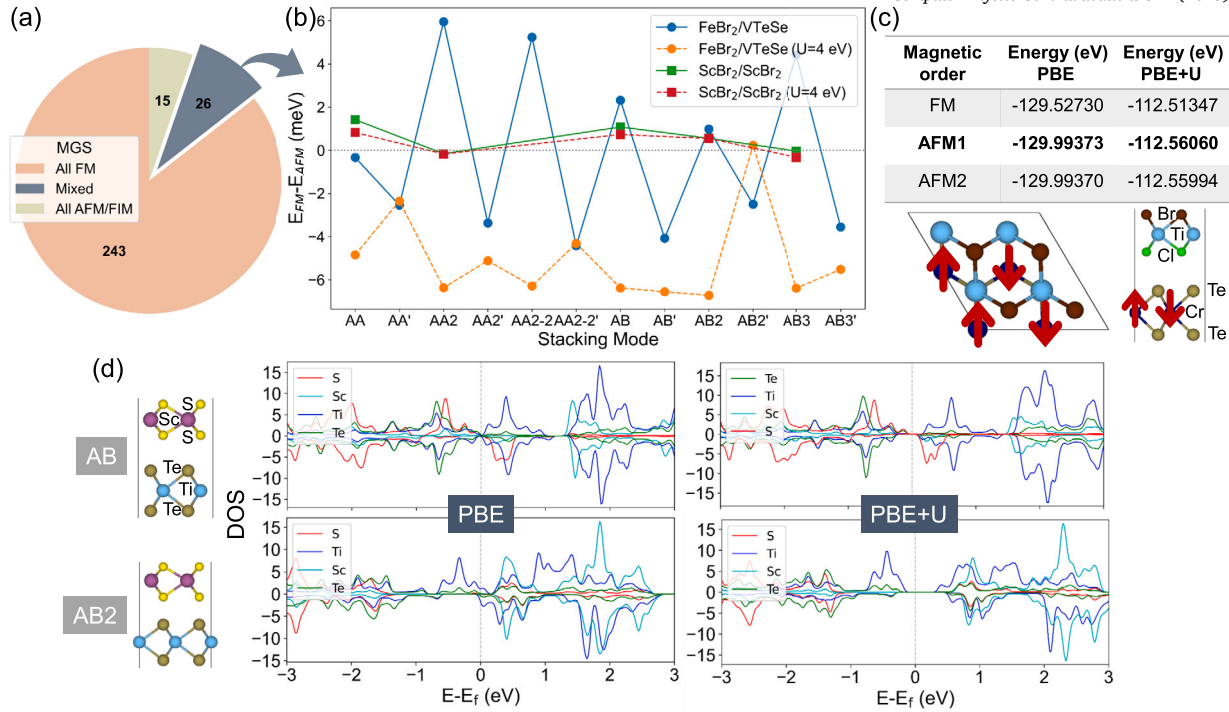
# Turn off U-value correction
MonoMagState(pU=False).genInputfile()
BiMagState(pU=False).genInputfile()

# Extract energy and determine MGS
MonoMagState().getEnergy()
BiMagState().getEnergy()
```

```
from pyhtstack2d.calcSets.\
    vaspINCARupdate import UpdateINCAR
```

```
# Update INCAR parameters
```





**Fig. 6.** Magnetic properties. (a) All involved monolayers exhibit a FM ground state, and their bilayer structures display various interlayer magnetic ordering. Among the bilayers, 15 and 246 structures exhibit AFM/FIM and FM coupling, respectively, across all high-symmetry stacking configurations considered. The remaining 26 bilayers show stacking-dependent interlayer magnetic ordering. (b) The homobilayer ScBr<sub>2</sub> and the heterostructure FeBr<sub>2</sub>/VTeSe demonstrate different MGSs across stacking modes. In pure PBE and PBE+U calculations, the magnetic ground state of ScBr<sub>2</sub> remains consistent, while FeBr<sub>2</sub>/VTeSe shows significant variation. For ScBr<sub>2</sub>, data are based on both pure PBE and PBE+U structure optimizations, whereas FeBr<sub>2</sub>/VTeSe, which encountered convergence issues with  $U = 4.0$  eV, used a pure PBE structure optimization followed by a PBE+U self-consistent calculation. (c) The bilayer heterostructure formed by stacking non-magnetic monolayers TiBrCl and CrTe<sub>2</sub> in a Br-Ti-Cl-Te-Cr-Te sequence induces Cr magnetization, and both PBE and PBE+U ( $U = 4.0$  eV) calculations reveal an AFM magnetic ground state. (d) In the heterostructure composed of magnetic ScS<sub>2</sub> and non-magnetic TiTe<sub>2</sub>, the AB stacking pattern shows magnetism primarily contributed by the S atoms, whereas in the AB2 stacking, magnetism shifts primarily to the Ti atoms. The density of states (DOS) under both pure PBE and  $U = 4.0$  eV corrections reveals this shift, likely related to the reduced interlayer distance in the AB2 stacking pattern.

