MoMa: A Simple Modular Learning Framework for Material Property Prediction

Anonymous authors
Paper under double-blind review

Abstract

Deep learning methods for material property prediction have been widely explored to advance materials discovery. However, the prevailing pre-train paradigm often fails to address the inherent diversity and disparity of material tasks. To overcome these challenges, we introduce MoMa, a simple **Mo**dular framework for **Ma**terials that first trains specialized modules across a wide range of tasks and then adaptively composes synergistic modules tailored to each downstream scenario. Evaluation across 17 datasets demonstrates the superiority of MoMa, with a substantial 14% average improvement over the strongest baseline. Few-shot and continual learning experiments further highlight MoMa's potential for real-world applications. Pioneering a new paradigm of modular material learning, MoMa will be open-sourced to foster broader community collaboration.

1 Introduction

Accurate and efficient material property prediction is critical for accelerating materials discovery. Key properties such as formation energy and band gap are fundamental in identifying stable and functional materials (Masood et al., 2023; Riebesell et al., 2025). While traditional approaches such as density functional theory offer high precision (Jain et al., 2016), their prohibitive computational cost limits their practicality for large-scale screening (Fiedler et al., 2022; Lan et al., 2023).

Recently, deep learning methods have been developed to expedite traditional approaches (Xie & Grossman, 2018; Griesemer et al., 2023). Pre-trained force field models, in particular, have shown remarkable success in generalizing to a wide spectrum of material property prediction tasks (Yang et al., 2024b; Shoghi et al., 2024; Rhodes et al., 2025), outperforming specialized models trained from scratch. These models are typically pre-trained on the potential energy surface (PES) data of materials (Barroso-Luque et al., 2024) and then fine-tuned for the target downstream task.

Despite these advances, we identify two key challenges that undermine the effectiveness of current deep learning models for material property prediction: **diversity** and **disparity**.

First, material tasks exhibit significant diversity (Fig. 1) which challenges the generalizability of existing models. For instance, prevailing force-field models are only trained on PES-derived properties (e.g., force, energy, and stress) mostly focusing on crystalline materials (Yang et al., 2024b; Barroso-Luque et al., 2024). However, material tasks span a much wider variety of systems (e.g., crystals, organic molecules) and properties (e.g., thermal stability, electronic behavior), making it difficult for methods trained on a limited set of data to generalize across the full spectrum of tasks.

Second, the disparate nature of material tasks presents huge obstacles for jointly training a broad span of tasks in one model. Material systems vary significantly in atomic composition, bonding and structural periodicity, while their properties are governed by distinct physical laws. For example, mechanical strength in metals is primarily influenced by atomic bonding and crystal structure, whereas electronic properties like conductivity are determined by the material's electronic structure. Consequently, training a single model across a wide range of tasks (Shoghi et al., 2024) may lead to knowledge conflicts, hindering the model's ability to effectively adapt to downstream scenarios.

Drawing inspiration from modular deep learning (Pfeiffer et al., 2023), we propose MoMa, a **Mo**dular framework for **Ma**terial property prediction. To accommodate the **diversity** challenge, MoMa trains multiple high-resource property prediction datasets into transferrable modules to sup-

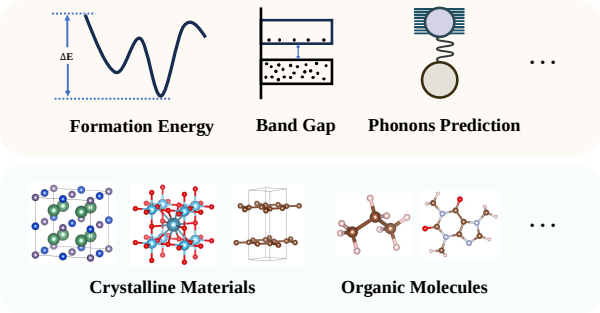

Figure 1: Illustration of the diversity of material properties (top) and systems (down). Material tasks are also disparate, with different laws governing diverse properties and systems. These characteristics pose challenges for material property prediction models.

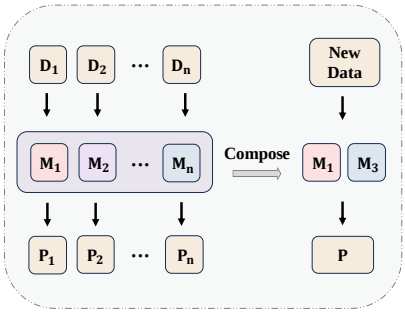

Figure 2: The modular learning scheme in MoMa trains and stores a broad spectrum of material tasks as modules, and adaptively composes them given a new material property prediction task.

port a wide-span of downstream tasks. In parallel, to address the **disparity** challenge, MoMa encapsulates each task within a specialized module during training to avoid interference. In adapting MoMa's modules to downstream tasks, we devise a novel composition strategy with high selectivity and efficiency to integrate the most synergistic modules to mitigate knowledge conflicts.

Specifically, MoMa comprises two major stages: (1) *Module Training & Centralization*. MoMa trains dedicated modules for a broad range of material tasks, offering two versions: a full module for superior performance and a memory-efficient adapter module. These trained modules are centralized in MoMa Hub, a repository designed to facilitate knowledge reuse while preserving proprietary data for privacy-aware material learning. (2) *Adaptive Module Composition (AMC) & Fine-tuning*. We devise AMC, a tailored algorithm that adaptively composes synergetic modules from MoMa Hub. AMC first estimates the performance of each module on the target task in a *training-free* manner, enabling efficient evaluation across a wide range of modules. Based on these predictions, AMC then optimizes a weighted combination of modules to explicitly minimize the target task loss, capturing relevant knowledge while mitigating interference. The resulting composed module is then fine-tuned for improved adaptation to the downstream task. Together, the two stages deliver a modular solution that enables MoMa to account for the diversity and disparity of material knowledge.

Empirical results across 17 downstream tasks showcase the superiority of MoMa, outperforming all baselines in **16/17** tasks, with an average improvement of **14%** compared to the second-best baseline. In *few-shot* settings, which are common in materials science, MoMa achieves even larger performance gains to the conventional pre-train then fine-tune paradigm. Additionally, we show that MoMa can expand its capability in *continual learning* settings by incorporating molecular tasks into MoMa Hub. The trained modules in MoMa Hub will be open-sourced, and we envision MoMa becoming a pivotal platform for the modularization and distribution of materials knowledge, fostering deeper community engagement to accelerate materials discovery.

2 Proposed Framework: MoMa

MoMa is a simple modular framework targeting the diversity and disparity of material property prediction tasks. A high-level abstraction of MoMa is provided in Fig. 2. Its modular solution allows for the flexible and scalable integration of diverse material knowledge modules, and the effective and tailored adaptation to material property prediction tasks. We now elaborate our proposed framework.

2.1 Overview

MoMa involves two major stages: (1) training and centralizing modules into MoMa Hub; (2) adaptively composing these modules to support downstream material tasks. A visual overview of MoMa is provided in Figure 3.

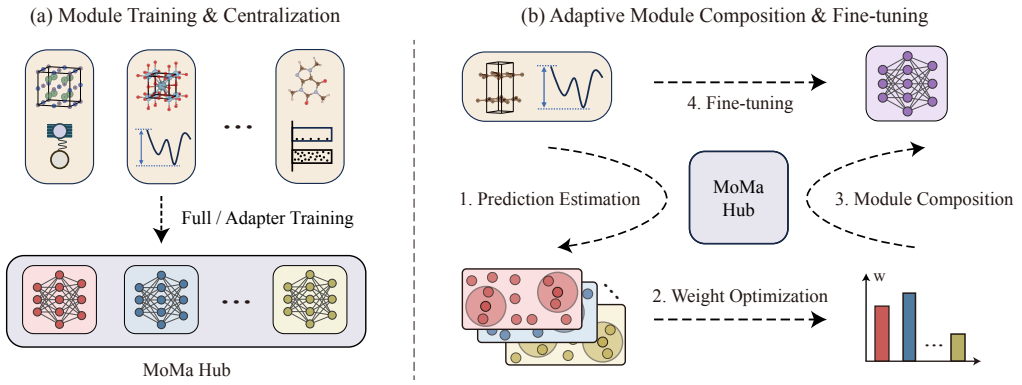

Figure 3: The MoMa framework. (a) During the Module Training & Centralization stage (Section 2.2), MoMa trains full and adapter modules for a wide spectrum of material tasks, constituting the MoMa Hub; (b) The Adaptive Module Composition (AMC) & Fine-tuning stage (Section 2.3) leverages the modules in MoMa Hub to compose a tailored module for each downstream task. The AMC algorithm comprises three steps: 1. module prediction estimation (with kNN); 2. module weight optimization; 3. module composition. The composed module is further fine-tuned on the task for better adaptation.

In the first stage (Section 2.2), we encompass a wide range of material properties and systems into MoMa Hub. This accommodates the diversity of material tasks and addresses the task disparity by training specialized modules for each.

