TOWARDS FASTER AND MORE COMPACT FOUNDATION MODELS FOR MOLECULAR PROPERTY PREDICTION

Yasir Ghunaim, Andrés Villa, Gergo Ignacz,
Gyorgy Szekely, Motasem Alfarra & Bernard Ghanem
King Abdullah University of Science and Technology (KAUST)
Correspondence to: yasir.ghunaim@kaust.edu.sa

ABSTRACT

Advancements in machine learning for molecular property prediction have improved accuracy but at the cost of increased complexity and longer training times. The recent Joint Multi-domain Pre-training (JMP) foundation model has demonstrated strong performance across various downstream tasks while reducing training time. However, fine-tuning on small-scale datasets remains time consuming, and larger datasets with more training samples pose even greater challenges. In this work, we investigate strategies to enhance efficiency by reducing model size while preserving performance. Through an analysis of layer contributions in JMP, we find that later interaction blocks provide diminishing returns, suggesting opportunities for model simplification. We explore block reduction strategies, where we prune the pre-trained model before fine-tuning, and assess their impact on efficiency and accuracy. Our findings reveal that removing two interaction blocks results in minimal performance drop, reducing model size by 32% while increasing inference throughput by 1.3×. This confirms that JMP-L is over-parameterized, and a smaller, more efficient variant can achieve comparable performance at a lower computational cost. Our study provides insights for developing lighter, faster, and more scalable foundation models for molecular and materials discovery. The code is publicly available at: github.com/Yasir-Ghunaim/efficient-jmp.

1 INTRODUCTION

Molecular property prediction using density functional theory (DFT) and molecular dynamics (MD) calculations plays a crucial role in the discovery of novel materials, including pharmaceutical drugs Sabe et al. (2021), catalysts Nørskov et al. (2011); Tran et al. (2023); Chanussot et al. (2021), metal-organic frameworks Rosen et al. (2022), and polymers Sharma et al. (2014). However, the high computational cost of DFT and MD calculations limits their feasibility for large-scale, high-throughput searches. To overcome this challenge, machine learning potentials have been developed to accelerate DFT and MD calculations Behler & Parrinello (2007); Bogojeski et al. (2020), based on the latest large-scale datasets Kolluru et al. (2022), such as Open Catalyst 2020 (OC20) Chanussot et al. (2021), Open Catalyst 2022 (OC22) Tran et al. (2023), and ODAC23 Sriram et al. (2023). However, training models from scratch for different tasks remains a major bottleneck for their widespread adoption. For example, datasets with different applied DFT theories, molecular system sizes, or chemical diversity increase complexity, thus hindering the generalizability and scalability of machine learning models in chemistry.

Recent progress in efficient pre-training strategies Zaidi et al. (2022); Zhou et al. (2023), the availability of extensive DFT and MD datasets Tran et al. (2023); Chanussot et al. (2021); Eastman et al. (2023); Smith et al. (2020), and the introduction of specialized chemical benchmarks Schreiner et al. (2022a); Dunn et al. (2020) have led to the emergence of foundation models for molecular property prediction. Foundation models such as the Joint Multi-domain Pre-training (JMP) model Shoghi et al. (2023) and MACE-MP-0 Batatia et al. (2023; 2022b;a) have demonstrated remarkable performance in diverse molecular tasks. In particular, JMP adapts the pre-train-then-finetune paradigm from vision and language tasks to molecular property prediction. By pre-training on large datasets,

JMP captures generalizable molecular representations that enable faster fine-tuning for downstream tasks, overcoming the need to train models from scratch for each new application.

Although the large variant of JMP (JMP-L) has outperformed state-of-the-art models on 34 out of 40 tasks, its efficiency in fine-tuning and inference has yet to be addressed. With 160M parameters, JMP-L achieves similar performance to MACE Batatia et al. (2022b), which uses only 3M parameters, suggesting potential over-parameterization. This over-parameterization increases memory and compute requirements and leads to higher carbon emissions Shoghi et al. (2023), reducing overall sustainability. Although 160M parameters are relatively small compared to vision and language models, the parameter-to-data ratio in molecular ML remains disproportionately large. For instance, MD17 contains only 1,000 training samples with an average of 13 nodes per graph Shoghi et al. (2023), making such a large model inefficient for small datasets.

To address these limitations, we perform an in-depth analysis of the efficiency of JMP-L. By examining its interaction block hierarchy, we find that higher-order blocks contribute less to overall performance. This observation aligns with recent findings in large language models, where deeper layers often yield diminishing returns Gromov et al. (2024). This motivates our exploration of block reduction, a pruning strategy that removes the least important layers to improve efficiency while maintaining accuracy. Additionally, we investigate knowledge distillation techniques tailored to molecular graph neural networks, integrating them with block reduction to assess their combined impact. Although pruning Liu et al. (2022) and distillation Zeng et al. (2023) are widely used in other domains, their application in molecular property prediction, particularly within the pre-train-then-finetune paradigm, remains underexplored.

Our findings reveal that pruning and distillation significantly improve the efficiency of JMP-L while preserving comparable performance for most tasks. Specifically, we show that a pruned and distilled variant of JMP-L achieves comparable accuracy to the original model across in-distribution and out-of-distribution downstream tasks. By removing two interaction blocks, we reduce the model size by 32%, decreasing parameter count from 160M to 108M, while improving inference throughput by 1.3×, compared to the baseline model. These results confirm that JMP-L is over-parameterized for many tasks, and smaller, more efficient versions can achieve similar performance with reduced computational cost.