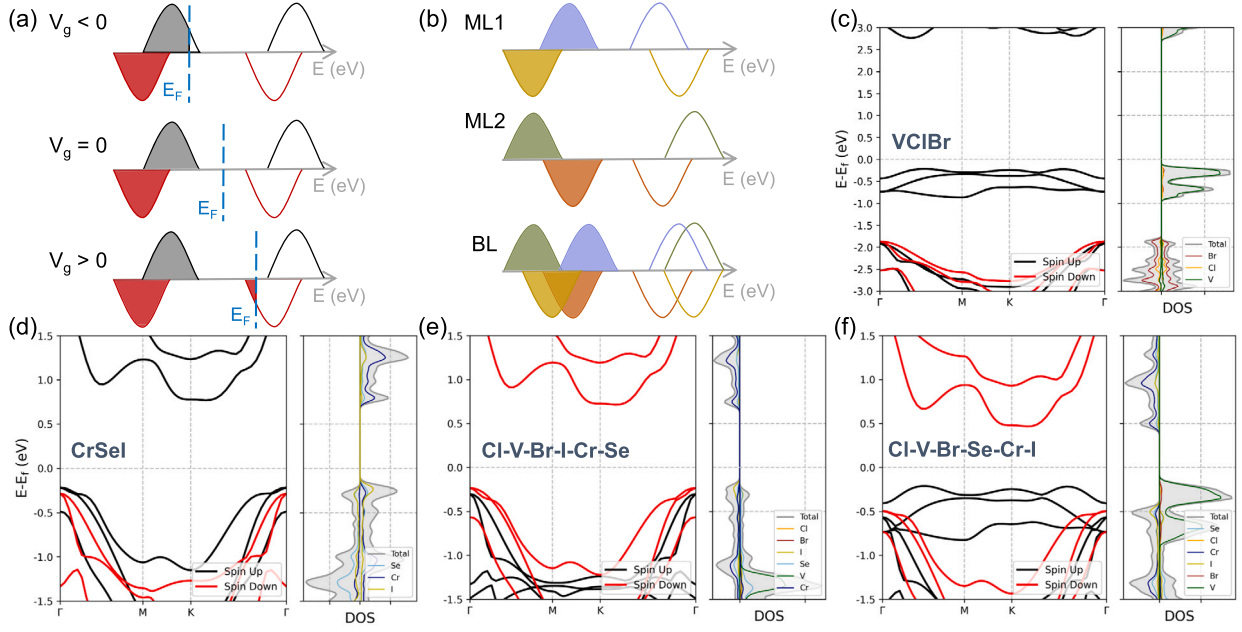
```
# Provide a new INCAR parameter file and
# the paths for magnetic materials
incar = 'INCAR-basic-new'
path = 'maglist.txt'
UpdateINCAR(materiallist=path,
             incarcbasic=incar).updateIncarfiles()
```

There are two methods available for batch creation of magnetic configuration directories. The first method requires spin-polarized calculations and ensures that output files include the (projected) atomic magnetic moments, which are then extracted by the package to identify magnetic atoms in each system and generate input files for different magnetic configurations. The second method allows the user to directly specify a list of magnetic atoms, where any elements matching this list are treated as magnetic, with non-zero initial magnetic moments assigned to these atoms. Each method has its advantages and disadvantages: the former requires an initial batch calculation but provides accurate magnetic atom identification for each system, while the latter bypasses this step but may lead to inaccuracies in identifying magnetic atoms. For instance, hexagonal monolayer CrTe<sub>2</sub> is non-magnetic, but Cr atoms are magnetic in many other systems. The execution code for these two methods can be found in Listing 6, where the parameter `skipmagnetfile` controls whether to skip the magnetic atom identification step. After running this process, each structure will contain FM and AFM\* configurations, where \* indicates the possibility of multiple distinct AFM or FIM orders. For identical magnetic atoms, antiparallel magnetic moments form AFM configurations, whereas non-identical magnetic atoms often produce FIM configurations due to partial cancellation of differing magnetic moment magnitudes. After computing these considered configurations, the `getEnergy` function can be used for automated energy extraction and comparison to determine the MGS order.

The results of the illustrative example are shown in Fig. 6(a), where the MGSs of 26 bilayer structures are found to depend on the stacking mode. We must acknowledge that these results were obtained using the PBE functional, which lacks precision for magnetic systems; thus, a  $U$  correction or hybrid functional would be more appropriate for accurate calculations. More specifically, due to the inability of the PBE functional to fully account for the strong correlation effects between localized electrons, it typically underestimates the local magnetic moments in magnetic materials and struggles to accurately describe the magnetic behavior of transition metals or rare-earth elements [64,68]. The DFT+ $U$  method addresses this limitation by introducing the Hubbard  $U$  correction term, thereby significantly improving the description of strongly correlated electron systems [69–71]. Additionally, hybrid functionals, which incorporate a portion of Hartree-Fock exchange energy, enhance computational accuracy for magnetic materials by effectively mixing exact exchange energies [72,73]. However, our primary goal here is to test the package's capabilities in generating input files and extracting results rather than conducting an in-depth material exploration. Additionally, it is important to note that the package defaults to including  $U$  corrections, but during testing, we observed convergence issues for certain materials and therefore opted to use the simpler PBE functional for preliminary MGS calculations.