In the second stage (Section 2.3), we devise the Adaptive Module Composition algorithm. Given the downstream material task, the algorithm heuristically optimizes the optimal combination of module weights for MoMa Hub and composes a customized module based on the weights, which is subsequently fine-tuned on the task for better adaptation. Respecting the diverse and disparate nature of material tasks, our adaptive approach automatically discovers synergistic modules and excludes conflicting combinations by the data-driven assignment of module weights.

2.2 Module Training & Centralization

To better exploit the transferrable knowledge of open-source material property prediction datasets, we first train distinctive modules for each high-resource material task, and subsequently centralize these modules to constitute MoMa Hub.

Module Training Leveraging the power of state-of-the-art material property prediction models, we choose to employ a pre-trained backbone encoder f as the initialization for training each MoMa module. Note that MoMa is independent of the backbone model choice, which enables smooth integration with other pre-trained backbones.

We provide two parametrizations for the MoMa modules: the **full** module and the **adapter** module. For the full module, we directly treat each fully fine-tuned model backbone as a standalone module. The adapter module, in contrast, serves as a parameter-efficient alternative where adapter layers (Houlsby et al., 2019) are inserted between each layer of the backbone. The adapters are updated and the rest of the backbone is frozen. All adapters trained for a given task are collectively treated as one module. This implementation trade-offs the downstream performance for a much lower GPU memory cost during training, making it especially suitable for compute-constrained settings. When training converges, all module parameters are stored into a centralized repository $\mathcal{H}$ termed MoMa Hub, formally:

$$\mathcal{H} = \{g_1, g_2, \ldots, g_N\}, \quad g_i = \begin{cases} \theta_f^i & \text{(full module)} \\ \Delta_f^i & \text{(adapter module)} \end{cases}$$

where θ_f^i and Δ_f^i denote the full and adapter module parameters for the i^{th} task and encoder f.

Module Centralization To support a wide array of downstream tasks, MoMa Hub needs to include modules trained on diverse material systems and properties. Currently, MoMa Hub encompasses 18 material property prediction tasks selected from the Matminer datasets (Ward et al., 2018) with over 10000 data points. These tasks span across a large range of material properties, including thermal properties (e.g. formation energy), electronic properties (e.g. band gap), mechanical properties (e.g. shear modulus), etc. For more details, please refer to Section B.1. To showcase the effect of scaling data diversity, we present the continual learning results in Section 3.5 after further incorporating molecular property prediction tasks into MoMa Hub. Note that MoMa is designed to be task-agnostic and may readily support a larger spectrum of tasks in the future.

An important benefit of the modular design of MoMa Hub is that it preserves proprietary data, which is prevalent in the field of materials, enabling privacy-aware contribution of new modules. Therefore, MoMa could serve as an open platform for the modularization of materials knowledge.

2.3 Adaptive Module Composition & Fine-tuning

Given a labeled material property prediction dataset $\mathcal{D}$ with m instances: $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, the second stage of MoMa customizes a task-specific model using the modules in MoMa Hub. We highlight the key desiderata in our setting:

- **Selective:** Material tasks are inherently disparate. Hence only the most relevant modules shall be selected to minimize interference and promote positive transfer to downstream tasks.
- **Data-driven:** As the diversity of tasks in MoMa Hub expands, it is impossible to rely solely on human expertise for module selection. A data-driven approach is required to mine the implicit relationships between the MoMa Hub modules and downstream tasks.
- **Efficient:** Enumerating all combinations of modules is impractical. Efficient algorithms shall be developed to return the optimal module composition with a reasonable amount of computation.

Unfortunately, to our best knowledge, none of the prevailing module composition methods fully satisfy the requirements outlined above in our setting. They either depend on human heuristics for module composition (Ilharco et al., 2022; Yu et al., 2024), or rely on assumptions on tasks (Yang et al., 2023; Zhu et al., 2025) or network structures (Ostapenko et al., 2024; Huang et al., 2024) not satisfiable in material property prediction settings, as further evidenced through a representative comparison in Section 3.3.

To address these limitations, we devise the Adaptive Module Composition (AMC) algorithm. AMC is a fast heuristic algorithm that first estimates the prediction of each module on the downstream task, then optimizes the module weights, and finally composes the selected modules to form the task-specific module. We now introduce AMC in detail, with its formal formulation in Algorithm 1.

Module Prediction Estimation We begin by estimating the predictive performance of each module in MoMa Hub $\mathcal{H}$ on the downstream task $\mathcal{D}$. More accurate predictions indicate stronger relevance to the task and intuitively warrant higher weights in the composition.

For each module g_j in $\mathcal{H}$, we first take it to encode each input materials in the train set of task $\mathcal{D}$ into a set of representation $\mathcal{X}^j = \{\mathbf{x}_1^j, \mathbf{x}_2^j, \ldots, \mathbf{x}_m^j\}$ in which $\mathbf{x}_i^j = g_j(x_i)$. Then we obtain the estimated prediction of g_j on $\mathcal{D}$ using a leave-one-out label propagation approach (Iscen et al., 2019). Specifically, we iteratively select one sample $\mathbf{x}_i^j$ from $\mathcal{X}^j$ and get the predicted label $\hat{y_i}^j$ by calculating the weighted sum of its K nearest neighbors' labels within $\mathcal{X}^j$:

$$\hat{y}_i^{\,j} = \sum_{k=1}^{K} \frac{f_d\left(\mathbf{x}_i^{\,j}, \mathbf{x}_k^{\,j}\right)}{Z_i^{\,j}} y_k, \quad Z_i^{\,j} = \sum_{k=1}^{K} f_d\left(\mathbf{x}_i^{\,j}, \mathbf{x}_k^{\,j}\right). \tag{1}$$

where $\mathbf{x}_k^j$ denotes the k-th nearest neighbors of $\mathbf{x}_i^j$. The distance function f_d is the exponential of cosine similarity between each embedding pair.

While other predictors are viable, we choose kNN due to its good trade-off in efficiency and accuracy. Also, its training-free nature enhances its flexibility in real-world scenarios, where the downstream data may be subject to updates.

Module Weight Optimization After estimating each module's prediction, we now have to select the optimal combination of modules tailored for the downstream task $\mathcal{D}$. To achieve this, the most straightforward approach is to compare the prediction error obtained after fine-tuning each combination of modules. However, this is infeasible due to the combinatorial explosion. Therefore, we reformulate the task as an optimization problem, using the prediction error before fine-tuning as a proxy metric (later referred to as *proxy error*). By optimizing the proxy error, we could obtain the optimal combination of weights.

Specifically, inspired by ensemble learning (Zhou et al., 2002; Zhou, 2016), we assign a weight w_j for each module g_j and calculate the output of the ensemble: $\sum_{j=1}^{NT} w_j \hat{y_i}^j$. We then estimate the proxy error on the train set of $\mathcal{D}$ for this weighted ensemble:

$$E_{\mathcal{D}} = \frac{1}{m}\sum_{i=1}^{m}(\sum_{j=1}^{N} w_j \hat{y}_i^j - y_i)^2 \tag{2}$$

To minimize the proxy error $E_{\mathcal{D}}$, we then utilize the open source cvxpy package (Diamond et al., 2014) to optimize the module weights. The objective is:

$$\underset{w_j}{\operatorname{argmin}}\ E_{\mathcal{D}},\ \text{ s.t. } \sum_{j=1}^{N} w_j = 1,\ w_j \geq 0 \tag{3}$$

Module Composition After the optimization converges, we can use the learned weights to compose a single customized module for the specific task. Inspired by the recent success of model merging (Wortsman et al., 2022; Ilharco et al., 2022; Yu et al., 2024; Yang et al., 2024a), we adopt a simple yet surprisingly effective method by weighted averaging the parameters of the selected modules: $g_{\mathcal{D}} = \sum_{j=1}^{N} w_j^* g_j$, where w_j^* represents the optimized weight for the j-th module in Eq. (3). Here, the weights underscore the relevance of each selected module to the downstream task.

While alternative composition methods, such as mixture-of-experts (Jacobs et al., 1991), are feasible, they incur high memory overhead as MoMa Hub expands, limiting their practical deployment under computational constraints. By contrast, our weighted-average composition uses fewer resources while effectively integrating knowledge from all modules. In the full-module setting, every module shares the same architecture and pre-trained backbone with identical initializations, providing a grounded foundation for successful knowledge composition (Zhou et al., 2024).

Downstream Fine-tuning To better adapt to the downstream task $\mathcal{D}$, the composed module $g_{\mathcal{D}}$ is appended with a task-specific head and then fine-tuned on $\mathcal{D}$ to convergence.

3 Experiments

In this section, we conduct comprehensive experiments to demonstrate the empirical effectiveness of MoMa. The experimental setup is outlined in Section 3.1. The main results, discussed in Section 3.2, show that MoMa **substantially outperforms** baseline methods. Additionally, we conduct a thorough ablation study on the AMC algorithm as detailed in Section 3.3. Confronted with the data scarcity challenge common in real-world materials discovery settings, we evaluate MoMa's few-shot learning ability in Section 3.4, where it achieves **even larger** performance gains compared to baselines. To further highlight the **flexibility and scalability** of MoMa, we extend MoMa Hub to include molecular datasets and present the continual learning results in Section 3.5. Finally, we visualize the module weights optimized by AMC in Section 3.6, highlighting MoMa's potential for providing **valuable insights** into material properties.