In summary, our contributions are three-fold:

- We develop a lightweight version of JMP-L with 108M parameters (32% reduction), achieving 1.3 times faster inference while maintaining performance.
- We evaluate the impact of block reduction and knowledge distillation on pre-training across in-distribution and out-of-distribution downstream tasks.
- We demonstrate that later interaction blocks of JMP-L contribute less to performance, supporting the case for model compression.

2 Related Work

2.1 Foundation Models in Molecular Property Prediction

Pre-trained models have significantly advanced the development of robust architectures across various domains. Notable examples include ResNet He et al. (2016) and ViT Dosovitskiy et al. (2021), which leverage large-scale datasets such as ImageNet Deng et al. (2009) to enhance image processing. In contrast, deep learning models for molecular property prediction have primarily been task-specific Batatia et al. (2023); Kovács et al. (2023), limiting their utility as general-purpose pre-trained models. Recently, JMP Shoghi et al. (2023) introduced a supervised pre-training strategy on large datasets, establishing a shared knowledge base for various downstream tasks. Built on GemNet-OC Gasteiger et al. (2022), JMP is the first large-scale foundation model for molecular property prediction. However, its fine-tuning efficiency remains a challenge, as it requires more than 275 GPU hours to converge Shoghi et al. (2023). In this work, we provide a comprehensive analysis of JMP and propose a more efficient approach to reduce its computational demands, enhancing its accessibility and scalability for broader applications.

2.2 EFFICIENT TRAINING

2.2.1 PRUNING

Pruning is a technique used to reduce the size and complexity of a neural network by eliminating weights, neurons, layers, or filters without compromising accuracy Sietsma & Dow (1988); Cheng et al. (2024); Blalock et al. (2020). It is particularly effective when a model is over-parameterized for its task Sietsma & Dow (1988). Structured pruning, which removes entire layers or filters, has been shown to improve memory and computational efficiency in various architectures, including large language models Zhang et al. (2024); Sun et al. (2024), vision transformers Yu et al. (2022), and graph neural networks (GNNs) Liu et al. (2022). JMP-L, based on the GemNet-OC architecture Gasteiger et al. (2022), consists of an embedding layer, six interaction layers, and three MLP layers. Drawing inspiration from pruning techniques in other domains, we investigate the impact of removing GemNet-OC interaction layers to accelerate fine-tuning and inference while maintaining model performance.

2.2.2 KNOWLEDGE DISTILLATION

Knowledge distillation (KD) is a model compression technique that transfers knowledge from a larger teacher model to a smaller student model, aiming to achieve similar performance with reduced computational costs Hinton (2015); Bucilua et al. (2006). Initially introduced by Bucilua *et al.* Bucilua et al. (2006) and later popularized by Hinton *et al.* Hinton (2015), KD has been widely applied in the language Xu et al. (2024), vision Habib et al. (2024), and general graph domains Tian et al. (2023), mainly in classification tasks. However, large-scale regression tasks such as molecular simulations introduce unique challenges for KD in density functional theory (DFT) and molecular dynamics (MD) simulations Ekström Kelvinius et al. (2024). Molecular GNNs operate on structured data with features distributed across nodes and edges, making direct knowledge transfer from teacher to student more challenging Ekström Kelvinius et al. (2024). These challenges are amplified when the teacher and student models differ significantly in architecture, making feature alignment more difficult. To address these issues, Ekström *et al.* Ekström Kelvinius et al. (2024) propose specialized loss functions—*node2node (n2n)*, *edge2edge (e2e)*, *edge2node (e2n)*, and *vector2vector (v2v)*—to supplement standard loss functions and enhance the effectiveness of KD in molecular GNNs. These strategies help bridge the gap between teacher and student models, improving knowledge transfer in complex molecular systems.

The standard loss function $\mathcal{L}_0$ for molecular GNNs, as outlined in Eq. 1, accounts for both energy and force predictions:

$$\mathcal{L}_0 = \alpha_{\mathrm{E}}\mathcal{L}_{\mathrm{E}}(\hat{E}, E) + \alpha_{\mathcal{F}}\mathcal{L}_{\mathrm{F}}(\hat{\boldsymbol{F}}, \boldsymbol{F}) \tag{1}$$

where E and F represent the ground truth energy and forces, while $\hat{E}$ and $\hat{F}$ denote their predicted counterparts. The terms $\mathcal{L}_E$ and $\mathcal{L}_F$ correspond to the energy and force loss functions, respectively, weighted by α_{E}, $\alpha_{\mathrm{F}} \in \mathbb{R}$.

For knowledge distillation Hinton (2015), the loss function in Eq. 1 is augmented with an auxiliary distillation loss $\mathcal{L}_{\mathrm{KD}}$, resulting in the following formulation:

$$\mathcal{L} = \mathcal{L}_0 + \lambda\mathcal{L}_{\mathrm{KD}}.$$

In this work, we aim to develop a more efficient variant of the foundational model JMP-L without compromising its performance. Since JMP is built on GemNet-OC Gasteiger et al. (2022), we focus on the *node2node* and *edge2edge* losses as key components of the distillation process. Unlike previous studies that apply distillation only at the final task level, our primary objective is to enhance the efficiency of the foundational model itself while systematically assessing its impact on downstream tasks. Specifically, we examine how distillation influences the model's generalization capabilities, providing deeper insights into its performance across diverse molecular property prediction tasks.