The setting of  $U$  values is a complex issue. Although the package applies a default  $U$  correction for transition metals, elements within the same period are assigned a common  $U$  value by default. For example, both V and Cr with “3d” orbitals are assigned a  $U$  value of 4.0 eV, which may not be suitable for all systems. Typically, in materials exploration, more accurate property calculations require  $U$  values that are either experimentally validated or determined based on linear re-





**Fig. 7.** Bipolar magnetic semiconductors (BMS). (a) The Fermi level position is adjusted by gate voltage to control the spin direction of charge carriers. The schematic DOS is shown for negative, zero, and positive gate voltages from top to bottom. Black and red represent opposite spin orientations. (b) ML1 and ML2 represent the DOS of FM monolayers, which are coupled through AFM stacking to form a bilayer (BL) structure, potentially exhibiting BMS properties. (c) and (d) Display the band structures and DOS of VCIBr and CrSeI monolayers, both in a FM ground state. (e) and (f) Present the band structures and DOS of bilayer heterostructures under different stacking sequences with AFM interlayer coupling. The electronic properties of the heterostructures, when the Se atom is located at the interlayer interface, show the characteristics required for BMS.

sponse methods. Therefore, we provide the `u_setting` parameter to allow specific U values to be assigned for particular elements as needed. Fig. 6(b) shows the magnetic ground state results obtained with both pure PBE and PBE+U ( $U = 4.0$  eV) calculations. By default, the package accurately identifies V, Fe, and Sc as magnetic atoms and applies the U correction accordingly. For the homo-bilayer  $\text{ScBr}_2$ , the magnetic ground state remains consistent between PBE and PBE+U calculations. However, the heterostructure  $\text{FeBr}_2/\text{VTeSe}$  exhibits notable differences. In fact, the  $\text{FeBr}_2/\text{VTeSe}$  heterobilayer encountered convergence issues during structural optimization with  $U = 4.0$  eV. As a result, we used a pure PBE-optimized structure, followed by a self-consistent PBE+U calculation to determine the energies of different magnetic configurations. This approach is not merely a workaround; in some studies [30], comparisons between pure PBE, PBE+U, and experimental data indicate that lattice constants may be less accurate with U corrections. Consequently, structural optimizations are sometimes performed with pure PBE, while PBE+U is applied for subsequent property calculations. To implement this approach, the `pU` parameter can be disabled to generate input files without U corrections. Additionally, if a self-consistent calculation with U correction is desired following a pure PBE structural optimization, the `UpdateINCAR` class in the `vaspINCARupdate` module allows for modification of parameters in an existing INCAR file without affecting other input files, thereby avoiding the need to recreate them entirely.

In our computational results, two phenomena have captured our attention. First, non-magnetic monolayers, upon stacking, can induce magnetization in one of the layers. As illustrated in Fig. 6(c), the hexagonal monolayer  $\text{CrTe}_2$ , which is intrinsically non-magnetic, exhibits magnetic behavior when stacked with the non-magnetic monolayer  $\text{TiBrCl}$ . Specifically, the magnetic moments of the Cr atoms are measured at  $1.25 \mu_B$  and  $3.41 \mu_B$  under PBE and U with 4.0 eV corrections, respectively. This phenomenon may be attributed to the relatively large lattice mismatch of approximately 2.65%. Second, when magnetic and non-magnetic monolayers are stacked, the primary magnetic contributors may shift depending on the stacking pattern. As shown in Fig. 6(d), the bilayer structure formed by stacking the magnetic monolayer  $\text{ScS}_2$  with the non-magnetic monolayer  $\text{TiTe}_2$  retains similarities to the in-

trinsic properties of the monolayers in the AB stacking pattern, with magnetism primarily arising from the S atoms. However, in the AB2 stacking pattern, the primary magnetic contribution shifts to the Ti atoms. The interlayer distance in the former case is  $3.13 \text{ \AA}$ , while in the latter it is only  $2.54 \text{ \AA}$ , suggesting that the reduction in interlayer spacing may lead to more significant electronic transfer, resulting in further changes in material properties. Although these particular phenomena are not commonly observed in our computational results, more careful investigations may reveal new insights. Such studies could enhance our understanding of the potential modulation of material magnetism and electronic structure through interlayer coupling.