3.1 Setup

Datasets To better align with real-world material property prediction settings where labels are usually scarce, we conduct experiments on 17 low-data material property prediction tasks from Matminer (Ward et al., 2018) adhering to Chang et al. (2022). This benchmark offers a comprehensive evaluation of model capability on a wide span of properties critical for material discovery. Refer to Section B.1 for more dataset details.

Table 1: **Main results for 17 material property prediction tasks.** The best MAE for each task is highlighted in **bold** and the second best result is underlined. The result for each task are the average of five data splits, reported to three significant digits. For each method, the standard deviation of the test MAE across five random seeds is shown in parentheses. Additionally, the average rank and its standard deviation across the 17 datasets are provided to reflect the consistency of each method.

Datasets	CGCNN	MoE-(18)	JMP-MT	JMP-FT	MoMa (Adapter)	MoMa (Full)
Experimental Band Gap (eV)	0.471 (0.008)	0.374 (0.008)	0.377 (0.005)	0.358 (0.014)	0.359 (0.009)	**0.305** (0.006)
Formation Enthalpy (eV/atom)	0.193 (0.015)	0.0949 (0.0016)	0.134 (0.001)	0.168 (0.007)	0.158 (0.009)	**0.0839** (0.0013)
2D Dielectric Constant	2.90 (0.12)	2.29 (0.01)	2.25 (0.06)	2.35 (0.07)	2.31 (0.04)	**1.89** (0.03)
2D Formation Energy (eV/atom)	0.169 (0.006)	0.106 (0.005)	0.140 (0.004)	0.125 (0.006)	0.112 (0.002)	**0.0495** (0.0015)
Exfoliation Energy (meV/atom)	59.7 (1.5)	52.5 (0.8)	42.3 (0.5)	**35.4** (2.0)	**35.4** (0.9)	36.3 (0.2)
2D Band Gap (eV)	0.686 (0.034)	0.532 (0.008)	0.546 (0.020)	0.582 (0.018)	0.552 (0.014)	**0.375** (0.006)
3D Poly Electronic	32.5 (1.1)	27.7 (0.1)	23.9 (0.2)	23.3 (0.3)	23.3 (0.2)	**23.0** (0.1)
3D Band Gap (eV)	0.492 (0.008)	0.361 (0.003)	0.423 (0.004)	0.249 (0.001)	0.245 (0.002)	**0.200** (0.001)
Refractive Index	0.0866 (0.0014)	0.0785 (0.0004)	0.0636 (0.0006)	0.0555 (0.0027)	0.0533 (0.0023)	**0.0523** (0.0010)
Elastic Anisotropy	3.65 (0.11)	3.01 (0.03)	2.53 (0.26)	**2.42** (0.36)	2.57 (0.61)	2.86 (0.28)
Electronic Dielectric Constant	0.168 (0.002)	0.157 (0.015)	0.137 (0.002)	0.108 (0.002)	0.106 (0.002)	**0.0885** (0.0048)
Dielectric Constant	0.258 (0.008)	0.236 (0.002)	0.224 (0.004)	0.171 (0.002)	0.168 (0.002)	**0.158** (0.002)
Phonons Mode Peak (cm^{-1})	0.127 (0.004)	0.0996 (0.0083)	0.0859 (0.0006)	0.0596 (0.0065)	0.0568 (0.0009)	**0.0484** (0.0026)
Poisson Ratio	0.0326 (0.0001)	0.0292 (0.0001)	0.0297 (0.0003)	0.0221 (0.0004)	0.0220 (0.0003)	**0.0204** (0.0002)
Poly Electronic	2.97 (0.10)	2.61 (0.13)	2.42 (0.03)	2.11 (0.04)	2.13 (0.03)	**2.09** (0.03)
Poly Total	6.54 (0.24)	5.51 (0.04)	5.52 (0.03)	4.89 (0.06)	4.89 (0.04)	**4.86** (0.07)
Piezoelectric Modulus	0.232 (0.004)	0.208 (0.003)	0.199 (0.002)	0.174 (0.004)	**0.173** (0.003)	0.174 (0.001)
Average Rank	6.00 (0.00)	4.12 (1.17)	3.94 (0.97)	2.88 (1.27)	2.47 (0.94)	**1.35** (0.86)

Implementation Details For the pre-trained backbone of MoMa, we employ the open-source JMP model (Shoghi et al., 2024) for representing material systems given its superior performance in property prediction tasks across both crystals and molecules. For a rigorous comparison, we present the MAE averaged across the five splits adopted from Chang et al. (2022). Each experiment is repeated with five random seeds, and the reported standard deviation is computed across the seed-level averages. Additional implementation details, including the details of module architecture, the hyper-parameters for MoMa, and the computational cost, are provided in Section B.2.

Baseline Methods We compare the performance of MoMa with four baseline methods: CGCNN (Xie & Grossman, 2018), MoE-(18) (Chang et al., 2022), JMP-FT, and JMP-MT (Shoghi et al., 2024). CGCNN represents a classical method without pre-training. MoE-(18) trains separate CGCNN models for the upstream tasks of MoMa, then ensembles them as one model in a mixture-of-experts approach for downstream fine-tuning. JMP-FT directly fine-tunes the JMP pre-trained checkpoint on the downstream tasks. JMP-MT trains all tasks in MoMa Hub with a multi-task pretraining scheme and then adapts to each downstream dataset with further fine-tuning. More discussions on baselines are included in Section B.3.

3.2 Main Results

Performance of MoMa As shown in Table 1, MoMa (Full) achieves the best performance with the lowest average rank of 1.35 and 14/17 best results. MoMa (Adapter) follows, with an average rank of 2.47. Together, the two variants hold **16/17** best results. They also exhibit the smallest rank deviations, indicating that MoMa consistently delivers reliable performance across tasks. Notably, MoMa (Full) outperforms JMP-FT in 14 tasks, with an impressive average improvement of 14.0%, highlighting the effectiveness of MoMa Hub modules in fostering material property prediction. Moreover, MoMa (Full) surpasses JMP-MT in 16 of 17 tasks with a substantial average margin of 24.8%, underscoring the advantage of MoMa's modular design in mitigating task interference. Further analyses in this section are done with MoMa (Full) due to its superior performance.

Performance of Baselines Among the baseline methods, JMP-FT performs the best with an average rank of 2.88, followed by JMP-MT with an average rank of 3.94. Though additionally trained on upstream tasks of MoMa Hub, JMP-MT still lags behind JMP-FT. We hypothesize that the inherent knowledge conflicts between the disparate material tasks pose a tremendous risk to the multi-task learning approach. We also observe that methods utilizing the JMP encoder outperform those based on CGCNN encoders, demonstrating the good transferability of large force field models to material tasks. We include results with more architectures and baselines in Section C.1 and Section C.2.

3.3 Ablation & Analysis of Adaptive Module Composition

Ablation Study We conduct a fine-grained ablation study of AMC with three variants : (1) *Select Average*, which retains the AMC-selected modules (nonzero weights) but averages them uniformly; (2) *All Average*, which averages all modules in MoMa Hub; (3) *Random Selection*, which picks a random set of modules in MoMa Hub with the same module number as AMC. A visualization of the ablation results is presented in Fig. 4. The three variants are inferior to AMC in 13, 15 and 15 out of 17 tasks, with an average test MAE increase of 11.0%, 18.0% and 20.2%, respectively. This highlight the effectiveness of both module selection and weighted composition in AMC.

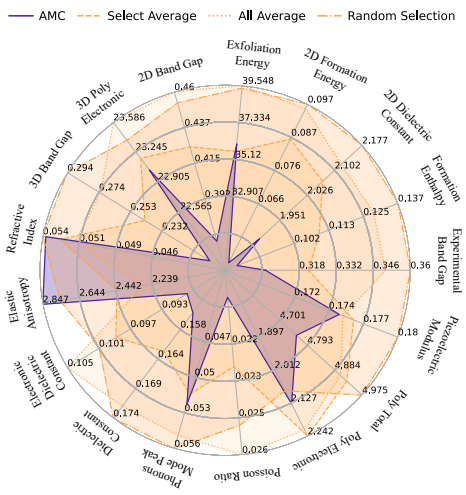

Figure 4: Ablation study of AMC. The axis represents test set MAE and **smaller area is better**.

Analysis Experiments AMC employs white-box optimization of module weights guided by the proxy error (Eq. (3)). To further analyze the importance of this scheme, we replace AMC with *LoraHub Learning* (Huang et al., 2024), a black-box optimization approach for module composition, and *Softmax Weighting*, a non-optimized heuristic based on kNN predicted performance. The two variants under-performs AMC in 15 and 12 out of 17 tasks, with 15.5% and 13.7% average increase in test MAE. This shows the benefit of the optimization scheme in AMC over both black-box search and performance-based heuristics. See more implementation details in Section B.4 and complete results and discussion in Section C.3.