2.2.3 PRUNING COUPLED WITH DISTILLATION

Aggressive structured pruning can severely degrade model performance. For instance, brute-force structural pruning methods, such as L2-based filter-wise pruning, have led to a 50-fold performance

drop in LLMPruner Ma et al. (2023). However, in large models, aggressive pruning combined with fine-tuning can significantly reduce the number of layers—sometimes by half—while incurring minimal performance loss Gromov et al. (2024). Techniques such as parameter-efficient fine-tuning, quantization, and low-rank adapters further help preserve model accuracy post-pruning Gromov et al. (2024). Recent KD approaches, such as those proposed by Ekström *et al.* Ekström Kelvinius et al. (2024), require training the student model from scratch, demanding substantial computational resources. To our knowledge, no prior work has explored the combined use of pruning and distillation for DFT and MD molecular property prediction. Our approach applies distillation to a pre-trained, block-reduced network, offering the potential for improved accuracy while significantly reducing both training and inference time.

3 Block Reduction for Efficient Foundation Models

3.1 Preliminaries

Our work builds upon the GemNet-OC architecture Gasteiger et al. (2022), though our analysis can be applied to similar architectures.

In particular, we define $f_\theta : \mathbb{R}^{4\times n} \to \mathbb{R}^{n\times d}$ as a function that maps a molecular graph—represented by the 3D positions and atomic numbers of n atoms—to a feature space. The feature extraction process is formulated as:

$$f(x) = \text{concat}(f_1(x), f_2 \circ f_1(x), \ldots, f_b \circ f_{b-1} \circ \cdots \circ f_2 \circ f_1(x)) \tag{2}$$

where the model extracts features through b sequential blocks. Each block $f_i : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ (for $i > 1$) refines the representations, while the initial block $f_1 : \mathbb{R}^n \to \mathbb{R}^{n\times d}$ is an embedding layer that performs the initial transformation. The resulting feature space is of dimension $\mathbb{R}^{n\times(d\times b)}$, obtained by concatenating outputs from all b blocks. This extracted feature representation is then processed through a sequence of multilayer perceptron (MLP) layers, known as FinalMLP in GemNet-OC.

$$g(x) = g_m \circ g_{m-1} \circ \cdots \circ g_2 \circ g_1(x)$$

where $g_1 : \mathbb{R}^{n\times(d\times b)} \to \mathbb{R}^{n\times d}$ transforms the concatenated features into d, with $g(x) : \mathbb{R}^{n\times(d\times b)} \to \mathbb{R}^{n\times d}$ providing the final transformation. Finally, the output of $g(x)$ is passed to a prediction head $h(x)$, which predicts the three-dimensional force vector for each atom and the molecule's energy. The full model is the composition between the feature extractor and the MLP layers given by the following:

$$F(x) = h \circ g \circ f(x). \tag{3}$$

For the GemNet-OC architecture used by JMP-L, $b = 7$ (one embedding layer and six interaction blocks) and $m = 5$, resulting in 160.1M parameters in total. JMP-L is shown on top in Figure 1.

3.2 Interaction Block Importance

Interaction blocks are a fundamental component of machine learning potential models (e.g., SchNet Schütt et al. (2017), GemNet-OC Gasteiger et al. (2021)), enabling richer representations and capturing long-range atomic interactions. These models stack multiple interaction blocks in a sequential manner to build higher-body representations and model complex atomic relationships effectively. However, quantifying each interaction block's contribution to the final prediction is not straightforward, as interactions are highly interdependent and difficult to isolate.

To address this, we propose an approach to measure interaction block importance within the GemNet-OC backbone used by JMP-L. We employ GradCAM Selvaraju et al. (2017) to assess each block's impact on the final output. First, we extract and concatenate the output features from all b blocks, as formulated in Eq.2. Using these features (f) and Eq. 3, we compute the model's output and its corresponding loss, $\mathcal{L}_0$. We then compute the gradient ∇_{CAM} of f with respect to $\mathcal{L}_0$ and determine each block's relevance r using:

$$r = \text{ReLU}\left(f \circ \nabla_{\text{CAM}}\right).$$

Following the GradCAM methodology, we apply a ReLU activation to emphasize features that positively contribute to the model's prediction. To quantify the contribution of each interaction block,

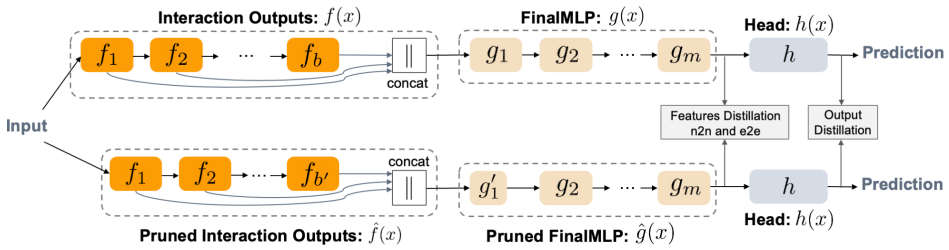

Figure 1: **Block Reduction for Efficient Foundation Models.** The top model represents the foundation model JMP-L, where interaction blocks extract features, which are concatenated and processed by FinalMLP before making predictions. The bottom model is its pruned version, constructed by removing low-importance blocks and adjusting FinalMLP. To mitigate performance degradation, we apply both feature distillation (node-to-node and edge-to-edge) and output distillation to transfer knowledge from the original model.

we decompose the relevance map r, which represents the overall importance of features, into b partitions. Each partition r_i corresponds to the feature dimension d of a specific interaction block. Finally, we compute the importance score for each block by averaging r_i across both the feature and batch dimensions, providing a measure of its overall contribution to the final prediction.

3.3 Block Reduction Strategies

Although the pre-trained GemNet-OC (JMP-L) achieves strong performance when fine-tuned across various tasks and datasets, its fine-tuning remains computationally expensive. This inefficiency arises from its large architecture and the high computational cost of each forward pass. For instance, fine-tuning JMP-L on rMD17—containing only 1,000 graphs—still requires 160.1M parameters. Given the small scale of rMD17, this parameter count appears disproportionately large, limiting the practicality of leveraging such a powerful pre-trained model efficiently.