Here, we introduce an application of magnetic vdWs materials in forming BMS. BMS materials are proposed to enable electric field control over the spin orientation of charge carriers [43]. Their intrinsic electronic properties exhibit opposite spin polarization at the VBM and CBM, allowing the Fermi level to be tuned with a gate voltage ( $V_g$ ), thus providing electrons with different spin orientations, as illustrated in Fig. 7(a). A feasible approach to constructing BMS is through heterostructure engineering, as shown in Fig. 7(b). This method involves stacking two half semiconductors (HSCs) — where the VBM and CBM belong to the same spin channel — into a bilayer material via AFM interlayer coupling [44]. This configuration could allow the CBM and VBM to reside in opposite spin channels, satisfying the electronic properties required for BMS. Deng et al. demonstrated this concept in  $\text{CrI}_3/\text{VI}_3$  under an  $\text{AC}_1$  stacking mode, achieving BMS. Given the limited availability of ideal HSCs in the 2H phase TMDH monolayers, where lattice mismatch among suitable layers exceeds 5%, we selected 1T phase monolayers VCIBr and CrSeI as examples, with a minimal lattice mismatch of 0.79%. The band structures of these monolayers are shown in Figs. 7(c) and (d), where both the VBM and CBM are in the same spin channel. Considering AFM interlayer coupling with an AA stacking pattern and four distinct stacking sequences, we find that if CrSeI is stacked with I atoms at the interlayer interface, it maintains HSC characteristics. Fig. 7(e) shows the band structure for the Cl-V-Br-I-Cr-Se stacking sequence. Alternatively, if Se atom is positioned at the interface, BMS properties emerge, as shown in Fig. 7(f) for the Cl-V-Br-Se-Cr-I stacking sequence. The

band structures for the Br-V-Cl-I-Cr-Se and Br-V-Cl-Se-Cr-I sequences are similar to those in Figs. 7(e) and (f), with minor variations in specific values but showing BMS characteristics only in the latter case. These band structures are calculated using PBE+U with  $U = 4.0$  eV. The pure PBE calculation also exhibits BMS characteristics when Se atom is at the interlayer interface. This finding suggests that the strategy is not only feasible but that exploring different stacking modes and sequences may broaden the search space for BMS materials. Regrettably, our calculations indicate that the MGS of the heterostructure composed of randomly selected VClBr/CrSeI layers exhibits greater stability under FM interlayer coupling. Despite this, we maintain optimism that through the application of these methodologies to high-throughput calculations of bilayer stacked structures, it is feasible to discover both ideal and stable BMS materials.

## 5. Conclusions and outlook

As research into 2D vdWs layered materials continues to deepen across various fields, the exploration of high-symmetry stacked structures in construction engineering has increasingly become a focal point. To facilitate the efficient stacking of diverse types of 2D materials, we have developed a package named PyHTStack2D, specifically designed to support this complex stacking process. Additionally, to meet the expanding needs of scientific research, we have integrated features that support high-throughput job submissions and batch extraction of results, significantly enhancing the efficiency of high-throughput computational simulations for material stacking. Through high-throughput stacking calculations on monolayers of 2H-TMDH, we have validated the applicability of the PyHTStack2D package and demonstrated the potential applications of these materials in fields such as photocatalysis and spintronic devices.

Further in-depth research facilitated by the PyHTStack2D tool will aid in the better design of these layered materials, markedly improving the efficiency of research efforts. However, the high-symmetry stacking modes currently considered in the package represent only a small fraction of the vast database of 2D materials available. Following more in-depth research, we plan to explore additional intriguing systems and their high-symmetry stacking modes, thereby broadening the scope and applicability of the PyHTStack2D package. Furthermore, in addition to the currently supported first-principles software interfaces, interfaces to other popular first-principles software packages, such as SIESTA [74], WIEN2k [75], and ABINIT [76,77], will be considered in future work. This will make the PyHTStack2D package more flexible and widely applicable.

## CRedit authorship contribution statement

**Qian Zhang:** Writing – original draft. **Jinlong Yang:** Writing – review & editing. **Wei Hu:** Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work is partly supported by the Innovation Program for Quantum Science and Technology (2021ZD0303306), the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB0450101), the National Natural Science Foundation of China (22173093, 22373096, 21688102, 22288201), the Laoshan Laboratory (LSKJ202300305), the Anhui Provincial Key Research and Development Program (2022a05020052), the National Key Research and Development Program of China (2021YFB0300600), the CAS Project for Young Scientists

in Basic Research (YSBR-005), and the Fundamental Research Funds for the Central Universities (WK2320000061). The authors thank the Hefei Advanced Computing Center, the Supercomputing Center of Chinese Academy of Sciences, the ORISE Supercomputing Center, the Supercomputing Center of USTC, the National Supercomputing Centers in Wuxi, Tianjin, Shanghai, and Guangzhou for the computational resources. The authors acknowledge the Beijing Beilong Super Cloud Computing Co., Ltd for providing HPC resources (<http://www.blsc.cn/>).

## Data availability

We have shared the link to our code and data in the Program Summary.

## References

- [1] N. Mounet, M. Gibertini, P. Schwaller, D. Campi, A. Merkys, A. Marrazzo, T. Sohier, I.E. Castelli, A. Cepellotti, G. Pizzi, N. Marzari, Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds, *Nat. Nanotechnol.* 13 (3) (2018) 246–252.
- [2] A. Geim, K. Novoselov, The rise of graphene, *Nat. Mater.* 6 (2007) 183–191.
- [3] B. Radisavljevic, A. Radenovic, J. Brivio, V. Giacometti, A. Kis, Single-layer  $\text{MoS}_2$  transistors, *Nat. Nanotechnol.* 6 (2011) 147–150.
- [4] F.H.L. Koppens, T. Mueller, P. Avouris, A.C. Ferrari, M.S. Vitiello, M. Polini, Photodetectors based on graphene, other two-dimensional materials and hybrid systems, *Nat. Nanotechnol.* 9 (2014) 780–793.
- [5] M. Chhowalla, H.S. Shin, G. Eda, L. Li, K.P. Loh, H. Zhang, The chemistry of two-dimensional layered transition metal dichalcogenide nanosheets, *Nat. Chem.* 5 (2013) 263–275.