Efficiency Analysis We highlight that AMC is highly efficient: it requires only a single round of forward embedding generation, followed by lightweight kNN prediction and convex optimization. For the largest dataset, AMC converges in under 30 seconds. This efficiency enables MoMa to scale to a larger number of modules in future applications. See Section C.4 for a detailed analysis.

3.4 Performance in Few-shot Settings

Motivation & Setup To better assess the performance of MoMa in real-world scenarios, where labeled material candidates are costly and often scarce (Abed et al., 2024), we construct a few-shot learning setting and compare MoMa with JMP-FT. For each downstream task, we down-sample the training data and apply AMC to compose modules from MoMa Hub, followed by fine-tuning on the sampled subset. The validation and test sets remain consistent with those in the standard setting for robust evaluation. Experiments are conducted under 10-shot and 100-shot conditions, representing few-shot and extremely few-shot scenarios.

Table 2: **Few-shot evaluation.** The average normalized test MAEs of MoMa and JMP-FT under varying data settings. MoMa consistently outperforms JMP-FT in all settings.

	10-shot	100-shot	Full data
JMP-FT	0.2217	0.4076	0.7003
MoMa	**0.1871**	**0.2990**	**0.5503**

Results The average normalized test MAEs[1] for the 17 downstream tasks of MoMa compared to JMP-FT across the full-data, 100-data, and 10-data settings are presented in Table 7. As expected, the test loss increases as the data size decreases, while MoMa consistently outperforms JMP-FT in all settings. Notably, the performance advantage of MoMa is more pronounced in the few-shot settings, with the normalized loss margin widening from 0.03 in the full-data setting to 0.11 and 0.15 in the 100-data and 10-data setting. This suggests that MoMa may offer even greater performance gains in real-world scenarios, where property labels are often limited, thereby hindering the effective fine-tuning of large pre-trained models. Complete results are shown in Section C.5.

[1]Computed by dividing the test MAE of each task by its standard deviation.

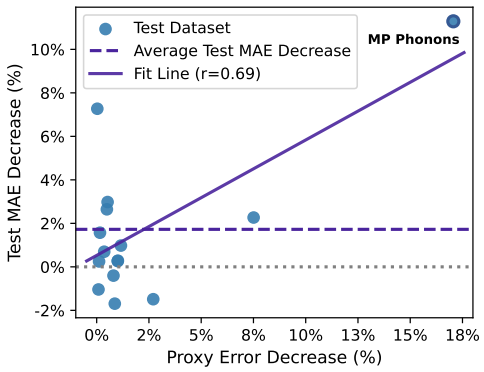

Figure 5: Scatter plot showing the relationship between the test MAE decrease and the proxy error (Eq. (3)) decrease after adding QM9 modules. The solid line represents a linear regression fit, yielding a Spearman correlation of 0.69.

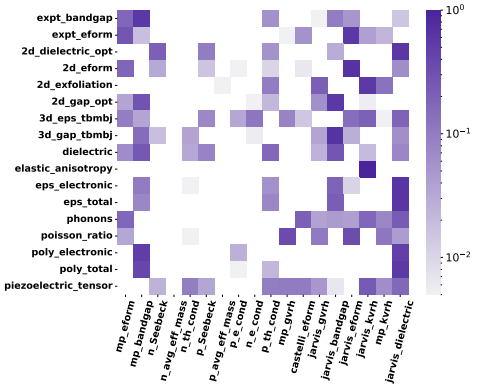

Figure 6: Heat map of AMC weights on one data split. The x-axis represents the task names of the MoMa Hub modules, while the y-axis shows the 17 material tasks in Table 1. Darker color indicate a stronger correlation.

3.5 Continual Learning Experiments

Motivation & Setup Continual learning refers to the ability of an intelligent system to progressively improve by integrating new knowledge (Wang et al., 2024). We investigate this capability of MoMa by incorporating new modules into MoMa Hub. Due to its modular nature, it is expected that MoMa will exhibit enhanced performance in tasks closely aligned with the new modules, while maintaining its performance when these additions are less relevant. We expand MoMa Hub to include 12 QM9 modules (Ramakrishnan et al., 2014) and evaluate on the 17 benchmark material tasks. QM9 comprises 12 quantum chemical properties (including geometric, electronic, energetic, and thermodynamic properties) for 134,000 stable small organic molecules, and is widely used as a benchmark for predicting structure–property relationships in small molecules.

Results We present the scatter plot of the reduction rate of test MAE w.r.t. the proxy error decrease in Fig. 5 across datasets where QM9 modules are selected. We observe that: (1) The integration of QM9 modules leads to an average of 1.7% decrease in test set MAE; (2) a larger reduction in the AMC-optimized proxy error correlates with greater performance improvements post-fine-tuning (with a Pearson correlation of 0.69). We highlight the task of MP Phonons prediction, which marks a significant 11.8% decrease in test set MAE following the expansion of MoMa Hub.

3.6 Materials Insights Mining

Motivation We argue that the AMC weights derived in Eq. (3) can provide valuable insights into the relationships of material properties. To explore this, we interpret the weights as indicators for the relationships between MoMa Hub modules and downstream tasks. Following Chang et al. (2022), we present a log-normalized visualization of these weights in Fig. 6.

Results We highlight several noteworthy observations. The weights assigned by AMC effectively capture physically intuitive relationships between material properties. For instance, in predicting electronic dielectric constants, MoMa assigns high weights to the band gap modules, which is reasonable given the inverse relationship between the dielectric constant and the square of the band gap (Ravichandran et al., 2016). At the same time, less-intuitive relationships also emerge. For the task of experimental band gap prediction (row 1), the formation energy module from the Materials Project (column 1) is assigned the second-highest weight. In the prediction of dielectric constant (row 9), modules related to thermoelectric and thermal properties (columns 5 and 6) are heavily weighted. However, the first-principles relationship between these tasks is indirect. We hypothesize that in addition to task relevance, other factors such as data distribution and size may also influence the weight assignments for AMC. Further investigation into these results is left to future work.

4 Related Work

4.1 Material Property Prediction with Deep Learning

Deep learning methods have been widely adopted for predicting material properties (De Breuck et al., 2021). The seminal CGCNN model (Xie & Grossman, 2018) represents crystalline materials with multi-edge graphs and applies graph neural networks for representation learning. Subsequent work (Choudhary & DeCost, 2021; Das et al., 2023; Yan et al., 2024; Taniai et al., 2024) has focused on improving neural network architectures to better model the inductive biases of crystals.

Another line of work develops pre-training strategies for materials (Jha et al., 2019; Magar et al., 2022; Wang et al., 2024; Song et al., 2024; Wang et al., 2025). Recently, a series of large force field models (Merchant et al., 2023; Batatia et al., 2023; Neumann et al., 2024; Wood et al., 2025) are trained on massive Potential Energy Surface data (Barroso-Luque et al., 2024) and achieve remarkable accuracy in material tasks (e.g. thermal stability prediction (Riebesell et al., 2025)). Notably, the JMP model (Shoghi et al., 2024), trained across multiple domains (small molecules, catalysts, etc.), performs impressively when fine-tuned on both molecular and crystalline tasks.

Extending beyond these methods, MoMa offers a modular strategy to centralize diverse material knowledge into modules and adaptively compose them, yielding superior downstream performance.

4.2 Modular Deep Learning

Modular deep learning (Pfeiffer et al., 2023; Xiao et al., 2024) represents a promising paradigm where parameterized modules (Jacobs et al., 1991; Houlsby et al., 2019; Hu et al., 2021) are composed, selected, and aggregated for function specialization and reuse. Notable examples of modular networks include mixture-of-experts (Jacobs et al., 1991; Shazeer et al., 2016), adapters (Houlsby et al., 2019) and LoRA (Hu et al., 2021). Recently, we have seen an increasing number of successful applications of modular methods across domains such as NLP (Pfeiffer et al., 2020; Huang et al., 2024; Tan et al., 2024) and CV (Puigcerver et al., 2020; Pham et al., 2024), where its strengths in flexibility and minimizing negative interference have been demonstrated.

However, for material property prediction, modular learning remains largely under-explored. The most related work is the mixture-of-experts framework MoE-(18) (Chang et al., 2022), which loads all pre-trained modules indiscriminately for each task and learns a routing network for embedding aggregation. In contrast, MoMa adaptively composes a subset of relevant modules with AMC into one tailored module, which is (1) explicitly selective for better mitigation of conflicting knowledge and (2) more efficient that allows for further scaling to include even more modules.

5 Conclusion

In this paper, we present MoMa, a simple modular learning framework for material property prediction. Motivated by the challenges of diversity and disparity in materials, MoMa first trains specialized modules across a wide spectrum of material tasks, constituting MoMa Hub. We then introduce the Adaptive Module Composition algorithm, which facilitates tailored adaptation from MoMa Hub to each downstream task by adaptively composing synergistic modules. Experimental results across 17 datasets demonstrate the superiority of MoMa, with few-shot and continual learning experiments further highlighting its data efficiency and scalability.