In this work, we aim to improve the efficiency of foundation pre-trained models through block reduction. Specifically, we explore different strategies to construct a reduced model $\hat{F}(x) = h \circ \hat{g} \circ \hat{f}(x)$, where:

$$\hat{f}(x) = \text{concat}(f_1(x), f_2 \circ f_1(x), \ldots, f_{b'} \circ f_{b'-1} \circ \cdots \circ f_2 \circ f_1(x))$$

and

$$\hat{g}(x) = g_m \circ g_{m-1} \circ \cdots \circ g_2 \circ g'_1(x)$$

with $g'_1 : \mathbb{R}^{n \times (d \times b')} \to \mathbb{R}^{n \times d}$ and $b' < b$.

This formulation reduces the original architecture by removing the last $b - b'$ interaction blocks and adjusting the dimensionality of the first MLP block (g'_1) accordingly. However, removing interaction blocks disrupts the alignment between the feature extractor $\hat{f}$ and the FinalMLP $\hat{g}$, as it alters the structure of the extracted features. To address this misalignment, we explore three main strategies to restore compatibility between the reduced feature extractor and the FinalMLP.

Random MLP. Random MLP is the simplest baseline, where we resize the first MLP layer g'_1 and randomly initialize its weights. This approach assumes that the features extracted by the remaining interaction blocks are still useful and that the weights of g'_1 can be learned effectively during fine-tuning. We refer to this strategy as RandomMLP in the experimental section.

Sliced MLP. In the Sliced MLP strategy, we retain the parameters of the first MLP layer g'_1 from the original model, truncating it to match the reduced dimensionality of the features. This assumes that the preserved parameters provide a good initialization for fine-tuning, maintaining continuity between the pre-trained and pruned model. Unless otherwise stated, all of our block reduction experiments follow this strategy.

Knowledge Distillation. In the knowledge distillation approach, we introduce a learning paradigm tailored for block reduction. Specifically, we follow Ekström Kelvinius et al. (2024) by distilling the force predictions of the pre-trained model F into its block-reduced counterpart $\hat{F}$. This is achieved by optimizing the following objective:

$$\min_{\theta} \mathbb{E}_{x\sim\mathcal{D}} \|\hat{F}_\theta(x) - F_{\theta_0}(x)\|_1$$

where $\mathcal{D}$ represents the data distribution used during the pre-training phase of F.

To further align the representations between the original and pruned models, we incorporate node-to-node (n2n) and edge-to-edge (e2e) distillation. This extends the objective to:

$$\min_{\theta} \mathbb{E}_{x\sim\mathcal{D}} \left[\|\hat{F}(x;\theta) - F(x;\theta_0)\|_1 + \sum_{i=1}^{n'} \|\hat{g}_i(\hat{f}(x);\theta) - g_i(f(x);\theta_0)\|_1 \right].$$

The first term represents output distillation, while the second term ensures feature-level consistency between the pruned and original models. Our full pipeline is illustrated in Figure 1.

It is important to note that all prior block reduction approaches operate on the pre-trained foundational model. Thus, block reduction produces a generalist model, which must still undergo fine-tuning on task-specific datasets to become a specialized model for a given application.

4 Experiments

4.1 Deeper layers contribute less

To develop a more efficient pre-trained version of JMP-L for diverse downstream tasks, we analyze the contribution of each interaction block to the final output prediction on the pre-training distribution, as outlined in Sec.3.2. Figure 2 reveals a gradual decline in relevance toward the deeper layers, with the sixth and seventh blocks exhibiting the lowest contribution. This suggests that these blocks can be removed with minimal impact on performance. While the embedding block (f_1) also shows low relevance, it directly feeds into the first interaction block (f_2), which initializes message passing and is crucial for propagating atomic interactions. Removing the embedding block could introduce significant structural disruptions, potentially degrading performance, and is therefore not considered for block reduction.

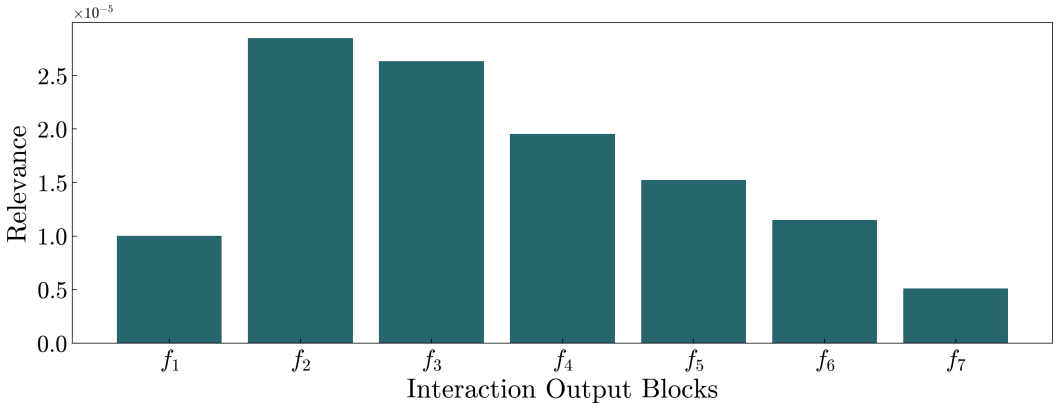

Figure 2: **Block Relevance Analysis.** We show the contribution of each output block in JMP-L to the final prediction. f_1 represents the embedding output, while f_2 to f_7 correspond to the six interaction blocks. We observe diminishing returns in deeper interaction blocks, making them strong candidates for pruning.