- [6] Q.H. Wang, K. Kalantar-zadeh, A. Kis, J.N. Coleman, M.S. Strano, Electronics and optoelectronics of two-dimensional transition metal dichalcogenides, *Nat. Nanotechnol.* 7 (2012) 699–712.
- [7] C. Gong, Y. Zhang, W. Chen, J. Chu, T. Lei, J. Pu, L. Dai, C. Wu, Y. Cheng, T. Zhai, L. Li, J. Xiong, Electronic and optoelectronic applications based on 2D novel anisotropic transition metal dichalcogenides, *Adv. Sci.* 4 (12) (2017) 1700231.
- [8] A.K. Geim, I.V. Grigorieva, Van der Waals heterostructures, *Nature* 499 (2013) 419–425.
- [9] K.S. Novoselov, A. Mishchenko, A. Carvalho, A.H.C. Neto, 2D materials and van der Waals heterostructures, *Science* 353 (6298) (2016), aac9439.
- [10] S.-J. Liang, B. Cheng, X. Cui, F. Miao, Van der Waals heterostructures for high-performance device applications: challenges and opportunities, *Adv. Mater.* 32 (27) (2020) 1903800.
- [11] F. Liu, R. Shi, Z. Wang, Y. Weng, C.-M. Che, Y. Chen, Direct Z-scheme hetero-phase junction of black/red phosphorus for photocatalytic water splitting, *Angew. Chem. Int. Ed.* 58 (34) (2019) 11791–11795.
- [12] Y. Liu, Y. Huang, X. Duan, Van der Waals integration before and beyond two-dimensional materials, *Nature* 567 (2019) 323–333.
- [13] S.M. Shinde, K.P. Dhakal, X. Chen, W.S. Yun, J. Lee, H. Kim, J.-H. Ahn, Stacking-controllable interlayer coupling and symmetric configuration of multilayered  $\text{MoS}_2$ , *NPG Asia Mater.* 10 (2018) e468.
- [14] F. Bertoldo, R.R. Unocic, Y.-C. Lin, X. Sang, A.A. Puzos, Y. Yu, D. Miakota, C.M. Rouleau, J. Schou, K.S. Thygesen, D.B. Geohegan, S. Canulescu, Intrinsic defects in  $\text{MoS}_2$  grown by pulsed laser deposition: from monolayers to bilayers, *ACS Nano* 15 (2) (2021) 2858–2868.
- [15] Z. Liu, L. Song, S. Zhao, J. Huang, L. Ma, J. Zhang, J. Lou, P.M. Ajayan, Direct growth of graphene/hexagonal boron nitride stacked layers, *Nano Lett.* 11 (5) (2011) 2032–2037.
- [16] J. Li, X. Yang, Y. Liu, B. Huang, R. Wu, Z. Zhang, B. Zhao, H. Ma, W. Dang, Z. Wei, K. Wang, Z. Lin, X. Yan, M. Sun, B. Li, X. Pan, J. Luo, G. Zhang, Y. Liu, Y. Huang, X. Duan, X. Duan, General synthesis of two-dimensional van der Waals heterostructure arrays, *Nature* 579 (2020) 368–374.
- [17] G. Yuan, W. Liu, X. Huang, Z. Wan, C. Wang, B. Yao, W. Sun, H. Zheng, K. Yang, Z. Zhou, Y. Nie, J. Xu, L. Gao, Stacking transfer of wafer-scale graphene-based van der Waals superlattices, *Nat. Commun.* 14 (2023) 5457.
- [18] Y. Zhao, Y. Wang, Y. Yang, J. Zhao, X. Jiang, Realization of 2D multiferroic with strong magnetoelectric coupling by intercalation: a first-principles high-throughput prediction, *npj Comput. Mater.* 10 (2024) 122.
- [19] A. Jain, S.P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G.-M. Rignanese, G. Hautier, D. Gunter, K.A. Persson, FireWorks: a dynamic workflow system designed for high-throughput applications, *Concurr. Comput. Pract. Exp.* 27 (17) (2015) 5037–5059.
- [20] S. Curtarolo, W. Setyawan, G.L. Hart, M. Jahnatek, R.V. Chepulskii, R.H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M.J. Mehl, H.T. Stokes, D.O. Demchenko, D. Morgan, AFLOW: an automatic framework for high-throughput materials discovery, *Compos. Mater. Sci.* 58 (2012) 218–226.
- [21] S. Kirklin, J.E. Saal, B. Meredig, A. Thompson, J.W. Doak, M. Aykol, S. Rühl, C. Wolverton, The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies, *npj Comput. Mater.* 1 (1) (2015) 15010.

- [22] J.E. Saal, S. Kirklin, M. Aykol, B. Meredig, C.M. Wolverton, Materials design and discovery with high-throughput density functional theory: the Open Quantum Materials Database (OQMD), *JOM* 65 (2013) 1501–1509.
- [23] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson, Commentary: the Materials Project: a materials genome approach to accelerating materials innovation, *APL Mater.* 1 (1) (2013) 011002.
- [24] Y. Wang, S. Liu, Q. Li, R. Quhe, C. Yang, Y. Guo, X. Zhang, Y. Pan, J. Li, H. Zhang, L. Xu, B. Shi, H. Tang, Y. Li, J. Yang, Z. Zhang, L. Xiao, F. Pan, J. Lu, Schottky barrier heights in two-dimensional field-effect transistors: from theory to experiment, *Rep. Prog. Phys.* 84 (5) (2021) 056501.