Limitations and Future Work The current scope of our study is limited to crystalline and organic materials. Future work includes expanding MoMa Hub with modules for a wider range of material data and prediction tasks, and examining how MoMa scales with hundreds or thousands of modules, which may yield deeper insights into the modularity of materials knowledge.

Broader Impact As an open-source platform for modularizing and distributing materials knowledge, MoMa enables secure sharing of modules without exposing proprietary data, efficient customization for downstream tasks, and improved prediction accuracy even in low-data scenarios. We envision MoMa fostering a new paradigm of modular material learning and driving broader community collaboration toward accelerated materials discovery.

REFERENCES

Jehad Abed, Jiheon Kim, Muhammed Shuaibi, Brook Wander, Boris Duijf, Suhas Mahesh, Hyeonseok Lee, Vahe Gharakhanyan, Sjoerd Hoogland, Erdem Irtem, et al. Open catalyst experiments 2024 (ocx24): Bridging experiments and computational models. *arXiv preprint arXiv:2411.11783*, 2024.

Luis Barroso-Luque, Muhammed Shuaibi, Xiang Fu, Brandon M Wood, Misko Dzamba, Meng Gao, Ammar Rizvi, C Lawrence Zitnick, and Zachary W Ulissi. Open materials 2024 (omat24) inorganic materials dataset and models. *arXiv preprint arXiv:2410.12771*, 2024.

Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M Elena, Dávid P Kovács, Janosh Riebesell, Xavier R Advincula, Mark Asta, Matthew Avaylon, William J Baldwin, et al. A foundation model for atomistic materials chemistry. *arXiv preprint arXiv:2401.00096*, 2023.

Ivano E Castelli, David D Landis, Kristian S Thygesen, Søren Dahl, Ib Chorkendorff, Thomas F Jaramillo, and Karsten W Jacobsen. New cubic perovskites for one-and two-photon water splitting using the computational materials repository. *Energy & Environmental Science*, 5(10):9034–9043, 2012.

Rees Chang, Yu-Xiong Wang, and Elif Ertekin. Towards overcoming data scarcity in materials science: unifying models and datasets with a mixture of experts framework. *npj Computational Materials*, 8(1):242, 2022.

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

Kamal Choudhary, Irina Kalish, Ryan Beams, and Francesca Tavazza. High-throughput identification and characterization of two-dimensional materials using density functional theory. *Scientific reports*, 7(1):5179, 2017.

Kamal Choudhary, Gowoon Cheon, Evan Reed, and Francesca Tavazza. Elastic properties of bulk and low-dimensional materials using van der waals density functional. *Physical Review B*, 98(1): 014107, 2018.

Kamal Choudhary, Kevin F Garrity, Andrew CE Reid, Brian DeCost, Adam J Biacchi, Angela R Hight Walker, Zachary Trautt, Jason Hattrick-Simpers, A Gilad Kusne, Andrea Centrone, et al. The joint automated repository for various integrated simulations (jarvis) for data-driven materials design. *npj computational materials*, 6(1):173, 2020.

Kishalay Das, Pawan Goyal, Seung-Cheol Lee, Satadeep Bhattacharjee, and Niloy Ganguly. Crysmmnet: multimodal representation for crystal property prediction. In *Uncertainty in Artificial Intelligence*, pp. 507–517. PMLR, 2023.

Pierre-Paul De Breuck, Geoffroy Hautier, and Gian-Marco Rignanese. Materials property prediction for limited datasets enabled by feature selection and joint learning with modnet. *npj computational materials*, 7(1):83, 2021.

Maarten De Jong, Wei Chen, Thomas Angsten, Anubhav Jain, Randy Notestine, Anthony Gamst, Marcel Sluiter, Chaitanya Krishna Ande, Sybrand Van Der Zwaag, Jose J Plata, et al. Charting the complete elastic properties of inorganic crystalline compounds. *Scientific data*, 2(1):1–13, 2015a.

Maarten De Jong, Wei Chen, Henry Geerlings, Mark Asta, and Kristin Aslaug Persson. A database to enable discovery and design of piezoelectric materials. *Scientific data*, 2(1):1–13, 2015b.

Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Steven Diamond, Eric Chu, and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization, version 0.2. `http://cvxpy.org/`, May 2014.

Alexander Dunn, Qi Wang, Alex Ganose, Daniel Dopp, and Anubhav Jain. Benchmarking materials property prediction methods: the matbench test set and automatminer reference algorithm. *npj Computational Materials*, 6(1):138, 2020.

Lenz Fiedler, Karan Shah, Michael Bussmann, and Attila Cangi. Deep dive into machine learning density functional theory for materials science and chemistry. *Physical Review Materials*, 6(4): 040301, 2022.

Johannes Gasteiger, Muhammed Shuaibi, Anuroop Sriram, Stephan Günnemann, Zachary Ward Ulissi, C Lawrence Zitnick, and Abhishek Das. Gemnet-oc: Developing graph neural networks for large and diverse molecular simulation datasets. *Transactions on Machine Learning Research*, 2022.

Sean D Griesemer, Yi Xia, and Chris Wolverton. Accelerating the prediction of stable materials with machine learning. *Nature Computational Science*, 3(11):934–945, 2023.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lorahub: Efficient cross-task generalization via dynamic lora composition. In *First Conference on Language Modeling*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5070–5079, 2019.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1), 2013.

Anubhav Jain, Yongwoo Shin, and Kristin A Persson. Computational predictions of energy materials using density functional theory. *Nature Reviews Materials*, 1(1):1–13, 2016.

Dipendra Jha, Kamal Choudhary, Francesca Tavazza, Wei-keng Liao, Alok Choudhary, Carelyn Campbell, and Ankit Agrawal. Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning. *Nature communications*, 10(1):5316, 2019.

George Kim, SV Meschel, Philip Nash, and Wei Chen. Experimental formation enthalpies for intermetallic phases and other inorganic compounds. *Scientific data*, 4(1):1–11, 2017.

Janice Lan, Aini Palizhati, Muhammed Shuaibi, Brandon M Wood, Brook Wander, Abhishek Das, Matt Uyttendaele, C Lawrence Zitnick, and Zachary W Ulissi. Adsorbml: a leap in efficiency for adsorption energy calculations using generalizable machine learning potentials. *npj Computational Materials*, 9(1):172, 2023.

Rishikesh Magar, Yuyang Wang, and Amir Barati Farimani. Crystal twins: self-supervised learning for crystalline material property prediction. *npj Computational Materials*, 8(1):231, 2022.

Hassan Masood, Tharmakulasingam Sirojan, Cui Ying Toe, Priyank V Kumar, Yousof Haghshenas, Patrick HL Sit, Rose Amal, Vidhyasaharan Sethu, and Wey Yang Teoh. Enhancing prediction accuracy of physical band gaps in semiconductor materials. *Cell Reports Physical Science*, 4(9), 2023.

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

Mark Neumann, James Gin, Benjamin Rhodes, Steven Bennett, Zhiyi Li, Hitarth Choubisa, Arthur Hussey, and Jonathan Godwin. Orb: A fast, scalable neural network potential. *arXiv preprint arXiv:2410.22570*, 2024.

Oleksiy Ostapenko, Zhan Su, Edoardo Maria Ponti, Laurent Charlin, Nicolas Le Roux, Matheus Pereira, Lucas Caccia, and Alessandro Sordoni. Towards modular llms by building and reusing a library of loras. *arXiv preprint arXiv:2405.11157*, 2024.

Ioannis Petousis, David Mrdjenovich, Eric Ballouz, Miao Liu, Donald Winston, Wei Chen, Tanja Graf, Thomas D Schladt, Kristin A Persson, and Fritz B Prinz. High-throughput screening of inorganic compounds for the discovery of novel dielectric and optical materials. *Scientific data*, 4(1):1–12, 2017.

Guido Petretto, Shyam Dwaraknath, Henrique PC Miranda, Donald Winston, Matteo Giantomassi, Michiel J Van Setten, Xavier Gonze, Kristin A Persson, Geoffroy Hautier, and Gian-Marco Rignanese. High-throughput density-functional perturbation theory phonons for inorganic materials. *Scientific data*, 5(1):1–12, 2018.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. Modular deep learning. *arXiv preprint arXiv:2302.11529*, 2023.

Chau Pham, Piotr Teterwak, Soren Nelson, and Bryan A Plummer. Mixturegrowth: Growing neural networks by recombining learned parameters. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2800–2809, 2024.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, Cedric Renggli, André Susano Pinto, Sylvain Gelly, Daniel Keysers, and Neil Houlsby. Scalable transfer learning with expert models. *arXiv preprint arXiv:2009.13239*, 2020.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Ram Ravichandran, Alan X Wang, and John F Wager. Solid state dielectric screening versus band gap trends and implications. *Optical materials*, 60:181–187, 2016.