4.2 Block Reduction and Distillation in Pre-training

We investigate the effectiveness of block reduction (BR) and knowledge distillation (KD) during the pre-training phase. As shown in our block importance analysis, the deeper interaction blocks in

Table 1: **Force MAE Evaluation During Pre-training.** We evaluate the impact of our block reduction (BR) approach on pre-training performance, comparing results with and without knowledge distillation (KD). The reported values represent the force MAE (meV/Å) across OC20, OC22, ANI-1x, and Transition-1x datasets. Note: The block count excludes the embedding output.

	# of Blocks	OC20	OC22	ANI-1x	Transition-1x
BR	2	106.6 (-89.6)	111.9 (-89.0)	711.8 (-689.3)	188.3 (-175.5)
+ KD	2	52.1 (-35.1)	56.3 (-33.4)	144.4 (-121.9)	53.3 (-40.6)
BR	3	94.1 (-77.1)	99.3 (-76.3)	615.1 (-592.6)	149.6 (-136.9)
+ KD	3	42.1 (-25.2)	45.6 (-22.7)	97.6 (-75.1)	39.1 (-26.4)
BR	4	65.3 (-48.4)	68.9 (-46.0)	443.4 (-421.0)	99.8 (-87.1)
+ KD	4	26.7 (-9.8)	34.5 (-11.6)	58.1 (-35.7)	24.2 (-11.4)
BR	5	39.1 (-22.1)	45.7 (-22.7)	220.0 (-197.6)	48.5 (-35.8)
+ KD	5	19.8 (-2.8)	25.8 (-2.9)	30.1 (-7.7)	16.0 (-3.3)
JMP-L (Teacher)	6	17.0	22.9	22.5	12.7

JMP-L contribute less to the final prediction compared to the earlier ones. This raises a key question: how much do these later blocks impact prediction performance, and if their removal leads to degradation, to what extent can knowledge distillation recover the lost performance? To answer this, we explore a combined approach of block reduction and knowledge distillation, assessing whether distilling knowledge from the full model into a block-reduced version can maintain performance while significantly reducing computational costs.

Settings: We use the same pre-training datasets as in Shoghi et al. (2023), including OC20 Chanussot et al. (2021), OC22 Tran et al. (2023), ANI-1x Smith et al. (2020), and Transition-1x Schreiner et al. (2022b), totaling 120M training samples. Following our block reduction (BR) strategy, we sequentially remove interaction blocks starting from the last one, as indicated by the importance analysis in the previous section. This allows us to construct progressively smaller versions of JMP-L, retaining 5, 4, 3, and 2 interaction blocks. To mitigate potential performance degradation, we apply a brief knowledge distillation (KD) phase to each pruned model using less than 1.5% of the pre-training datasets. We find that running KD for under 2 GPU-days on an A100 is sufficient for convergence. We report performance using the mean absolute error (MAE).

Observations: Table 1 presents the results of block reduction (BR) and knowledge distillation (KD) during the pre-training stage. As expected, removing interaction blocks leads to a performance drop proportional to the number of blocks removed. Notably, OC20 and OC22 show smaller performance drops, likely due to their higher force loss weight during pre-training Shoghi et al. (2023), making the model less sensitive to deeper block removal. Applying both BR and KD significantly reduces the performance gap with the teacher model. For instance, with 5 blocks, the performance difference narrows to just -2.8 and -2.9 meV/Å for OC20 and OC22, respectively. These results demonstrate the effectiveness of combining BR and KD in maintaining predictive accuracy during pre-training. Next, we analyze how well these pruned models perform on downstream tasks.

4.3 Main Results

We now evaluate the pruned versions of JMP-L across different downstream tasks and knowledge transfer strategies, aiming to identify the most efficient fine-tuning approach for optimal downstream performance. A key question is whether distilling during pre-training is more effective than applying block reduction alone. To answer this, we compare against the following baselines.

Baselines: We evaluate the following fine-tuning strategies:

- BR (Block Reduction): Remove interaction blocks and slice their corresponding weights in the first layer of FinalMLP.
- BR/RandomMLP: A simpler variant of BR, where instead of pruning the first layer of FinalMLP, we randomly initialize a smaller version to match the reduced number of interaction blocks.

- BR+KD (Block Reduction + Knowledge Distillation): Load the pruned and distilled version of JMP-L, where knowledge distillation has been applied during pre-training.

In addition, we report the original performance of JMP-L from Shoghi et al. (2023) alongside our reproduced version. The key distinction is that our fine-tuning process is constrained by a fixed computational budget, which we describe next.

Settings: We evaluate the baselines on a representative set of targets from the datasets used in Shoghi et al. (2023), specifically: Aspirin (rMD17 Chmiela et al. (2017)), U_0 (QM9 Ramakrishnan et al. (2014)), Solvated Amino Acids (SPICE Eastman et al. (2023)), Ac-Ala3-NHMe (MD22 Chmiela et al. (2023)), and Band Gap (QMOF Rosen et al. (2021)). For a fair comparison, since the baselines differ in computational demands, we fine-tune each model for 1 GPU-day on a V100, except for QM9, which requires 2 GPU-days to approach convergence. We then evaluate the models on the test set of the corresponding dataset and target. We present our findings in Figure 3.