- [25] Q. Tang, G. Hu, V. Fung, D.-e. Jiang, Insights into interfaces, stability, electronic properties, and catalytic activities of atomically precise metal nanoclusters from first principles, *Acc. Chem. Res.* 51 (11) (2018) 2793–2802.
- [26] Q. Tang, D.-e. Jiang, Mechanism of hydrogen evolution reaction on 1T-MoS<sub>2</sub> from first principles, *ACS Catal.* 6 (8) (2016) 4953–4961.
- [27] B. Liu, Y. Liu, C. Zhu, H. Xiang, H. Chen, L. Sun, Y. Gao, Y. Zhou, Advances on strategies for searching for next generation thermal barrier coating materials, *J. Mater. Sci. Technol.* 35 (5) (2019) 833–851.
- [28] C.J. Pickard, I. Errea, M.I. Erements, Superconducting hydrides under pressure, *Annu. Rev. Condens. Matter Phys.* 11 (2020) 57–76.
- [29] R.K. Barik, L.M. Woods, High throughput calculations for a dataset of bilayer materials, *Sci. Data* 10 (2023) 232.
- [30] S. Pakdel, A. Rasmussen, A. Taghizadeh, M. Kruse, T. Olsen, K.S. Thygesen, High-throughput computational stacking reveals emergent properties in natural van der Waals bilayers, *Nat. Commun.* 15 (2024) 932.
- [31] R. Dong, A. Jacob, S. Bourdais, S. Sanvito, High-throughput bandstructure simulations of van der Waals hetero-bilayers formed by 1T and 2H monolayers, *npj 2D Mater. Appl.* 5 (2021) 1–12.
- [32] W. Zhou, J. Chen, Z. Yang, J. Liu, F. Ouyang, Geometry and electronic structure of monolayer, bilayer, and multilayer Janus WS<sub>2</sub>, *Phys. Rev. B* 99 (2019) 075160.
- [33] J. He, K. Hummer, C. Franchini, Stacking effects on the electronic and optical properties of bilayer transition metal dichalcogenides MoS<sub>2</sub>, MoSe<sub>2</sub>, WS<sub>2</sub>, and WSe<sub>2</sub>, *Phys. Rev. B* 89 (7) (2014) 075409.
- [34] X. Hu, L. Kou, L. Sun, Stacking orders induced direct band gap in bilayer MoSe<sub>2</sub>-WSe<sub>2</sub> lateral heterostructures, *Sci. Rep.* 6 (2016) 31122.
- [35] C. Zhang, C.-P. Chuu, X. Ren, M.-Y. Li, L.-J. Li, C. Jin, M.-Y. Chou, C.-K. Shih, Interlayer couplings, Moiré, patterns, and 2D electronic superlattices in MoS<sub>2</sub>/WSe<sub>2</sub> hetero-bilayers, *Sci. Adv.* 3 (1) (2017) e1601459.
- [36] Z. Lin, C. Si, S. Duan, C. Wang, W. Duan, Rashba splitting in bilayer transition metal dichalcogenides controlled by electronic ferroelectricity, *Phys. Rev. B* 100 (2019) 155408.
- [37] S. Wang, X. Cui, C. Jian, H. Cheng, M. Niu, J. Yu, J. Yan, W. Huang, Stacking-engineered heterostructures in transition metal dichalcogenides, *Adv. Mater.* 33 (16) (2021) 2005735.
- [38] X. Zhang, H. Nan, S. Xiao, X. Wan, X. Gu, A. Du, Z. Ni, K.K. Ostrikov, Transition metal dichalcogenides bilayer single crystals by reverse-flow chemical vapor epitaxy, *Nat. Commun.* 10 (1) (2019) 598.
- [39] A. Chaves, J.G. Azadani, H. Alsalman, D.R. da Costa, R. Frisenda, A. Chaves, S. Song, Y.D. Kim, D. He, J. Zhou, A. Castellanos-Gomez, F.M. Peeters, Z. Liu, C.L. Hinkle, S. Oh, P.D. Ye, S.J. Koester, Y.H. Lee, P. Avouris, X. Wang, T. Low, Bandgap engineering of two-dimensional semiconductor materials, *npj 2D Mater. Appl.* 4 (2020) 1–21.
- [40] H. Shin, D. Hong, H. Cho, H. Jang, G.Y. Kim, K.M. Song, M.-J. Choi, D. Kim, Y.S. Jung, Indirect-to-direct bandgap transition in GaP semiconductors through quantum shell formation on ZnS nanocrystals, *Nat. Commun.* 15 (1) (2024) 1–10.
- [41] N. Ferdous, M.S. Islam, J. Park, A resilient type-III broken gap Ga<sub>2</sub>O<sub>3</sub>/SiC van der Waals heterogeneous bilayer with band-to-band tunneling effect and tunable electronic property, *Sci. Rep.* 14 (12748) (2024) 2045–2322.
- [42] Q. Xu, L. Zhang, J. Yu, S. Wageh, A.A. Al-Ghamdi, M. Jaroniec, Direct Z-scheme photocatalysts: principles, synthesis, and applications, *Mater. Today* 21 (10) (2018) 1042–1063.