Benjamin Rhodes, Sander Vandenhaute, Vaidotas Šimkus, James Gin, Jonathan Godwin, Tim Duignan, and Mark Neumann. Orb-v3: atomistic simulation at scale. *arXiv preprint arXiv:2504.06231*, 2025.

Francesco Ricci, Wei Chen, Umut Aydemir, G Jeffrey Snyder, Gian-Marco Rignanese, Anubhav Jain, and Geoffroy Hautier. An ab initio electronic transport database for inorganic materials. *Scientific data*, 4(1):1–13, 2017.

Janosh Riebesell, Rhys EA Goodall, Philipp Benner, Yuan Chiang, Bowen Deng, Gerbrand Ceder, Mark Asta, Alpha A Lee, Anubhav Jain, and Kristin A Persson. A framework to evaluate machine learning crystal stability predictions. *Nature Machine Intelligence*, 7(6):836–847, 2025.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468. PMLR, 2020.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2016.

Nima Shoghi, Adeesh Kolluru, John R Kitchin, Zachary W Ulissi, C Lawrence Zitnick, and Brandon M Wood. From molecules to materials: Pre-training large generalizable models for atomic property prediction. In *The Twelfth International Conference on Learning Representations*, 2024.

Zixing Song, Ziqiao Meng, and Irwin King. A diffusion-based pre-training framework for crystal property prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 8993–9001, 2024.

Shaomu Tan, Di Wu, and Christof Monz. Neuron specialization: Leveraging intrinsic task modularity for multilingual machine translation. *arXiv preprint arXiv:2404.11201*, 2024.

Tatsunori Taniai, Ryo Igarashi, Yuta Suzuki, Naoya Chiba, Kotaro Saito, Yoshitaka Ushiku, and Kanta Ono. Crystalformer: infinitely connected attention for periodic structure encoding. *arXiv preprint arXiv:2403.11686*, 2024.

Amanda Wang, Ryan Kingsbury, Matthew McDermott, Matthew Horton, Anubhav Jain, Shyue Ping Ong, Shyam Dwaraknath, and Kristin A Persson. A framework for quantifying uncertainty in dft energy corrections. *Scientific reports*, 11(1):15496, 2021.

Han Wang, Duo Zhang, Chun Cai, Wentao Li, Yuanchang Zhou, Jinzhe Zeng, Mingyu Guo, Chengqian Zhang, Bowen Li, Hong Jiang, et al. A graph neural network for the era of large atomistic models. 2025.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Logan Ward, Alexander Dunn, Alireza Faghaninia, Nils ER Zimmermann, Saurabh Bajaj, Qi Wang, Joseph Montoya, Jiming Chen, Kyle Bystrom, Maxwell Dylla, et al. Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, 152:60–69, 2018.

Brandon M Wood, Misko Dzamba, Xiang Fu, Meng Gao, Muhammed Shuaibi, Luis Barroso-Luque, Kareem Abdelmaqsoud, Vahe Gharakhanyan, John R Kitchin, Daniel S Levine, et al. Uma: A family of universal models for atoms. *arXiv preprint arXiv:2506.23971*, 2025.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Chaojun Xiao, Zhengyan Zhang, Chenyang Song, Dazhi Jiang, Feng Yao, Xu Han, Xiaozhi Wang, Shuo Wang, Yufei Huang, Guanyu Lin, et al. Configurable foundation models: Building llms from a modular perspective. *arXiv preprint arXiv:2409.02877*, 2024.

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

Keqiang Yan, Cong Fu, Xiaofeng Qian, Xiaoning Qian, and Shuiwang Ji. Complete and efficient graph transformers for crystal material property prediction. *arXiv preprint arXiv:2403.11857*, 2024.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. *arXiv preprint arXiv:2310.02575*, 2023.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *arXiv preprint arXiv:2408.07666*, 2024a.

Han Yang, Chenxi Hu, Yichi Zhou, Xixian Liu, Yu Shi, Jielan Li, Guanzhi Li, Zekun Chen, Shuizhou Chen, Claudio Zeni, et al. Mattersim: A deep learning atomistic model across elements, temperatures and pressures. *arXiv preprint arXiv:2405.04967*, 2024b.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

Zhanpeng Zhou, Zijun Chen, Yilan Chen, Bo Zhang, and Junchi Yan. On the emergence of cross-task linearity in pretraining-finetuning paradigm. In *Forty-first International Conference on Machine Learning*, 2024.

Zhi-Hua Zhou. Learnware: on the future of machine learning. *Frontiers Comput. Sci.*, 10(4):589–590, 2016.

Zhi-Hua Zhou, Jianxin Wu, and Wei Tang. Ensembling neural networks: many could be better than all. *Artificial intelligence*, 137(1-2):239–263, 2002.

Didi Zhu, Yibing Song, Tao Shen, Ziyu Zhao, Jinluan Yang, Min Zhang, and Chao Wu. Remedy: Recipe merging dynamics in large vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

A Algorithm for Adaptive Module Composition

The formal description of the Adaptive Module Composition algorithm is included in Algorithm 1.

Algorithm 1 Adaptive Module Composition (AMC)

Input: MoMa Hub $\mathcal{H} = \{g_j\}_{j=1}^{N}$, Downstream training set $\mathcal{D} = \{(X_i, y_i)\}_{i=1}^{m}$.
Output: Composed module $g_{\mathcal{D}}$.
{*1. Module Prediction Estimation*}
for $j = 1 \to N$ **do**
  Generate embeddings $\mathcal{X}^j \leftarrow \{g_j(X_i) \mid (X_i, y_i) \in \mathcal{D}\}$.
  Compute prediction vector $\hat{\mathbf{Y}}^j = (\hat{y}_1^j, \ldots, \hat{y}_m^j)$ via leave-one-out k-Nearest Neighbors.
end for
{*2. Module Weight Optimization*}
Let $\mathbf{Y} = (y_1, \ldots, y_m)$ be the vector of true labels from $\mathcal{D}$.
Find optimal weights $\boldsymbol{w}^* = (w_1^*, \ldots, w_N^*)$ by solving the convex optimization problem:
$\boldsymbol{w}^* \leftarrow \arg\min_{\boldsymbol{w}} \left\| \sum_{j=1}^{N} w_j \hat{\mathbf{Y}}^j - \mathbf{Y} \right\|_2^2$
  subject to: $\sum_{j=1}^{N} w_j = 1$ and $w_j \geq 0$ for all $j \in \{1, \ldots, N\}$.
{*3. Module Composition*}
$g_{\mathcal{D}} \leftarrow \sum_{j=1}^{N} w_j^* g_j$
Return $g_{\mathcal{D}}$

B Experimental Details

Here we provide more experimental details regarding the datasets, baselines, and implementation.

B.1 Dataset Details

We primarily adopt the dataset setup proposed by Chang et al. (2022). Specifically, we select 35 datasets from Matminer (Ward et al., 2018) for our study, categorizing them into 18 high-resource material datasets, with sample sizes ranging from 10,000 to 132,000 (an average of 35,000 samples), and 17 low-data datasets, with sample sizes ranging from 522 to 8,043 (an average of 2,111 samples).

The high-resource datasets are utilized for training the MoMa Hub modules, as their larger data volumes are likely to encompass a wealth of transferrable material knowledge. A detailed introduction of these MoMa Hub datasets is included in Table 3.

The low-data datasets serve as downstream tasks to evaluate the effectiveness of MoMa and its baselines. A detailed introduction is included in Table 4. This setup mimics real-world materials

Table 3: Datasets for training MoMa Hub modules. **Num** stands for the number of samples in each dataset.

Datasets	Num	Description
MP E_f	132752	The energy change during the formation of a compound from its elements. Data from Jain et al. (2013).
MP E_g	106113	The PBE band gaps, calculated using the Perdew-Burke-Ernzerhof (PBE) functional, represent the energy difference between the valence and conduction bands in a material. Data from Jain et al. (2013).
MP G_{VRH}	10987	VRH-average shear modulus, an approximate value obtained by averaging the shear modulus of polycrystalline materials. Data from Jain et al. (2013).
MP K_{VRH}	10987	VRH-average bulk modulus, calculated by averaging the Voigt (upper bound) and Reuss (lower bound) bulk moduli. Data from Jain et al. (2013).
n-type σ_e	37390	n-type σ_e measures the material's conductivity performance when electrons are the primary charge carriers. Data from Ricci et al. (2017).
p-type σ_e	37390	Similar to n-type σ_e, with holes as carriers. Data from Ricci et al. (2017).
n-type κ_e	37390	n-type κ_e evaluates the efficiency of n-type materials that can conduct both electricity and heat, which is crucial for understanding its performance in thermoelectric applications. Data from Ricci et al. (2017).
p-type κ_e	37390	Similar to n-type κ_e, with holes as carriers. Data from Ricci et al. (2017).
n-type S	37390	n-type S denotes the average conductivity eigenvalue, which measures thermoelectric conversion efficiency in the hole-conducting state when electrons act as the primary charge carriers. Data from Ricci et al. (2017).
p-type S	37390	Similar to n-type S, with holes as carriers. Data from Ricci et al. (2017).
n-type $\overline{m}_e^*$	21037	n-type $\overline{m}_e^*$ denotes the average eigenvalue of conductivity effective mass, which measures the impact of the electron's effective mass on the electrical conductivity. Data from Ricci et al. (2017).
p-type $\overline{m}_e^*$	20270	Similar to n-type $\overline{m}_e^*$, with holes as carriers. Data from Ricci et al. (2017).
Perovskite E_f	18928	Perovskite E_f refers to the heat of formation of perovskite, the amount of heat released or absorbed when the perovskite structure is formed from its constituent elements. Data from Castelli et al. (2012).
JARVIS E_f	25923	Formation energy from the JARVIS dataset (Choudhary et al., 2020).
JARVIS dielectric constant (Opt)	19027	Dielectric constant measures the material's ability to polarize in response to an electric field in two-dimensional systems. Data from Choudhary et al. (2020).
JARVIS E_g	23455	PBE band gaps from the JARVIS dataset (Choudhary et al., 2020).
JARVIS G_{VRH}	10855	VRH-average shear modulus from the JARVIS dataset (Choudhary et al., 2020).
JARVIS K_{VRH}	11028	VRH-average bulk modulus from the JARVIS dataset (Choudhary et al., 2020).