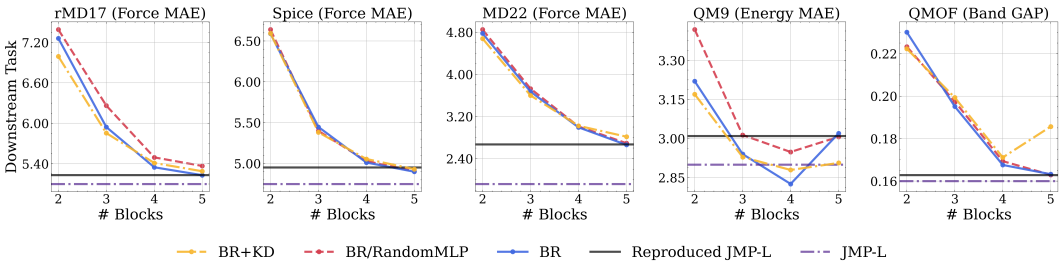

Figure 3: **Evaluation on downstream tasks.** We evaluate the performance across various downstream tasks using different block reduction strategies: block reduction (BR), block reduction with a randomly initialized MLP (BR/RandomMLP), and block reduction combined with knowledge distillation (BR+KD). Performance is measured in MAE: meV/Å for force targets, meV for the QM9 energy target, and eV for the QMOF band gap target. The original JMP-L model utilizes 6 blocks.

Block Reduction is a Strong Baseline: Figure 3 highlights the surprising effectiveness of BR (shown in blue) across datasets. In particular, the 5-block model matches the performance of the 6-block reproduced JMP-L baseline. Even the 4-block model remains competitive across most tasks and outperforms the original JMP-L on the QM9 target. These results suggest that for tasks with longer convergence times, such as QM9, a compressed model not only reduces computational costs but may also converge faster and even surpass the full model's performance. However, when further reducing to 3 or 2 blocks, we observe a more pronounced drop in performance, indicating that the model may be underfitting the task.

Pre-training Distillation Works in Certain Scenarios: While KD improved performance on the pre-training datasets (as discussed in Section 4.2), its effectiveness on downstream tasks varies, as shown by BR+KD (yellow in Figure 3). For example, KD improves performance on rMD17 when using 2 or 3 blocks, but it hurts the performance with 4 and 5 blocks. This could be due to the fact that the last two blocks of JMP-L contribute less to downstream tasks (as indicated by BR results), meaning that distilling from these less relevant blocks during pre-training may introduce noise and distort useful features. Interestingly, KD improves performance for the 5-block model in QM9, suggesting a potential edge case where distillation benefits from specific task characteristics.

JMP-L's FinalMLP Layer May Indicate Distribution Shifts: The BR/RandomMLP baseline (red in Figure 3) exhibits inconsistent behavior across different tasks. In QMOF, SPICE, and MD22, randomly initializing the first layer of FinalMLP had little impact on performance, suggesting a distribution shift between pre-training and downstream tasks. Conversely, BR/RandomMLP shows a noticeable performance drop in rMD17 and QM9, indicating that the learned FinalMLP features are more relevant to these tasks. This observation also aligns with the improved performance of BR+KD in rMD17 and QM9, where KD effectively preserves useful representations. These findings suggest that FinalMLP features could serve as indicators of distribution differences across tasks, highlighting variations in task similarity to the pre-training distribution.

4.4 Evaluating Training and Inference Efficiency

We complement our analysis with both training and inference times. For training time, Figure 4 illustrates the convergence speed over a fixed budget of 1 GPU-day for models with 3, 4, and 6 blocks. Compared to the 6-block (full JMP-L) model, the 4-block model demonstrates faster convergence on QM9 after 8 hours of training, lags behind on MD22, and achieves comparable convergence on the other datasets.

We complement our analysis with both training and inference times. For training efficiency, Figure 4 shows the convergence speed over a fixed 1 GPU-day budget for models with 3, 4, and 6 blocks. On QM9, the 4-blocks model achieves a lower loss than the 6-blocks model after 8 hours of training, though neither has fully converged. In contrast, it lags behind on MD22 and shows comparable convergence on the other datasets.

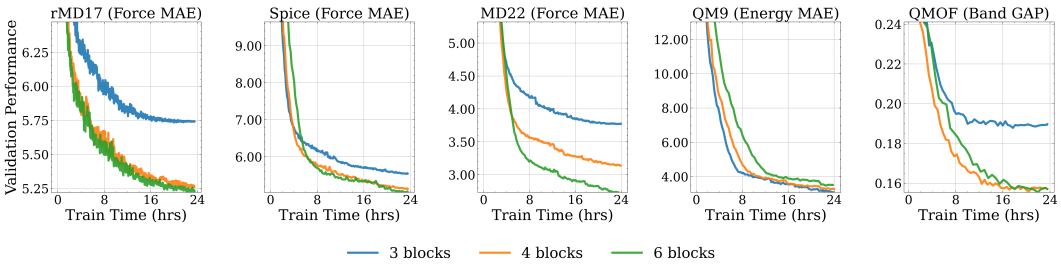

Figure 4: **Convergence Speed Analysis.** We present the training time and corresponding performance of JMP-L with 3, 4, and 6 blocks. Performance is measured in MAE: meV/Å for force targets, meV for the QM9 energy target, and eV for the QMOF band gap target.

To assess inference efficiency, we use the QMOF dataset, which has a large average graph size. Using a V100 GPU, we evaluate the models on a subset of the QMOF validation set and report the results in Table 2. Reducing the model from 6 blocks to 5 blocks slightly improves inference throughput, increasing it from 19.1 to 21.8 samples/s, with further gains as more blocks are removed—though at the cost of some performance degradation. Among the pruned models, the 4-block model achieves the best trade-off, offering a significant increase in throughput while maintaining competitive performance.

Table 2: **Inference Efficiency Analysis.** We evaluate the impact of block reduction on JMP-L's efficiency using a subset of the QMOF validation set. Reducing interaction blocks lowers computational cost and improves inference speed. The 4-block model provides the best trade-off, achieving a 1.3x speedup with a 32% reduction in parameters.