- [43] J. Li, X. Li, J. Yang, A review of bipolar magnetic semiconductors from theoretical aspects, *Fundam. Res.* 2 (4) (2022) 511–521.
- [44] J. Deng, J. Pan, Y. Zhang, Y. Li, W. Dong, J. Sun, S. Du, Screening and design of bipolar magnetic-semiconducting monolayers and heterostructures, *ACS Appl. Electron. Mater.* 4 (7) (2022) 3232–3239.
- [45] G. Kresse, J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set, *Phys. Rev. B* 54 (16) (1996) 11169.
- [46] S. Haastруп, M. Strange, M. Pandey, T. Deilmann, P.S. Schmidt, N.F. Hinsche, M.N. Gjerding, D. Torelli, P.M. Larsen, A.C. Riis-Jensen, J. Gath, K.W. Jacobsen, J.J. Mortensen, T. Olsen, K.S. Thygesen, The computational 2D materials database: high-throughput modeling and discovery of atomically thin crystals, *2D Mater.* 5 (4) (2018) 042002.
- [47] M.N. Gjerding, A. Taghizadeh, A. Rasmussen, S. Ali, F. Bertoldo, T. Deilmann, N.R. Knøsgaard, M. Kruse, A.H. Larsen, S. Mantl, et al., Recent progress of the computational 2D materials database (C2DB), *2D Mater.* 8 (4) (2021) 044002.
- [48] G. Constantinescu, A. Kuc, T. Heine, Stacking in bulk and bilayer hexagonal boron nitride, *Phys. Rev. Lett.* 111 (2013) 036104.
- [49] X. Mao, J. Li, Z. Liu, G. Wang, Q. Zhang, Y. Jin, T-Phase and H-Phase coupled TMD van der Waals heterostructure ZrS<sub>2</sub>/MoTe<sub>2</sub> with both Rashba spin splitting and Type-III band alignment, *J. Phys. Chem. C* 126 (25) (2022) 10601–10609.
- [50] B. Sa, R. Hu, Z. Zheng, R. Xiong, Y. Zhang, C. Wen, J. Zhou, Z. Sun, High-throughput computational screening and machine learning modeling of Janus 2D III–VI van der Waals heterostructures for solar energy applications, *Chem. Mater.* 34 (15) (2022) 6687–6701.
- [51] J. Zhou, L. Shen, M. Costa, K. Persson, S. Ong, P. Huck, Y. Lu, X. Ma, Y. Chen, H. Tang, Y.P. Feng, 2D MatPedia, an open computational database of two-dimensional materials from top-down and bottom-up approaches, *Sci. Data* 6 (2019) 86.
- [52] W. Hu, L. Lin, A.S. Banerjee, E. Vecharynski, C. Yang, Adaptively compressed exchange operator for large-scale hybrid density functional calculations with applications to the adsorption of water on silicene, *J. Chem. Theory Comput.* 13 (3) (2017) 1188–1198.
- [53] J. Feng, L. Wan, J. Li, S. Jiao, X. Cui, W. Hu, J. Yang, Massively parallel implementation of iterative eigensolvers in large-scale plane-wave density functional theory, *Comput. Phys. Commun.* 299 (2024) 109135.
- [54] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. De Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougousis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. Seitsonen, A. Smogunov, P. Umari, R. Wentzcovitch, QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials, *J. Phys. Condens. Matter* 21 (39) (2009) 395502.
- [55] V. Wang, N. Xu, J.-C. Liu, G. Tang, W.-T. Geng, VASPKIT: a user-friendly interface facilitating high-throughput computing and analysis using VASP code, *Comput. Phys. Commun.* 267 (2021) 108033.
- [56] S.P. Ong, W.D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V.L. Chevrier, K.A. Persson, G. Ceder, Python Materials Genomics (pymatgen): a robust, open-source python library for materials analysis, *Compos. Mater. Sci.* 68 (2013) 314–319.
- [57] A.B. Yoo, M.A. Jette, M. Grondona, Slurm: simple Linux utility for resource management, in: D. Feitelson, L. Rudolph, U. Schwiigelshohn (Eds.), *Job Scheduling Strategies for Parallel Processing*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 44–60.
- [58] R.L. Henderson, Job scheduling under the portable batch system, in: D.G. Feitelson, L. Rudolph (Eds.), *Job Scheduling Strategies for Parallel Processing*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1995, pp. 279–294.
- [59] B. Huang, M. McGuire, A. May, D. Xiao, P. Jarillo-Herrero, X. Xu, Emergent phenomena and proximity effects in two-dimensional magnets and heterostructures, *Nat. Mater.* 19 (2020) 1276–1289.
- [60] M. Gibertini, M. Koperski, A.F. Morpurgo, K.S. Novoselov, Magnetic 2D materials and heterostructures, *Nat. Nanotechnol.* 14 (5) (2019) 408–419.
- [61] O.T. Unke, S. Chmiela, H.E. Sauceda, M. Gastegger, I. Poltavsky, K.T. Schütt, A. Tkatchenko, K.-R. Müller, Machine learning force fields, *Chem. Rev.* 121 (16) (2021) 10142–10186.
- [62] J. Behler, Four generations of high-dimensional neural network potentials, *Chem. Rev.* 121 (16) (2021) 10037–10072.