discovery scenarios, where downstream data are often scarce. Compared to the benchmark in Chang et al. (2022), we exclude two low-data datasets with exceptionally small data sizes (fewer than 20 test samples) from our experiments, as their limited data could lead to unreliable conclusions.

Following Chang et al. (2022), all datasets are split into training, validation, and test sets with a ratio of 7:1.5:1.5. For the downstream low-data datasets, we follow the exact splitting provided by Chang et al. (2022) to ensure a fair comparison.

B.2 Implementation Details of MoMa

Module Architecture Details We now introduce the architectural details of MoMa modules. Across all our experiments in the main text, the JMP (Shoghi et al., 2024) backbone is adopted due to its comprehensive strength across a wide range of molecular and crystal tasks. JMP is pre-trained on $\sim$ 120 million DFT-generated force-field data across large-scale datasets on catalyst and small molecules. It is a 6-layer GNN model with around 160M parameters which is based on the GemNet-OC architecture (Gasteiger et al., 2022). Note that MoMa is backbone-agnostic and we include results with the Orb model (Neumann et al., 2024) in Section C.1.

For the full module parametrization, we exclude the output layer and treat the entire GNN backbone as a single module. For the adapter components, we follow the standard implementation of adapter

Table 4: Downstream evaluation datasets.

Datasets	Num	Description
Experimental Band Gap (eV)	2481	The band gap of a material as measured through physical experiments. Data from Ward et al. (2018).
Formation Enthalpy (eV/atom)	1709	The energy change for forming a compound from its elements, crucial for defining Gibbs energy of formation. Data from Wang et al. (2021); Kim et al. (2017).
2D Dielectric Constant	522	The dielectric constant of 2D materials from Choudhary et al. (2017).
2D Formation Energy (eV/atom)	633	The energy change associated with the formation of 2D materials from their constituent elements. Data from Choudhary et al. (2017).
Exfoliation Energy (meV/atom)	636	The energy required to separate a single or few layers from bulk materials. Data from Choudhary et al. (2017).
2D Band Gap (eV)	522	The band gap of 2D materials from Choudhary et al. (2017).
3D Poly Electronic	8043	Poly electronic of 3D materials from Choudhary et al. (2018).
3D Band Gap (eV)	7348	The band gap of 3D materials from Choudhary et al. (2018).
Refractive Index	4764	The quantitative change of the speed of light as it passes through different media. Data from Dunn et al. (2020); Petousis et al. (2017).
Elastic Anisotropy	1181	The directional dependence of a material's elastic properties. Data from De Jong et al. (2015a).
Electronic Dielectric Constant	1296	Electronic dielectric constant refers to the dielectric response caused by electronic polarization under an applied electric field. Data from Petretto et al. (2018).
Dielectric Constant	1296	Dielectric constant of materials from Petretto et al. (2018).
Phonons Mode Peak	1265	Phonon mode peak refers to the peak in the phonon spectrum caused by specific phonon modes. Data from Petretto et al. (2018).
Poisson Ratio	1181	Poisson Ratio quantifies the ratio of transverse strain to axial strain in a material under uniaxial stress, reflecting its elastic deformation behavior. Data from De Jong et al. (2015a).
Poly Electronic	1056	The Average eigenvalue of the dielectric tensor's electronic component, where the dielectric tensor links a material's internal and external fields. Data from Petousis et al. (2017).
Poly Total	1056	The Average dielectric tensor eigenvalue. Data from Petousis et al. (2017).
Piezoelectric Modulus	941	Piezoelectric modulus measures a material's ability to convert mechanical stress into electric charge or vice versa. Data from De Jong et al. (2015b).

layers (Houlsby et al., 2019). Specifically, an adapter layer is inserted between every two layers of the JMP backbone. Each adapter consists of a downward projection to a bottleneck dimension, followed by an upward projection back to the original dimension. We adopt BERT-style initialization (Devlin, 2018), with the bottleneck dimension set to half of the input embedding dimension. Note that the merging process for adapters is performed in a layer-wise manner. For each backbone layer containing adapters, we compute a weighted average of the parameters from all selected adapter modules. A single scalar weight for each module, determined by AMC, is applied uniformly across all adapter layers belonging to that module.

Hyper-parameters For the training of JMP backbone, we mainly follow the hyper-parameter configurations in Shoghi et al. (2024), with slight modifications to the learning rate and batch size. During the module training stage of MoMa, we use a batch size of 64 and a learning rate of 5e-4 for 80 epochs. During downstream fine-tuning, we adopt a batch size of 32 and a learning rate of 8e-5. We set the training epoch as 60, with an early stopping patience of 10 epochs to prevent over-fitting. We adopt mean pooling of embedding for all properties since it performs significantly better than sum pooling in certain tasks (e.g. band gap prediction), which echos the findings in Shoghi et al. (2024).

For the Adaptive Module Composition (AMC) algorithm, we set the number of nearest neighbors (K in Eq. (1)) to 5. For the optimization problem formulated in Eq. (3), we utilize the CPLEX optimizer from the cvxpy package (Diamond et al., 2014). AMC is applied separately for each random split of the downstream tasks to avoid data leakage.

Computational Cost Experiments are conducted on NVIDIA A100 80 GB GPUs. During the module training stage, training time ranges from 30 to 300 GPU hours, depending on the dataset size. While this training process is computationally expensive, it is a one-time investment, as the trained models are stored in MoMa Hub as reusable material knowledge modules. Downstream fine-tuning requires significantly less compute, ranging from 2 to 8 GPU hours based on the dataset scale. The full module and adapter module require similar training time; however, the adapter module greatly reduces memory consumption during training. The time cost of AMC is discussed in Section C.4.

B.3 Baseline Details

The CGCNN baseline refers to fine-tuning the CGCNN model (Xie & Grossman, 2018) separately on 17 downstream tasks. Conversely, MoE-(18) involves training individual CGCNN models for each dataset in MoMa Hub and subsequently integrating these models using mixture-of-experts (Jacobs et al., 1991; Shazeer et al., 2016). For the baseline results of CGCNN and MoE-(18), we reproduce the results with the open-source codebase provided by Chang et al. (2022) and follow the exactly same hyper-parameters as reported in their papers.

For JMP-FT, we use the JMP (large) checkpoint from the codebase open-sourced by Shoghi et al. (2024) and fine-tune it directly on the downstream tasks with a batch size of 64. JMP-MT adopts a multi-task pre-training strategy, training on all 18 MoMa Hub source tasks without addressing the conflicts between disparate material tasks. Starting from the same pre-trained checkpoint as JMP-FT, JMP-MT employs proportional task sampling and trains for 5 epochs across all tasks with a batch size of 16. The convergence of multi-task pre-training is indicated by a lack of further decrease in validation error on most tasks after 5 epochs. For downstream fine-tuning, both JMP-FT and JMP-MT adopt the same training scheme as the fine-tuning stage in MoMa.

B.4 Implementation Details of LoraHub Learning & Softmax Weighting

In our analysis experiments (Section 3.3), we compare AMC against two alternative module composition strategies: LoraHub Learning, a black-box optimization approach, and Softmax weighting, a non-optimized performance-based heuristic.

For the implementation of LoraHub Learning, we strictly follow the hyper-parameters and black-box optimization scheme in its official repository except that we use a training-free kNN predictor to obtain the metric in each round of optimization, which is aligned with AMC. This is because current capabilities pre-trained models cannot enable zero-shot prediction of material tasks as in LLMs.

For the implementation of Softmax weighting, we convert the predicted MAE from the same initial kNN evaluation into a weight for each module. The goal is to directly assign higher weights to modules with better predicted individual performance (i.e., lower MAE). Formally, the weight w_j for module j is calculated as:

$$w_j = \frac{\exp(-\text{MAE}_j/T)}{\sum_{k=1}^{N} \exp(-\text{MAE}_k/T)} \tag{4}$$

where the temperature T is set to 1.