Blocks	Throughput (samples/s)	GFlops (Billion)	Parameters (M)
6-blocks (JMP-L)	19.1	1.74	160.9
5-blocks	21.8 (+2.7)	1.45 (-0.29)	134.5 (-16.4%)
4-blocks	25.6 (+6.5)	1.16 (-0.58)	108.2 (-32.7%)
3-blocks	30.8 (+11.7)	0.87 (-0.87)	81.9 (-49.1%)
2-blocks	38.0 (+18.9)	0.59 (-1.15)	55.5 (-65.5%)

5 Conclusion

In this work, we explored strategies to enhance the efficiency of foundation models for molecular property prediction. By analyzing the role of individual layers in JMP-L, we found that deeper interaction blocks contribute less to predictive accuracy, making them suitable candidates for pruning. Our results show that reducing JMP-L's parameter count by 32% improves inference throughput by 1.3× while maintaining comparable performance. Additionally, we demonstrated that knowledge distillation can help mitigate performance degradation in certain tasks. We hope this study inspires further research into efficient training and inference for molecular property prediction, paving the way for lighter models in molecular and materials discovery.

ACKNOWLEDGMENTS

This work is supported by the KAUST Center of Excellence for Generative AI under award number 5940. The computational resources are provided by IBEX, which is managed by the Supercomputing Core Laboratory at KAUST. Yasir is supported by Saudi Aramco.

REFERENCES

Ilyes Batatia, Simon Batzner, Dávid Péter Kovács, Albert Musaelian, Gregor N. C. Simm, Ralf Drautz, Christoph Ortner, Boris Kozinsky, and Gábor Csányi. The design space of e(3)-equivariant atom-centered interatomic potentials, 2022a.

Ilyes Batatia, David Peter Kovacs, Gregor N. C. Simm, Christoph Ortner, and Gabor Csanyi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=YPpSngE-ZU.

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

Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical Review Letters*, 98(14), April 2007. ISSN 1079-7114. doi: 10.1103/physrevlett.98.146401. URL http://dx.doi.org/10.1103/PhysRevLett.98.146401.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146, 2020.

Mihail Bogojeski, Leslie Vogt-Maranto, Mark E. Tuckerman, Klaus-Robert Müller, and Kieron Burke. Quantum chemical accuracy from density functional approximations via machine learning. *Nature Communications*, 11(1), October 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-19093-1. URL http://dx.doi.org/10.1038/s41467-020-19093-1.

Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pp. 535–541, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933395. doi: 10.1145/1150402.1150464. URL https://doi.org/10.1145/1150402.1150464.

Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, et al. Open catalyst 2020 (oc20) dataset and community challenges. *Acs Catalysis*, 11(10):6059–6072, 2021.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning-taxonomy, comparison, analysis, and recommendations, 2024. URL https://arxiv.org/abs/2308.06767.

Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.

Stefan Chmiela, Valentin Vassilev-Galindo, Oliver T Unke, Adil Kabylda, Huziel E Sauceda, Alexandre Tkatchenko, and Klaus-Robert Müller. Accurate global machine learning force fields for molecules with hundreds of atoms. *Science Advances*, 9(2):eadf0873, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

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.

Peter Eastman, Pavan Kumar Behara, David L Dotson, Raimondas Galvelis, John E Herr, Josh T Horton, Yuezhi Mao, John D Chodera, Benjamin P Pritchard, Yuanqing Wang, et al. Spice, a dataset of drug-like molecules and peptides for training machine learning potentials. *Scientific Data*, 10(1):11, 2023.

Filip Ekström Kelvinius, Dimitar Georgiev, Artur Toshev, and Johannes Gasteiger. Accelerating molecular graph neural networks via knowledge distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

Johannes Gasteiger, Florian Becker, and Stephan Günnemann. Gemnet: Universal directional graph neural networks for molecules. *Advances in Neural Information Processing Systems*, 34:6790–6802, 2021.

Johannes Gasteiger, Muhammed Shuaibi, Anuroop Sriram, Stephan Günnemann, Zachary Ulissi, C Lawrence Zitnick, and Abhishek Das. Gemnet-oc: developing graph neural networks for large and diverse molecular simulation datasets. *arXiv preprint arXiv:2204.02782*, 2022.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers, 2024. URL https://arxiv.org/abs/2403.17887.

Gousia Habib, Tausifa Jan Saleem, and Brejesh Lall. Knowledge distillation in vision transformers: A critical review, 2024. URL https://arxiv.org/abs/2302.02108.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Adeesh Kolluru, Muhammed Shuaibi, Aini Palizhati, Nima Shoghi, Abhishek Das, Brandon Wood, C. Lawrence Zitnick, John R Kitchin, and Zachary W Ulissi. Open challenges in developing generalizable large scale machine learning models for catalyst discovery, 2022. URL https://arxiv.org/abs/2206.02005.

Dávid Péter Kovács, J Harry Moore, Nicholas J Browning, Ilyes Batatia, Joshua T Horton, Venkat Kapil, Ioan-Bogdan Magdău, Daniel J Cole, and Gábor Csányi. Mace-off23: Transferable machine learning force fields for organic molecules. *arXiv preprint arXiv:2312.15211*, 2023.

Chuang Liu, Xueqi Ma, Yibing Zhan, Liang Ding, Dapeng Tao, Bo Du, Wenbin Hu, and Danilo Mandic. Comprehensive graph gradual pruning for sparse training in graph neural networks, 2022. URL https://arxiv.org/abs/2207.08629.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-pruner: On the structural pruning of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=J8Ajf9WfXP.