- [63] P.E. Blöchl, Projector augmented-wave method, *Phys. Rev. B* 50 (24) (1994) 17953.
- [64] J.P. Perdew, K. Burke, M. Ernzerhof, Generalized gradient approximation made simple, *Phys. Rev. Lett.* 77 (18) (1996) 3865.
- [65] S. Grimme, J. Antony, S. Ehrlich, H. Krieg, A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H–Pu, *J. Chem. Phys.* 132 (15) (2010) 154104.
- [66] H. Ma, Z. Wang, W. Zhao, H. Ren, H. Zhu, Y. Chi, W. Guo, Enhancing the photoinduced interlayer charge transfer and spatial separation in Type-II heterostructure of WS<sub>2</sub> and asymmetric Janus-MoSe with intrinsic self-build electric field, *J. Phys. Chem. Lett.* 13 (36) (2022) 8484–8494.
- [67] Q. Zhang, Y. Xiong, Y. Gao, J. Chen, W. Hu, J. Yang, First-principles high-throughput inverse design of direct momentum-matching band alignment van der Waals heterostructures utilizing two-dimensional indirect semiconductors, *Nano Lett.* 24 (12) (2024) 3710–3718.
- [68] V.I. Anisimov, J. Zaanen, O.K. Andersen, Band theory and Mott insulators: Hubbard U instead of Stoner I, *Phys. Rev. B* 44 (1991) 943–954.
- [69] B. Himmetoglu, A. Floris, S. de Gironcoli, M. Cococcioni, Hubbard-corrected dft energy functionals: the lda+u description of correlated systems, *Int. J. Quant. Chem.* 114 (1) (2014) 14–49.
- [70] S.L. Dudarev, G.A. Botton, S.Y. Savrasov, C.J. Humphreys, A.P. Sutton, Electron-energy-loss spectra and the structural stability of nickel oxide: an LSDA+U study, *Phys. Rev. B* 57 (1998) 1505–1509.
- [71] A.I. Liechtenstein, V.I. Anisimov, J. Zaanen, Density-functional theory and strong interactions: orbital ordering in Mott-Hubbard insulators, *Phys. Rev. B* 52 (1995) R5467–R5470.
- [72] J. Heyd, G.E. Scuseria, M. Ernzerhof, Hybrid functionals based on a screened Coulomb potential, *J. Chem. Phys.* 118 (18) (2003) 8207–8215.
- [73] R. Haunschild, G.E. Scuseria, Range-separated local hybrids, *J. Chem. Phys.* 132 (22) (2010) 224106.
- [74] J.M. Soler, E. Artacho, J.D. Gale, A. García, J. Junquera, P. Ordejón, D. Sánchez-Portal, The SIESTA method for ab initio order-N materials simulation, *J. Phys. Condens. Matter* 14 (11) (2002) 2745.
- [75] P. Blaha, K. Schwarz, F. Tran, R. Laskowski, G.K.H. Madsen, L.D. Marks, WIEN2k: an APW+lo program for calculating the properties of solids, *J. Chem. Phys.* 152 (7) (2020) 074101.

- [76] X. Gonze, B. Amadon, G. Antonius, F. Arnardi, L. Baguet, J.-M. Beuken, J. Bieder, F. Bottin, J. Bouchet, E. Bousquet, N. Brouwer, F. Bruneval, G. Brunin, T. Cavignac, J.-B. Charraud, W. Chen, M. Côté, S. Cottenier, J. Denier, G. Geneste, P. Ghosez, M. Giantomassi, Y. Gillet, O. Gingras, D.R. Hamann, G. Hautier, X. He, N. Helbig, N. Holzwarth, Y. Jia, F. Jollet, W. Lafargue-Dit-Hauret, K. Lejaeghere, M.A.L. Marques, A. Martin, C. Martins, H.P.C. Miranda, F. Naccarato, K. Persson, G. Petretto, V. Planes, Y. Pouillon, S. Prokhorenko, F. Ricci, G.-M. Rignanese, A.H. Romero, M.M. Schmitt, M. Torrent, M.J. van Setten, B.V. Troeye, M.J. Verstraete, G. Zérah, J.W. Zwanziger, The Abinit project: impact, environment and recent developments, *Comput. Phys. Commun.* 248 (2020) 107042.
- [77] A.H. Romero, D.C. Allan, B. Amadon, G. Antonius, T. Applencourt, L. Baguet, J. Bieder, F. Bottin, J. Bouchet, E. Bousquet, F. Bruneval, G. Brunin, D. Caliste, M. Côté, J. Denier, C. Dreyer, P. Ghosez, M. Giantomassi, Y. Gillet, O. Gingras, D.R. Hamann, G. Hautier, F. Jollet, G. Jomard, A. Martin, H.P.C. Miranda, F. Naccarato, G. Petretto, N.A. Pike, V. Planes, S. Prokhorenko, T. Rangel, F. Ricci, G.-M. Rignanese, M. Royo, M. Stengel, M. Torrent, M.J. van Setten, B.V. Troeye, M.J. Verstraete, J. Wiktor, J.W. Zwanziger, X. Gonze, ABINIT: overview, and focus on selected capabilities, *J. Chem. Phys.* 152 (2020) 124102.