C More Experimental Results

C.1 Results with More Architectures

To verify whether MoMa offers consistent benefits in other model backbones beyond JMP, we conduct additional experiments on the architecture used by the Orb model (Neumann et al., 2024). Note that Orb is based on the GNS architecture (Sanchez-Gonzalez et al., 2020) which is not equivariant and much less complex than the GemNet-based architecture (Gasteiger et al., 2022) in JMP. Specifically, we load the pre-trained checkpoint from the Orb repository, and fine-tune on the datasets in Table 3 to construct an Orb-based MoMa Hub. Then we run AMC and downstream adaptation identically as in Section 2.3. The results (Orb-MoMa) are compared with directly fine-tuning the pre-trained Orb model (Orb-FT).

As shown in Fig. 8, MoMa outperforms in 14/17 tasks and achieves a 8.2% average boost to direct fine-tuning. This indicates that the effectiveness of MoMa is consistent across GemNet-based and GNS-based architectures.

C.2 Results on More Baselines

We implement an additional baseline that adopt a mixture-of-experts scheme with the JMP backbone, termed JMP-(18), where each expert is a full JMP module in MoMa Hub. Full fine-tuning all parameters induces formidable memory cost, and is impractical considering MoMa Hub may further scale in the future. Hence, resembling Chang et al. (2022), we only unfreeze the final MLP layer as well as the router network in downstream fine-tuning.

The results comparing MoMa (Full) and JMP-(18) on all 17 benchmark datasets are shown in Fig. 9. Results are reported on one random seed. We see that MoMa beats JMP-(18) on all datasets, with a substantial 21.8% test MAE decrease, which shows the effectiveness of MoMa's adaptive module selection scheme.

C.3 Complete Results & Discussion for AMC analysis experiments

The complete results for the analysis experiments is shown in Fig. 7. LoraHub Learning and Softmax Weighting fall short of AMC in 15 and 12 out of 17 tasks, exhibiting average increases of 15.5% and 13.7% in test MAE.

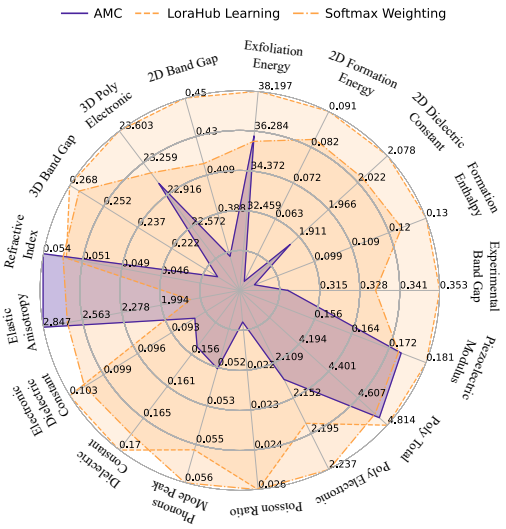

Figure 7: Analysis results for AMC. The axis shows test set MAE and **smaller area is better**.

We conjecture that AMC outperforms LoraHub Learning for two main reasons. First, AMC directly minimizes the proxy loss with *white-box* optimization, whereas LoraHub relies on *black-box* search. The availability of gradient information enables AMC to reliably converge to an optimal set of weights. Second, the JMP backbone network is complex and constitute a rugged optimization landscape, making it hard for black-box methods to navigate effectively.

The advantage of AMC over the Softmax Weighting highlights the importance of optimizing for synergy. Softmax Weighting determines each module's contribution based solely on its isolated performance, overlooking potential synergistic interactions. In contrast, AMC explicitly optimizes for the weight configuration that maximizes collective performance and captures such interactions.

C.4 Efficiency Analysis of AMC

Time Cost For the prediction estimation stage, we further divide it into the embedding generation and kNN prediction step. While these steps should be conducted separately for each module and each downstream dataset, the process can be parallelized and the runtime mainly depends on the size of the downstream dataset. As shown in Table 5, the maximum total time is below 30 seconds. For the weight optimization stage, we report the minimum and maximum time required for convergence of each downstream task (Eq. (3)). As shown in Table 6, the time cost is negligible and remains roughly constant as the number of modules scales.

Memory Cost During embedding generation, only one module is loaded into GPU at a time, requiring approximately 1.8 GB of memory. The generated embeddings are stored on CPU, with the largest set requiring about 5.5 MB. Overall, AMC is lightweight in memory usage and scales well with the number of modules.

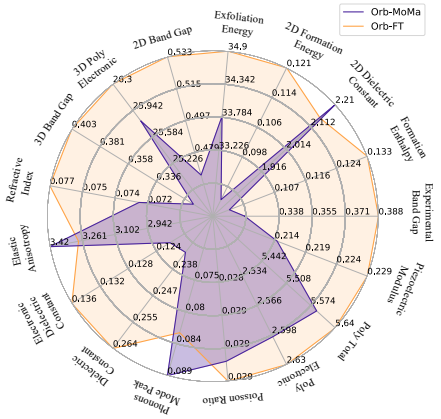

Figure 8: Orb-based MoMa results (purple) compared with the Orb-FT baseline (orange).

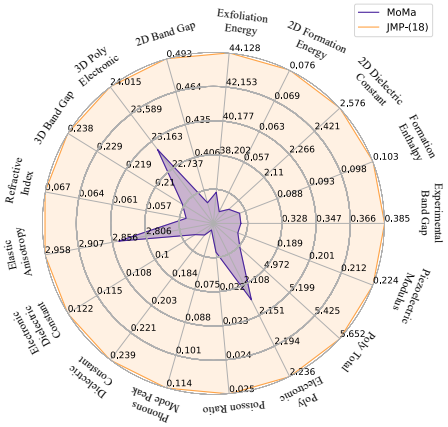

Figure 9: The comparison of MoMa (purple) with JMP-(18) (orange).

Table 5: Module prediction estimation time

	Min time (s)	Max time (s)
Embedding generation	7.29	24.06
KNN prediction	0.05	4.02
Total time	7.34	28.08

Table 6: Weight optimization time

Module #	Min time (s)	Max time (s)
3	0.07	0.08
9	0.12	0.15
18	0.14	0.25

Table 7: Test set MAE and average test loss of JMP-FT and MoMa under the full-data, 100-data, and 10-data settings. Results are averaged over five random data splits on one random seed. Results are preserved to the third significant digit.

Datasets	**JMP-FT**	**MoMa**	**JMP-FT (100)**	**MoMa (100)**	**JMP-FT (10)**	**MoMa (10)**
Experimental Band Gap	0.380	0.305	0.660	0.469	1.12	1.245
Formation Enthalpy	0.156	0.0821	0.273	0.101	0.514	0.143
2D Dielectric Constant	2.45	1.90	3.19	2.35	7.74	3.31
2D Formation Energy	0.135	0.0470	0.366	0.113	0.842	0.214
2D Exfoliation Energy	38.9	36.1	54.4	56.1	118	87.3
2D Band Gap	0.611	0.366	0.890	0.517	1.23	1.05
3D Poly Electronic	23.7	23.0	33.6	24.8	54.0	48.9
3D Band Gap	0.249	0.201	1.71	0.686	2.10	1.47
Dielectric Constant	0.0552	0.0535	0.134	0.102	0.289	0.231
Elastic Anisotropy	2.11	2.85	4.85	3.79	4.02	5.26
Electronic Dielectric Constant	0.108	0.0903	0.260	0.178	0.568	0.500
Total Dielectric Constant	0.172	0.155	0.361	0.287	0.543	0.527
Phonons Mode Peak	0.0710	0.0521	0.221	0.199	0.493	0.485
Poisson Ratio	0.0221	0.0203	0.0345	0.0317	0.0466	0.057
Poly Electronic	2.10	2.13	3.24	2.88	6.08	5.10
Total Poly	4.83	4.76	6.54	6.32	11.2	10.1
Piezoelectric Modulus	0.169	0.175	0.248	0.258	0.303	0.290
Average Normalized Test Loss	0.222	0.187	0.408	0.299	0.700	0.550

C.5 Complete Few-shot Learning Results

We present the complete results of the few-shot learning experiments in Table 7. MoMa consistently shows performance improvements across all settings, with the margin of normalized test loss increasing as dataset size shrinks. These results highlight MoMa's strong potential to retain a performance advantage in few-shot scenarios, which are prevalent in material property prediction tasks.

D Potential Societal Impact

MoMa is visioned to be an open-source platform for the sharing of materials knowledge as modules. Potential positive societal impacts include the acceleration of the discovery of new materials with desirable properties, which benefit industries such as energy, electronics, and manufacturing. However, there are risks associated with the mal-intended use of material knowledge to develop harmful or unsafe materials. To mitigate these risks, it is crucial to ensure that the application of this work adheres to ethical guidelines. Although we do not foresee significant negative consequences in the near future, we recognize the importance of responsible usage and oversight in the application of these technologies.