Jens K. Nørskov, Frank Abild-Pedersen, Felix Studt, and Thomas Bligaard. Density functional theory in surface chemistry and catalysis. *Proceedings of the National Academy of Sciences*, 108 (3):937–943, January 2011. ISSN 1091-6490. doi: 10.1073/pnas.1006652108. URL http://dx.doi.org/10.1073/pnas.1006652108.

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.

Andrew S Rosen, Shaelyn M Iyer, Debmalya Ray, Zhenpeng Yao, Alan Aspuru-Guzik, Laura Gagliardi, Justin M Notestein, and Randall Q Snurr. Machine learning the quantum-chemical properties of metal–organic frameworks for accelerated materials discovery. *Matter*, 4(5):1578–1597, 2021.

Andrew S. Rosen, Victor Fung, Patrick Huck, Cody T. O'Donnell, Matthew K. Horton, Donald G. Truhlar, Kristin A. Persson, Justin M. Notestein, and Randall Q. Snurr. High-throughput predictions of metal–organic framework electronic properties: theoretical challenges, graph neural networks, and data exploration. *npj Computational Materials*, 8(1), May 2022. ISSN 2057-3960. doi: 10.1038/s41524-022-00796-6. URL http://dx.doi.org/10.1038/s41524-022-00796-6.

Victor T. Sabe, Thandokuhle Ntombela, Lindiwe A. Jhamba, Glenn E.M. Maguire, Thavendran Govender, Tricia Naicker, and Hendrik G. Kruger. Current trends in computer aided drug design and a highlight of drugs discovered via computational techniques: A review. *European Journal of Medicinal Chemistry*, 224:113705, November 2021. ISSN 0223-5234. doi: 10.1016/j.ejmech.2021.113705. URL http://dx.doi.org/10.1016/j.ejmech.2021.113705.

Mathias Schreiner, Arghya Bhowmik, Tejs Vegge, Jonas Busk, and Ole Winther. Transition1x - a dataset for building generalizable reactive machine learning potentials. *Scientific Data*, 9(1), December 2022a. ISSN 2052-4463. doi: 10.1038/s41597-022-01870-w. URL http://dx.doi.org/10.1038/s41597-022-01870-w.

Mathias Schreiner, Arghya Bhowmik, Tejs Vegge, Jonas Busk, and Ole Winther. Transition1x-a dataset for building generalizable reactive machine learning potentials. *Scientific Data*, 9(1):779, 2022b.

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

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017. doi: 10.1109/ICCV.2017.74.

Vinit Sharma, Chenchen Wang, Robert G. Lorenzini, Rui Ma, Qiang Zhu, Daniel W. Sinkovits, Ghanshyam Pilania, Artem R. Oganov, Sanat Kumar, Gregory A. Sotzing, Steven A. Boggs, and Rampi Ramprasad. Rational design of all organic polymer dielectrics. *Nature Communications*, 5(1), September 2014. ISSN 2041-1723. doi: 10.1038/ncomms5845. URL http://dx.doi.org/10.1038/ncomms5845.

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. *arXiv preprint arXiv:2310.16802*, 2023.

Sietsma and Dow. Neural net pruning-why and how. In *IEEE 1988 International Conference on Neural Networks*, pp. 325–333 vol.1, 1988. doi: 10.1109/ICNN.1988.23864.

Justin S Smith, Roman Zubatyuk, Benjamin Nebgen, Nicholas Lubbers, Kipton Barros, Adrian E Roitberg, Olexandr Isayev, and Sergei Tretiak. The ani-1ccx and ani-1x data sets, coupled-cluster and density functional theory properties for molecules. *Scientific data*, 7(1):134, 2020.

Anuroop Sriram, Sihoon Choi, Xiaohan Yu, Logan M. Brabson, Abhishek Das, Zachary Ulissi, Matt Uyttendaele, Andrew J. Medford, and David S. Sholl. The open dac 2023 dataset and challenges for sorbent discovery in direct air capture, 2023. URL https://arxiv.org/abs/2311.00341.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models, 2024. URL https://arxiv.org/abs/2306.11695.

Yijun Tian, Shichao Pei, Xiangliang Zhang, Chuxu Zhang, and Nitesh V. Chawla. Knowledge distillation on graphs: A survey, 2023. URL https://arxiv.org/abs/2302.00219.

Richard Tran, Janice Lan, Muhammed Shuaibi, Brandon M Wood, Siddharth Goyal, Abhishek Das, Javier Heras-Domingo, Adeesh Kolluru, Ammar Rizvi, Nima Shoghi, et al. The open catalyst 2022 (oc22) dataset and challenges for oxide electrocatalysts. *ACS Catalysis*, 13(5):3066–3084, 2023.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models, 2024. URL https://arxiv.org/abs/2402.13116.

Fang Yu, Kun Huang, Meng Wang, Yuan Cheng, Wei Chu, and Li Cui. Width & depth pruning for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3143–3151, 2022.

Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via denoising for molecular property prediction, 2022. URL https://arxiv.org/abs/2206.00133.

Liang Zeng, Lanqing Li, and Jian Li. Molkd: Distilling cross-modal knowledge in chemical reactions for molecular property prediction, 2023. URL https://arxiv.org/abs/2305.01912.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. Loraprune: Structured pruning meets low-rank parameter-efficient fine-tuning, 2024. URL https://arxiv.org/abs/2305.18403.

Gengmo Zhou, Zhifeng Gao, Qiankun Ding, Hang Zheng, Hongteng Xu, Zhewei Wei, Linfeng Zhang, and Guolin Ke. Uni-mol: A universal 3d molecular representation learning framework. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6K2RM6wVqKu.